

## Máquina de estado finito

### ¿Qué es?

Una máquina de estado finito es como se comporta un sistema que puede cambiar de un estado a otro. Cuando cambia de un estado a otro se llama transición. Son muy útiles en sistemas reactivos y embebidos que encontramos en dispositivos electrónicos.

### Diagrama de Estado Finito o Diagrama de Transición de Estados

Como ejemplo, consideremos un muy simplificado sistema de control de un ascensor:



**Estados:** El sistema está formado por tres estados: DETENIDO, YENDO\_ARRIBA y YENDO\_ABAJO. Los diferentes estados se los representa mediante bloques cuadrados (como en este caso) o círculos.

**Transiciones:** Las transiciones se las representa mediante flechas que indican la dirección de transición de un estado a otro.

**Eventos:** Los eventos para el sistema en este ejemplo son los siguientes:

- seleccion\_piso: Es un evento externo que se genera toda vez que un usuario selecciona un piso o llama al ascensor desde otro piso.
- arribado\_nuevo\_piso: Es un evento interno que se genera cada vez que los sensores detectan que se ha arribado al nuevo piso seleccionado por el usuario.

**Condiciones de Transición:** Dos transiciones en este sistema de ejemplo tienen asociadas sus respectivas Condiciones de Transición. No todas las transiciones poseen Condiciones de Transición.

- $\text{piso\_nuevo} > \text{piso\_actual}$ : Es la condición necesaria para que se produzca una transición del estado DETENIDO al estado YENDO\_ARRIBA.
- $\text{piso\_nuevo} < \text{piso\_actual}$ : Es la condición necesaria para que se produzca una transición del estado DETENIDO al estado YENDO\_ABAJO.

Las Condiciones de Transición se anotan por debajo de las flechas de transición.

Una pseudo transición inicial del punto rojo al estado DETENIDO identifica a este último como el estado inicial de la MEF.

```

01.  /* Ejemplo de Máquina de Estado Finito para un ascensor simplificado */
02.  enum eEstados { // Los estados de la MEF se definen en una enumeración
03.      DETENIDO,
04.      YENDO_ARRIBA,
05.      YENDO_ABAJO
06.  } estado;
07.
08.  int piso_nuevo, piso_actual;
09.  main() {
10.      estado = DETENIDO; // La MEF se inicializa en estado DETENIDO
11.      while (TRUE) {
12.          switch (estado) {
13.              case DETENIDO: { //
14.                  // Verificar si se ha seleccionado un nuevo piso:
15.                  piso_nuevo = seleccion_piso();
16.                  if (piso_nuevo > piso_actual)
17.                      estado = YENDO_ARRIBA;
18.                  else if (piso_nuevo < piso_actual)
19.                      estado = YENDO_ABAJO;
20.              } break;
21.              case YENDO_ARRIBA: {
22.                  piso_actual = comprobacion_piso();
23.                  if (piso_nuevo == piso_actual)
24.                      estado = DETENIDO;
25.              } break;
26.              case YENDO_ABAJO: {
27.                  piso_actual = comprobacion_piso();
28.                  if (piso_nuevo == piso_actual)
29.                      estado = DETENIDO;
30.              } break;
31.          }
32.      }
33.  }

```

### ¿Cómo lo implementamos en nuestro proyecto de RPG?

En un combate el personaje tiene 4 estados:

- Ataque
- Defensa
- Contraataque
- Estado normal (que se quede quieto)

#### **Transiciones:**

Si un personaje decide atacar, pasa al estado de Ataque.

Si un personaje decide defenderse, pasa al estado de Defensa.

Si un personaje recibe un ataque mientras está en estado de Ataque, pasa al estado de Contraataque.

Si un personaje decide no hacer nada, permanece en su estado normal.

Solamente sería implementar esta lógica en el programa para que podamos tener un combate más chido.