

Bloque 1: Introducción práctica

1.4. Pruebas y depuración

José Ignacio Huertas Fernández

@jihuefer

Jesús Moreno León

@J_MorenoL

<http://programamos.es>

info@programamos

Twitter: [@programamos](https://twitter.com/programamos)

<http://facebook.es/programamos>

©Programamos, octubre de 2015

Algunos derechos reservados. Este artículo se distribuye bajo la licencia "Reconocimiento-CompartirIgual 3.0 España" de Creative Commons, disponible en <http://creativecommons.org/licenses/by-sa/3.0/es/deed.es>

Este documento (o uno muy similar) está disponible en (o enlazado desde) <http://programamos.es>

1. Recalcar fundamentos del Pensamiento Computacional.
2. Localizar, comprender y resolver errores de nuestros proyectos.
3. Buenas prácticas de programación para evitar errores.
4. Desarrollo iterativo e incremental.

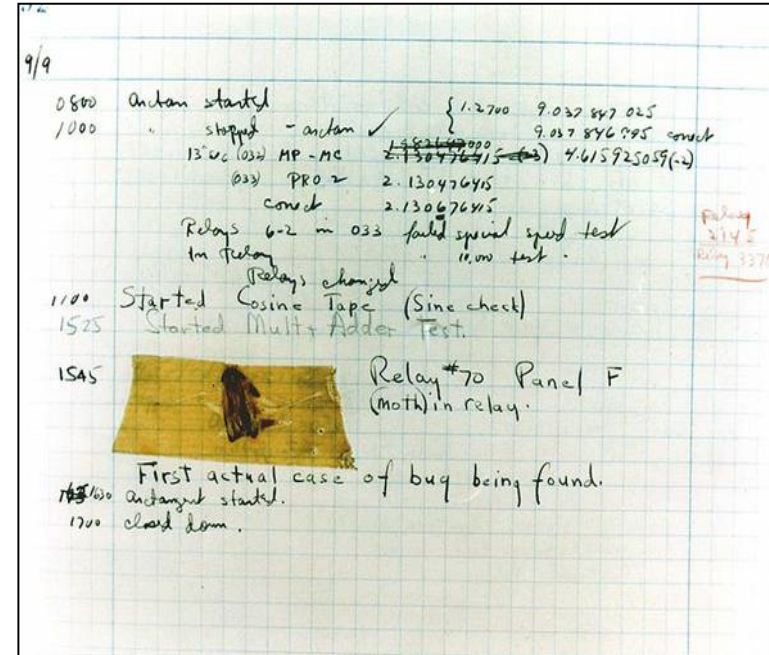
Errores en los programas

Todo programador tiene que enfrentarse a programas que parecen perfectamente correctos pero que no se comportan como se espera de ellos.

Para encontrar los errores, o **bugs**, es necesario **depurar** los programas (**debugging**)

Origen del término *bug*

«First actual case of bug being found».



Fuente: http://es.wikipedia.org/wiki/Grace_Murray_Hopper

Errores típicos en Scratch

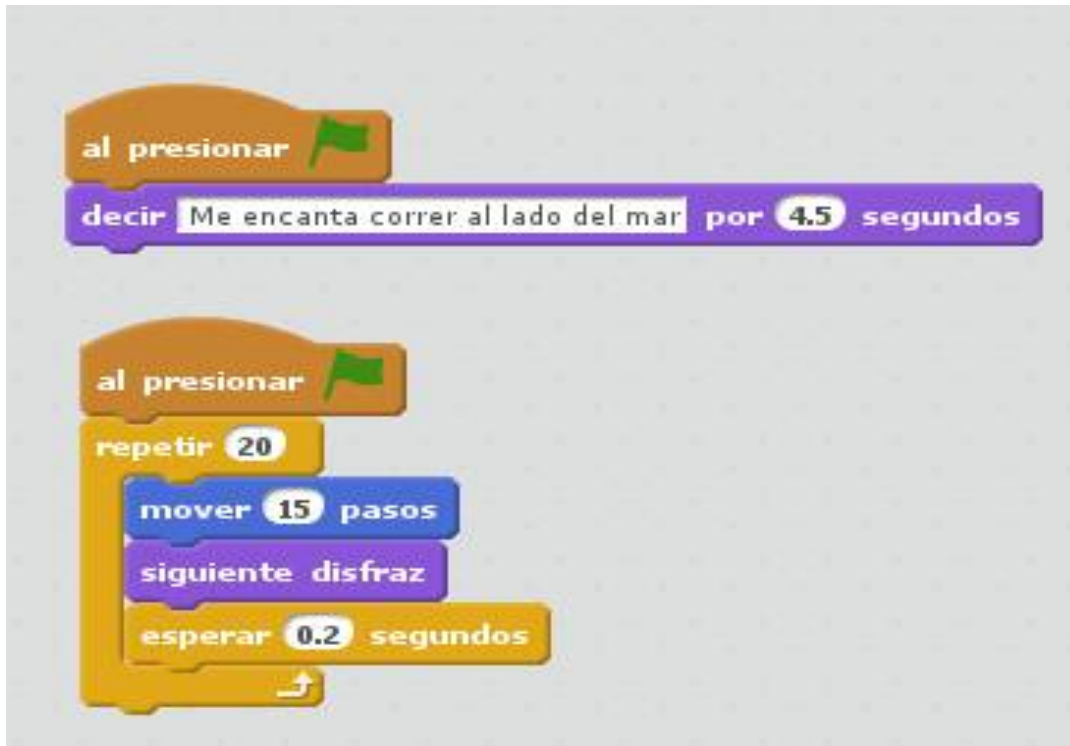
Error 1



Error 1: <http://scratch.mit.edu/projects/22825299>

Errores típicos en Scratch

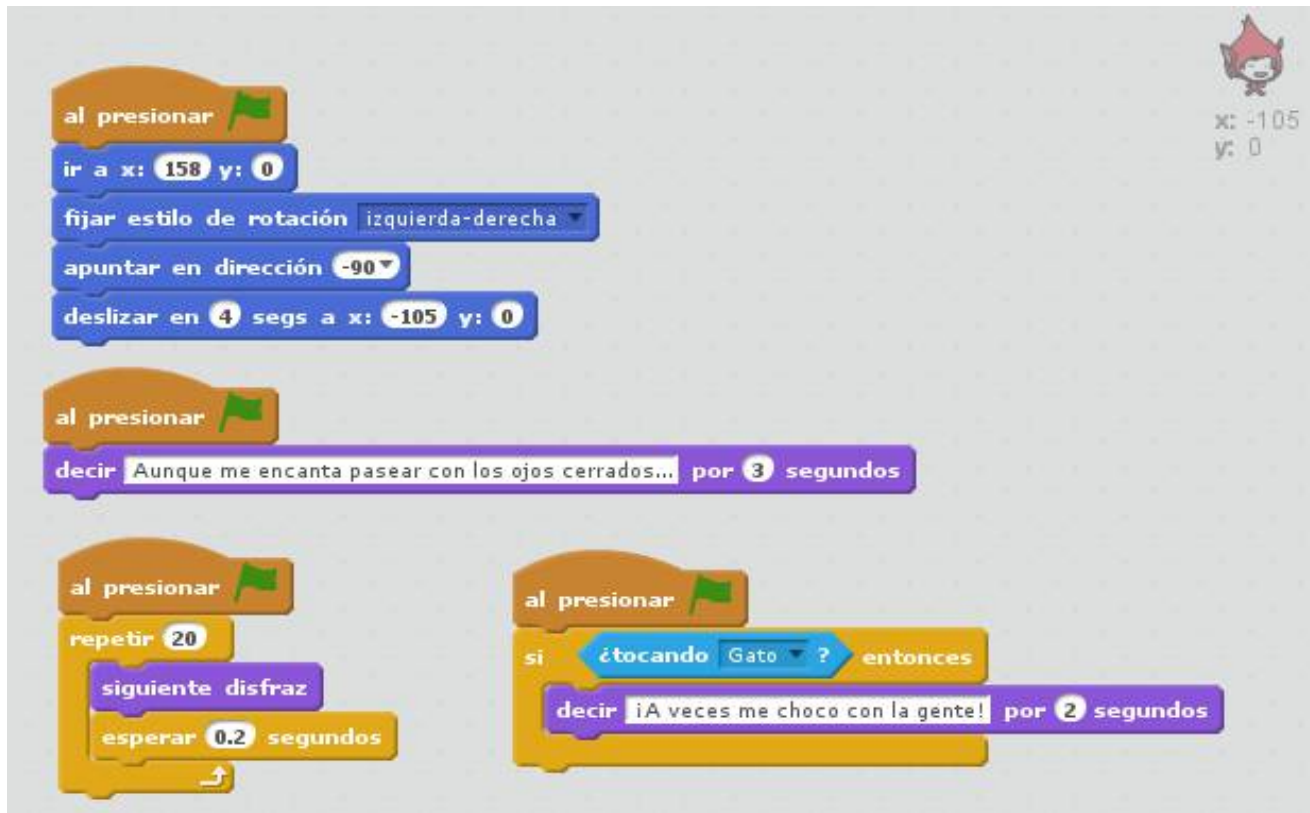
Error 2



Error2: <http://scratch.mit.edu/projects/22826068>

Errores típicos en Scratch

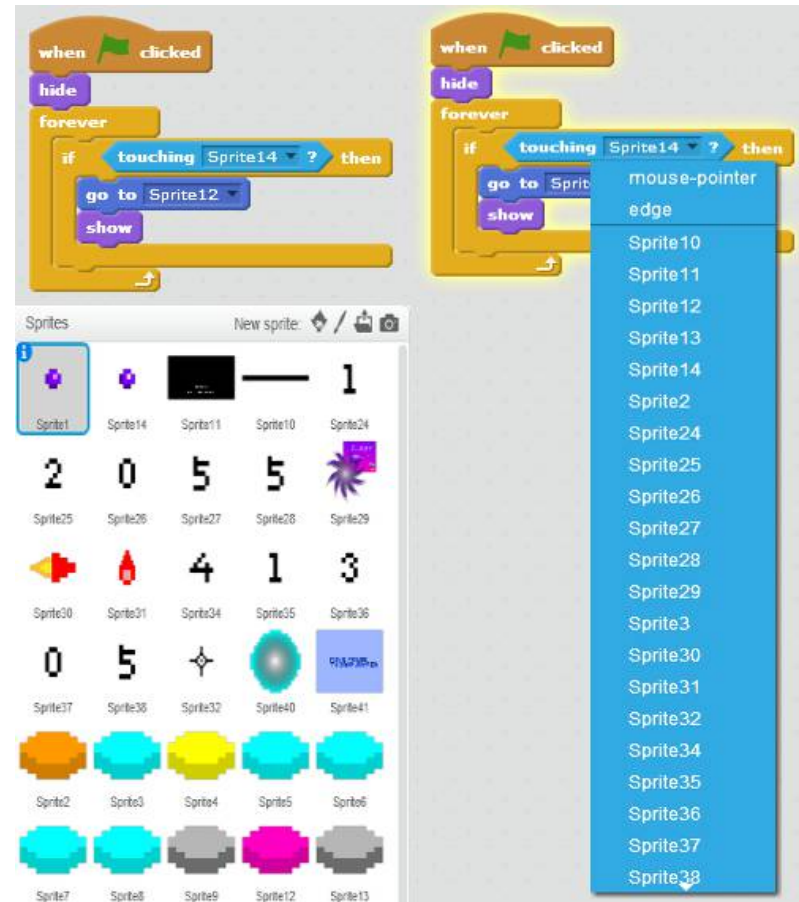
Error 3



Error3: <http://scratch.mit.edu/projects/22826700/>

Nombrado de personajes

- Scratch asigna un nombre automáticamente a cada personaje de un proyecto: Objeto1, Objeto2....
- Es muy recomendable asignar un nombre significativo a cada personaje.



Comentarios

- Los comentarios son bloques que permiten introducir anotaciones legibles para los programadores dentro del código de un programa.
- Estos bloques son ignorados en compilación y no tienen efecto en ejecución.
- El objetivo de los comentarios es conseguir que el código sea más fácil de entender por un programador que tenga que mantenerlo o actualizarlo.

Comentarios



The image shows a Scratch script with the following blocks:

- cuando el fondo cambie a Fondo2
- mostrar
- ir a x: 0 y: 0
- por siempre
- si ¿tocando el color []? entonces
- girar 180 grados
- mover 10 pasos

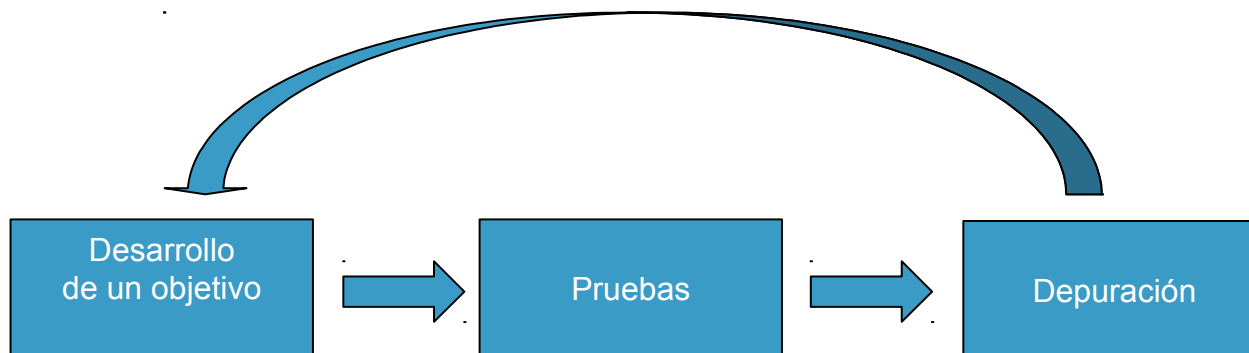
Two yellow callout boxes provide explanations:

- The first callout points to the 'cuando el fondo cambie a Fondo2' block and contains the text: "Comienza la partida porque el usuario ha pinchado en 'Start'".
- The second callout points to the 'si ¿tocando el color []? entonces' block and contains the text: "Está tocando un borde y debe retroceder automáticamente".

Desarrollo iterativo e incremental

El desarrollo de un proyecto debe realizarse de forma iterativa e incremental:

- Se divide el proyecto en objetivos pequeños que puedan desarrollarse y que puedan ser probados y depurados de forma sencilla.
- Cuando se comprueba que un objetivo funciona correctamente, se pasa al siguiente objetivo

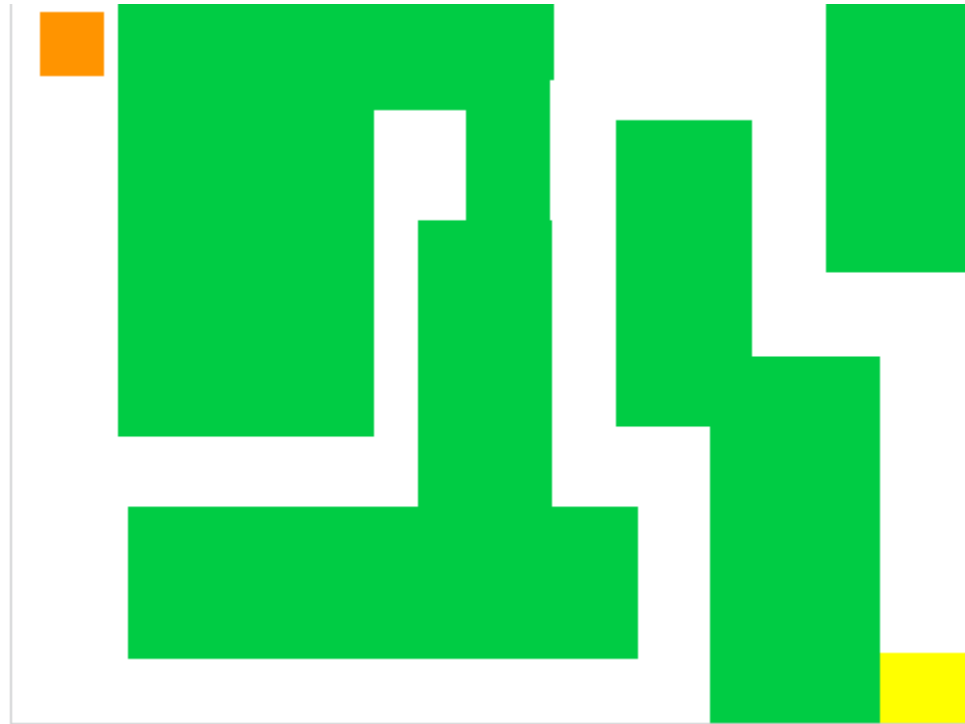


Desafío

- Vamos a desarrollar un videojuego sencillo, un laberinto, de manera iterativa e incremental y tratando de seguir las buenas prácticas recomendadas para evitar errores.
- Ejemplo de laberinto: <http://scratch.mit.edu/projects/2042736/>
- Objetivos:
 - Diseño del personaje y del escenario
 - Estados iniciales
 - Navegación interactiva
 - Rebotar en las paredes
 - Condición final del programa

Desafío

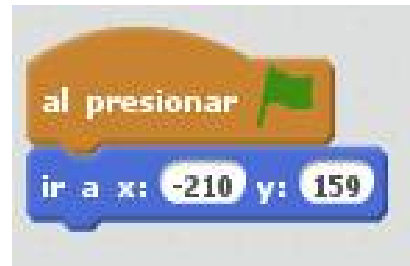
- Diseño del personaje y del escenario



Desarrollo iterativo e incremental

Desafío

- Estados iniciales



Desarrollo iterativo e incremental

Desafío

- Navegación interactiva



Desafío

- Rebotar en las paredes



Desarrollo iterativo e incremental

Desafío

- Establecer condición final del programa

