

# Tema 6

# SISTEMAS OPERATIVOS

Gestión de Entrada/Salida

---

# Introducción

---

## Sistema Operativo

### Concepto

Software que controla la ejecución de los programas de aplicación y que actúa como interfaz entre el usuario y el hardware del computador.

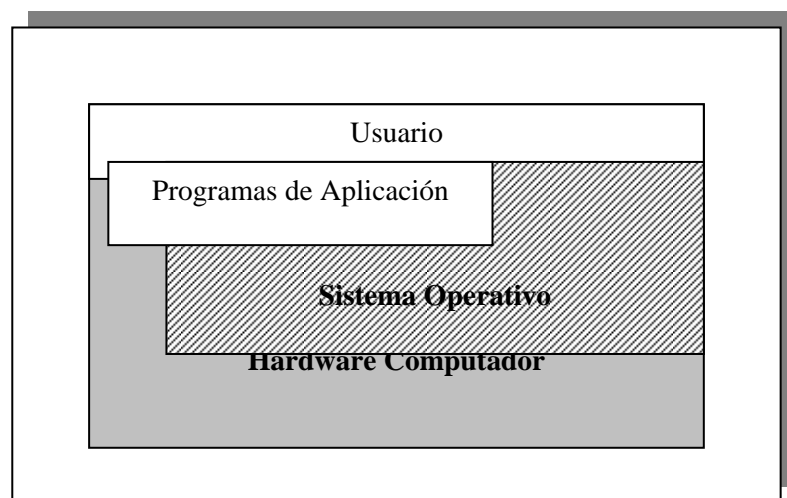
### Objetivos

Comodidad. Un Sistema Operativo (SO) hace que un computador sea más fácil de utilizar.

Eficiencia. Un SO permite que los recursos del computador se aprovechen de una manera más eficiente.

Capacidad de evolución. Un SO debe construirse de modo que permita el desarrollo de nuevas funciones.

### Programa interfaz



## Servicios del Sistema Operativo

Creación de programas.  
Ejecución de programas.  
Gestión de Entrada/Salida.  
Gestión del Sistema de Archivos.  
Acceso al sistema.  
Detección y respuesta de errores.  
Contabilidad.

## Gestión de Entrada/Salida

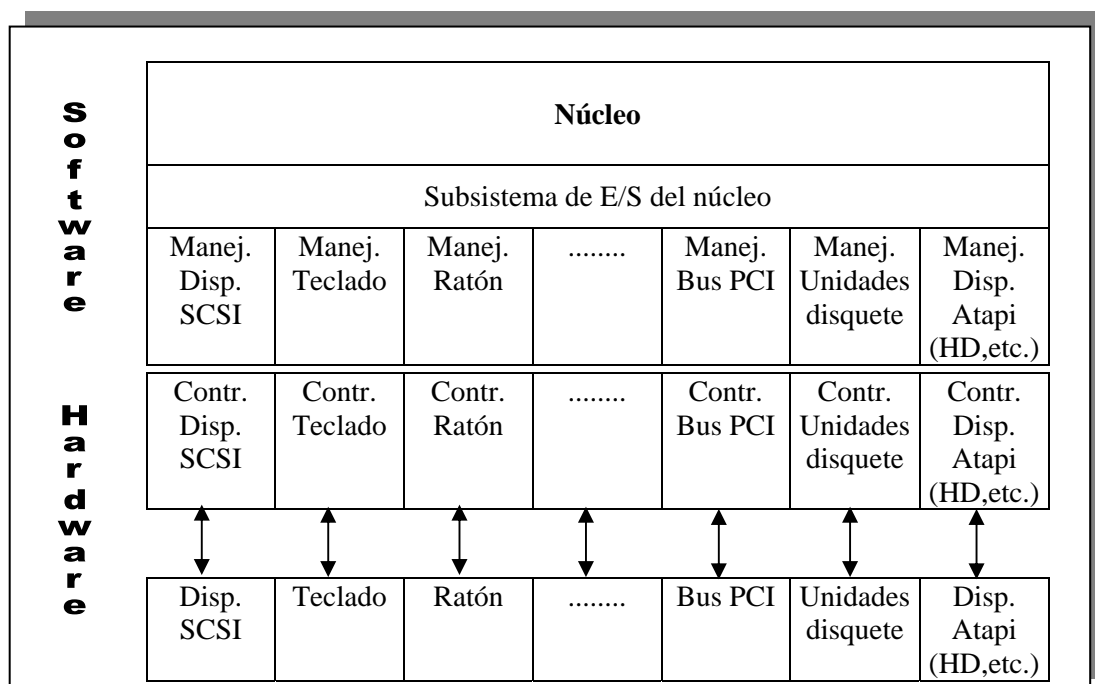
Parte del SO encargada de la gestión de los dispositivos de E/S que actúa como interfaz entre los usuarios del sistema y los dispositivos de E/S, de forma que en las capas superiores los dispositivos se tratan de una manera uniforme, sencilla, segura y eficaz (**Independencia de Dispositivo**).

## Sistema de Entrada/Salida

Se divide en:

Hardware de E/S. Formado por el dispositivo físico de E/S que es la parte electromecánica y el controlador del dispositivo que es la parte electrónica.

Software de E/S. Formado por el software que permite la comunicación con el controlador del dispositivo.



# Hardware de E/S

---

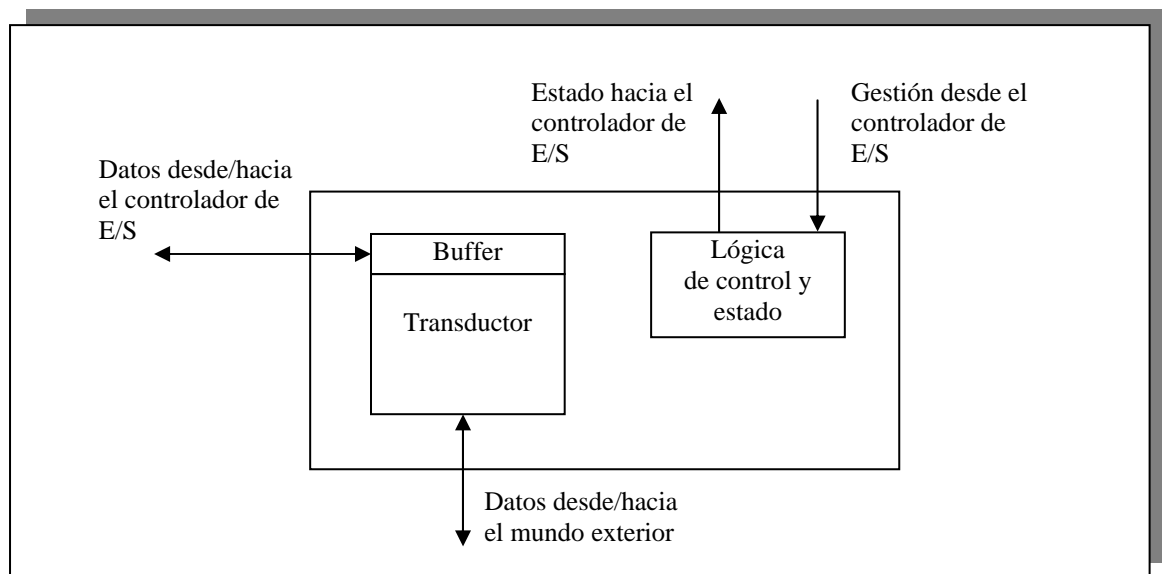
## Dispositivo de E/S

### Concepto

Un dispositivo de E/S o dispositivo externo está formado básicamente por los siguientes elementos:

- Una lógica de control para gobernar su funcionamiento.
- Un transductor.
- Una pequeña memoria (registro/buffer) para almacenar temporalmente los datos que transfiere.
- Los datos que transfiere constituidos por un conjunto de bits que se envían o reciben desde el controlador de E/S.
- Señales de control que determinan la función que realiza el dispositivo. Estas señales pueden enviar el dato al controlador de E/S (leer), aceptar el dato del controlador de E/S (escribir), realizar alguna función de control particular para ese dispositivo como posicionar la cabeza de una unidad de disco, etc.
- Señales de estado que indiquen el estado en que se encuentra el dispositivo.

### Estructura



## Clasificación general

Adaptados al usuario. Son apropiados para comunicar información al usuario. Por ejemplo, las consolas, los terminales de vídeo, las impresoras, etc.

Adaptados a la máquina. Permiten comunicarse con el sistema. A este tipo pertenecen las unidades de almacenamiento secundario como por ejemplo, las unidades de disco, las unidades de cinta magnética, etc.

De comunicación. Permiten la transferencia de información entre dispositivos remotos. Estos dispositivos pueden estar adaptados al usuario, a la máquina o incluso ser otro computador.

## Clasificación desde el punto de vista de programación

Dispositivos de bloques. Organizan la información en bloques direccionables de tamaño fijo. Ej.: discos, cdrom, etc.

Dispositivos de caracteres. La unidad de trasiego es el carácter, en función del cual no se establece ninguna organización. Ej.: terminales, cintas, etc.

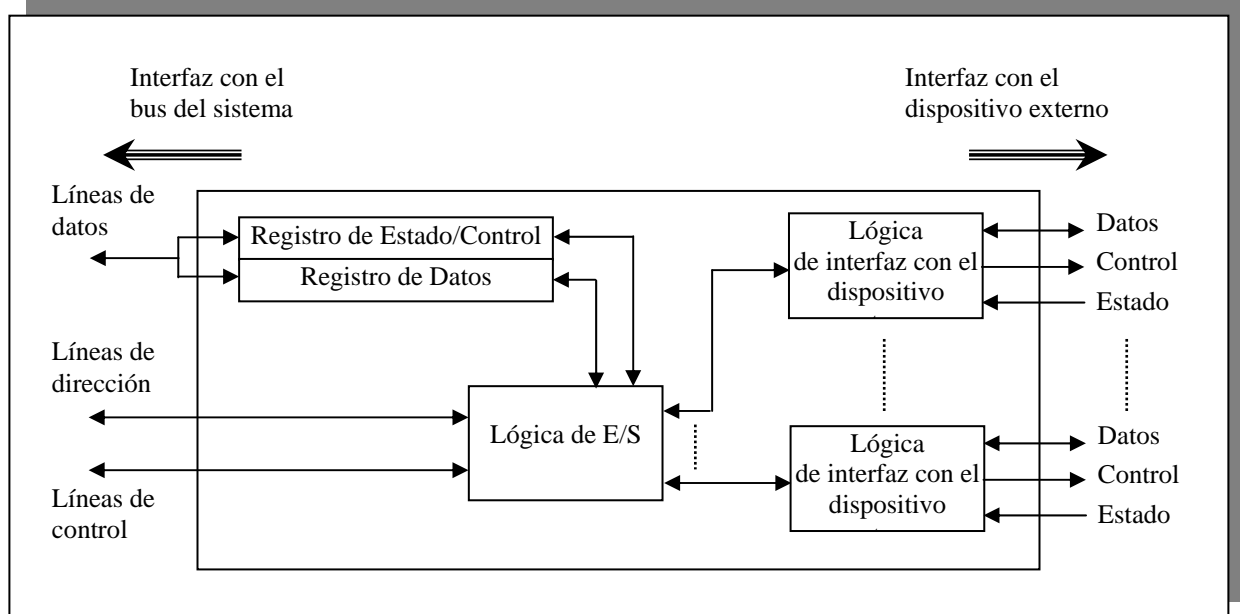
Algunos no admiten esta clasificación. Ej.: Reloj. No tiene direcciones de bloque ni aceptan ni generan trasiego de información.

## Controlador de E/S

### Concepto

Elemento encargado del control de uno o más dispositivos externos y del intercambio de datos entre dichos periféricos con la memoria principal o con la CPU.

## Estructura



## Funciones

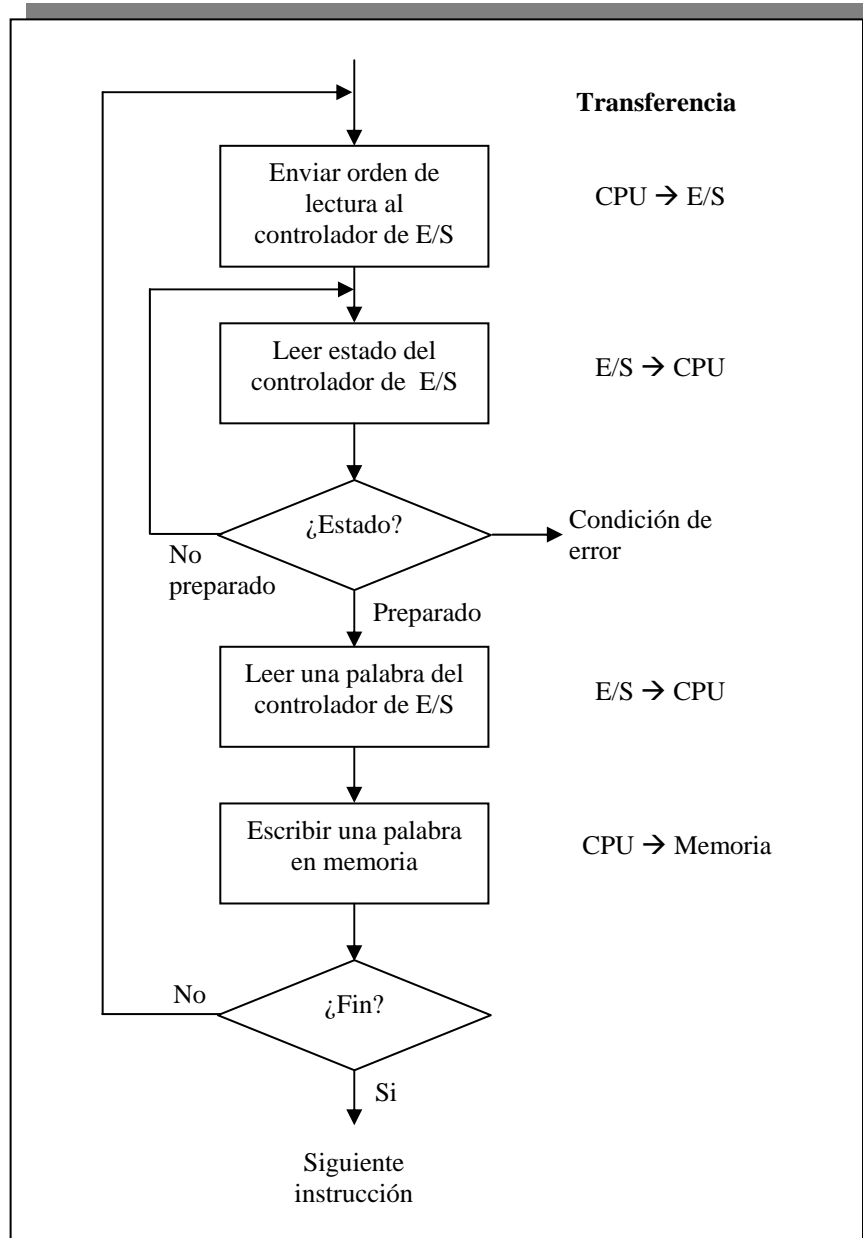
- Comunicación con la CPU
- Comunicación con el dispositivo externo
- Almacenamiento temporal de datos
- Control y temporización
- Detección de errores

## Técnicas de Transferencia de E/S

Existen diferentes métodos para copiar la información desde un dispositivo externo a la memoria o desde la memoria a un dispositivo externo.

### Entrada/Salida controlada por programa

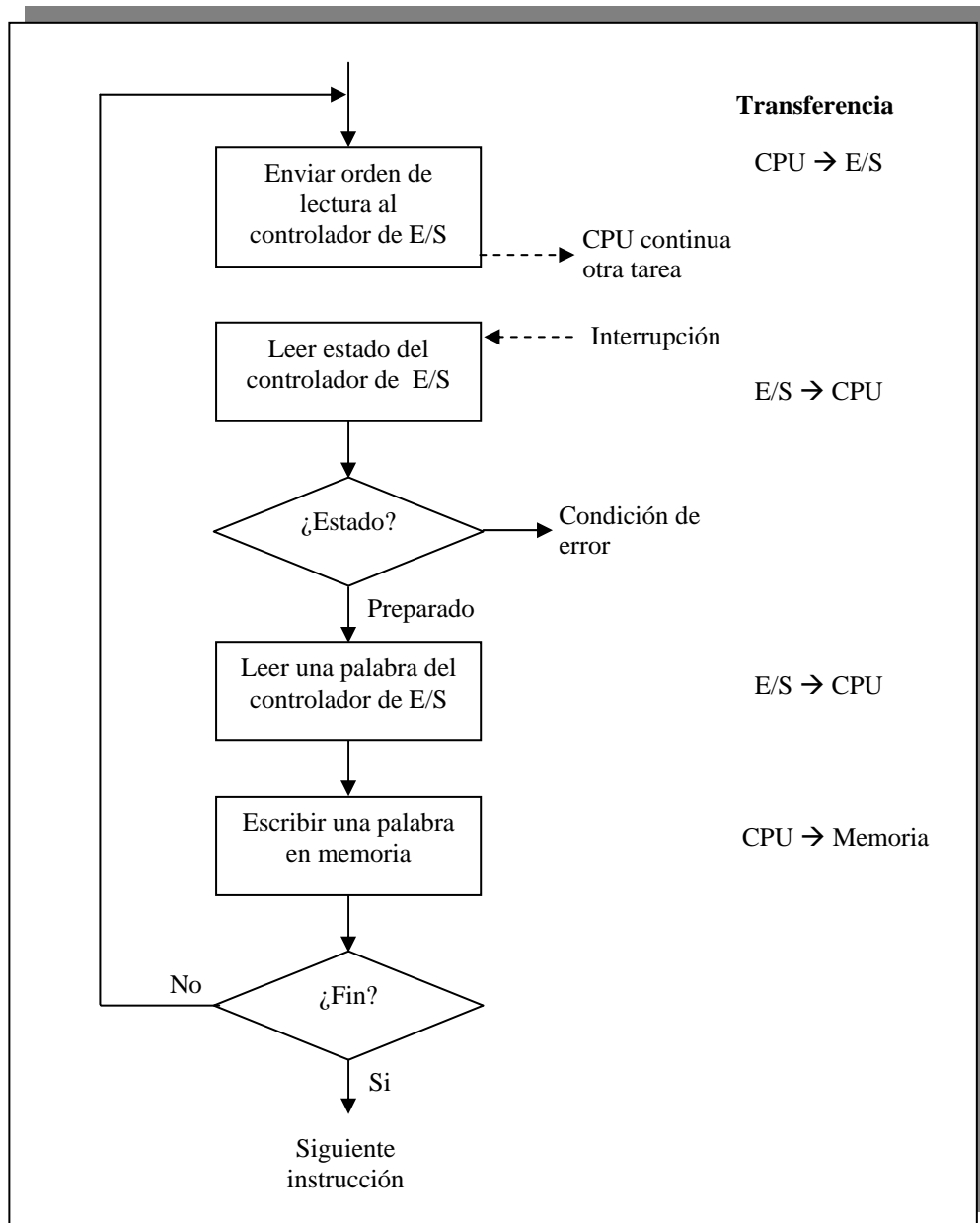
#### **Lectura de un bloque de datos**



Desventaja. El procesador no realiza ningún trabajo útil mientras permanece en el bucle de espera, por lo que el rendimiento del sistema disminuye.

## Entrada/Salida mediante interrupciones

### **Lectura de un bloque de datos**



Ventaja: La CPU ya no espera como en la entrada/salida controlada por programa.

Desventaja. La CPU emplea tiempo en transferir datos entre el dispositivo externo y la memoria.



## Entrada/Salida mediante Acceso Directo a Memoria

### **Controlador de DMA**

Permite transferir información sin que la CPU espere y sin que la CPU pierda el tiempo copiando los datos a memoria.

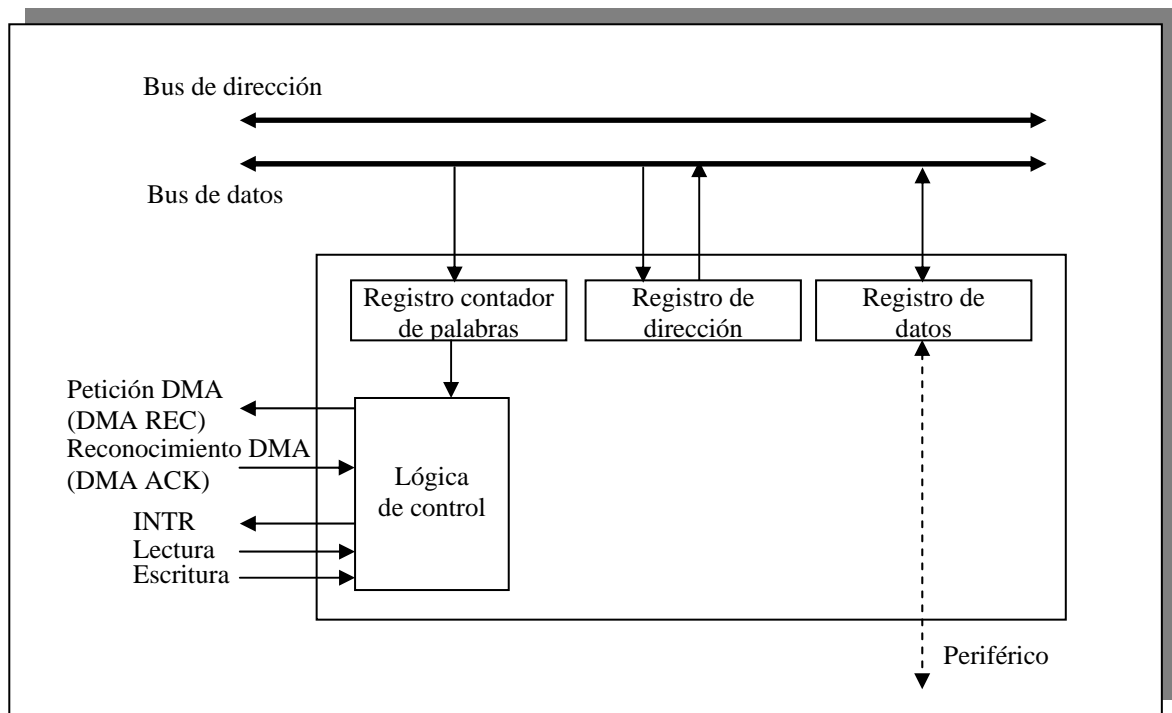


Diagrama de bloques de un controlador de DMA.

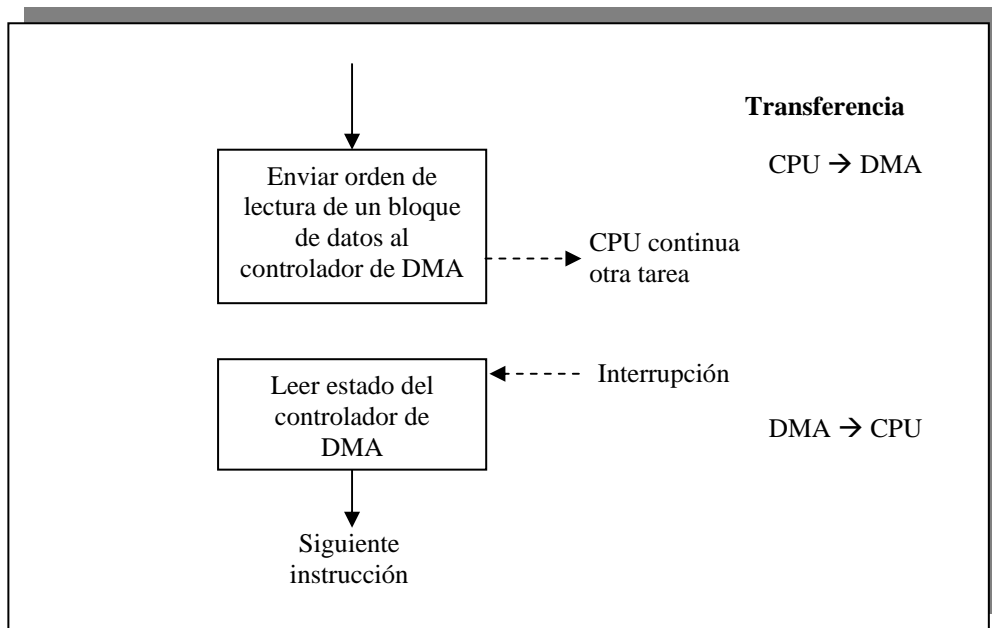
El controlador de DMA (Direct Memory Access) está formado por:

Registro de datos. Almacena los datos temporalmente.

Registro de dirección. Se utiliza para almacenar la dirección de la siguiente palabra que se va a transmitir y se incrementa de forma automática después de cada transferencia.

Registro contador de palabras. Almacena el número de palabras que quedan por enviar y también se decrementa automáticamente después de cada transferencia.

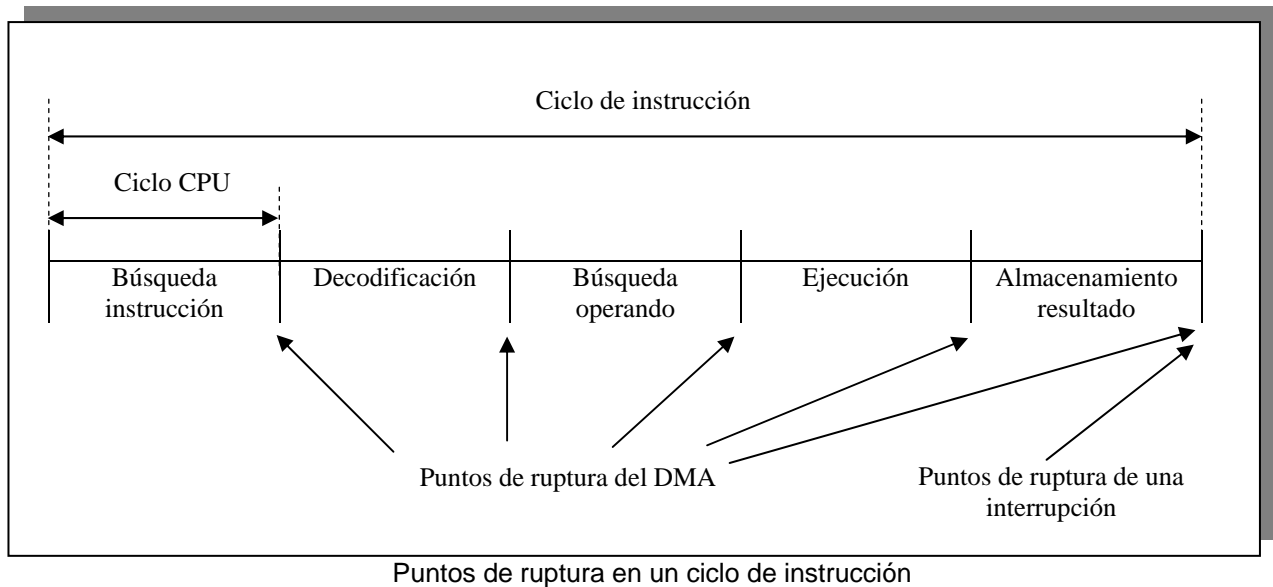
La unidad de control del DMA comprueba si el contenido del registro contador de palabras es 0 y cuando alcanza este valor para la transferencia, envía una señal de interrupción a la CPU indicándole que la transferencia ha finalizado.



## Control del bus

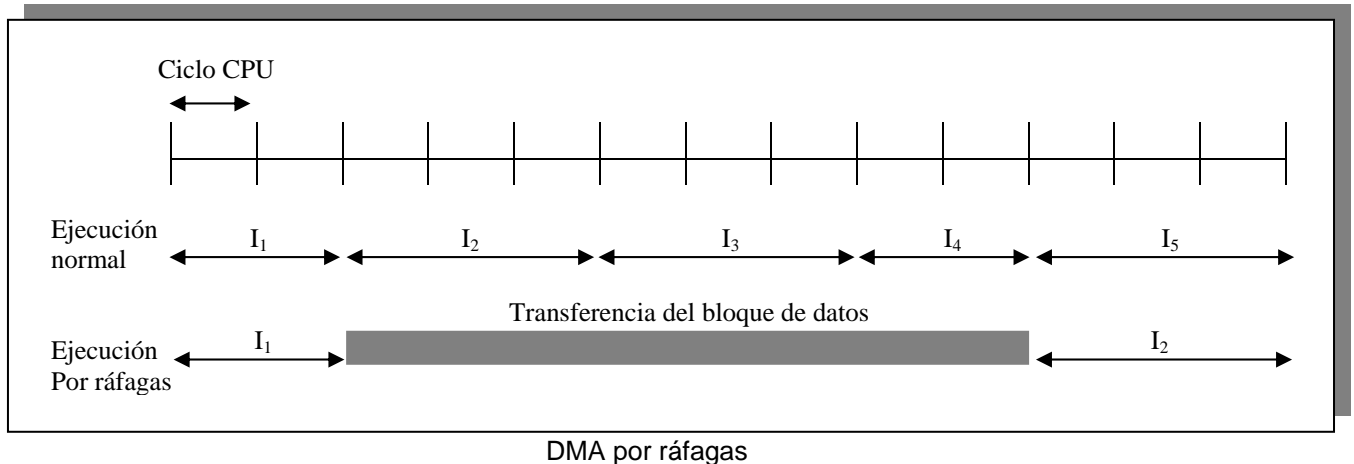
En este tipo de transferencia, el controlador de DMA necesita tener el control del bus para poder transferir datos hacia o desde la memoria.

El controlador de DMA puede interrumpir la ejecución de una instrucción en diferentes momentos como se muestra en la siguiente figura.

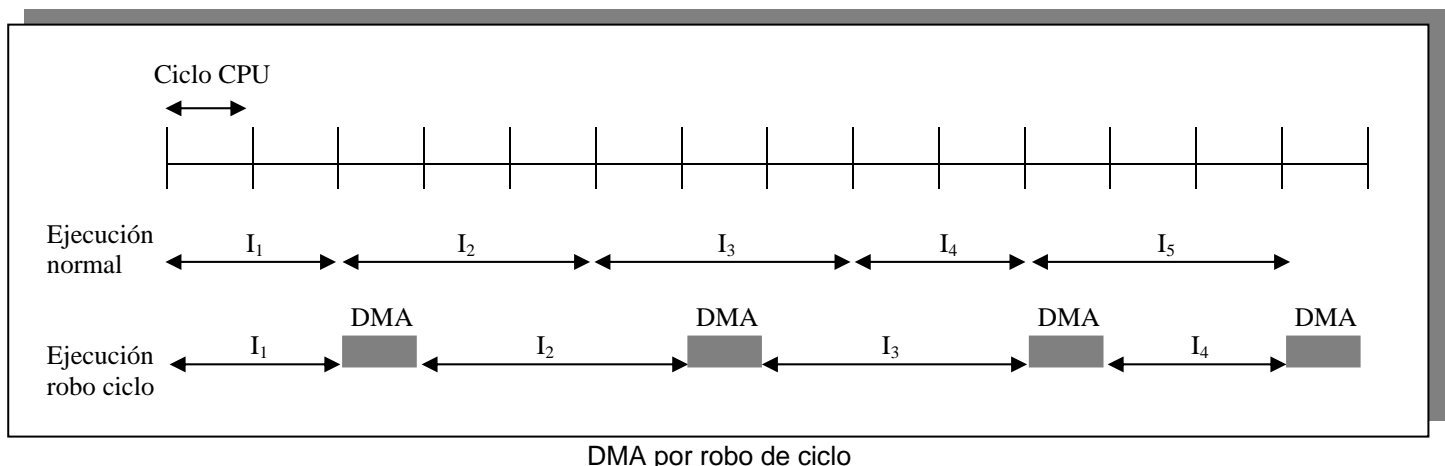


Existen diferentes formas de obtener el control del bus que suponen distintos compromisos entre velocidad de transferencia y actividad de la CPU:

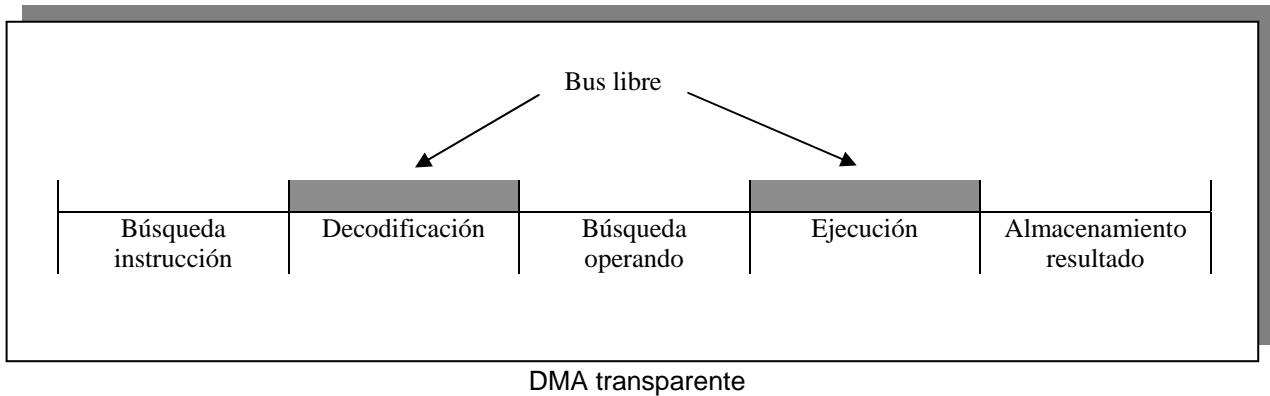
**Por ráfagas.** Cuando el DMA toma el control del bus no lo libera hasta haber transmitido el bloque de datos pedido. Con este método se consigue la mayor velocidad de transferencia pero se tiene a la CPU inactiva durante períodos relativamente largos.



**Por robo de ciclos.** Cuando el DMA toma el control del bus lo retiene durante un solo ciclo. Transmite una palabra y libera el bus. Es la forma más usual de transferencia y en ella el DMA “roba” ciclos a la CPU. El robo de ciclos reduce al máximo la velocidad de transferencia y la interferencia del controlador de DMA sobre la actividad de la CPU.



**DMA transparente.** Es posible eliminar completamente la interferencia entre el controlador de DMA y la CPU. Se consigue si se diseña el DMA de forma que solamente se roban ciclos cuando la CPU no está utilizando el bus del sistema. No se obtiene la ventaja de una velocidad de transferencia muy elevada propia del DMA.



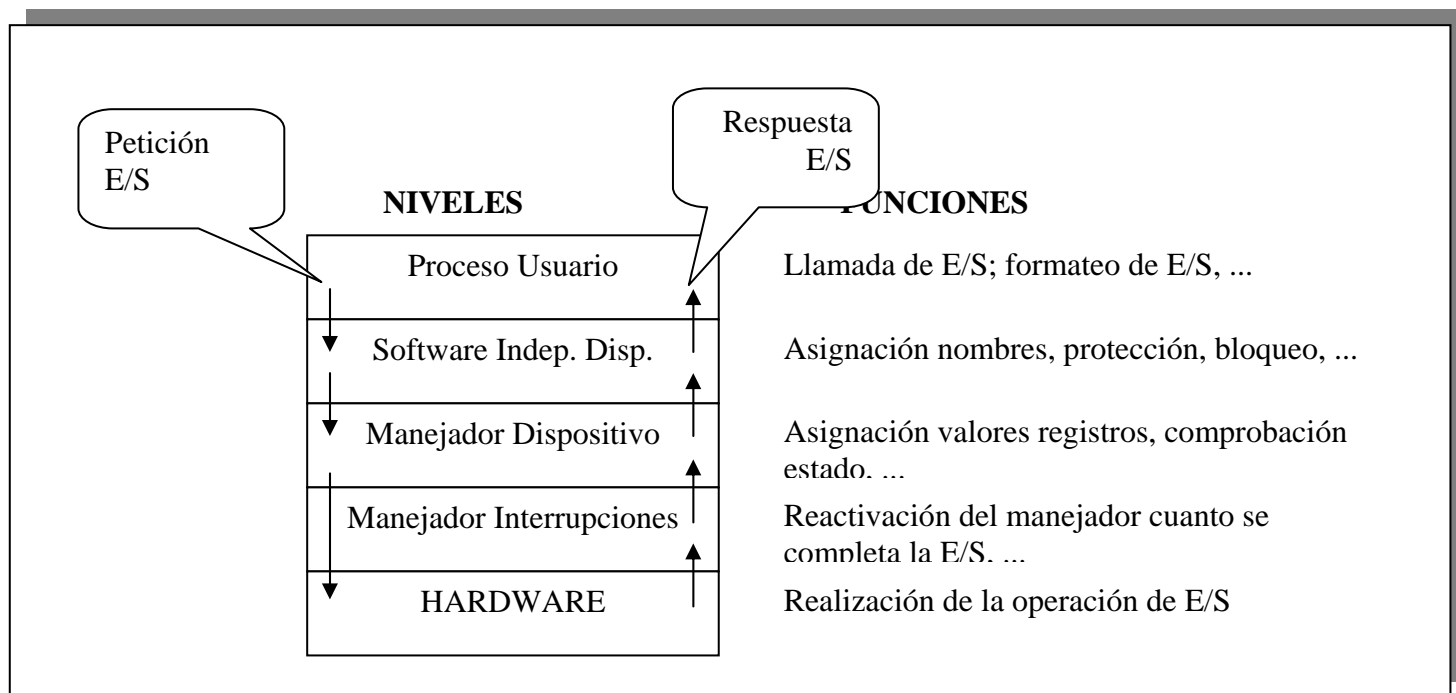
# Software de E/S

---

## Objetivos

- Proporcionar independencia de dispositivo
- Nombres uniformes (del archivo o dispositivo y no depender del dispositivo)
- Manejo de errores (se debe manejar lo más cerca del Hardware)
- Independencia del tipo de transferencia (síncrona mediante bloqueo o asíncrona mediante interrupciones)
- Dispositivos compartidos o exclusivos

## Niveles



En una operación de Entrada/Salida intervienen 4 tipos o niveles de programas

Pasos en la **lectura de un bloque** de un archivo:

1. Se invoca al SO para llevar a cabo la llamada.
2. El Software independiente del dispositivo busca, por ejemplo, en la caché.
3. Se llama al manejador de dispositivos, si el bloque no estuviera en la caché.
4. El manejador pasa la solicitud al Hardware.
5. El proceso se bloquea hasta que termina la operación.
6. Una vez terminada la operación, el Hardware genera una interrupción.
7. Se ejecuta el manejador de interrupciones para descubrir lo ocurrido.
8. Extrae el estado del dispositivo y lo despierta para enviarle los datos y que continúe.

## Manejador de interrupciones

Es el encargado de ocultar las interrupciones a los niveles superiores y que se gestionen en los niveles inferiores.

Es el encargado de bloquear y despertar al proceso que solicita la entrada/salida.

## Manejador de dispositivos

Funciones:

- Recibir peticiones abstractas de niveles superiores

- Enviar las órdenes al controlador y comprobar que se realizan adecuadamente

- Devolver datos, información de estado y error

Es el software que contiene todo el código dependiente de dispositivo. Es el único programa que tiene acceso a los registros del controlador.

Suele manejar solo una clase de dispositivos.

Después de enviar las órdenes al controlador, el manejador puede:

- Bloquearse hasta recibir interrupción si la operación tarda mucho

- No bloquearse si la operación ejecuta rápidamente

## Programas del SO independientes del dispositivo

Funciones:

- Proporcionar operaciones de E/S comunes a todos los dispositivos

- Proporcionar interfaz uniforme a los programas de usuario

- Asignar nombre a los dispositivos

- Proteger los dispositivos

- Proporcionar tamaño de bloque independiente de los dispositivos

- Almacenar temporalmente los datos en las transferencias E/S (buffering)

- Gestionar la asignación de espacio en los dispositivos de bloques

- Reservar y liberar los dispositivos

- Informar de los errores y actuar en consecuencia si el tratamiento es independiente del dispositivo

## Nivel de usuario

La mayoría del software de E/S están dentro del SO, pero una pequeña parte está fuera de él:

- Bibliotecas que se enlazan con los programas de usuario. Llamadas al sistema, etc.

- Sistemas de spooling. Consisten en un directorio para almacenar los ficheros a transferir y un proceso especializado (**daemon**) que es el único que puede utilizar el manejador del dispositivo para que se envíen los ficheros. Ej.: impresora.

# Discos

---

Presentan tres ventajas respecto al uso de la Memoria Principal:

La capacidad del espacio de almacenamiento disponible es mucho más grande

El precio por bit es más barato

No son volátiles

## Hardware de Disco

En el hardware de disco pueden distinguirse dos partes:

La unidad de disco que es el dispositivo electromecánico. Ej: FDD: Floppy Disk Drive.

El controlador de la unidad de disco. Ej: FDC: Floppy Disk Controller.

### El disco

Un **disco magnético** es un cilindro recubierto en las superficies planas de material magnético sobre el que se efectúa la grabación de datos.

El **cabezal del disco** es un transductor que permite leer/escribir datos en el material magnético.

#### **Tipos de discos:**

- Desplazamiento cabezas: fija y móvil.
- Transportabilidad de disco: fijo (rígido) y extraíble (flex.)
- Superficies: única y doble.
- Número disco: único y múltiples.
- Mecanismo de la cabeza: contacto, fija y aerodinámica

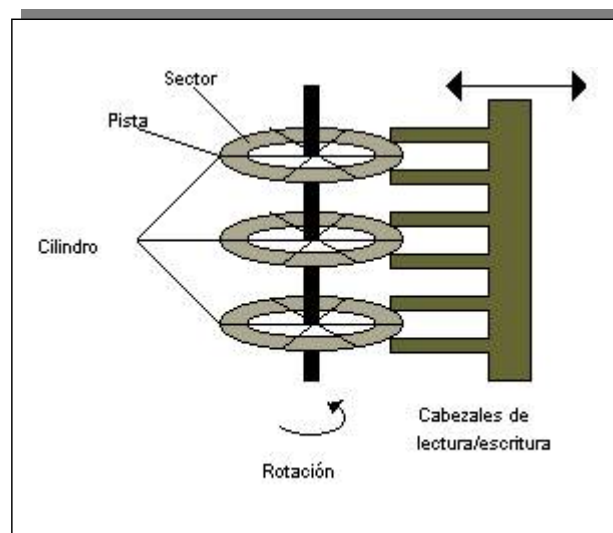


## Organización del disco

**Cilindro:** Conjunto de pistas accesibles en una posición del brazo. Se supone que hay una cabeza por cada cara.

**Pista:** Círculo determinado por una cabeza. Está formada por sectores.

**Sectores:** Unidad básica de trasiego de información.



La organización en cilindros, pistas y sectores se efectúa mediante la operación de dar **formato**.

Ejemplos:

Unidad de Disco	Número Cilindros	Pistas por Cilindro	Sectores por Pista	Sectores por Disco	Bytes por sector	Bytes por disco
5 1/4" Doble Densidad	40	2	9	720	512	368640
5 1/4" Alta Densidad	80	2	15	2400	512	1228800
3 1/2" Doble Densidad	80	2	9	1440	512	737280
3 1/2" Alta Densidad	80	2	18	2880	512	1474560
3 1/2" Extra Densidad	80	2	36	5760	512	2949120

## Direccionamiento

A un determinado sector físico se puede acceder mediante su dirección lógica. Por ejemplo, para el mismo disco flexible

Dirección física sector			Dirección lógica sector
Cilindro	Pista	Sectores	
0	0	1..9	0..8
0	1	1..9	9..17
1	0	1..9	18..26
...	...	...	...
39	1	1..9	711..719

El controlador numera los sectores dentro de una pista desde 1 a 9, y no desde 0 a 8. Cilindro y cabezales sí son numerados a partir del 0.

### El dispositivo

El dispositivo electromecánico está constituido por motores, cabezas de lectura y escritura, carcasa, etc. y la lógica relacionada. Típicamente consta de:

Motores: para el giro del disco y para el posicionamiento del brazo.

Detectores de posición: de inicio de pista, para posición de referencia de las cabezas (pista 00) y de protección contra escritura.

Circuitos para efectuar las lecturas y las escrituras.

Ante una operación de lectura o escritura, la unidad, en un esquema simplificado, opera de la siguiente manera:

1. Enciende el motor.
2. Posiciona las cabezas y espera un tiempo para que estas apoyen sobre la superficie.
3. Localiza el sector y efectúa la operación.
4. Para el motor.

Los tiempos que intervienen en las operaciones de entrada/salida a disco son:

**Tiempo de búsqueda** del cilindro o tiempo de posicionamiento de brazo. Es el tiempo necesario para mover el brazo del disco hasta la pista solicitada. Este tiempo incluye el tiempo de arranque y el tiempo de estabilización de velocidad.

**Tiempo de latencia** o retardo de rotación. Tiempo que transcurre desde que la cabeza está situada en el cilindro hasta que la cabeza se sitúa en el sector solicitado. El tiempo de acceso es igual al tiempo de búsqueda más el tiempo de latencia.

**Tiempo de transferencia.** Tiempo que se necesita para realizar la transferencia de los datos solicitados.

## El controlador

Es la parte electrónica que establece la interacción entre el procesador y el disco recibiendo peticiones del primero, gestionando el trasvase y notificando al procesador el fin de operación.

Realiza las siguientes operaciones:

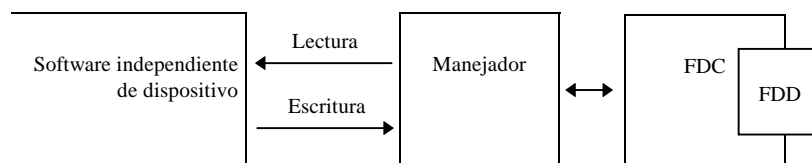
Control de la unidad de disco. Movimiento del brazo para localizar el cilindro, selección de la cabeza correspondiente a la pista, operación sobre el sector interesado, recalibrado, etc.

Trasvase de información. Comprobación de la corrección de los datos, interpretación y ejecución de las órdenes relacionadas con el disco.

## Software de Disco

### Esquema general

Función principal del manejador es proporcionar un interfaz independiente del dispositivo.



El esquema del manejador es un bucle infinito que atiende peticiones, encarga su realización e informa del resultado.

---

```
mensaje mens;                                /*buffer del mensaje */
manejador_disco ()
{
    int r, peticionario;

    iniciar();
    while (TRUE)
    {
        recibir_petición (&mens);             /*esperar petición */
        peticionario = mens.peticionario;      /*proceso que envió mensaje */
        switch(mens.m_tipo)
        {                                       /*tratamiento tipo petición */
            case READ:    r = leer(); break;
            case WRITE:   r = escribir(); break;
            default:      r = ERROR;
        }
        mens.tipo_op = RESPUESTA;
        mens.STATUS = r;                      /* código de resultado */
        enviar(peticionario, &mens);          /* mensaje respuesta remitente*/
    }
}
```

---

## Conversión de direcciones lógicas a direcciones físicas

Para la conversión es necesario conocer la numeración de los cilindros, pistas y sectores. Generalmente:

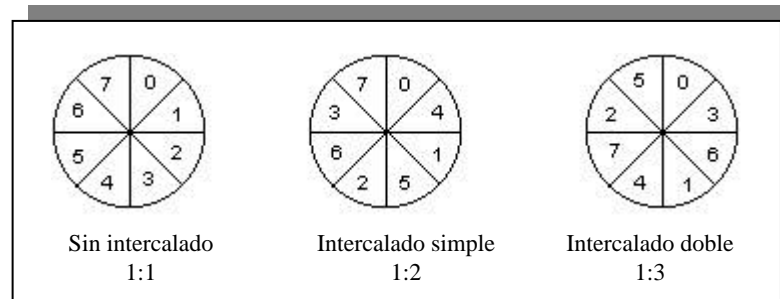
Numeración de cilindros comenzando desde el más externo (cilindro físico 0).

Numeración de pistas comenzando desde la superior (cabeza 0)

Numeración de sectores a partir de la marca de inicio de pista y considerando el factor de intercalación.

El **factor de intercalación** indica la posición que ocupa un sector lógico respecto de su predecesor en una pista.

Este mecanismo permite reducir el tiempo que necesita la unidad de disco para localizar un sector en una pista:



La función que convierte la dirección lógica que recibe el manejador al formato de cilindro-pista-sector para acceso al disco será:

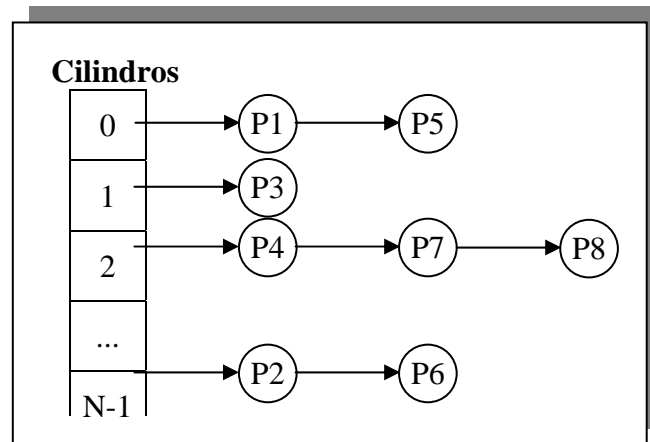
```
int convertir_dirección(long dir_disco) /*dirección en palabra, no en bloque
*/
{
    ...
    bloque = (int) (dir_disco / tam_sector);
    cilindro = (int) (bloque / (n_pist * n_sect));
    pista = (int) ((bloque % (n_pist * n_sect)) / n_sect);
    sector_lógico = bloque - cilindro * n_pist * n_sect - pista * n_sect;
    sector_físico = (sector_lógico * intercalación) % n_sect;
    ...
}
```

Por ejemplo, en un disco de 2 pistas, 9 sectores por pista y 512 bytes por sector (se suponen los bloques del mismo tamaño que los sectores), las posiciones en el disco para varias direcciones se muestran en la tabla:

intercalado	dir_disco	bloque	cil.	pista	s._lóg.	sect._fís.
1	34309	67	3	1	4	4
2						8
1	57212	111	6	0	3	3
2						6
1	40450	79	4	0	7	7
2						5

## Planificación del Movimiento del Brazo

El Sistema Operativo mantiene una cola de peticiones de acceso para cada dispositivo de Entrada/Salida. Dependiendo de cómo se atiendan esas peticiones el tiempo de posicionamiento será mayor o menor. Por ejemplo, para un determinado



dispositivo

### **FCFS: Primero en llegar, primero en ser servido**

Las peticiones son atendidas en el orden de llegada.

**Ejemplo:** Sea un disco con 200 pistas y se supone que las peticiones en un instante determinado son en orden de llegada: 55, 58, 39, 18, 90, 160, 150, 38 y 184. Suponer que inicialmente la cabeza está situada en la pista 100 y se mueve en direcciones crecientes de pista. Indicar en qué orden se atenderán.

Pistas Solicitados								
055	058	039	018	090	160	150	038	184
1								
	2							
		3						
			4					
				5				
					6			
						7		
							8	
								9

Es el algoritmo más fácil de implementar.

Movimientos bruscos.

Recorridos totales elevados.

### SSTF (SSF) : Primero más corto

Se atiende primero las peticiones más próximas a la posición actual de la cabeza. Si dos peticiones son equidistantes, se puede usar un algoritmo aleatorio para decidir o bien se da prioridad al sentido de movimiento actual de la cabeza.

Ejemplo:

Pistas Solicitadas								
055	058	039	018	090	160	150	038	184
				1				
	2							
3								
		4						
							5	
			6					
						7		
					8			
								9

Se minimiza el tiempo medio de posicionamiento. Mejor que FCFS.

Problema de inanición

El brazo tiende a permanecer en la mitad del disco.

### SCAN: Algoritmo del ascensor

El brazo sólo se puede mover en un sentido, resolviendo todas las peticiones pendientes en su ruta hasta que alcance la última pista. A continuación, la cabeza se mueve en el otro sentido atendiendo las peticiones que se encuentre.

Ejemplo:

Pistas Solicitadas								
055	058	039	018	090	160	150	038	184
						1		
					2			
								3
				4				
	5							
6								
		7						
							8	
			9					

Presenta la ventaja de que atiende cualquier petición en un tiempo máximo.

Normalmente, es peor que el algoritmo SSTF.

## CSCAN: Algoritmo del ascensor circular

El brazo sólo se puede mover en un sentido, resolviendo todas las peticiones pendientes en su ruta. Cuando llega a la última pista, se desplaza a la primera y empieza a resolver las nuevas peticiones.

Ejemplo:

Pistas Solicitadas								
055	058	039	018	090	160	150	038	184
						1		
					2			
								3
			4					
							5	
		6						
7								
	8							
				9				

## LOOK

Es una variante del algoritmo SCAN. Cambia de sentido cuando se ha atendido la última petición y no tiene que llegar hasta la última pista.

Ejemplo:

Pistas Solicitadas								
055	058	039	018	090	160	150	038	184
						1		
					2			
								3
				4				
	5							
6								
		7						
							8	
			9					

Los recorridos totales son inferiores al algoritmo SCAN

Más complejo que el algoritmo SCAN.

## CLOOK

Es una variante del algoritmo CSCAN. Cambia de sentido cuando se ha atendido la última petición y no tiene que llegar hasta la última pista.

Ejemplo:

Pistas Solicitadas								
055	058	039	018	090	160	150	038	184
						1		
					2			
								3
			4					
							5	
		6						
7								
	8							
				9				

Los recorridos totales son inferiores al algoritmo CSCAN pero más elevados que LOOK.

Más complejo que el algoritmo CSCAN.

## Transferencia

La transferencia es una operación sencilla a nivel del software porque el trabajo se realiza entre el controlador y el DMA. Todo lo que se requiere es indicar:

- el tipo de operación (lectura o escritura),
- la unidad de disco,
- la dirección física (cilindro, pista, sector),
- el número de sectores por pista,
- el espacio entre sectores y
- la longitud de datos.

Seguidamente, el manejador debe bloquearse en espera de la interrupción del controlador indicando el final de operación. Ocurrida la interrupción, se obtiene el estado del controlador y se comprueba el resultado de la transferencia.



## Tratamiento de Errores

Se pueden distinguir dos tipos de errores:

Sistemático, porque algún elemento haya perdido operatividad o alguna de las actividades manuales se haya realizado incorrectamente (ausencia de disco, protección contra escritura en su caso, etc.). Se emite un mensaje de error y no se reintenta.

Transitorio, desde una mota de polvo sobre la superficie hasta un error en el controlador. Se resuelven reintentando la operación hasta un número limitado de veces. Caso de no subsanarse, se entenderá que se trata de un error sistemático.

Cuando la causa del error es de deterioro de un sector, se toman soluciones como:

Crear un archivo de sectores defectuosos cuyo contenido son los sectores que no pueden ser utilizados. Este fichero es transparente al usuario.

Reemplazar las pistas dañadas por otras al final del disco reservadas a tal efecto.

Los más comunes:

De programación (algún parámetro incorrecto). Ej.: solicitar sector inexistente.

- algunos son detectados por el controlador
- en general, el manejador deberá devolver error

Transitorios de transferencia. Ej.: provocado por suciedad.

- se detectan mediante sumas de control de los datos trasvasados
- se reintenta la operación varias veces

Permanentes de transferencia. Ej.: sectores dañados.

- los bloques inservibles deben estar caracterizados
- se pueden utilizar ficheros de bloques inservibles (ojo a operaciones que no operan sobre ficheros)

De búsqueda (posicionamiento incorrecto del brazo)

- algunos controladores los corrigen automáticamente
- otros controladores ponen a 1 un bit de error y declinan la responsabilidad en el manejador (suele enviar comando RECALIBRATE)

Del controlador. Ej.: el controlador no acepta los comandos.

- caso de malfunción, es típico poner a 1 un bit de error para forzar la reinicialización del propio controlador

## Otras Operaciones

### **Iniciación del dispositivo**

El manejador informa al controlador del número de unidades de disco, calibrado del brazo, unidad de disco por defecto, parámetros de estabilización de disco, etc.

Comprobar que las operaciones se han realizado satisfactoriamente.

### **Carga de DMA**

Sentido de la transferencia (Controlador → Memoria y Controlador ← Memoria).

Dirección de inicio en memoria principal.

Cantidad de palabras a transferir.

### **Parada del motor**

No se para el motor después de cada transferencia sino que se realiza una **parada retardada** con un temporizador de guardia.

### **Encendido del motor**

Para encender el motor hay que activar un bit y esperar un tiempo de estabilización.

---

```
int arrancar_motor()
{
    ...
    if (motor_girando)
    {
        inhabilitar_parada();
        return;
    }
    escribir_en_puerto ( );
    esperar (tiempo_estabilización); /* temporizador de guardia */
}
```

---

# Mejoras

## Mejoras de Eficiencia

Se suelen utilizar otras técnicas para mejorar las prestaciones globales.

Solapamiento de búsquedas. Solapadas entre sí y solapadas con operación lectura o escritura.

Otras planificaciones del brazo. Ascensor combinado con datos de uso más frecuente en cilindros centrales.

Cache interna del controlador. Cuando se da orden de leer un sector, se lee toda la pista y se almacena en la cache.

Defragmentar.

Intercalación entre pistas.

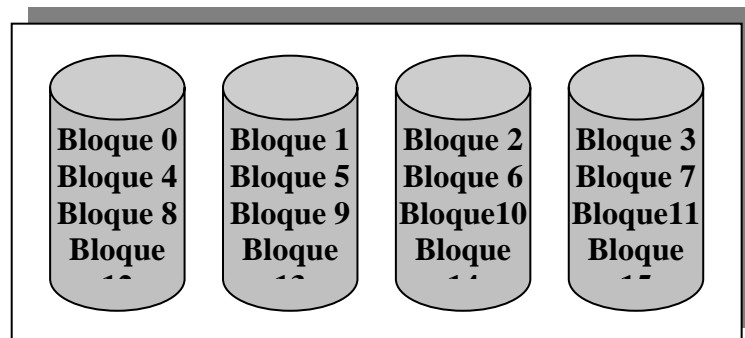
Intercalación de discos. Acceso a varios discos en paralelo.

## Mejoras de Fiabilidad

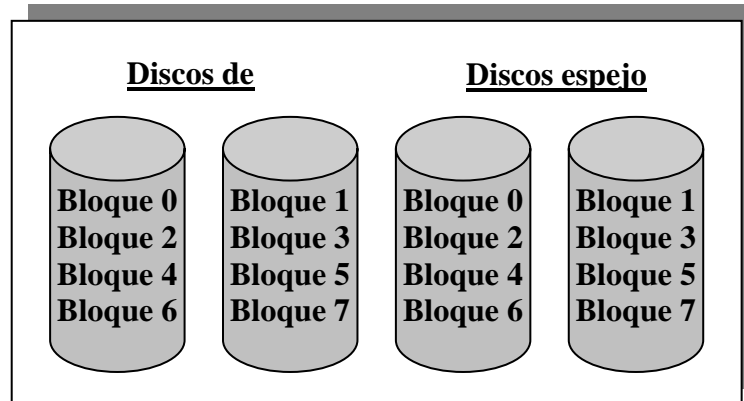
Empleo de grupos redundantes de discos de bajo coste **Sistemas RAID** (redundant arrays of inexpensive disks).

Esta redundancia puede organizarse con diverso rendimiento y rentabilidad. Existen diferentes niveles o configuraciones RAID dependiendo del grado de redundancia:

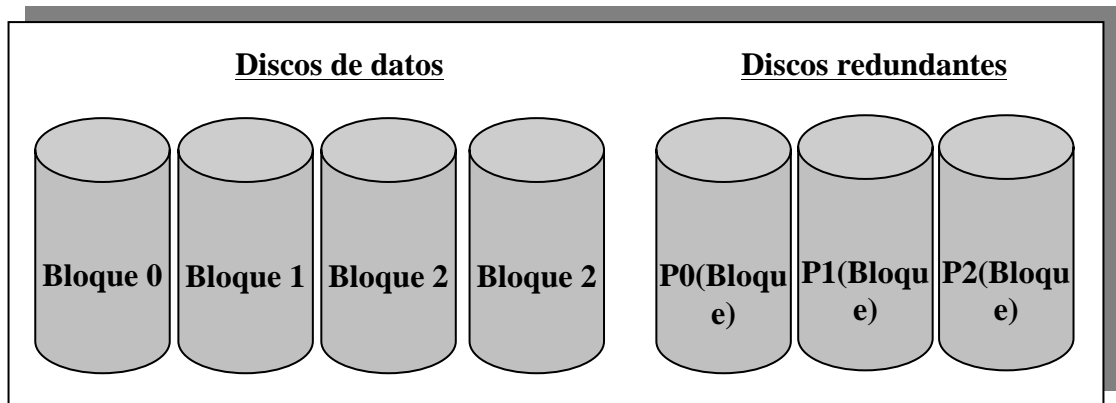
- Nivel 0. Se basa en la técnica de intercalación de discos. Este nivel se emplea en aplicaciones que requieren altas prestaciones con datos no críticos.



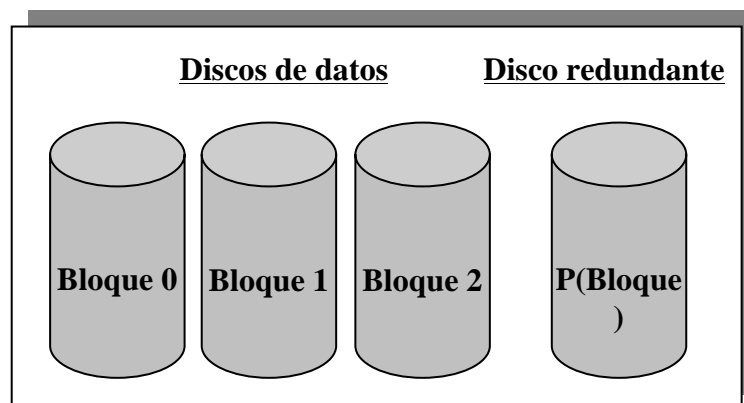
- Nivel 1. Llamado “espejo” o “sombra”, consiste en conservar un duplicado de cada disco. Su principal inconveniente es el coste. Es muy aconsejable cuando se trabaje con datos muy críticos. Sistema muy caro.



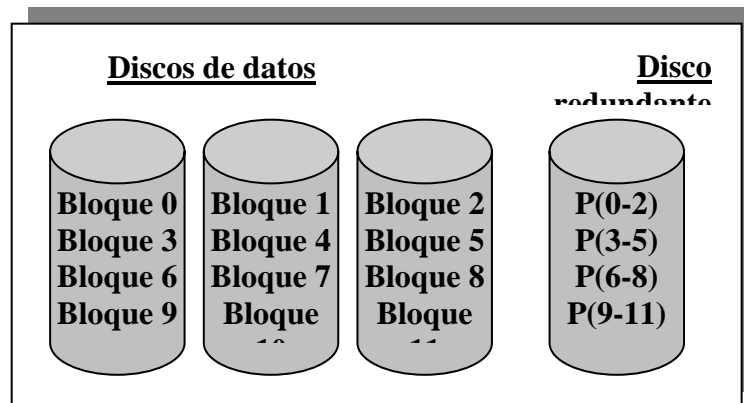
- Nivel 2. Este nivel usa una técnica de acceso paralela. Se emplea un código de corrección de errores (Hamming) que se calcula a partir de los bits de cada disco, y los bits del código se almacenan en las correspondientes posiciones de bit en varios discos redundantes. Sistema bastante caro.



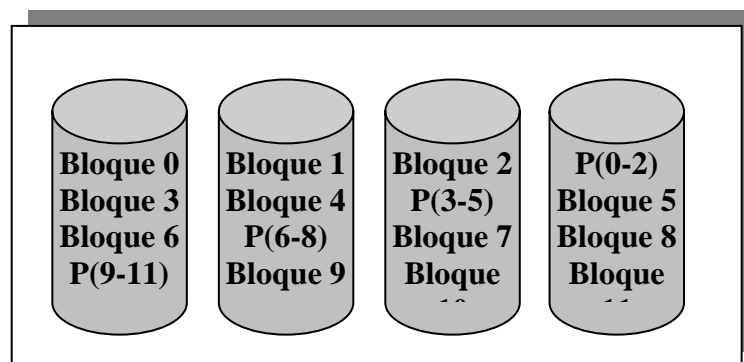
- Nivel 3. También usa una técnica de acceso paralela. Código de corrección de errores más sencillo (paridad). Velocidades de transferencia de datos más altas que RAID 2. Menos caro que RAID 2. No simultaneidad peticiones.



- Nivel 4. Cada disco opera independientemente de forma que peticiones de E/S separadas se atienden en paralelo. Velocidades de petición de E/S altas. Velocidades bajas de transferencias de datos. Presenta como inconveniente el hecho de que el disco de paridad se convierte en cuello de botella.



- Nivel 5. Está organizado de manera similar al Nivel 4. La diferencia es que este nivel distribuye los bloques de paridad a lo largo de todos los discos con lo que se evita el cuello de botella.

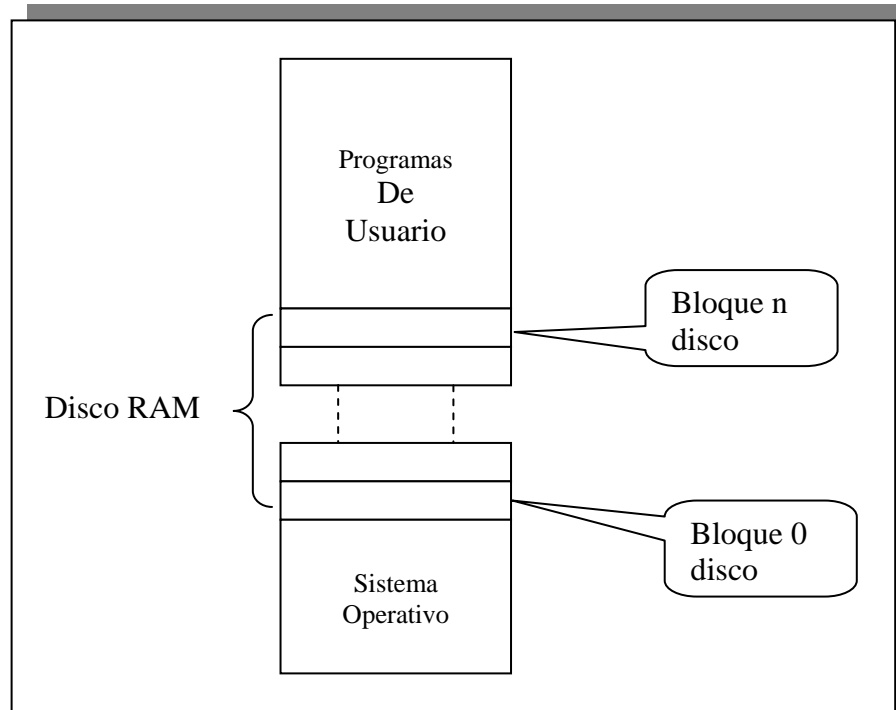


Por tanto el uso de sistemas RAID disminuye las pérdidas de datos y aumenta el rendimiento del sistema. Se ha demostrado que, con un array de 100 discos de bajo coste y 10 discos de paridad, el tiempo promedio entre pérdidas de datos del array es 90 años, frente a 2 o 3 años para los discos grandes y caros.

# Discos RAM

---

Son discos que utilizan una parte de memoria principal para almacenar los bloques.



Adecuado para el almacenamiento de programas o datos de uso frecuente (ficheros temporales).

La principal ventaja es que su tiempo de acceso es muy reducido (memoria RAM).

Los inconvenientes que presenta son volatilidad, poca capacidad, no búsquedas solapadas.

# Relojes

---

Todos los computadores disponen de un reloj (*timer*) que se utiliza para:

- Mantener fecha y hora
- Disparar alarmas
- Gestionar los requisamientos de procesador
- Proporcionar la base para la contabilización

El reloj es considerado por el sistema como un dispositivo y tiene asignado un manejador y un controlador.

El controlador es un circuito integrado que funciona basándose en el reloj interno del computador.

## Hardware

### Tipos

Generación de pulsos en función de la señal de alimentación. (220 v; 60 Hz)

Oscilador de cristal.(100 MHz).

Componentes:

- oscilador (cuarzo)
- contador (descendente)
- registro de carga

Funcionamiento:

- Iniciar una cuenta copiando el registro de carga al contador
- El oscilador genera un pulso periódico de decremento del contador
- Si el contador llega a cero dispara una interrupción

### Modos de operación de los relojes programables

No repetitivo: El contador se carga e inicia el decremento externamente y se detiene al llegar a cero.

Onda cuadrada: El contador se carga automáticamente al llegar a cero y la cuenta no para de forma que el reloj genera una secuencia de interrupciones periódicas (pulsos de reloj o ticks).

# El Manejador

## Funciones

Mantener fecha y hora actual. Formatos de almacenamiento (pulsos, segundos, por referencia al tiempo de arranque).

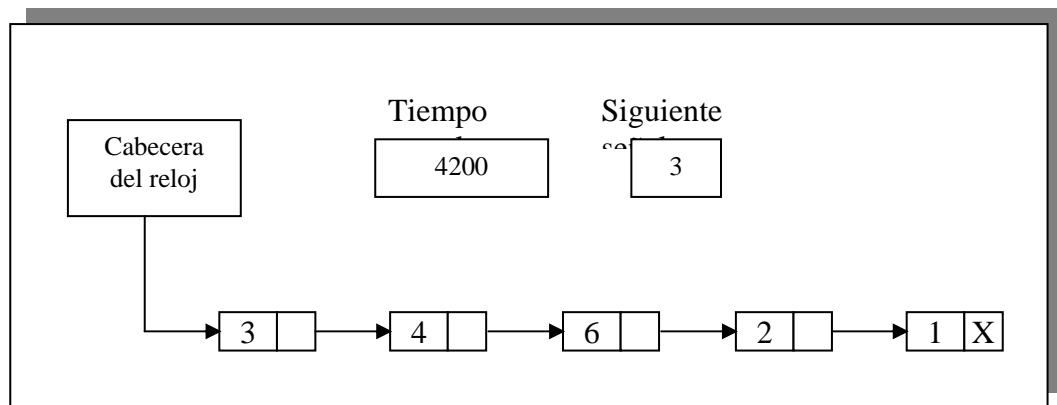
Proporcionar interrupciones para liberación de procesador (Quatum)

Contabilizar los tiempos de uso del procesador.

Implementar la llamada al sistema *alarm(...)* para los procesos de usuario

Se deben simular múltiples relojes virtuales, con un sólo reloj físico.

Para ello se mantiene una tabla o una lista enlazada de los tiempos pendientes



Las señales se han de generar en los instantes 4203, 4207, 4213, 4215 y 4216

Proporcionar temporizadores de guarda.

El propio sistema necesita también temporizadores. Ej.: motor disquetera.

Medir perfiles de ejecución y reunir estadísticas. Ej.: histograma del contador de programa

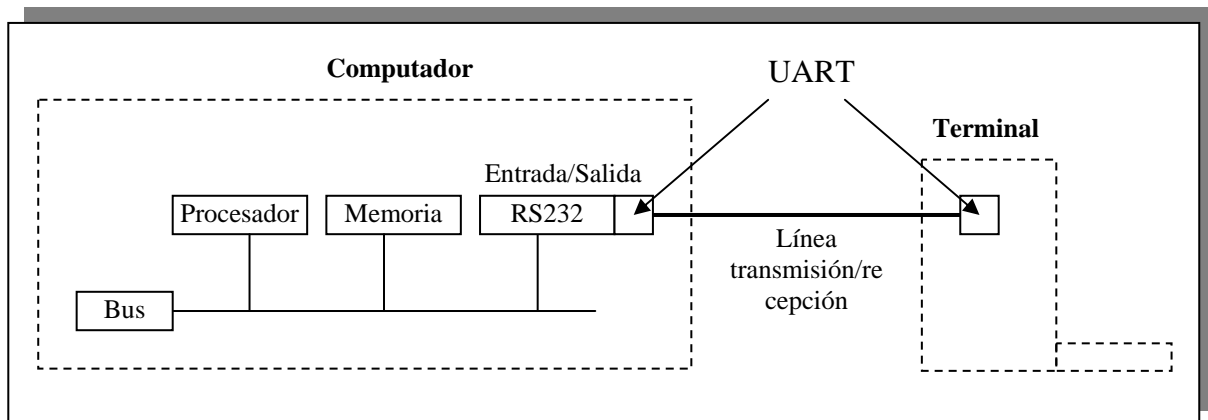


# Terminales

## Tipos

### Terminales serie (RS-232)

- Formados por un teclado, un monitor y una tarjeta controladora.
- La transmisión es bit a bit.
- El manejador escribe un carácter en la RS-232 y se bloquea en espera de la interrupción de fin de transmisión
- El UART envía el carácter bit a bit. Después de enviarlo, genera una interrupción indicando que ha sido transmitido el carácter.
- Algunos controladores tienen buffer y procesador para efectuar la transmisión.



### Terminales mapeadas a memoria

- La RAM de vídeo es parte de la memoria central.
- En la tarjeta gráfica está incorporado un controlador gráfico que genera la señal que va al monitor.
- El controlador recorre la RAM de vídeo y la muestra en pantalla.
- Estos terminales son más rápidos que los anteriores.

### Terminales de red

- Formados por un teclado, un monitor, una tarjeta de red, el procesador y la memoria.
- La comunicación entre computador y terminal se realiza a través de una red (ethernet, token ring, etc.).
- Estos terminales son más rápidos y caros que los serie.

### Ejemplo: IBM PC

- Teclado y pantalla son independientes
- Terminal mapeada a memoria (0xB8000 dirección base de pantalla).
- Un carácter en pantalla ocupa 2 bytes RAM (código ASCII y atributo)
- El teclado envía una interrupción al pulsar una tecla y otra al soltarla
- El controlador proporciona un código por cada tecla pulsada pero no facilita la conversión ASCII

## El Manejador

El manejador de la terminal se encarga de ofrecer un interfaz común para acceder a los terminales. Oculta las diferencias entre los distintos tipos de terminales.

## Teclado

### Función

Obtener las señales del teclado.

Elaborar los códigos y otras operaciones sencillas.

pasarlos a los programas de usuario.

### Modos de operación

**Puro:** envía los caracteres aisladamente sin efectuar modificaciones sobre ellos ni procesarlos

**Crudo:** envía los caracteres aisladamente pero interpretando:

DEL (Genera la señal SIGINT)

^\  
(Genera la señal SIGQUIT)

^S (Parar desplazamiento pantalla)

^Q (Continuar desplazamiento pantalla)

**Elaborado:** soporta operaciones sencillas de edición sobre una línea de caracteres ya que tiene un buffer interno. Interpreta más caracteres que el modo crudo.

### Buffer de teclado

Se utiliza para:

- almacenar la entrada si se produce por adelantado, en el caso de que el manejador continúe ocupado con el carácter anterior.
- operar en modo elaborado.

Tipos:

- Dedicados: un buffer para cada terminal.
- Compartidos: pool buffer central.

### Manejador de teclado

Se encarga de:

Enviar a pantalla la tecla pulsada (el eco por hardware no es deseable).

Sincronización. La interpretación de las teclas no siempre tardan lo mismo.  
(scroll, retornos de carro, retroceso, tabulador, etc.)

Permite la programación del teclado. En un sistema UNIX la programación se efectúa mediante la llamada al sistema **ioctl** y establece:

- Los caracteres especiales.

- Los parámetros de la transmisión.
- La palabra de modo del terminal.

## Monitor

### Interfaz serie

Suele utilizar buffers, ya sea dedicados o comunes al teclado.

Modo de operación:

1. La salida se copia al buffer.
2. El manejador saca un carácter del buffer y lo envía al puerto serie.
3. El manejador se bloquea en espera de interrupción de fin de transmisión antes de enviar el siguiente carácter.
4. El controlador del monitor se ocupa de operar sobre el dispositivo como si a su vez utilizara interfaz por memoria.

### Interfaz por RAM vídeo

La salida se copia en la RAM de vídeo.

Es responsabilidad del manejador:

- procesar los caracteres especiales de operación sobre pantalla
- gestionar la RAM de vídeo
- posicionar el cursor (utiliza un registro contador)

