

# Arquitectura y Sistemas Operativos

Tecnicatura Universitaria en  
Programación

ion: absolute; z-index: 999;  
x 5px #ccc}.gbrtl .gbm{-m  
display: block; position: a  
capacity: 1; \*top: -2px; \*left:  
/; top: -4px\0/; left: -6px\0  
ne-box; display: inline-bloc  
display: block; list-style: r  
ne-block; line-height: 27px;  
pointer; display: block; tex  
ative; z-index: 1000}.gbtm{\*  
padding-right: 9px}#gbz .g  
ad:url(//

# Gestión de Entrada Salida

Unidad 9



# Agenda



1. Introducción
2. Uso de buffers
3. Planificación de disco
4. RAID





# 1. Introducción

Evolución del sistema de E/S

# Técnicas de E/S

## E/S programada.

- El procesador envía un mandato de E/S a un módulo de E/S; a continuación, ese proceso realiza una espera activa hasta que se complete la operación antes de continuar.

## E/S dirigida por interrupciones.

- El procesador emite un mandato de E/S y continúa ejecutando las instrucciones siguientes, siendo interrumpido por el módulo de E/S cuando éste ha completado su trabajo.

## Acceso directo de memoria (DMA).

- Un módulo de DMA controla el intercambio de datos entre la memoria principal y un módulo de E/S. El procesador manda una petición de transferencia de un bloque de datos al módulo de DMA y resulta interrumpido sólo cuando se haya transferido el bloque completo.

# Evolución en el tiempo

1. El procesador controla directamente un dispositivo periférico. Esta situación se presenta en dispositivos simples controlados por un microprocesador.



2. Se añade un controlador o módulo de E/S. El procesador usa E/S programada sin interrupciones. Con este paso, el procesador se independiza de los detalles específicos de las interfaces de los dispositivos externos.



3. Se utiliza la misma configuración que en la etapa anterior, pero empleando interrupciones. El procesador no necesita gastar tiempo esperando a que se realice una operación de E/S, incrementando de esta manera la eficiencia.



4. Al módulo de E/S se le da control directo de la memoria mediante DMA. Con ello, puede mover un bloque de datos a la memoria sin involucrar el procesador, excepto al principio y al final de la transferencia.



5. Se mejora el módulo de E/S para convertirse en un procesador independiente. La CPU hace que el procesador ejecute un programa de E/S residente en la memoria principal. El procesador de E/S lee y ejecuta estas instrucciones sin la intervención del procesador.



6. El módulo de E/S tiene su propia memoria local. y se pueden controlar un gran conjunto de dispositivos de E/S, con una intervención mínima por parte del procesador.





## 2. Uso de buffers



# Situación hipotética

Un proceso debe leer datos de disco y escribirlos en una zona de datos del espacio de direcciones del proceso de usuario.

Dos opciones: espera activa, o suspender el proceso hasta que se complete la petición de E/S.





# Situación hipotética

- Espera activa: hace un uso ineficiente del procesador
- Suspender el proceso: si se utiliza paginación de memoria, la página de la zona de datos del espacio de direcciones donde deben escribirse los datos debe estar en memoria. Si se suspende el proceso y la página se expulsa, se **genera un interbloqueo de un solo proceso.**

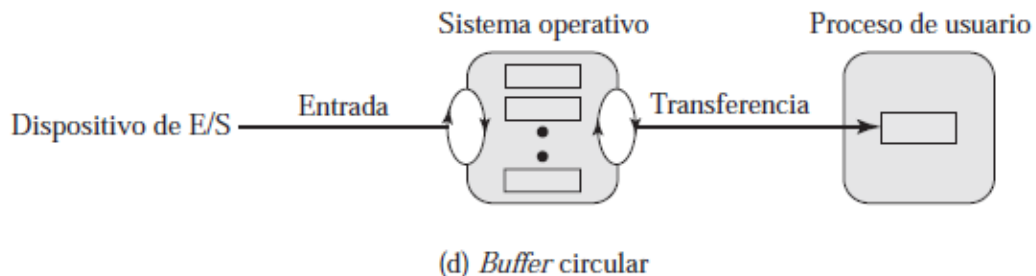
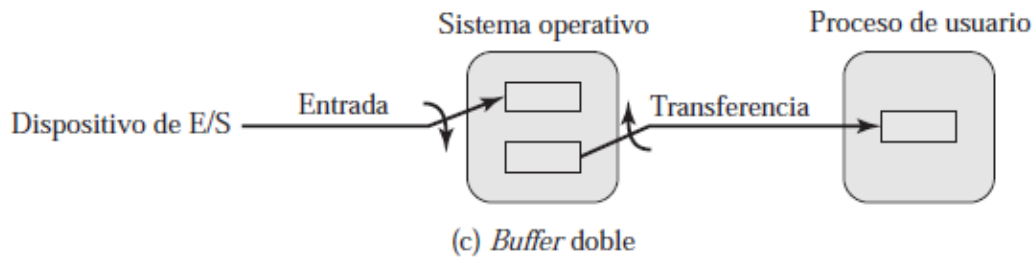
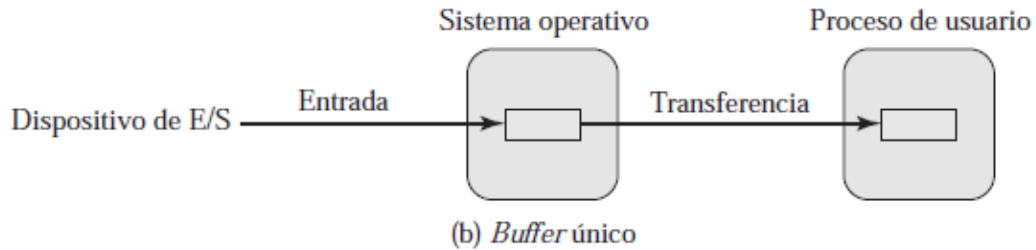
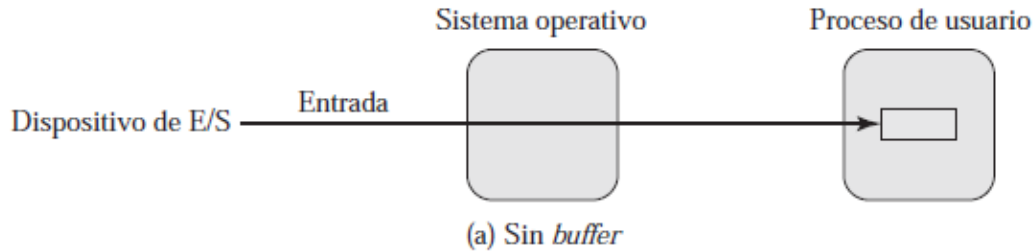
# Uso de buffers

El uso de buffers es una técnica que si bien no permitirá que un dispositivo de E/S funcione a la velocidad del procesador o de la memoria principal, amortigua los picos en la demanda de E/S.

En un entorno multiprogramado, el uso de buffers es una técnica que puede incrementar la eficiencia del sistema operativo y el rendimiento de los procesos individuales.



# Tipos de buffers



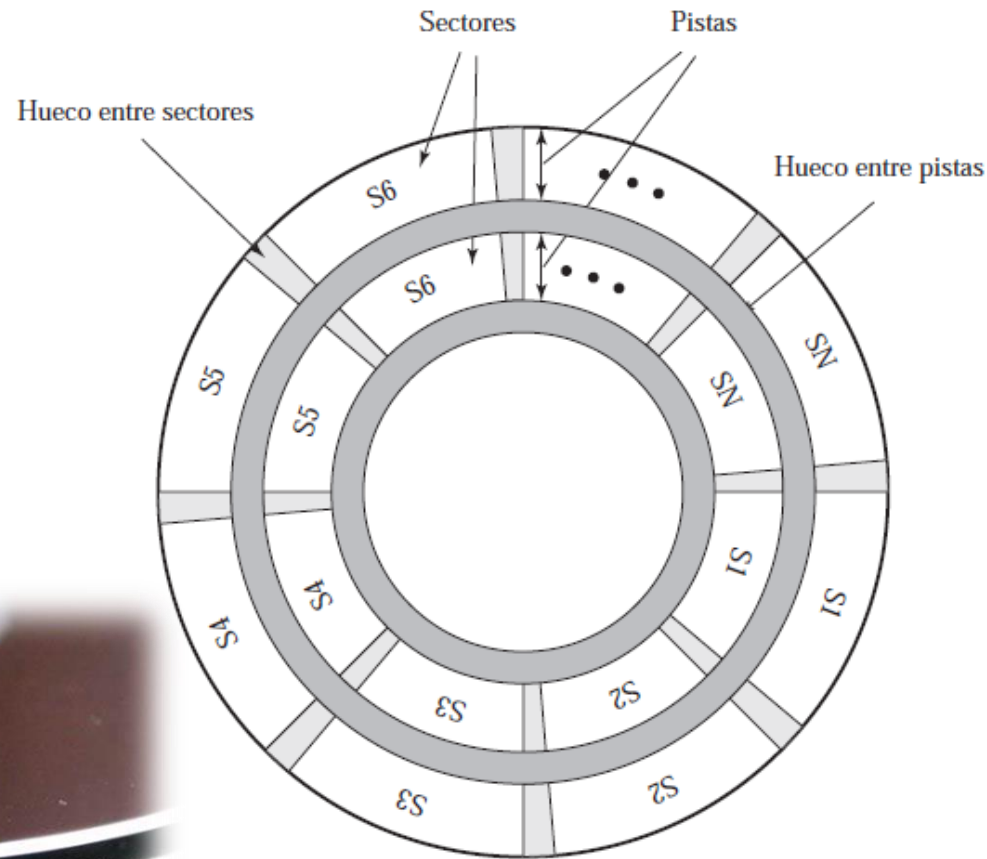
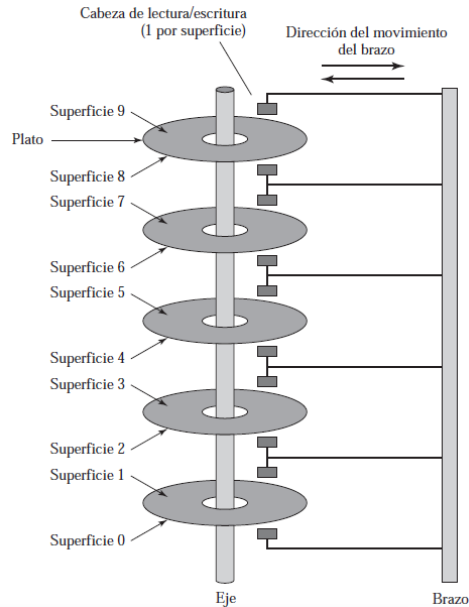
- **Buffer Único:** Los datos se ingresan hasta llenar el buffer del SO. Luego son enviados al área de datos del proceso de usuario. Permite expulsar el proceso
- **Buffer Doble:** Los datos cargados en un buffer son consumidos por el proceso mientras el SO llena el otro buffer.
- **Buffer Circular:** Similar a buffer doble pero con más buffers. Esta técnica permite “suavizar” la demanda de E/S



### 3. Planificación de disco



# El disco.

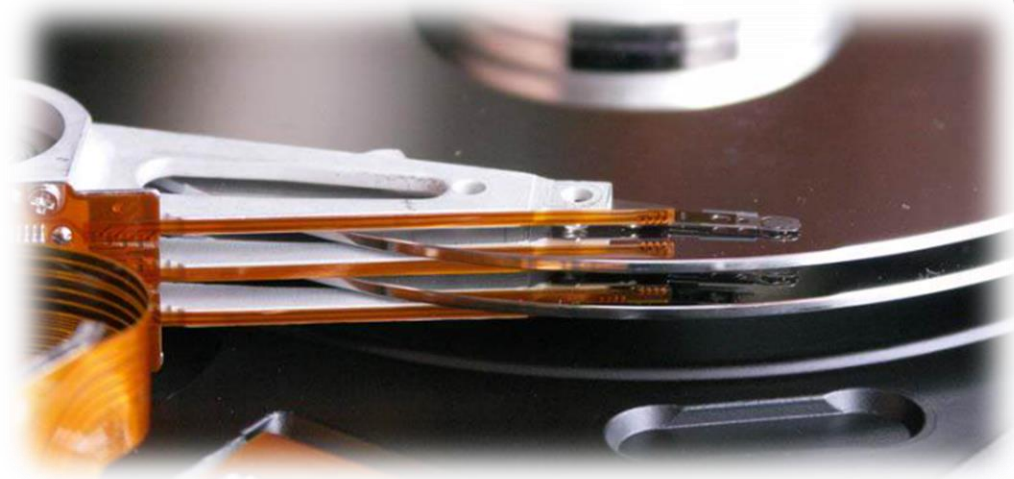


**Tiempo de búsqueda:** el tiempo que se tarda en situar la cabeza en la pista

**Retardo rotacional:** el tiempo que se tarda en llegar al comienzo del sector

**Tiempo de acceso:** Tiempo de búsqueda + Retardo rotacional

**Tiempo de transferencia:** Tiempo que transcurre mientras que el sector se desplaza debajo de la cabeza de lectura/escritura



# Políticas de planificación

Tienen como objetivo principal **minimizar el tiempo de búsqueda**, es decir reducir la cantidad de movimientos que realiza el cabezal de lectura.



# Políticas de planificación

## FIFO

- Simple y equitativa. Rendimiento pobre

## SSTF

- Primero el de tiempo de servicio más corto. Mejor rendimiento que FIFO. Posible starvation

## SCAN

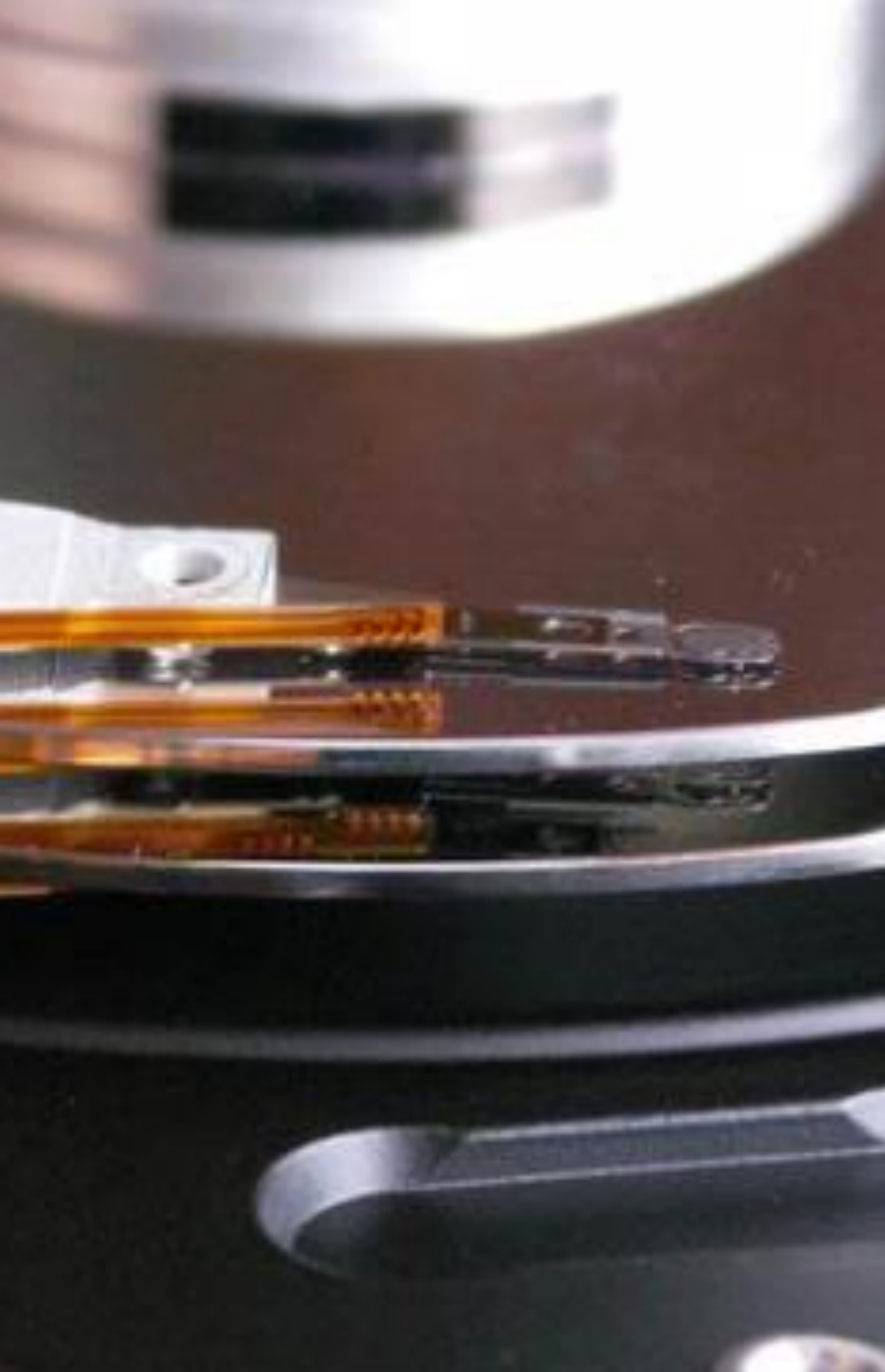
- Algoritmo del ascensor. El brazo sólo se mueve en una dirección. No sufre starvation.

## C-SCAN

- Similar a SCAN, restringiendo el movimiento del brazo en una dirección. Reduce los tiempos de SCAN para algunas peticiones



## 4. RAID



# RAID

“Redundant array of independent disks”  
o array redundante de discos Independientes.

Es un conjunto de discos que a partir de distintas configuraciones proporcionan mejoras en cuanto a la velocidad de acceso, redundancia y consistencia de los datos.



# Tres características fundamentales

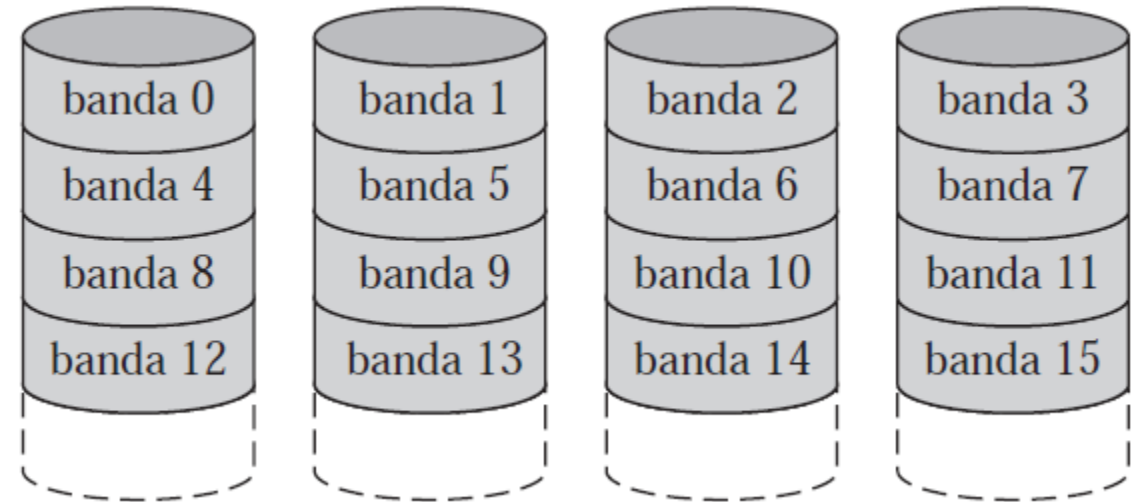
1. El sistema operativo los percibe como un único dispositivo lógico.

2. Los datos se distribuyen a lo largo de las unidades físicas de un vector

3. La capacidad de redundancia del disco se utiliza para almacenar información de paridad, que garantiza la recuperación de los datos en caso de falla.

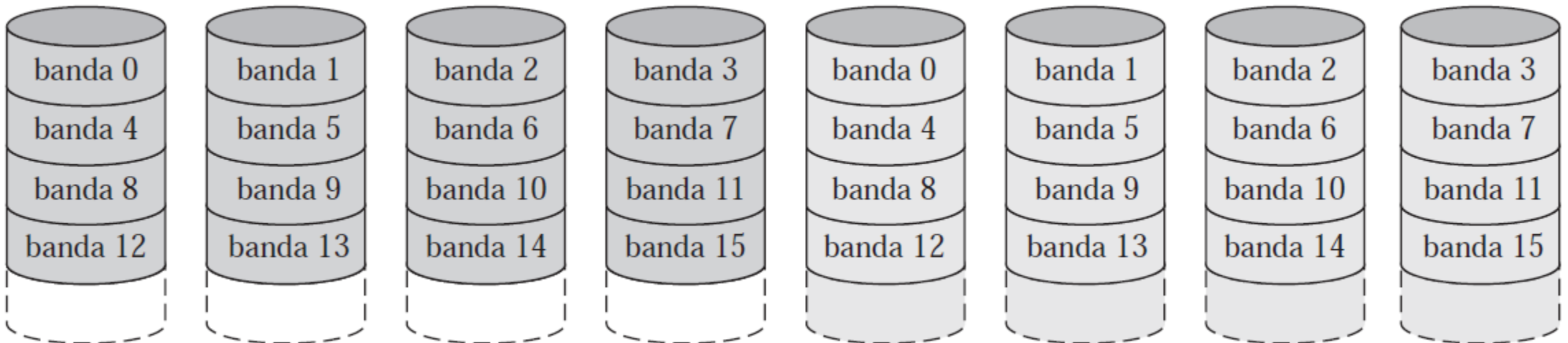
# Raid 0

- No presenta redundancia
- Los datos están distribuidos a lo largo de todos los discos
- Facilita las lecturas en paralelo en el caso de que los datos no se encuentren en el mismo disco.



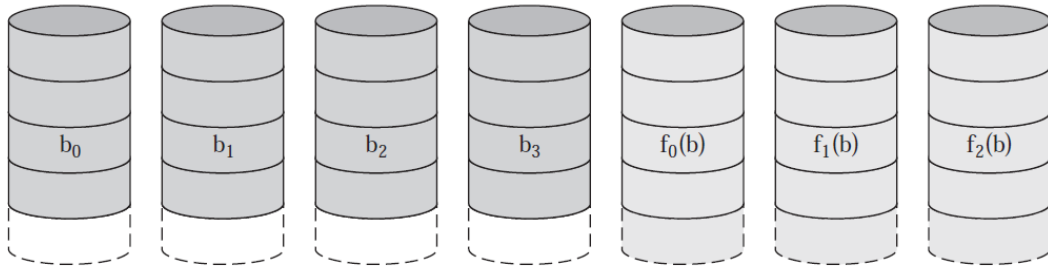
# Raid 1 (mirroring)

- Una petición de lectura puede servirse de cualquiera de los 2 discos
- Una petición de escritura requiere actualizar ambas bandas.
- Proporciona tolerancia a fallos. Si un disco se rompe, se pueden recuperar los datos desde el otro disco.
- Requiere doble de espacio

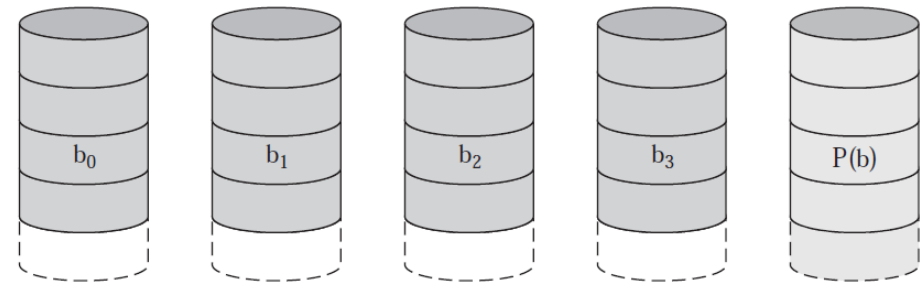




# RAID 2 y RAID 3

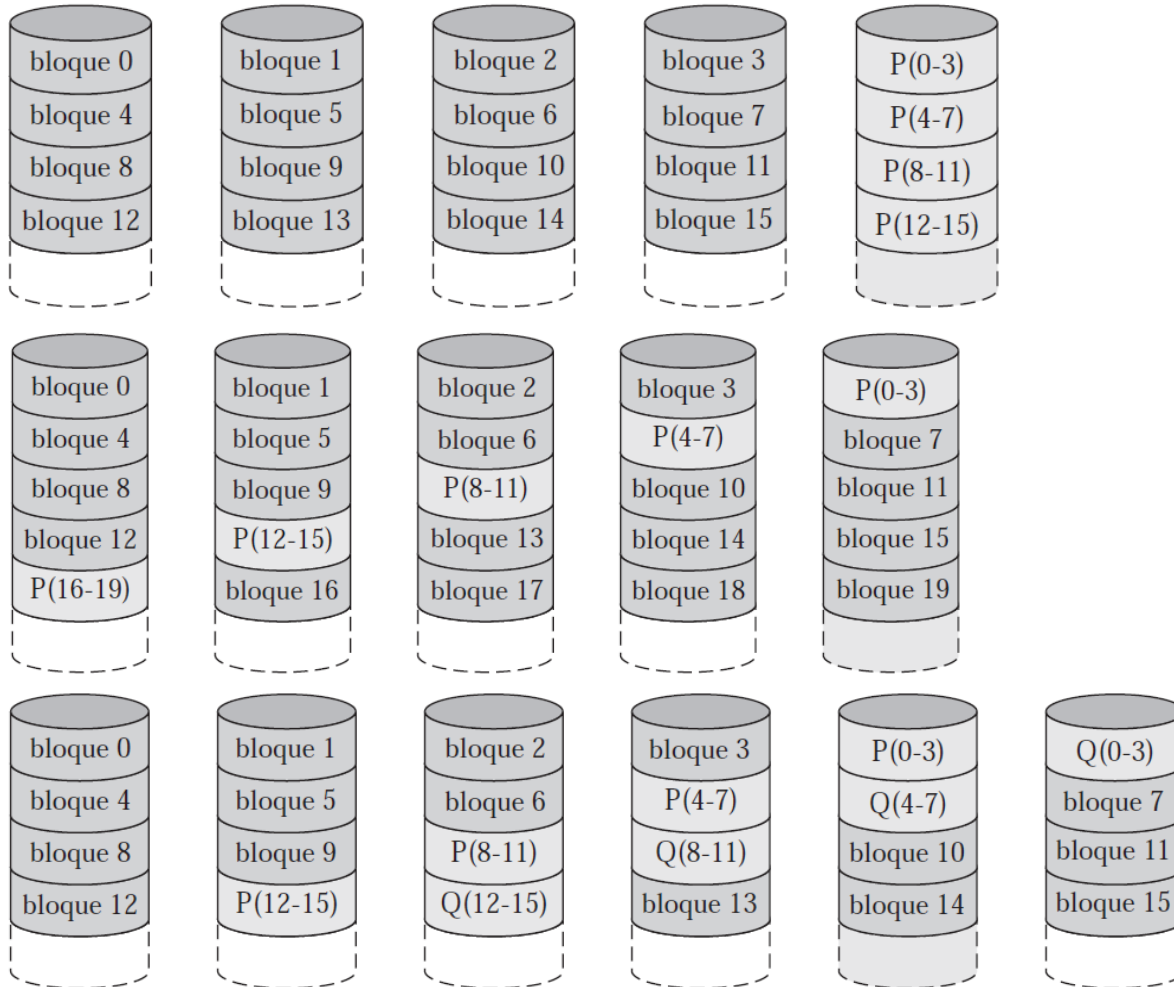


RAID 2 – Corrección de errores mediante código Hamming



RAID 3 – Paridad a nivel bit

# RAID 4,5 y 6



- RAID 4 – Paridad distribuida a nivel bloque
- RAID 5 – Paridad distribuida a nivel bloque
- RAID 6 – Redundancia dual

Muchas gracias