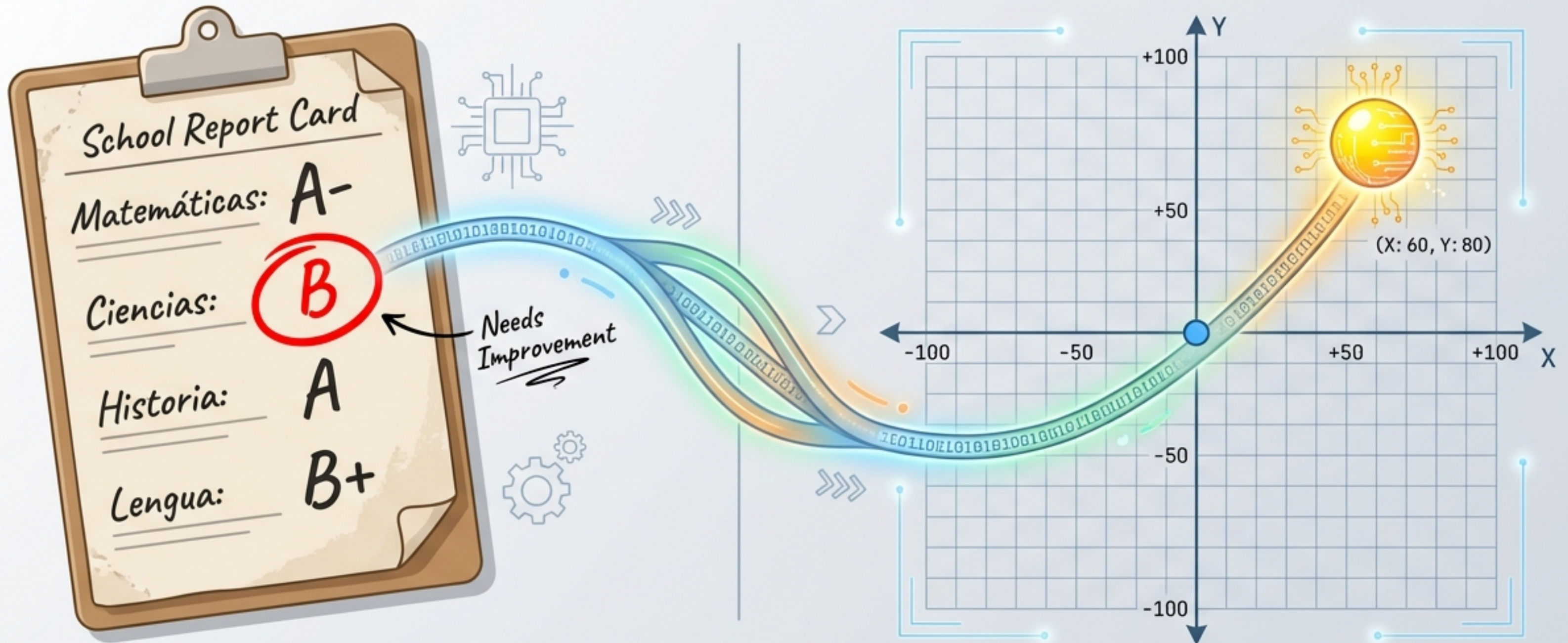


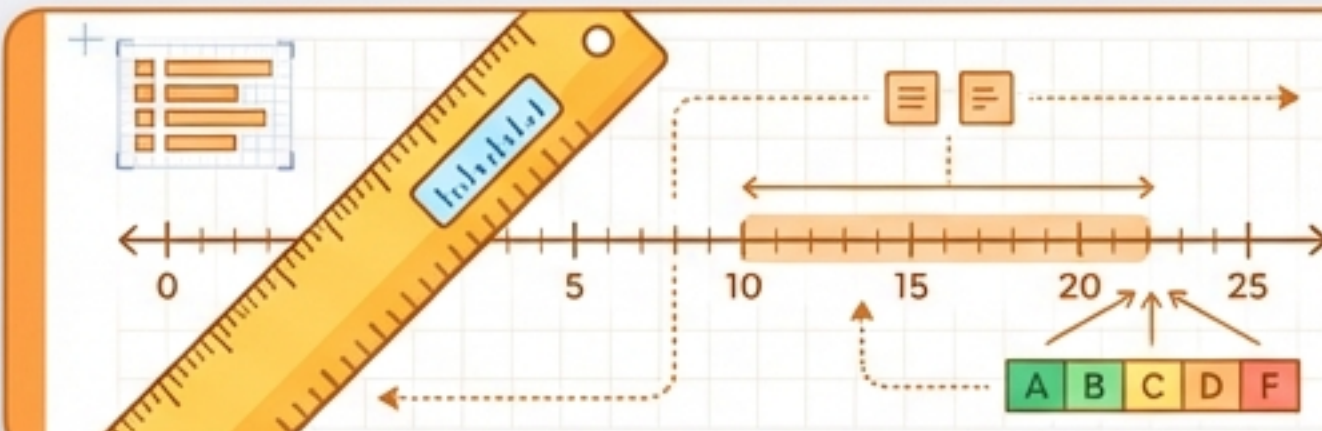
Domina las Condiciones: Lógica Avanzada en Scratch

Guía 1, Sesión 3 – Grado 7º



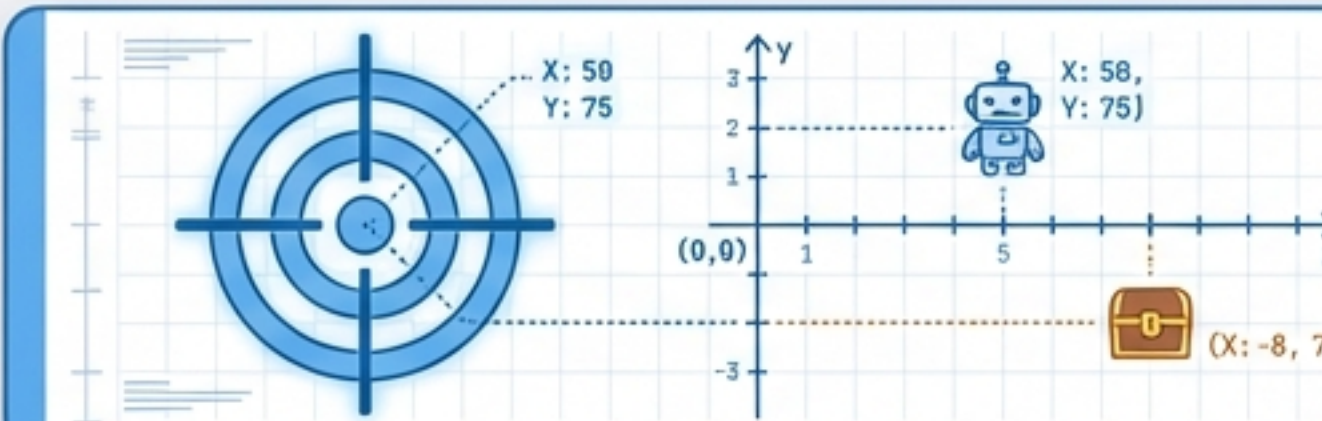
Hoy pasamos de preguntas simples de 'Sí o No' a enseñar a la computadora a entender **rangos** y **espacio**. Aprenderás a automatizar decisiones complejas, desde calcular tu nota final hasta ubicar objetos en un plano cartesiano.

Tu Misión de Hoy: Tres Objetivos Clave



Establecer Rangos:

Usar secuencias numéricas para clasificar datos (como notas escolares).



Ubicar Objetos:

Entender y programar coordenadas en el plano cartesiano.



Lógica Anidada:

Aplicar 'condicionales anidados' en Scratch para filtrar información paso a paso.

El Desafío de Gabriela: ¿Cómo clasificamos las notas intermedias?

Gabriela tiene su nota final. Si saca menos de 60, es fácil: "Deficiente". Si saca más de 86, es "Excelente".



Tabla de Clasificación de Notas

Deficiente: Menor a 60

Básico: Entre 60 y 75

Alto: Entre 76 y 85

Excelente: Entre 86 y 100

Pero, ¿cómo le decimos a la computadora que detecte los números que están *en el medio* (como un **65** o un **80**)? Para esto, necesitamos **límites**.

La Herramienta Secreta: Condicionales Anidados

Una condicional anidada es simplemente una pregunta **dentro** de otra pregunta. Nos permite crear filtros jerárquicos.

Pregunta 1 (Límite Inferior)

Pregunta 2 (Límite Superior)

Respuesta
Correcta

En lugar de preguntar todo a la vez, vamos paso a paso:

1. ¿La nota es mayor que el límite inferior? **(SÍ) → Entramos.**
2. Ahora, ¿la nota es menor que el límite superior? **(SÍ) → ¡Lo encontramos!**

Construyendo el Código: El Rango "Básico"



Para el rango de 60 a 75:

Para el rango de 60 a 75:

El **Límite Inferior** es 60

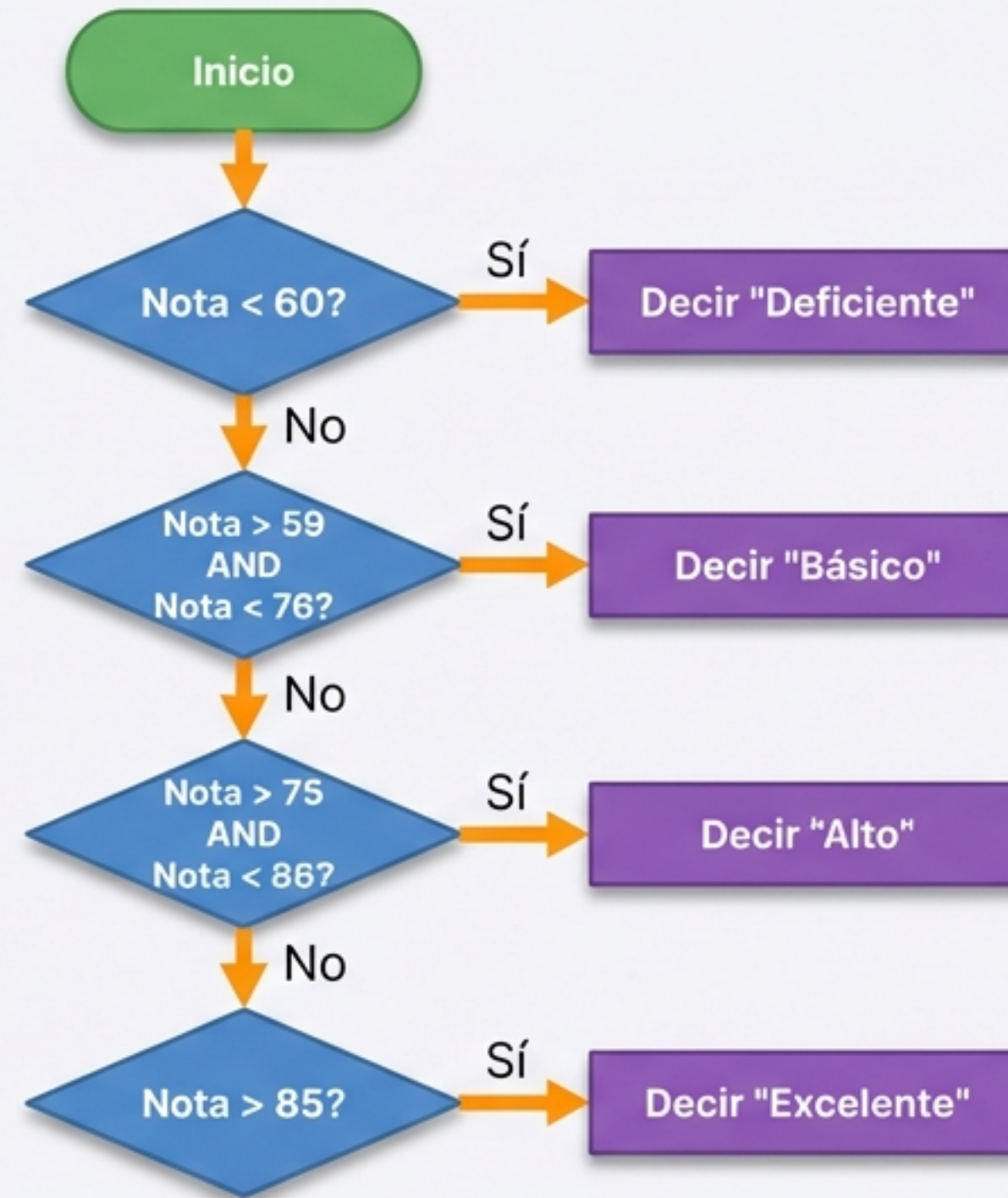
(preguntamos > 59) y el

Límite Superior es 75

(preguntamos < 76).

Solo si *ambas* condiciones son verdaderas, la computadora dirá "Básico".

Solución Completa: El Programa Unificado

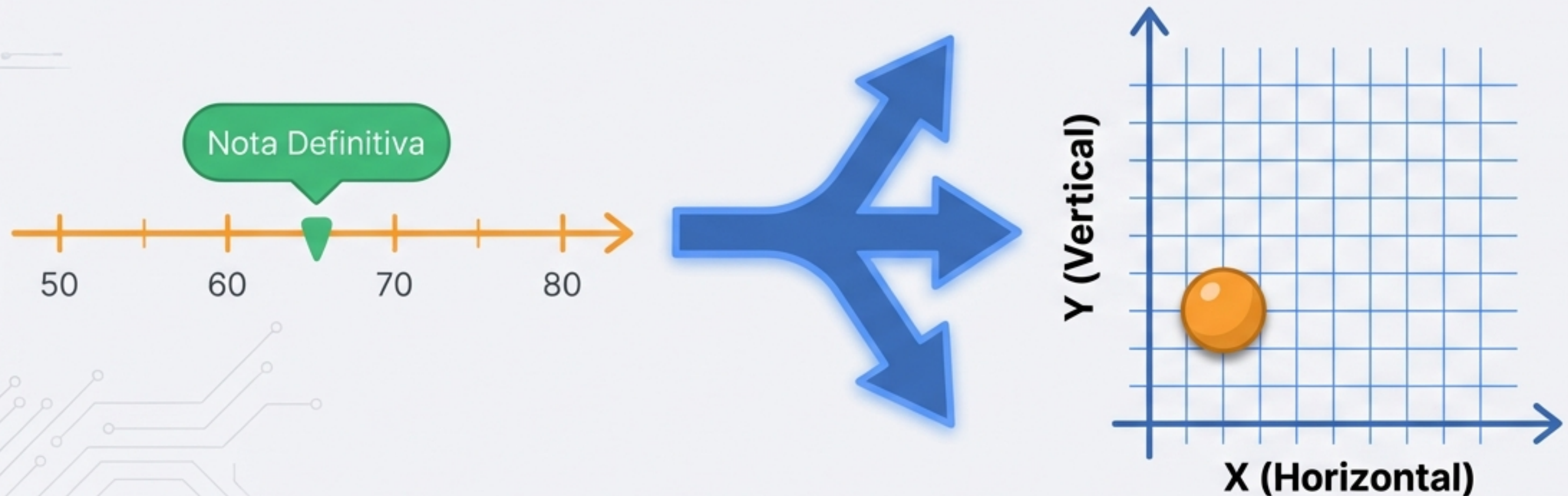


****Reto:**** ¿Cómo podrías pedirle al usuario que **ingrese su propia nota?**
(Pista: **Usa el bloque azul 'Preguntar' en Sensores.**)

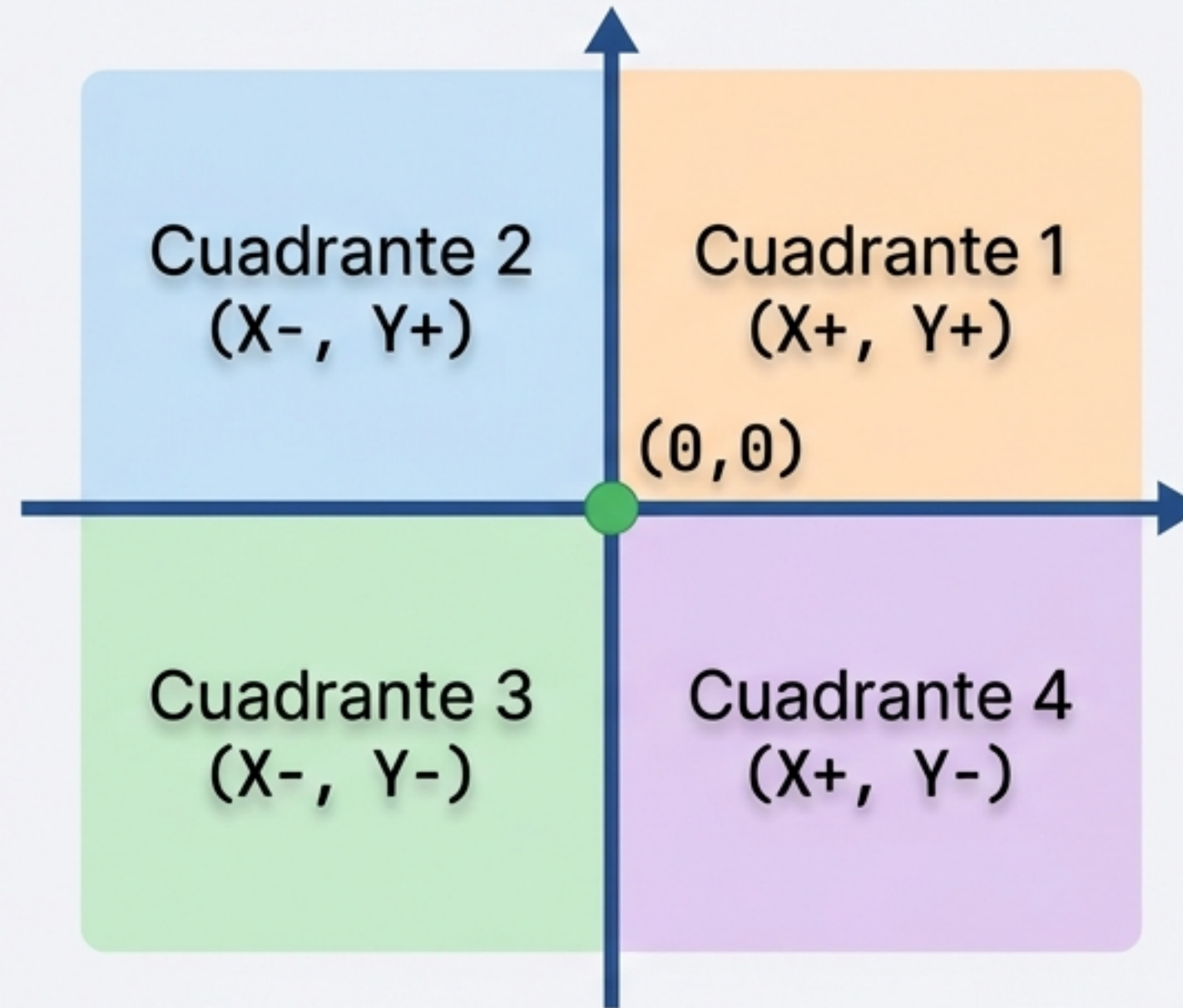
Siguiente Nivel: Entrando a la Segunda Dimensión

Las notas son lineales (una sola variable). Pero el espacio tiene **dos dimensiones**: **X** (horizontal) y **Y** (vertical).

Ahora usaremos la misma lógica de “preguntas anidadas” para ubicar un objeto “Ball” (Pelota) en el espacio.



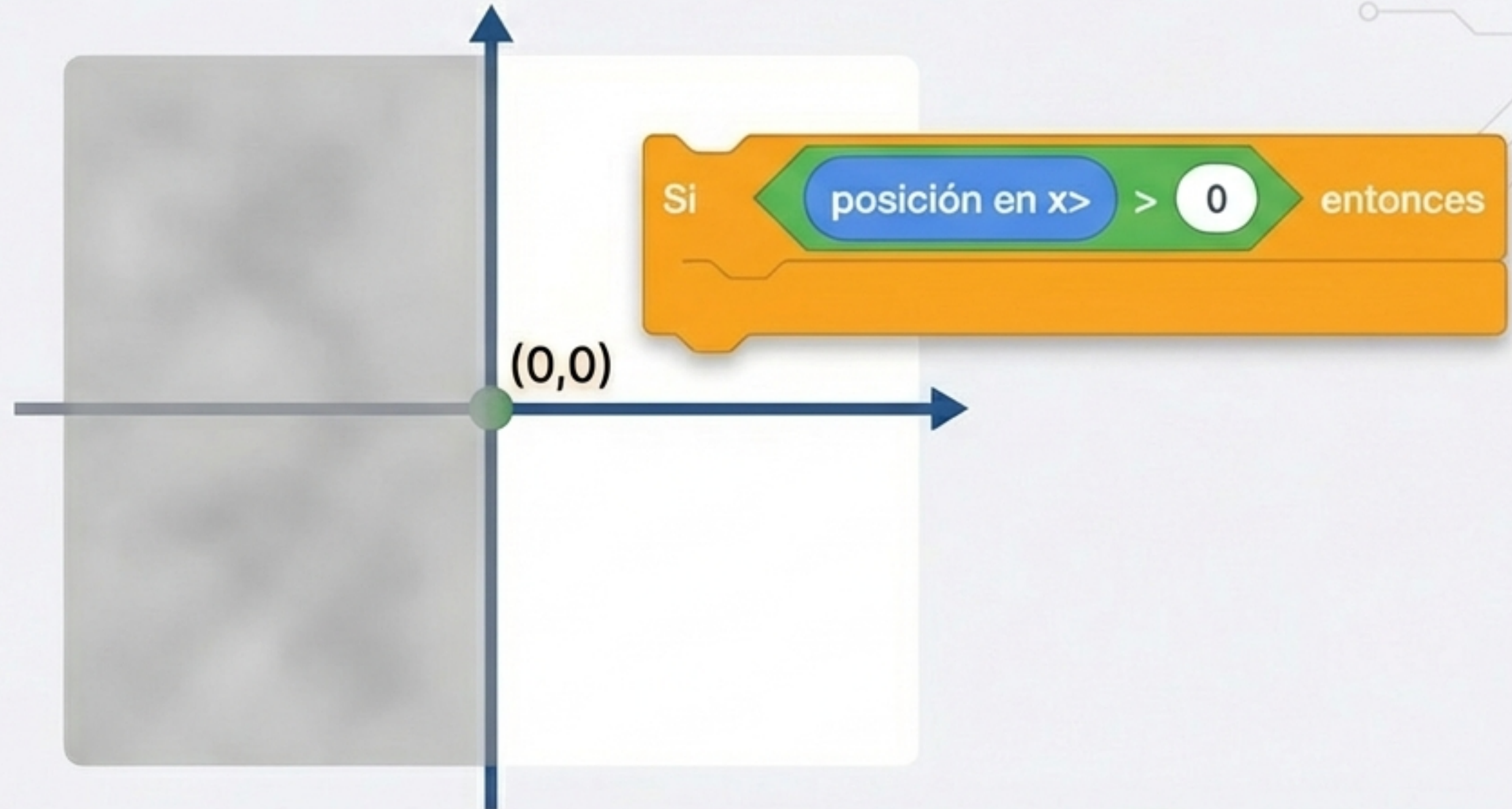
El Mapa: Entendiendo los Cuadrantes



El plano se divide en cuatro sectores basados en si X y Y son positivos (+) o negativos (-). Nuestro objetivo es que la pelota sepa en qué sector está automáticamente.

Estrategia de Ubicación: Divide y Vencerás

La primera pregunta divide el mundo en dos. Si **X es mayor que 0**, sabemos que la pelota está a la **DERECHA**. Pero, ¿está arriba o abajo? Para saberlo, necesitamos una segunda pregunta anidada.



Detectando el Lado Derecho

Dentro de la zona derecha ($X > 0$), preguntamos por Y. Si es positivo, estamos arriba. **Si no** ('Else'), significa que Y es negativo, por lo tanto, estamos abajo.

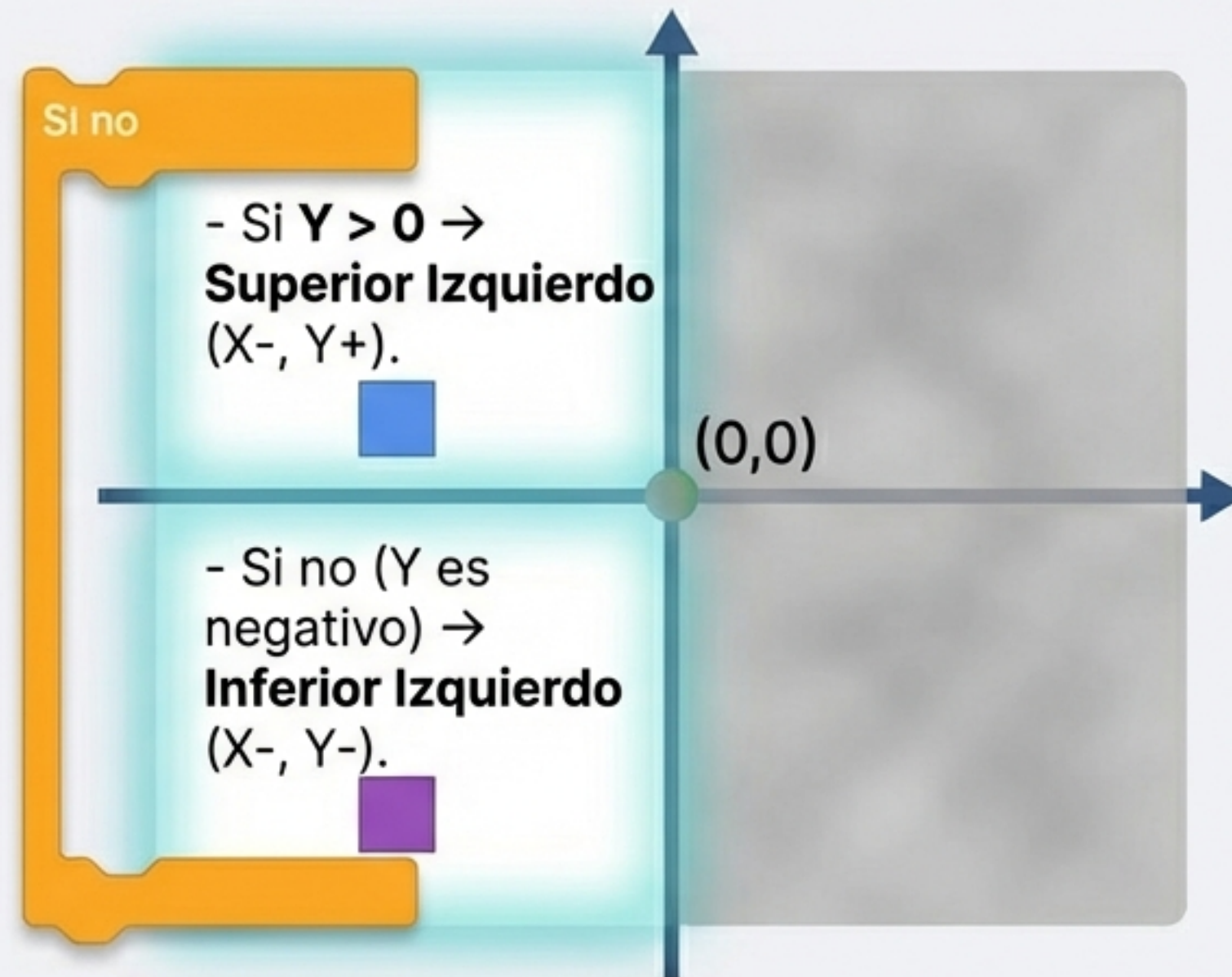


Objeto	Ball	x	-75	y	-180
		Tamaño	100	Dirección	90

El Laberinto del 'Si No': El Lado Izquierdo

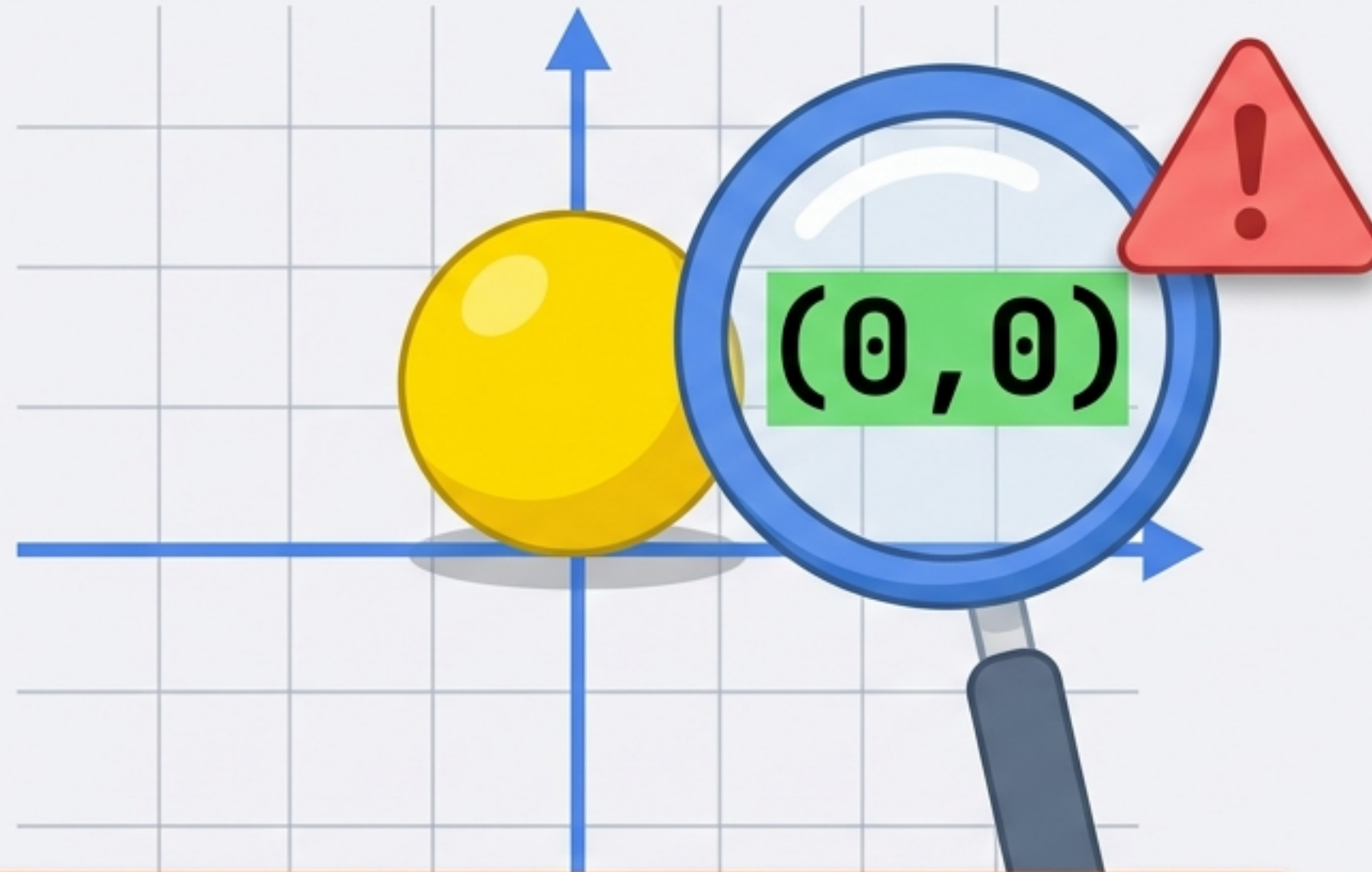
Si la respuesta a ' $X > 0$ ' es **NO**, entramos al bloque 'Si no'.

Aquí aplicamos la misma lógica:



El 'Bug': ¿Qué pasa con el Cero?

Nuestro código anterior usa `> 0`. Pero 0 no es mayor que 0. Si X es 0, el código podría enviarnos erróneamente al bloque "Si no" y confundirnos.

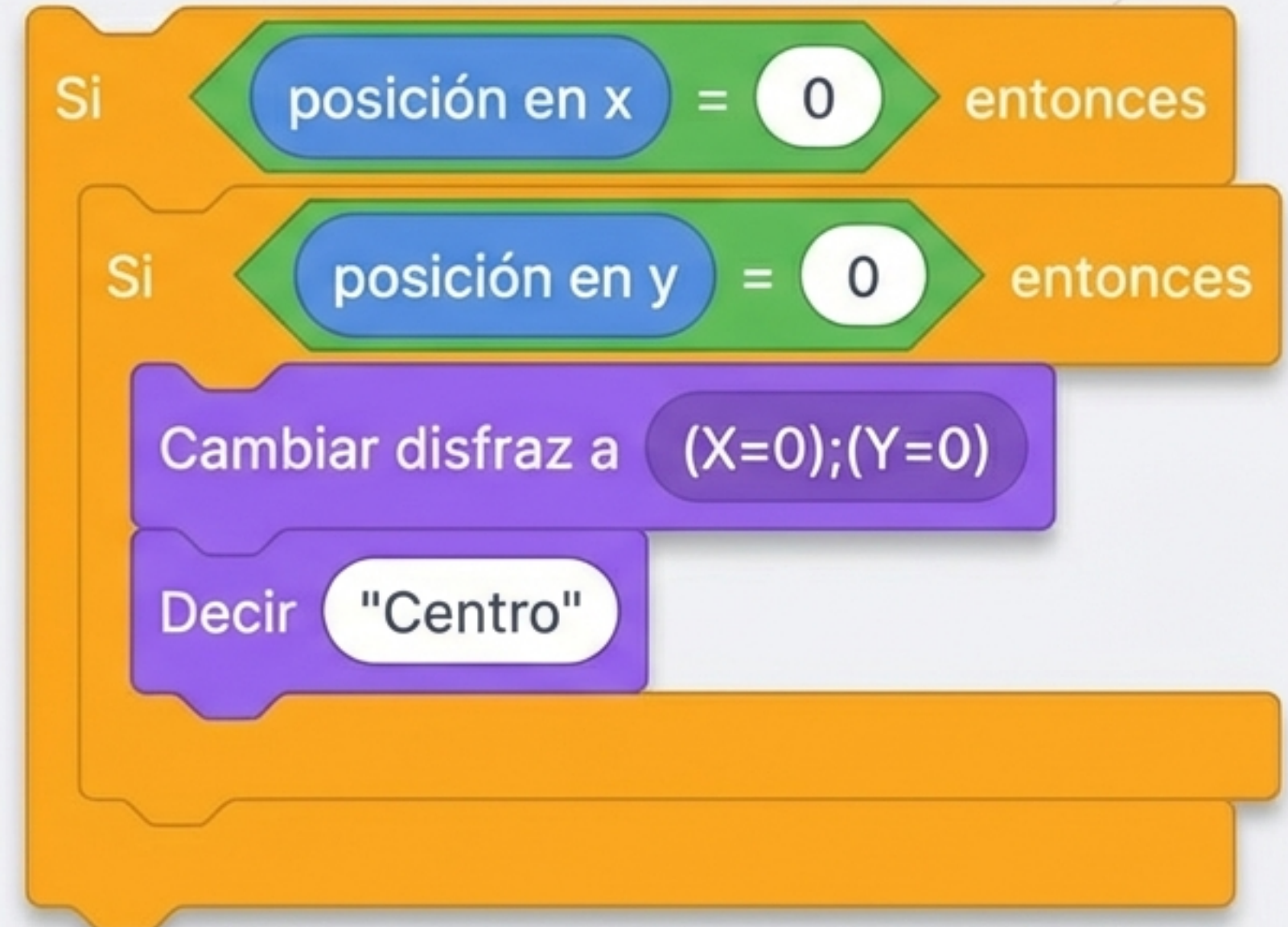


¡Necesitamos ser precisos!

Solución de Precisión: Detectando el Origen

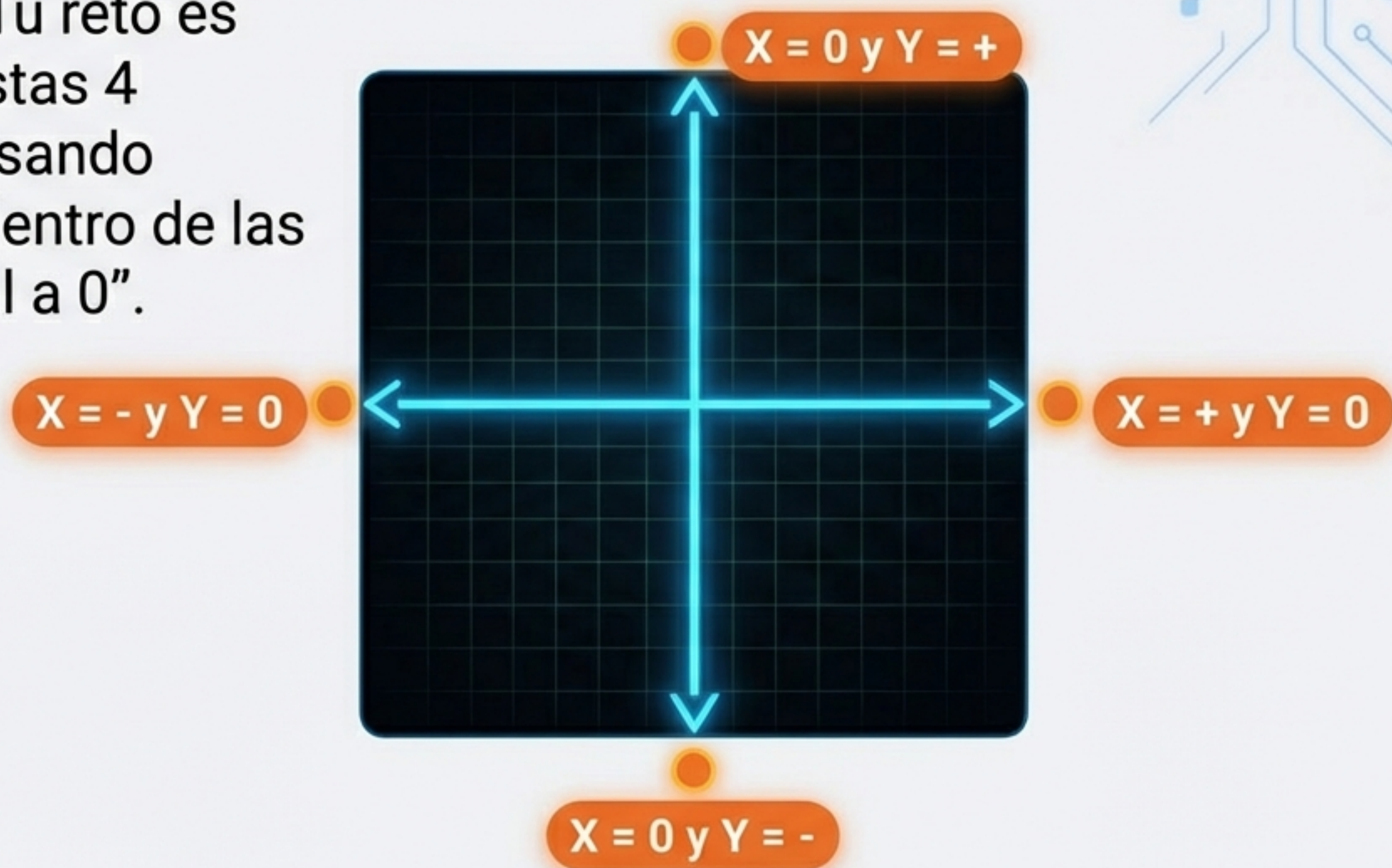
Para encontrar el centro, usamos el operador de igualdad (=).

1. Preguntamos si **X** es igual a 0.
2. Si es así, verificamos si **Y** *también* es igual a 0.
3. ¡Solo entonces estamos en el origen!

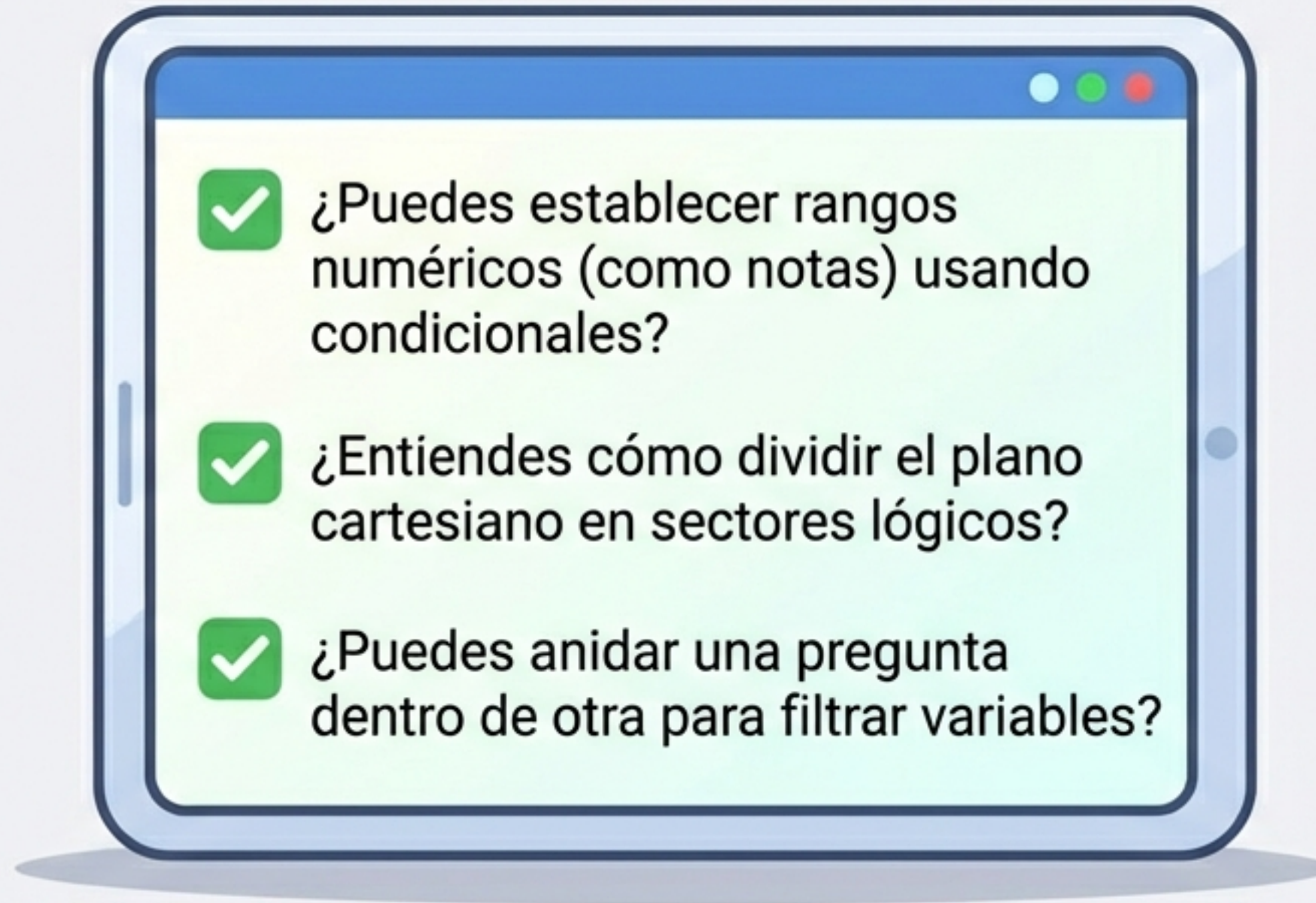


Reto Final: Los Ejes Invisibles

El código aún tiene “puntos ciegos” en las líneas de los ejes. Tu reto es agregar mensajes para estas 4 situaciones específicas usando condicionales anidados dentro de las comprobaciones de “Igual a 0”.



Lista de Verificación: ¿Dominaste las Condiciones?



La lógica es el lenguaje de la programación. Ahora lo hablas con mayor fluidez.
Si tienes dudas, revisa tu código y experimenta con los valores de X y Y.