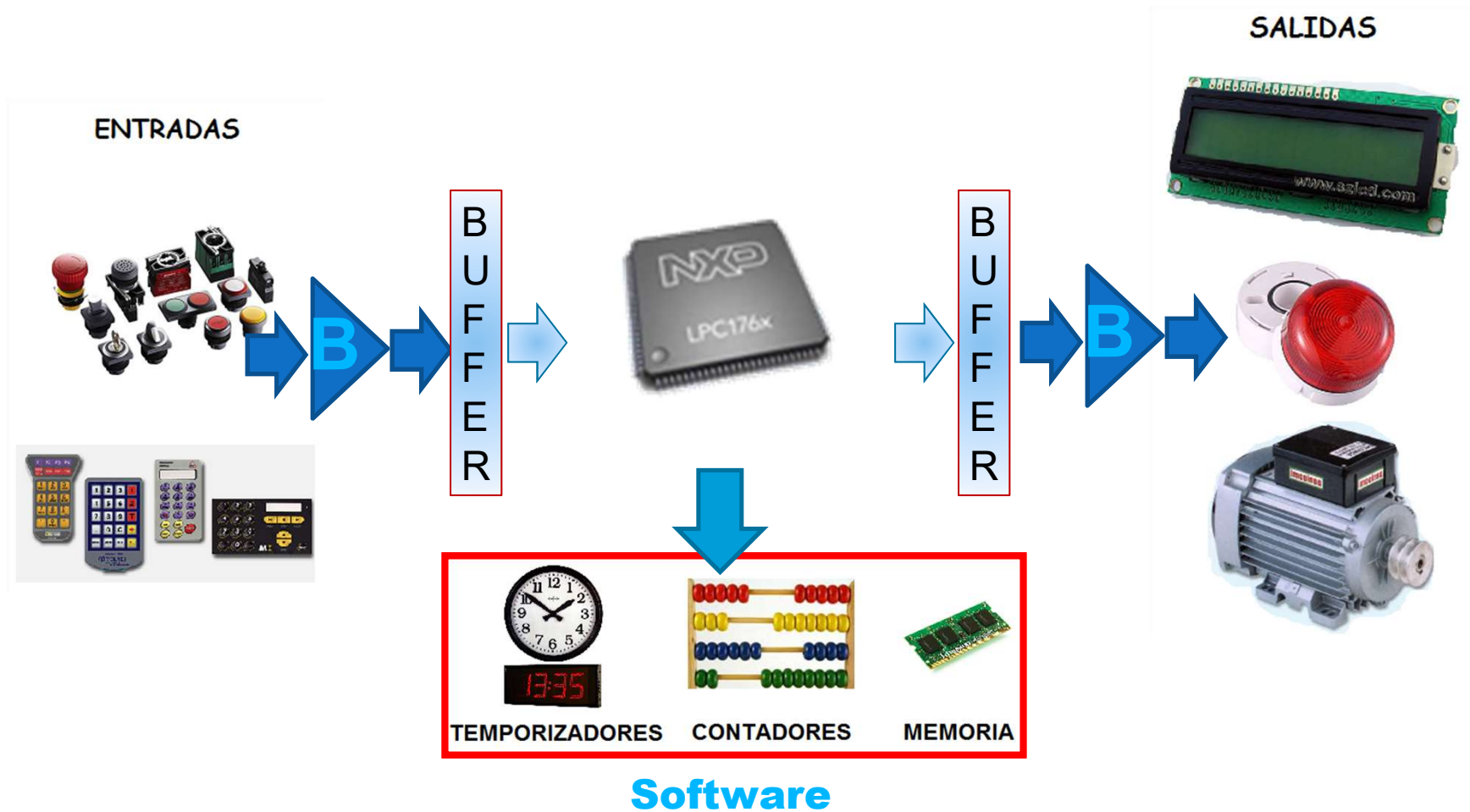


Diplomatura Arduino

Maquina de Estados



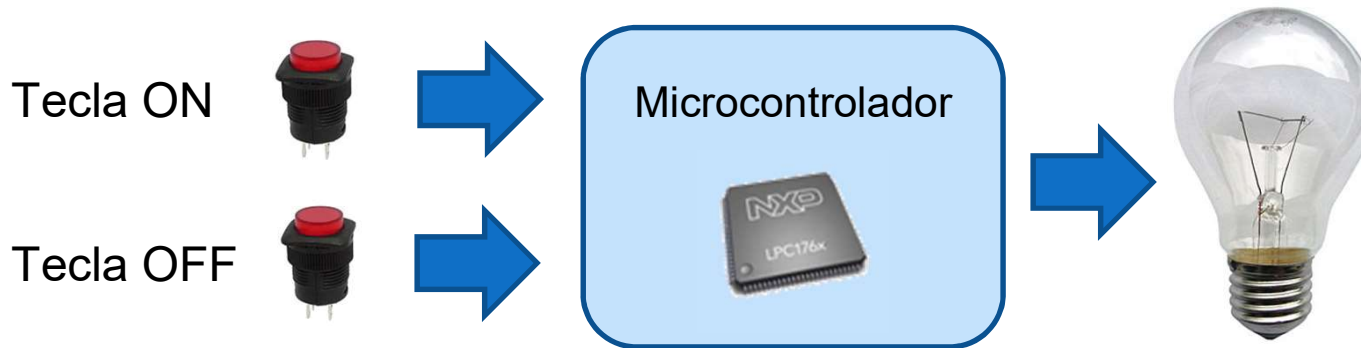
De donde partimos...



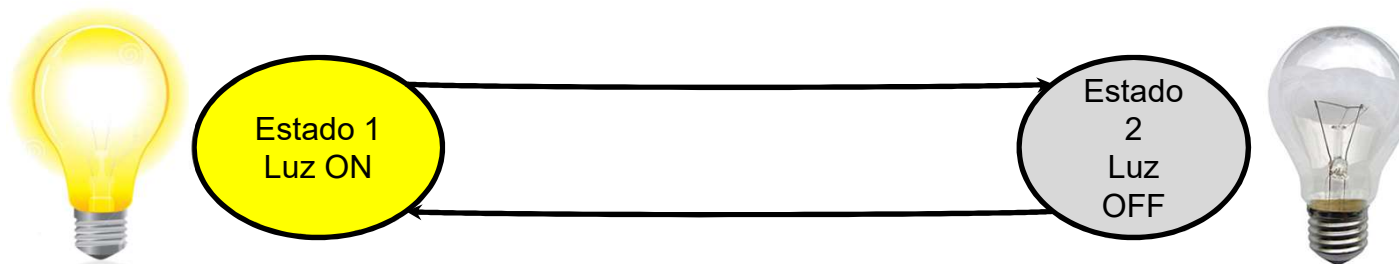
Maquina de Estados

Vamos a realizar un primer ejemplo:

Encendido y apagado de una luz cada vez que presionan unas teclas



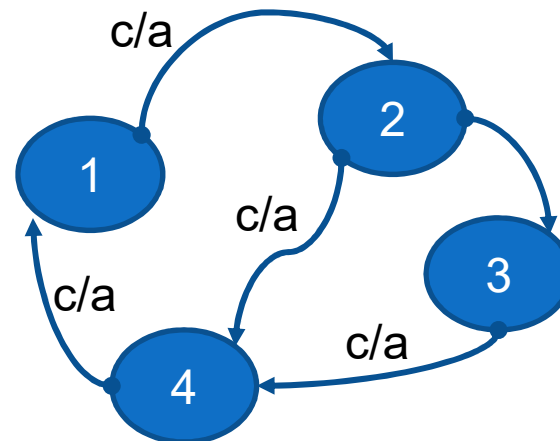
Tecla OFF presionada / Apagar Luz



Tecla ON presionada / Encender Luz

Maquina de Estados

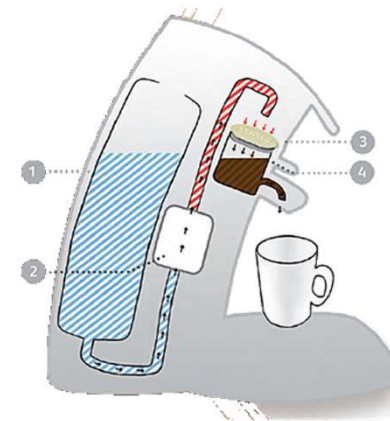
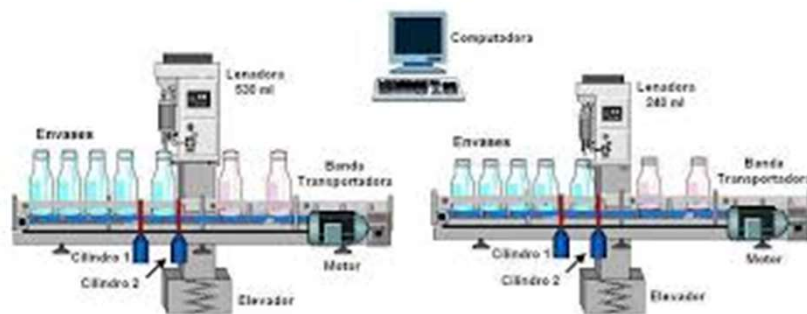
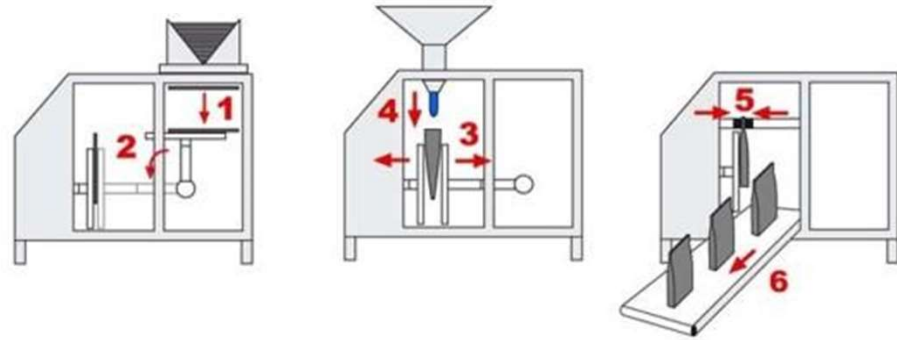
Es un MODELO que describe a un *sistema* mediante:



- **Estados** (comportamiento estático), hay un numero finito de estados posibles.
- **Transiciones** (cambios entre estados)
- **Excitaciones** (que condicionan las transiciones)
- **Acciones** (asociadas a las transiciones)

Maquina de Estados

¿Qué podemos resolver?



Maquina de Estados

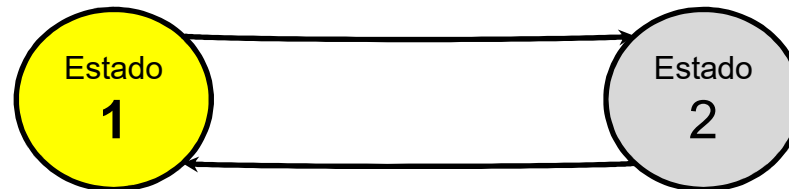
¿ Porque usarlas ?

- Brinda una rigurosa descripción del comportamiento del sistema
- Ofrece sobre situaciones muy complejas soluciones sencillas. Incluso compuestas por la operación en paralelo de múltiples máquinas
- Su implementación genera código simple, eficiente y preciso.
- Fácil de depurar, modificar, expandir y bien organizado.
- Sin duda facilitan el análisis, el diseño y la implementación

Maquina de Estados

Vamos a comenzar a codificar!! Comencemos por cada estado en particular

Tecla OFF presionada / **Apagar Luz**



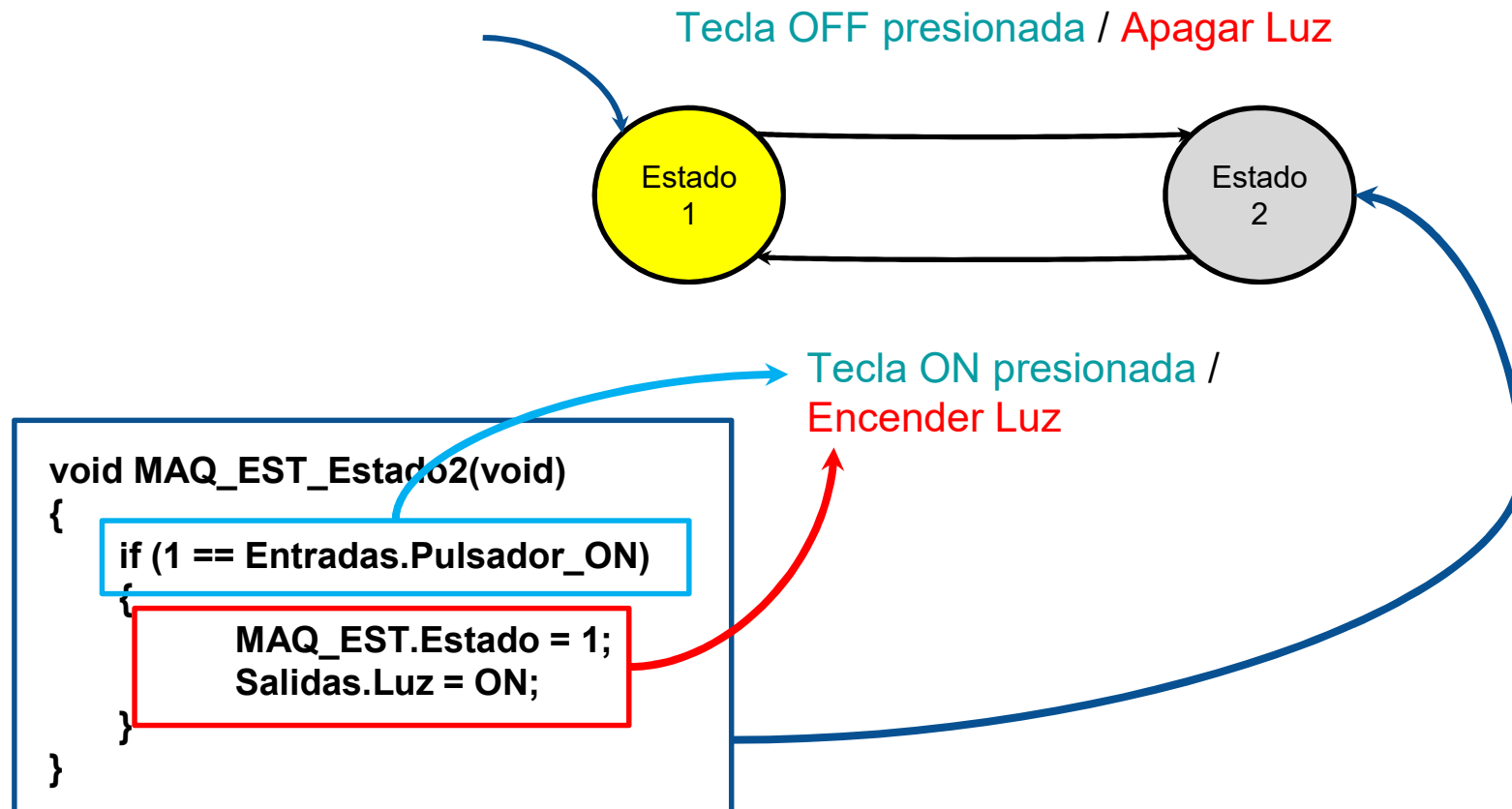
Tecla ON presionada /
Encender Luz

Truco!
Por cada
estado
vamos a
crear una
función

Re-truco!
Ahora por
cada "flecha"
que sale del
estado es un
"if"

Maquina de Estados

Analicemos nuevamente lo anterior con mas detalle!



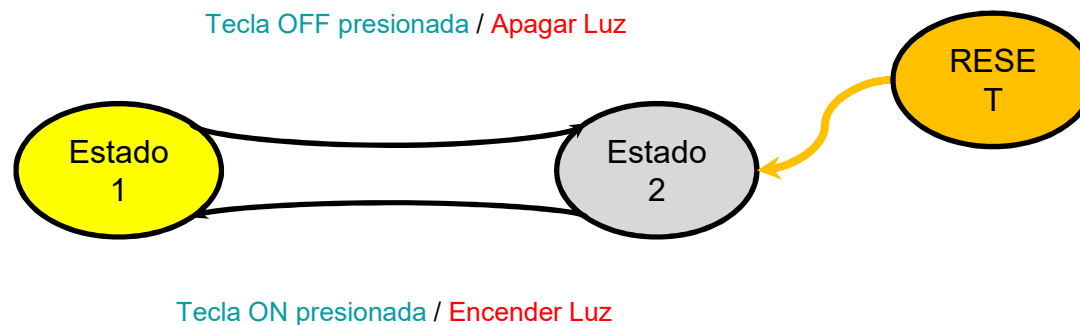
Maquina de Estados

*Sigamos con la inicialización de nuestra maquina de estados....
Pero primero que es inicialización?*

```
void MAQ_EST_Inicializar(void)
{
    MAQ_EST.Estado = 2;
    Salidas.Luz = OFF;
    return;
}
```



Nos referimos al estado de inicialización al estado donde cargamos a todas las variables para que el sistema arranque inicialmente.



Maquina de Estados

Ahora vamos a comenzar a vincular los estados

```
typedef struct
{
    uint8_t Estado;                // Estado de Contador
}STRUCT_MAQ_EST;

STRUCT_MAQ_EST MAQ_EST;          // Estructura principal de la maquina de estado

void main (void)
{
    MAQ_EST_Inicializar();         // Llamado a la función de inicialización

    while(1)
    {
        MAQ_EST_Analisis();       // Llamado a la función de análisis
    }
}
```

Maquina de Estados

Para implementar el análisis de la maquina de estados tengo varios métodos:

- Switch's
- Múltiples if
- Punteros a función

```
void MAQ_EST_Analisis (void)
{
    switch (estado)
    {
        case 1:
            MAQ_EST_Estado1();
            break;
        case 2:
            MAQ_EST_Estado2();
            break;
    }
}
```

```
void MAQ_EST_Analisis (void)
{
    if(1 == MAQ_EST.Estado)
    {
        MAQ_EST_Estado1();
    }
    else
    {
        if(2 == MAQ_EST.Estado)
            MAQ_EST_Estado2();
    }
}
```

Maquina de Estados

Con un vector de punteros a función

```
const void (*MAQ_EST_ArrayFunciones [ ]) (void) = {    MAQ_EST_Estado1,  
                                                         MAQ_EST_Estado2};
```

```
void main (void)  
{  
    MAQ_EST_Inicializar();  
  
    while(1)  
    {  
        (*MAQ_EST_ArrayFunciones [MAQ_EST.Estado]) ();  
    }  
}
```

Maquina de Estados

Sigamos avanzando sobre el tema.

Podemos manejar dos maquinas de estado al mismo tiempo?



```
void main (void)
{
    MAQ_EST1_Inicializar();
    MAQ_EST2_Inicializar();

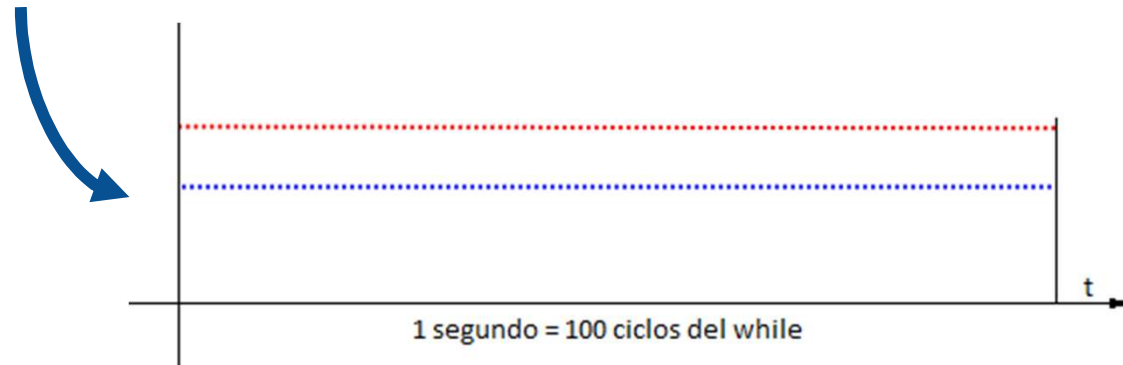
    while(1)
    {
        MAQ_EST1_Analisis();
        MAQ_EST2_Analisis();
    }
}
```



Maquina de Estados



*Estipulemos que el tiempo de cada maquina de estado es de aproximadamente 5ms.
(solo para entender el concepto que vamos a explicar)*

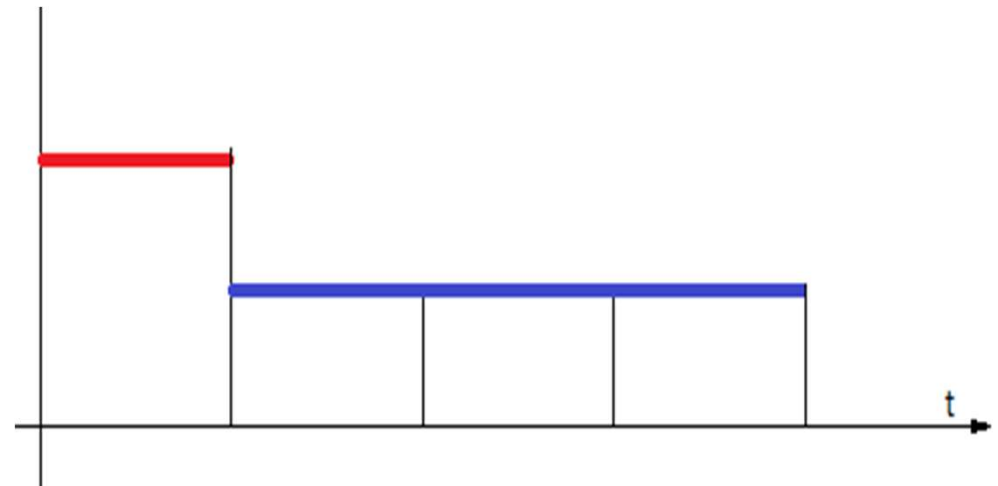


Los dos procesos están corriendo en "paralelo" desde el punto de vista del usuario, ya que no ve la conmutación de cada una de las maquinas de estado.

Maquina de Estados

```
void main (void)
{
    MAQ_EST1_Inicializar();
    MAQ_EST2_Inicializar();

    while(1)
    {
        MAQ_EST1_Analisis();
        MAQ_EST2_Analisis();
    }
}
```



*Es fundamental que ninguna de las maquinas de estado sea **bloqueante**, ya que esto impide la ejecución de las demás.*

*También no se pueden realizar **demoras** o **esperas** asincrónicas.*

Maquina de Estados

Ahora hay un máximo de maquinas de estado?



```
void main (void)
{
    MAQ_EST1_Inicializar();
    MAQ_EST2_Inicializar();

    while(1)
    {
        MAQ_EST1_Analisis();
        MAQ_EST2_Analisis();
    }
}
```



Maquina de Estados

Esto es todo lo que se puede hacer con maquinas de estados?

Y si le agregamos un par de cositas al vector de punteros a función....

```
const void (*MAQ_EST_ArrayFunciones [ ]) (void) = {    MAQ_EST_Estado1,
                                                         MAQ_EST_Estado2};
```

```
void main (void)
{
    MAQ_EST_Inicializar();

    while(1)
    {
        if(MAQ_EST_ArrayFunciones [MAQ_EST.Estado] != 0)
            (*MAQ_EST_ArrayFunciones [MAQ_EST.Estado]) ();
    }
}
```

Maquina de Estados

Hablar sobre maquinas de estados en sistemas operativos

Como ejemplo, en *Windows 95/98*, los desplazamientos de tiempo se establecieron en 20 milisegundos, lo que suponía 50 cambios de hilo en cada segundo. Este intervalo era lo suficientemente corto para que el usuario tuviese la impresión de una ejecución paralela y lo suficientemente largo para que cada hilo avanzase bastante en su intervalo asignado de ejecución.

La multitarea puede realizarse de dos maneras diferentes:

Cooperativa (cooperative multitasking)
Preventiva o preferente (Preemptive multitasking)