

Escuelas de la ZER El Moianès Llevant

- ❖ l'Estany
- ❖ Collsuspina
- ❖ Sant Quirze Safaja

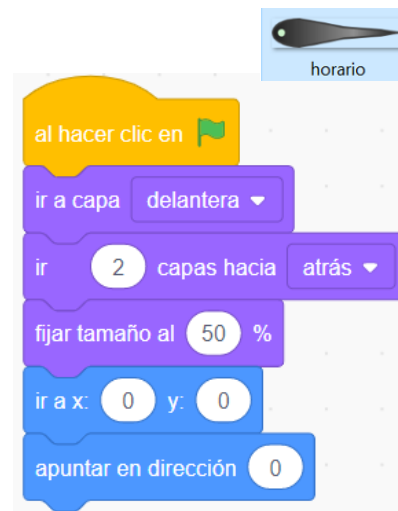
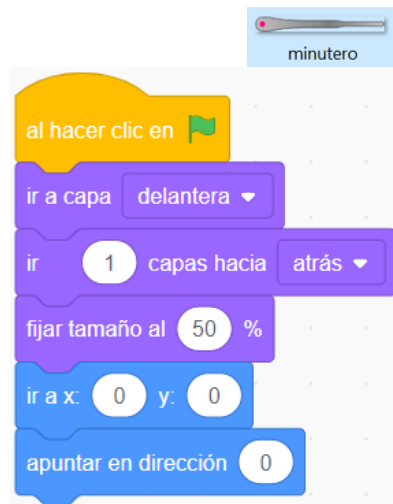
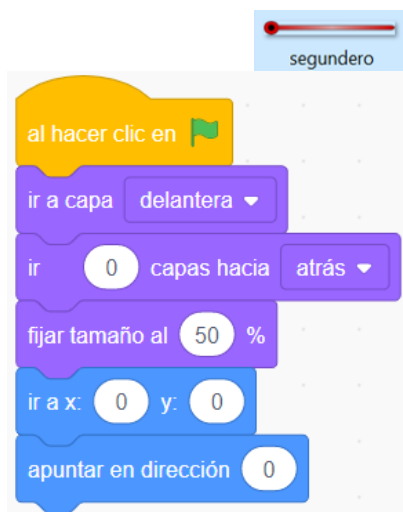
STEAM-CS

SCRATCH – Reloj con sincronización manual

Objetivo de la práctica

- Este trabajo permitirá al alumno resolver problemas en contenidos clave como:
 1. Cálculo
 2. Espacio y forma
 3. Medida
- Y lograr con la ayuda de las TIC:
 1. Situar figuras en un plano determinado por dos ejes de coordenadas
 2. Estudiar las unidades de tiempo horas, minutos y segundos
 3. Comparar y relacionar las medidas de ángulos y de indicadores de tiempo situadas en dos esferas
 4. Reconocimiento y construcción de ángulos a partir de giros
 5. Traducir un texto a expresión matemática para resolver un problema
 6. Utilizar variables para poner en hora el reloj
 7. Construcción de ángulos apuntando en una dirección (Una propiedad vectorial)

Situar las agujas en diferentes planos o capas



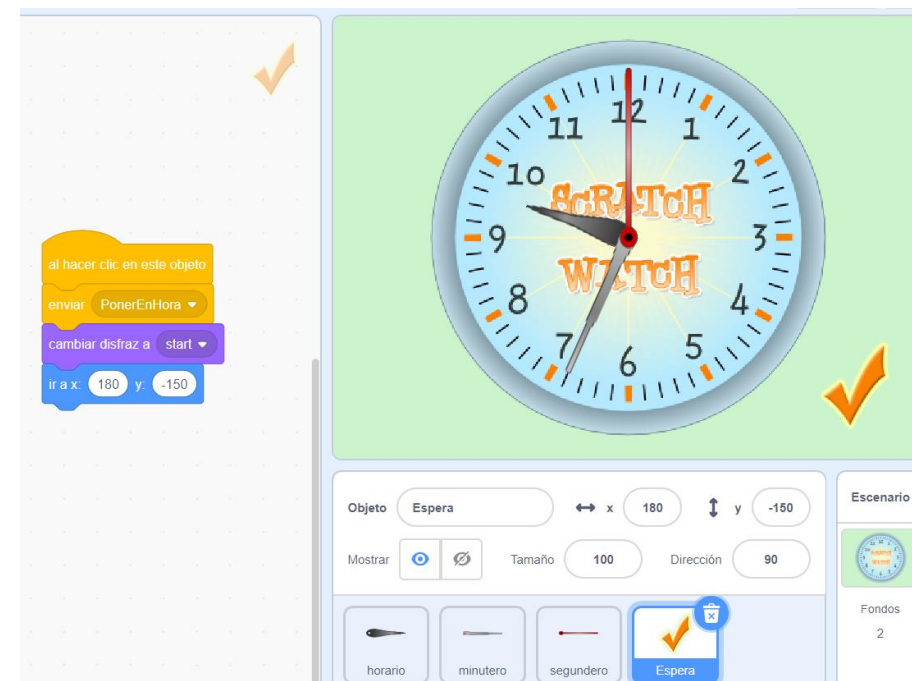
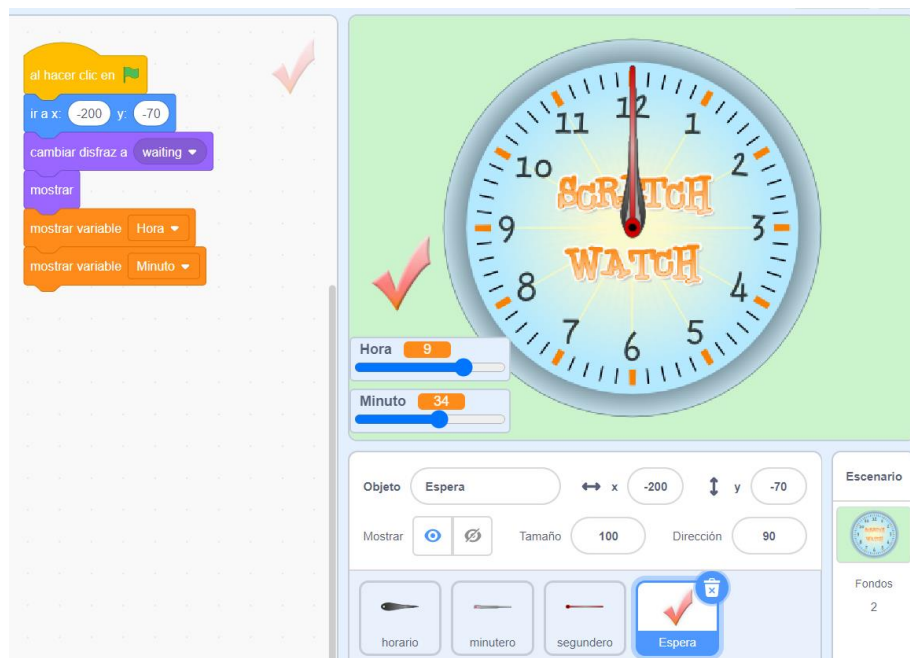
Introducir la hora actual al reloj

- Se crearán dos variables globales (Podrán ser leídas y escritas por cualquier sprite)

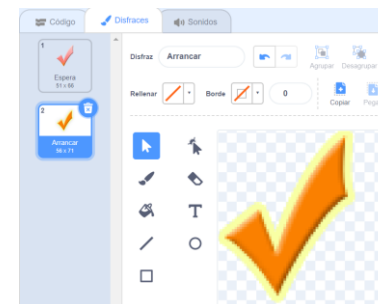
The image illustrates the process of creating global variables and configuring sliders in Scratch to display the current time. It is divided into several sections:

- Scratch Interface:** Shows the 'Variables' category in the left sidebar. The 'Hora' and 'Minuto' variables are checked. In the 'Control' section, two 'set to value' blocks are shown: 'set Hora to 0' and 'sumar a Hora 1'.
- Variable Creation Dialogs:** Two 'Nueva variable' (New Variable) dialog boxes are shown. The first is for 'Hora' with a range of 0 to 12. The second is for 'Minuto' with a range of 0 to 60. Both are set to 'Para todos los objetos' (For all objects).
- Slider Range Configuration Dialogs:** Two 'Cambiar rango del deslizador' (Change Slider Range) dialog boxes are shown. The first is for 'Hora' with a range of 0 to 12. The second is for 'Minuto' with a range of 0 to 60.
- Slider UI Elements:** Six slider controls are shown, representing the visual output of the configuration:
 - Hora slider with value 12
 - Hora slider with value 6
 - Hora slider with value 0
 - Minuto slider with value 60
 - Minuto slider with value 30
 - Minuto slider with value 0

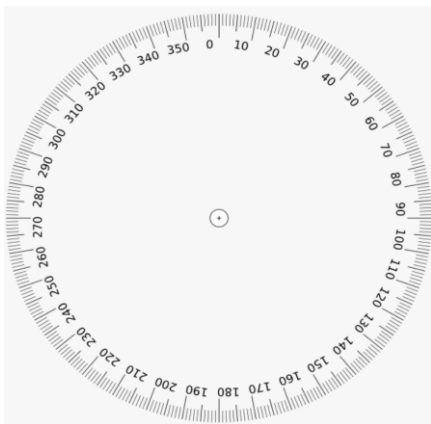
Algoritmo introducir hora y minuto actual



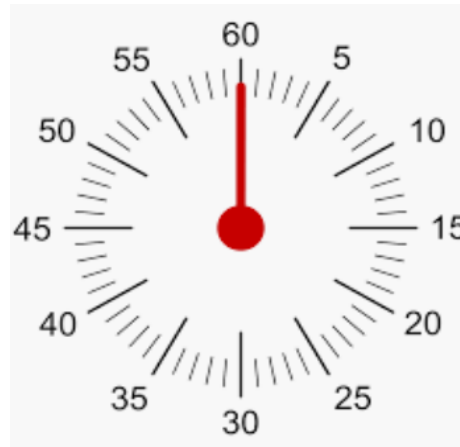
Previamente se habrá importado, subido, el objeto “stick” que tiene dos disfraces, el que representa “esperar” y el que representa “arrancar”



Giro segundos controlado por ángulos

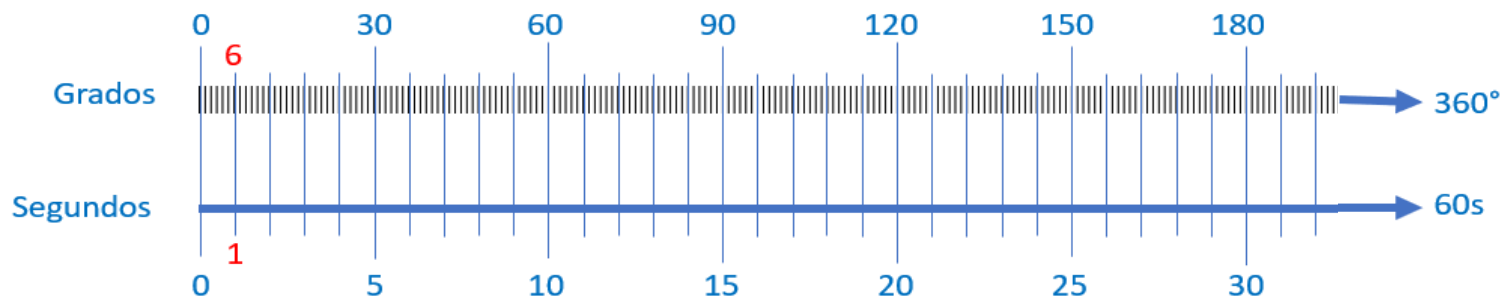


Un grado es cada una de las 360 partes en que se divide una circunferencia



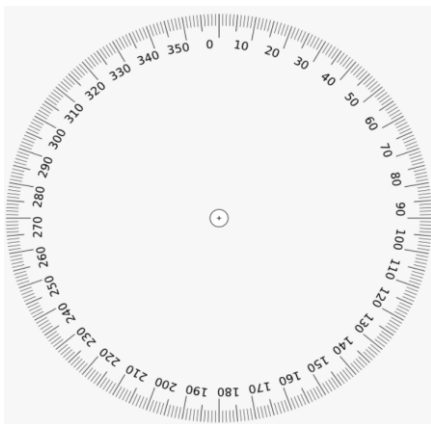
En un reloj analógico, un segundo o un minuto es cada una de las 60 partes en que se divide una circunferencia

Aproximación a un reloj analógico

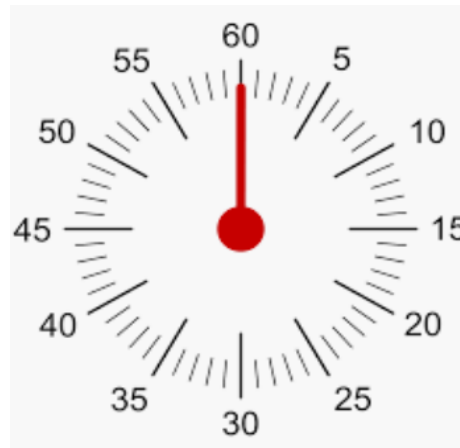


Repartir 360 grados entre 60 segundos $\rightarrow 360^\circ$ dividido $60\text{ s} \rightarrow \frac{360^\circ}{60\text{ s}} = 6^\circ/\text{s}$

Giro minutos controlado por ángulos

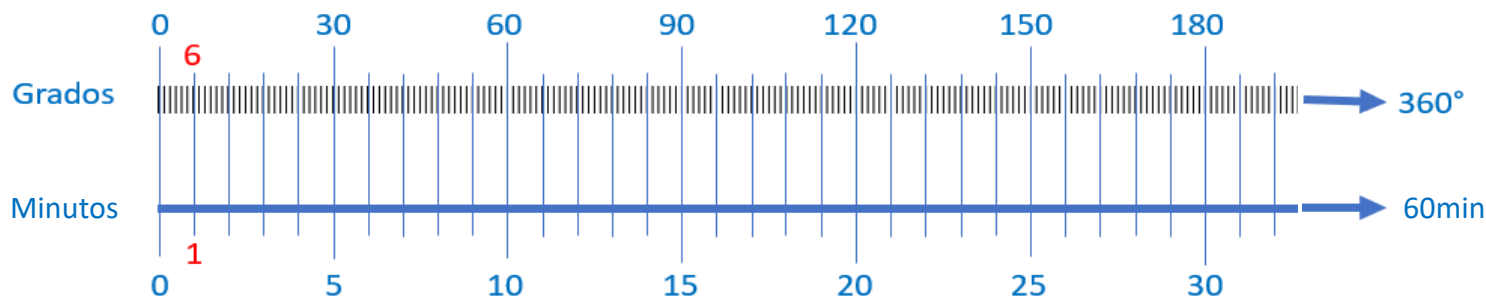


Un grado es cada una de las 360 partes en que se divide una circunferencia



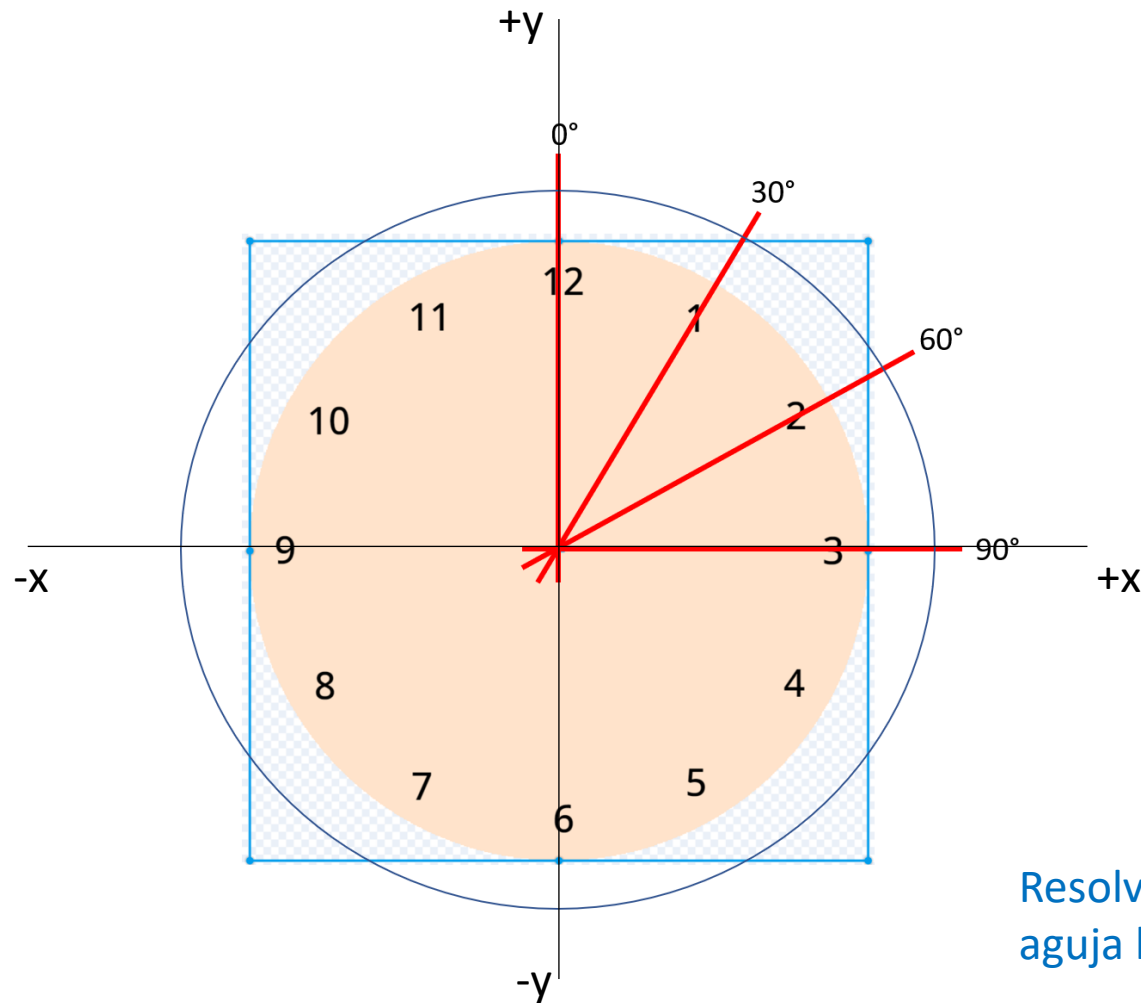
En un reloj analógico, un minuto es cada una de las 60 partes en que se divide una circunferencia

Aproximación a un reloj analógico



$$\text{Repartir 360 grados entre 60 minutos} \rightarrow 360^\circ \text{ dividido } 60 \text{ min} \rightarrow \frac{360}{60} \frac{^\circ}{\text{min}} = 6^\circ/\text{min}$$

Relación grados / horas



Respecto al eje vertical:

1 hora \rightarrow 30°

2 horas \rightarrow 60°

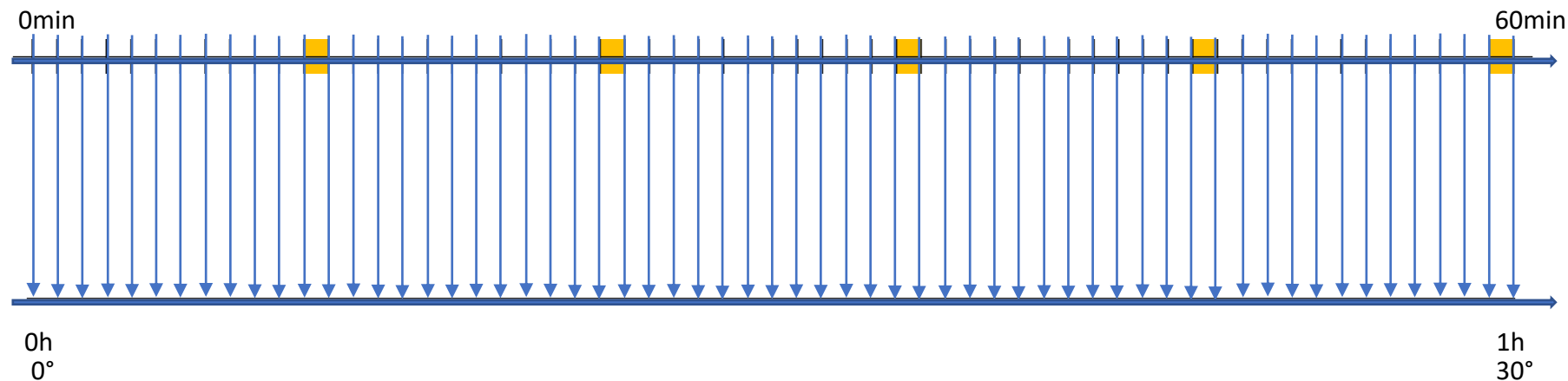
3 horas \rightarrow 90°

$$\text{Grados por hora} = \frac{360^\circ}{12 \text{ h}} = 30^\circ/\text{h}$$

Resolver: Cuantos grados tiene el ángulo que forma la aguja horaria cuando son las 6 horas en punto:

$$6 * 30^\circ = 180^\circ$$

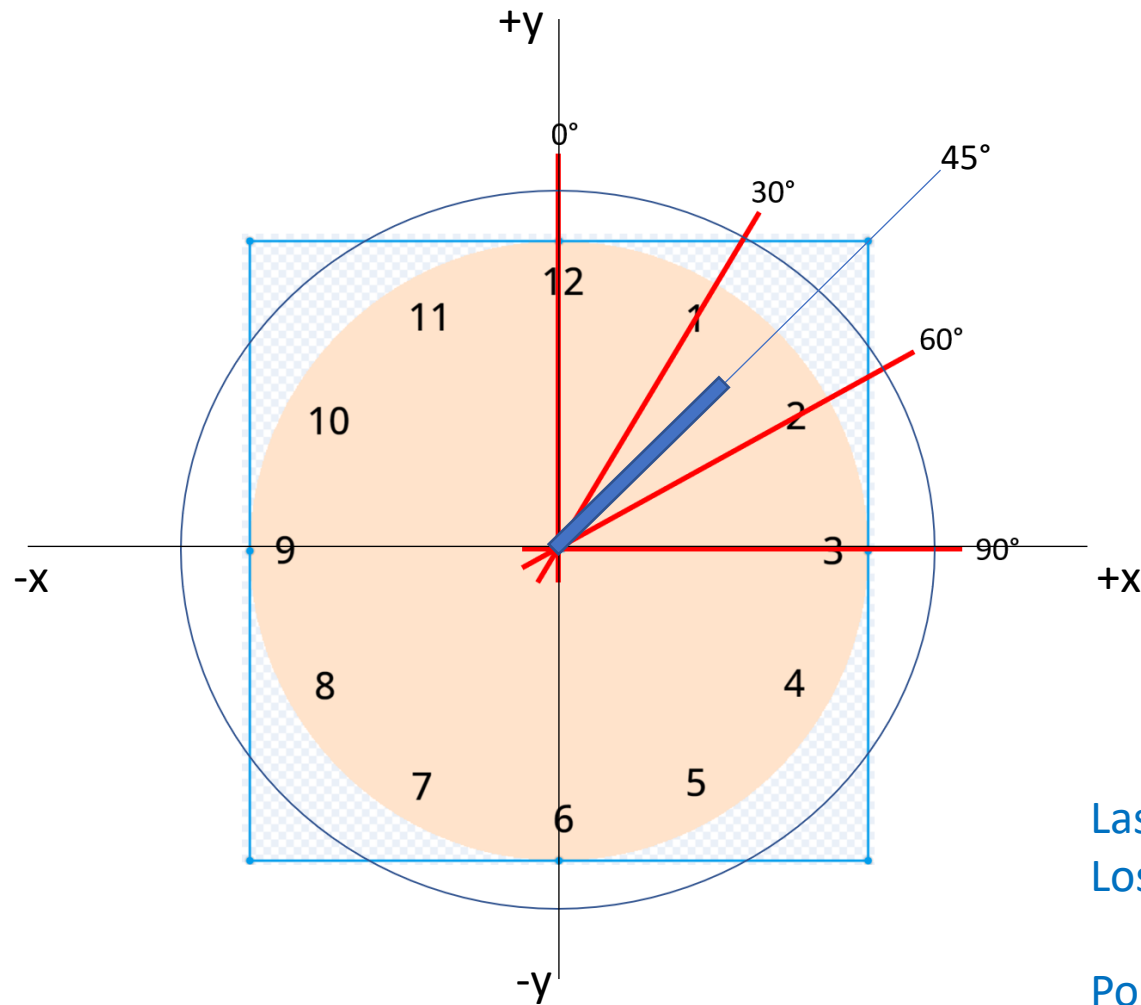
Relación minutos con los grados de la esfera horaria



¿Cuántos grados de la esfera horaria se corresponden con un minuto de tiempo?

$$¿ \text{Grados} / \text{minuto} ? = \frac{30^\circ}{60 \text{min}} = 0,5^\circ / 1 \text{min} = 0,5^\circ / \text{min}$$

Situar la aguja horaria a la 01h:30min

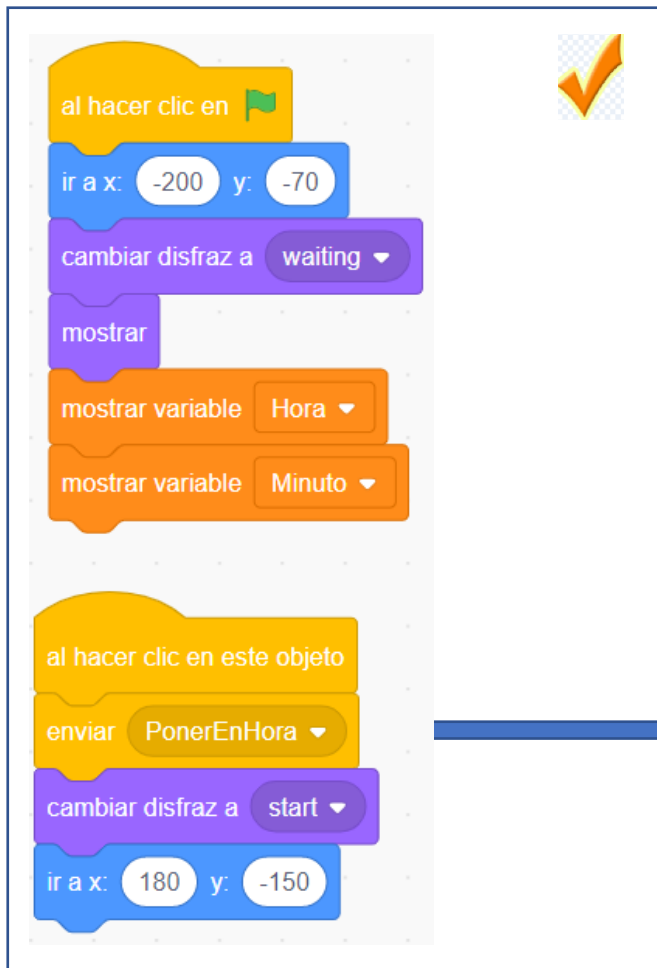



Las horas equivalen a: $1 * 30^\circ = 30^\circ$

Los minutos equivalen a: $30 * 0,5^\circ = 15^\circ$

Por lo tanto la aguja horaria estará en $30^\circ + 15^\circ = 45^\circ$

Algoritmo poner el reloj en hora



al hacer clic en 

ir a x: -200 y: -70

cambiar disfraz a waiting

mostrar

mostrar variable Hora


mostrar variable Minuto

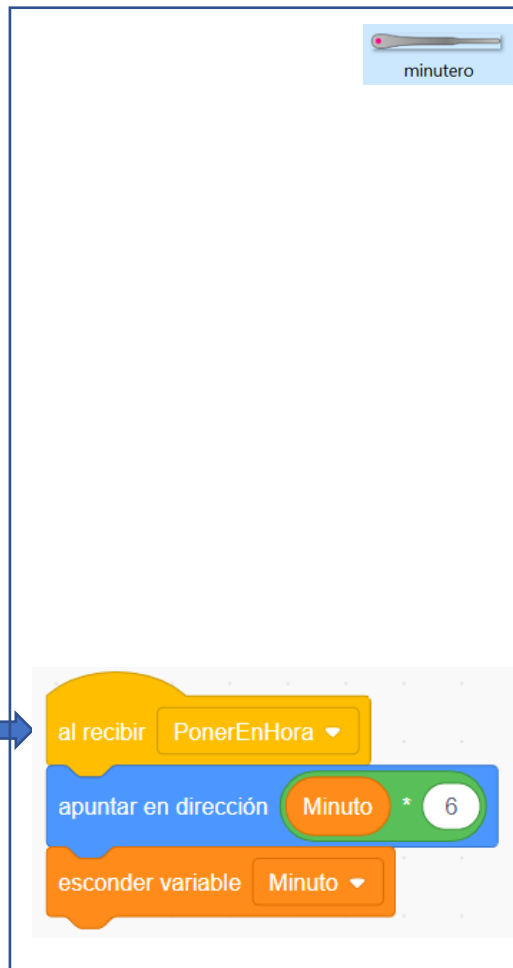
al hacer clic en este objeto

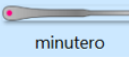
enviar PonerEnHora

cambiar disfraz a start

ir a x: 180 y: -150





 minutero

al recibir PonerEnHora

apuntar en dirección $\text{Minuto} * 6$

esconder variable Minuto



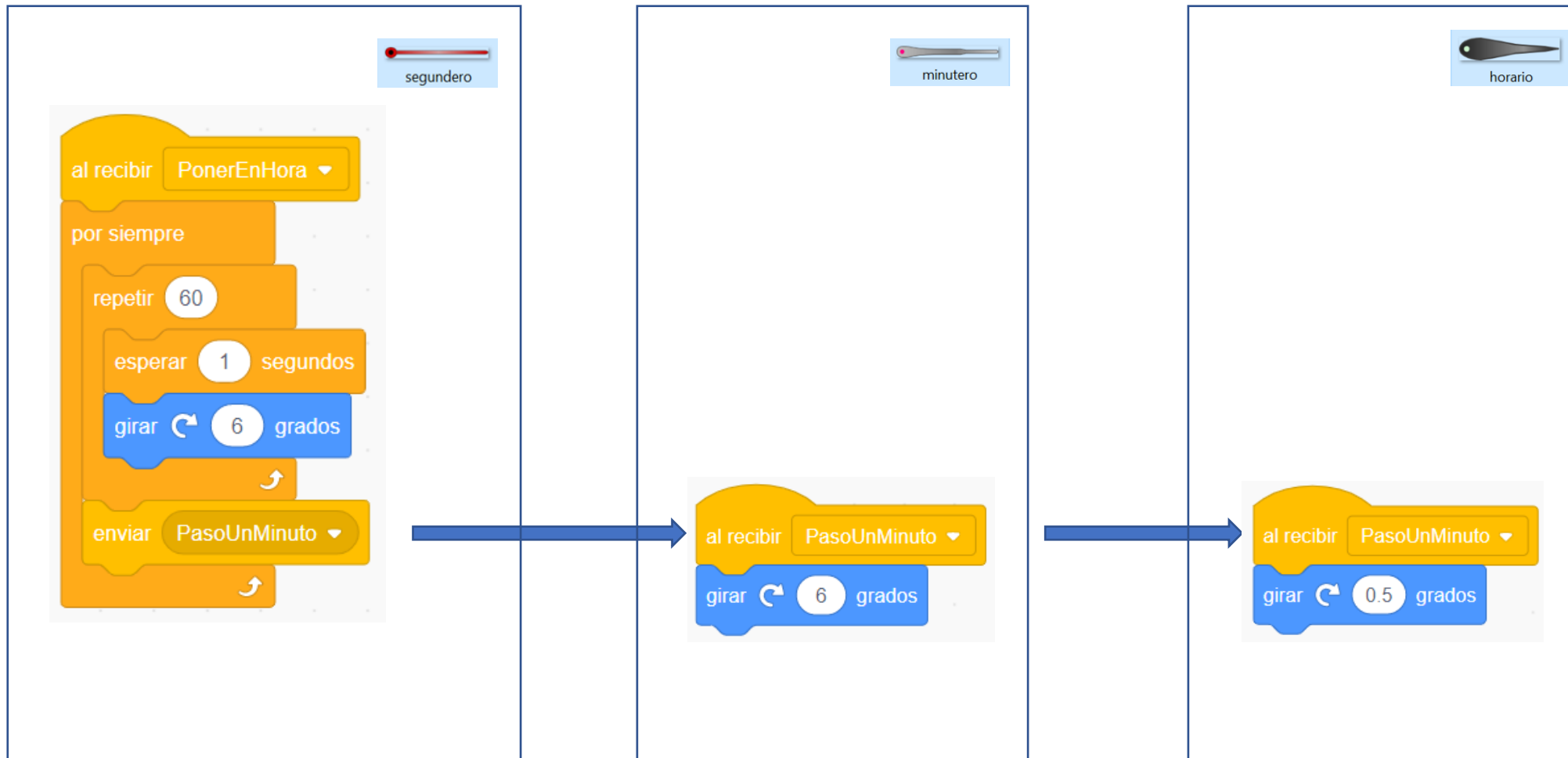
 horario

al recibir PonerEnHora

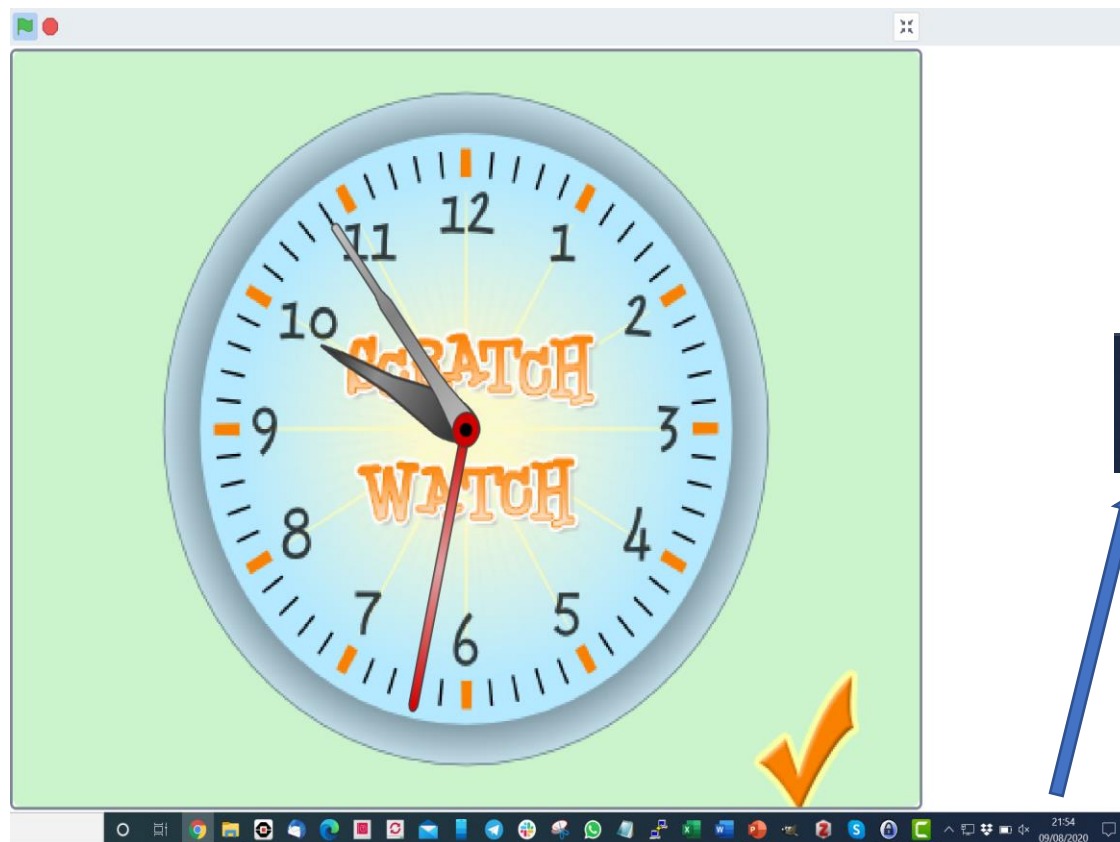
apuntar en dirección $\text{Hora} * 30 + \text{Minuto} * 0.5$

esconder variable Hora

Sincronizar la aguja de las horas por paso de mensajes



Comparar esta solución contra un reloj con sincronismo UTC (Coordinación de Tiempo Universal)



Pasado un tiempo...

Se observará que “nuestro” reloj atrasa respecto a la hora UTC. Los motivos son:

- El propio programa que gestiona el reloj introduce retrasos al gestionar otros bloques de código diferentes al bloque de espera 1 segundo.
- El programa si bien puede parecernos que está siempre en marcha no lo está, ya que el ordenador reparte su tiempo atendiendo a todos los programas que se están ejecutando en paralelo al programa del reloj

21:54
09/08/2020

Aspectos a mejorar

- No intervención humana para poner el reloj en hora
- Implementar una solución que mantenga el reloj siempre ajustado a la hora UTC (Coordinación de tiempo universal)
- Aprovechar el NTP (Network Time Protocol) que utilizan los sistemas operativos (Windows, Linux, Unix,...)

Créditos

- El reloj de fondo del escenario y las agujas de segundos, minutos y horas han sido obtenidas de:

➤ <https://scratchcodelab.blogspot.com/p/meca-1.html>

steam 4all

Este documento se ofrecen bajo licencia *Creative Commons Atribución/Reconocimiento, NoComercial, CompartirIgual 4.0* Licencia Pública Internacional — CC BY-NC-SA 4.0

