

RETO: Cantidad de Impares

El gato pedirá qué cantidad de impares quieres que te escriba.

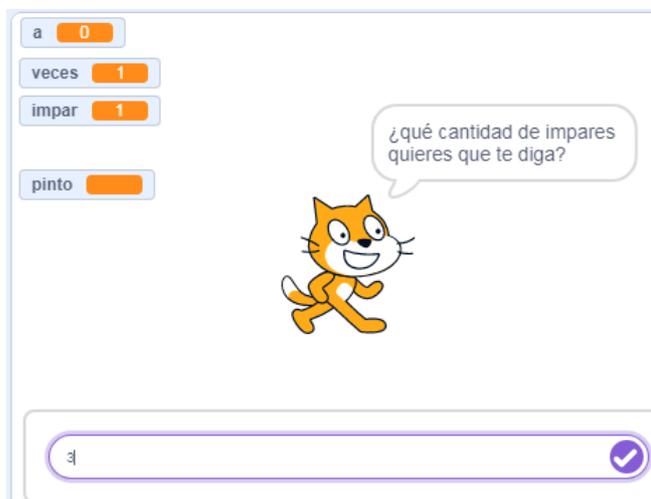
Si le pones 1, dirá 1.

Si le pones 2, te dirá 1 3.

Si le pones 3, te dirá 1 3 5.

Etc.

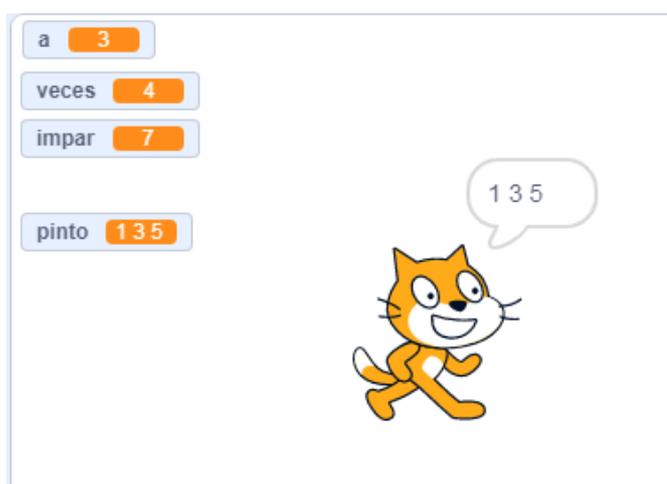
Ejemplo, le ponemos 3:



The image shows a Scratch workspace with the following elements:

- Variables: 'a' (0), 'veces' (1), 'impar' (1), and 'pinto' (empty).
- Sprite: A cat character with a speech bubble that says "¿qué cantidad de impares quieres que te diga?".
- Input: A text box containing the number '3' with a checkmark icon on the right.

Y responderá con 1 3 5



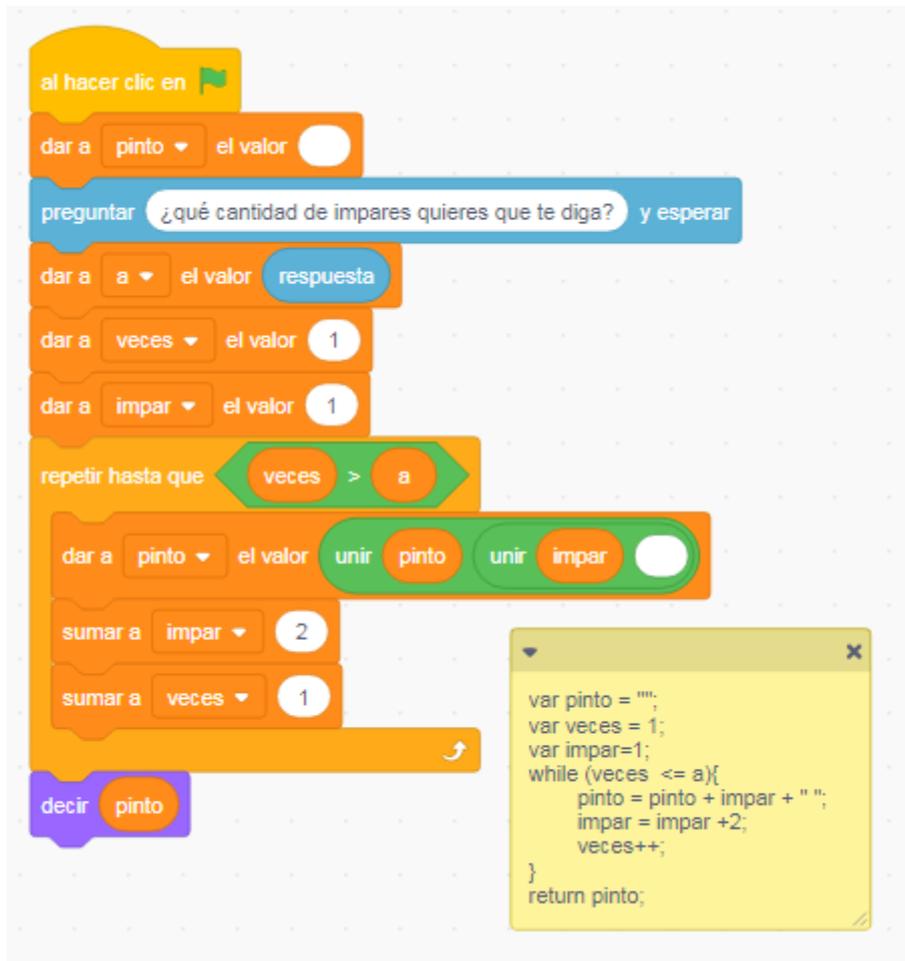
The image shows the same Scratch workspace after the user's input:

- Variables: 'a' (3), 'veces' (4), 'impar' (7), and 'pinto' (135).
- Sprite: The cat character with a speech bubble that says "1 3 5".

Observa el código en Javascript.

Tengo un argumento de entrada (a)

Necesito **2 variables (veces e impar)** y un bucle **repetir hasta que** (parecido a mientras que = while) para gestionar la respuesta.



The image displays a Scratch script and a corresponding JavaScript code window. The Scratch script starts with a 'when clicked' event, followed by setting 'pinto' to an empty string. It then asks the user '¿qué cantidad de impares quieres que te diga?' and waits for a response. The response is stored in 'a'. Two variables, 'veces' and 'impar', are both initialized to 1. A 'repeat until' loop is set up with the condition 'veces > a'. Inside the loop, 'pinto' is updated by concatenating 'impar' to it, 'impar' is incremented by 2, and 'veces' is incremented by 1. After the loop, 'pinto' is spoken.

```
var pinto = "";  
var veces = 1;  
var impar=1;  
while (veces <= a){  
  pinto = pinto + impar + " ";  
  impar = impar +2;  
  veces++;  
}  
return pinto;
```

También se puede definir un Bloque o **Función**

Cambia el programa y crea el BLOQUE

The image shows a Scratch code editor with a script on the left and a custom block definition on the right. The script starts with a 'when clicked' event, followed by setting 'pinto' to an empty string, asking the user for the number of odd numbers, and then calling the custom block 'pintaTantosNumerosImpares'. The custom block definition sets 'veces' to 1 and 'impar' to 1, then enters a 'repeat until' loop where 'veces' is greater than 'a'. Inside the loop, 'pinto' is updated with the current 'impar' value, 'impar' is incremented by 2, and 'veces' is incremented by 1. After the loop, 'pinto' is spoken.

```
var pinto = "";  
var veces = 1;  
var impar=1;  
while (veces <= a){  
  pinto = pinto + impar + " ";  
  impar = impar +2;  
  veces++;  
}  
return pinto;
```