

# Programas interactivos

Cecilia Manzino

9 de abril de 2024

# ¿Qué es un programa interactivo?

- ▶ Programa que acepta entradas, datos u órdenes de personas u otros programas durante su ejecución.
- ▶ Ejemplos:
  - ▶ Procesador de texto
  - ▶ Programa de mensajería
  - ▶ Navegador
  - ▶ Juegos
  - ▶ etc

- ▶ La comunicación de un programa interactivo se realiza mediante **eventos**, los cuales pueden ser generados por:
  - ▶ **Personas**: a través del mouse, el teclado, pantalla táctil, etc.
  - ▶ **Otro programa**: reloj interno de la computadora (tick), arribo de mensaje, etc.

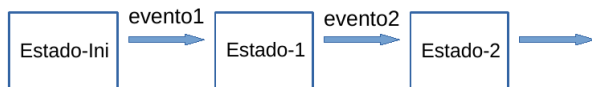
## Ejemplo 1

Escribiremos un programa que reaccione ante paso del tiempo.

” El programa mostrará en una escena vacía un círculo de diámetro 50 pixeles de color rojo. El círculo cambiará de color con cada 3 segundos, pasando por los colores amarillo, verde y de nuevo rojo.”

# Estado

- ▶ Para diseñar un programa interactivo el primer paso es identificar el **estado** del programa.
- ▶ El **estado** de un programa es el conjunto de **propiedades o valores** que cambian cuando ocurre un evento.



- ▶ Luego, tendremos que pensar cómo representar el estado, dar el diseño de datos.
- ▶ Y definir el estado el inicial.

- ; El estado del sistema será una cadena
- ; que representa el color del círculo.
- ; Posibles valores: "red", "green" y "yellow"
- ; Estado inicial  
(define INICIAL "red")

## Función que interpreta el estado

; interpretar: Estado -> Image

; dado un estado, devuelve la imagen a mostrar

```
(define (interpretar e)
  (place-image (circle 50 "solid" s)
               150
               150
               (empty-scene 300 300)))
```

## Expresión big-bang

```
(big-bang INICIAL ; estado inicial  
  [to-draw interpretar] ; evento obligatorio  
  [on-tick cambiar-color 3]))
```



; cambiar-color: Estado -> Estado

; dado un color, devuelve el siguiente color del círculo

```
(check-expect (cambiar-color "red") "yellow")  
(check-expect (cambiar-color "yellow") "green")  
(check-expect (cambiar-color "green") "red")
```

```
(define (cambiar-color c)  
  (cond [(string=? c "red") "yellow"]  
        [(string=? c "yellow") "green"]  
        [(string=? c "green") "red"])))
```