

8. Cuando una clave "x" y otra clave "y" generan, por función de dispersión, distinta dirección, entonces:
- Una de los dos no será almacenada en el archivo.
 - A puede producir un desborde.**
 - Se produce una colisión.
 - "x" e "y" son sinónimos respecto a la función dispersión.
 - Hay más de una opción correcta.

b. "Se puede producir un desborde" como posible,
porque aunque vayan a diferentes direcciones, cada dirección puede llenarse si la zona primaria está saturada.

9. La Doble Dispersión:
- Evita que sucedan desbordes. *no evita desbordes*.
 - Evita que sucedan colisiones. *no evita colisiones, solo ayuda a manejarlas*
 - No es aplicable a la dispersión estática. *si aplicable*
 - Siempre requiere más espacio físico que la Saturación Progresiva. *usan lo mismo*
 - Ninguna de las opciones anteriores es correcta.

10. Una colisión:
- Siempre produce saturación.
 - Solo ocurre cuando la dirección se encuentra vacía.
 - Algunas veces produce saturación.** *no siempre produce saturacion*
 - Nunca puede producir saturación
 - Ninguna de las opciones anteriores es correcta.

7. Si se quiere dispersar un archivo de 40.000 elementos

- Se requiere un archivo de 40.000 cubetas
- Se requiere un archivo de al menos 40.000 cubetas
- Se requiere un archivo de menos de 40.000 cubetas
- Todas las anteriores
- Algunas de las anteriores
- Ninguna de las anteriores**

Clave: la cantidad de cubetas NO está predeterminada por la cantidad de elementos.
Depende de la tasa de carga, del espacio deseado, del método de dispersión, etc.

Porque nada en el enunciado fija la cantidad de cubetas que se deben usar.
Podés usar 10.000, 20.000, 40.000 ó 80.000.
No hay número obligatorio.

8. Si ocurrió saturación
- No hubo colisión *falso, Si hubo saturación, si o hubo colisiones.*
 - Pudo haber ocurrido colisión *falso, Si hubo saturación, si o hubo colisiones.*
 - Hay más de 2 claves sinónimo *No necesariamente.*
 - Todas las anteriores *Podría saturarse una cubeta de tamaño 1 con solo dos claves sinónimas.*
 - Algunas de las anteriores
 - Ninguna de las anteriores**

9. La densidad de empaquetamiento
- Es un parámetro de eficiencia en cualquier tipo de hashing
 - Es un parámetro de eficiencia sólo para un tipo de hashing** *estático*
 - Permite detectar si la cantidad de espacio libre en el archivo puede crecer
 - Permite detectar si la cantidad de elementos del archivo puede crecer
 - Algunas de las anteriores
 - Ninguna de las anteriores

c. falso. La densidad mide cuán lleno está el archivo, no predice si el espacio libre va a crecer.

d. La densidad no te dice si "puede crecer" solo te dice cuánto está ocupando ahora.

El crecimiento depende de la técnica de hashing, no de la densidad.

¿Qué es la densidad de empaquetamiento en hashing?
Es la relación entre la cantidad de registros almacenados y la capacidad total del archivo hash.
Se usa para medir qué tan lleno está el archivo.
No se usa en todos los hashing.. sólo en ciertos métodos.

1. Cual de las siguientes definiciones corresponden al método de hash:
- Técnica para generar una dirección base única para una clave dada.
 - Técnica que convierte la clave asociada a un registro de datos en un número aleatorio, que se utiliza para determinar donde se almacena el registro.
 - Técnica de almacenamiento y recuperación que usa una función para mapear registros en direcciones de almacenamiento en memoria secundaria. *Esto describe el proceso completo de hashing (almacenar + recuperar usando la función hash).*
 - Todas las anteriores son aplicables.

2. La técnica de hash
- Entorpece la inserción y el borrado de elementos *El hashing justamente acelera inserciones y búsquedas.*
 - La localización de un registro siempre debe utilizar una tabla adicional en memoria *pero en archivos hash NO hace falta ninguna tabla adicional.*
 - No es conveniente de aplicar sobre claves secundarias**
 - Requiere al menos dos funciones de hash para el tratamiento de los desbordes.

e. No es conveniente de aplicar sobre claves secundarias." *Correcto.*
El hashing se usa solo sobre claves primarias porque:
- Necesita valores únicos
- Una clave secundaria puede repetirse → generar colisiones inevitables.
- No tendría sentido para acceso directo.

d. Requiere al menos dos funciones de hash para el tratamiento de los desbordes." *Incorrecto.*
Solo algunos métodos usan más de una función (dispersión doble),
pero no es un requisito general del hashing.

3. Cuál de los siguientes conceptos corresponden con parámetros de la dispersión
- Capacidad de almacenamiento de cada sector del archivo
 - Densidad de empaquetamiento *Es un indicador directo del desempeño del hashing.*
 - Método de tratamiento de desbordes
 - Todos los anteriores**

a. Capacidad de almacenamiento de cada sector del archivo *Si.*
Es un parámetro fundamental: define cuántos registros caben por bloque → afecta colisiones y desempeño

b. Densidad de empaquetamiento *Si.*
Es otro parámetro clave: indica cuán lleno está el archivo → afecta probabilidad de overflow.

c. Método de tratamiento de desbordes *Si.*
Esto también es un parámetro de la técnica de hashing → determina qué pasa cuando hay colisiones.

4. La densidad de empaquetamiento se define como
- El cociente entre cantidad de registro y espacio disponible en el archivo
 - El cociente entre la cantidad de registros y la cantidad de nodos del archivo
 - El cociente entre la cantidad de registros, y el producto entre la cantidad de nodos y el contenido posible de registros de cada nodo.
 - Hay más de una respuesta correcta

La densidad de empaquetamiento = cantidad de registros / capacidad total del archivo
(capacidad total = cantidad de nodos * registros por nodo)

5. Una colisión se produce
- Cuando dos registros diferentes obtienen de la función de hash la misma dirección de disco
 - Cuando dos registros iguales obtienen de la función de hash direcciones diferentes de disco
 - Cuando un registro no cabe en el lugar donde debe almacenarse de acuerdo al resultado de la función de hash. *Eso es overflow, no colisión.* La colisión es lógica (dos claves → misma dirección)
 - Cuando dos registros diferentes obtienen de la función de hash direcciones diferentes de disco.

6. El hash asistido por tabla *también llamado "hashing por tabla auxiliar"*

- Utiliza un espacio estático de memoria
- Utiliza un espacio dinámico de memoria
- No requiere de espacio adicional en memoria *requiere espacio adicional para la tabla auxiliar.*
- Utiliza solo una función de hash para su desarrollo *Generalmente usa dos pasos*

La tabla auxiliar se reserva completa desde el inicio, con tamaño fijo → espacio estático.

7. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

8. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Orden Lineal
 - Algunas veces es uno**
 - Siempre es uno
 - Orden Logarítmico

9. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

10. La técnica de hash extensible
- Presenta una variante de hash que permite no solo ubicar rápidamente los registros, sino que además permite el acceso secuencial a los mismos. *El hash extensible no da acceso secuencial*
 - Siempre inserta un registro con un y solo un acceso a disco *La inserción a veces implica split del bucket,*
 - Siempre se recupera un registro con un y solo un acceso a disco. *La búsqueda SIEMPRE son 1 acceso a disco*
 - En algunos casos recuperará un registro con un y solo un acceso a disco. *Es siempre.*

11. Cual de los siguientes parámetros no corresponde a hashing
- Función de hash
 - Tamaño del nodo (capacidad para almacenar registros)
 - Densidad de búsqueda** *Lo correcto es densidad de empaquetamiento,*
 - Forma de tratar los desbordes
 - Todas las anteriores corresponden

La catedra SOLO considera 3 parámetros como "parámetros de eficiencia":

- Densidad de empaquetamiento
- Función de dispersión
- Tratamiento de los sinónimos

Las opciones a y b NO son "parámetros de eficiencia" según definición estricta.

27. Si se tiene una política de hash con espacio de direcccionamiento dinámico
- la densidad de empaquetamiento puede ser mayor que 1 *La densidad SIEMPRE está entre 0 y 1.*
 - cuando la densidad de empaquetamiento supera el 75% se debe reacomodar al archivo *Es un límite rígido de densidad como en hash estático.*
 - cuando la densidad de empaquetamiento supera el 75% se debe utilizar más espacio para nodos *Es otro límite rígido de densidad como en hash estático.*
 - cuando la densidad de empaquetamiento supera el 75% debe activarse una política de tratamiento de overflow, como por ejemplo área de desborde separado.
 - ninguna de las anteriores es válida**

En hash dinámico, el archivo crece solo cuando hace falta, pero NO por un porcentaje fijo como 75%.

28. A partir de un archivo dispersado con hashing extensible

- Siempre es posible agregar elementos al archivo *si*
- Algunas veces no es posible agregar elementos al archivo *no*
- Se puede utilizar Saturación Progresiva Encadenada para tratar registros en saturación *no*
- No se puede utilizar Densidad Doble para tratar registros en saturación *si*
- Algunas de las anteriores**
- Ninguna de las anteriores es correcta

En hashing extensible:

Siempre se puede insertar → porque el archivo crece dinámicamente.

No existe un límite rígido de densidad como en hash estático.

Y NO usan políticas de overflow tipo "área de desborde separada".

29. La técnica de hashing extensible
- Utiliza una función de hash porque debe cambiar dinámicamente *no*
 - Reduce la densidad de empaquetamiento *solo en hashing estático*
 - Utiliza una segunda función de hash para ubicar rápidamente los registros, pero no es posible incorporar más elementos.
 - Siempre se aplica la dispersión extensible.
 - Afecta la densidad de empaquetamiento sólo en casos especiales
 - Todas las anteriores son validas
 - Algunas de las anteriores
 - Ninguna de las anteriores**

30. La eficiencia de búsqueda de un registro en un archivo organizado mediante dispersión estática tiene:

- Es Orden Lineal
- Algunas veces es uno**
- Siempre es uno
- Orden Logarítmico

31. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

32. El hash asistido por tabla *también llamado "hashing por tabla auxiliar"*
- Utiliza un espacio estático de memoria
 - Utiliza un espacio dinámico de memoria
 - No requiere de espacio adicional en memoria *requiere espacio adicional para la tabla auxiliar.*
 - Utiliza solo una función de hash para su desarrollo *Generalmente usa dos pasos*

La tabla auxiliar se reserva completa desde el inicio, con tamaño fijo → espacio estático.

33. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

34. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Es Orden Lineal
 - Algunas veces es uno**
 - Siempre es uno
 - Orden Logarítmico

35. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

36. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

37. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Es Orden Lineal
 - Algunas veces es uno**
 - Siempre es uno
 - Orden Logarítmico

38. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

39. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

40. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Es Orden Lineal
 - Algunas veces es uno**
 - Siempre es uno
 - Orden Logarítmico

41. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

42. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

43. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Es Orden Lineal
 - Algunas veces es uno**
 - Siempre es uno
 - Orden Logarítmico

44. Cuando la Densidad de empaquetamiento de un archivo tiende a uno
- Es necesario redefinir el espacio disponible únicamente. *No alcanza con "redefinir espacio": ya no hay lugar físico real.*
 - El archivo se completa y no es posible incorporar más elementos.
 - Se debe cambiar la política de hash de estática a dinámica. *Eso sería una decisión de diseño*
 - Es necesario redefinir el espacio disponible y rehashear todo el archivo** *Correcta, y es justo lo que se hace en hashing estático cuando se llena.*
hay que aumentar la capacidad
y hay que rehash para que los registros se redistribuyan correctamente en el archivo nuevo

Cuando tiende a 1 → el archivo está prácticamente lleno.

45. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:
- Orden lineal
 - Orden logarítmico
 - Orden constante**
 - No dispongo datos para contestar la pregunta

46. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:
- Es Orden Lineal
 - Algun**

15. Para tener un bajo porcentaje de densidad de empaquetamiento

- a. **se deben tener disponibles muchos nodos de los necesarios**
- b. se debe utilizar una función de hash que genere pocas colisiones
- c. se deben tener nodos con capacidad de almacenar más de 100.000 registros
- d. se debe utilizar hash con espacio de direccionamiento dinámico
- e. todas las anteriores sirven para bajar el porcentaje de la densidad de empaquetamiento
- f. ninguna respuesta es correcta

Usa una estructura auxiliar llamada directorio (SIEMPRE está en memoria)

12. Cuando una clave "x" y otra clave "y" generan, por función de dispersión, diferente dirección, entonces:

- a. Una de las dos no será almacenada en el archivo.
- b. **Se puede producir un desborde.**
- c. Se produce una colisión.
- d. Hay más de una opción correcta.
- e. Ninguna es correcta

15. El hash con espacio de direccionamiento dinámico

- a. **Encuentra los registros con un solo acceso a disco**
- b. Se puede aplicar a cualquier clave
- c. No requiere de espacio adicional
- d. Todas las anteriores
- e. Algunas de las anteriores son válidas
- f. Ninguna de las anteriores es válida

22. Una función de hash en teoría debe ser

- a. **Uniforme y aleatoria**
- b. Uniforme pero no aleatoria
- c. Aleatoria pero no uniforme
- d. No debe ser ni uniforme ni aleatoria

23. La densidad de empaquetamiento

- a. Mide la relación entre espacio disponible y espacio utilizado
- b. Mide la relación entre la cantidad de registros que tiene un archivo y la capacidad de almacenamiento del mismo
- c. Es el cociente entre el número de registros del archivo disponible y el espacio disponible
- d. **Todas las anteriores son válidas**
- e. Algunas de las anteriores son válidas
- f. Ninguna de las anteriores es válida

24. En caso de que se produzca overflow cuando se inserta sobre un nodo cualquiera

- a. Se puede utilizar saturación progresiva para su tratamiento
- b. Se puede utilizar saturación progresiva encadenada para su tratamiento
- c. Se puede utilizar doble paginación para su tratamiento
- d. Se puede utilizar área de desborde por separado para su tratamiento
- e. **Todas las anteriores son válidas**
- f. Algunas de las anteriores son válidas
- g. Ninguna de las anteriores es válida

25. El método de área de desborde por separado es

- a. Más eficiente que el método de saturación progresiva
- b. Más eficiente que el método de saturación progresiva encadenada
- c. Más eficiente que el método de doble dispersión por separado
- d. Todas las anteriores son válidas
- e. Algunas de las anteriores son válidas
- f. **Ninguna de las anteriores son válidas**

27. En hashing extensible, cuando la Densidad de empaquetamiento tiende a 1:

- a. Se debe pasar por hash estático a dinámico
- b. Se debe reformular el tamaño del nodo
- c. Se debe aumentar la cantidad de registros por nodo
- d. Algunas de las anteriores
- e. **Ninguna de las anteriores**

28. El método de dispersión con espacio de direccionamiento dinámico, denominada has extensible:

- a. Guarda los registros de forma ordenada por algún criterio
- b. **Siempre requiere el uso de una estructura auxiliar**
- c. Siempre requiere variar el tamaño del espacio de direcciones
- d. Necesita de dos funciones de dispersión
- e. Ninguna de las opciones anteriores es correcta

29. Con 10000 direcciones con capacidad para 4 registros cada una y 30000 claves para dispersar, entonces

- a. La densidad de empaquetamiento es inferior al 50% pero mayor o igual al 25%
- b. La densidad de empaquetamiento es inferior al 75% pero mayor o igual al 50%
- c. La densidad de empaquetamiento es mayor que uno
- d. **La densidad de empaquetamiento es superior o igual al 75%**
- e. No se puede calcular

30. De los cuatro métodos de tratamiento de desbordes en hash estático vistos, cual de ellos potencialmente no genera zonas de overflow contiguas dentro del espacio de direccionamiento de la función de hash.

- a. Saturación progresiva
- b. Saturación progresiva encadenada
- c. **Doble dispersión**
- d. Área de desborde separado

3. Según el estado de un archivo directo para dispersión extensible y bloques o cubetas con capacidad para 3 registros:

- a. El alta de la clave 30845 (...1) no produce overflow.
- b. **El alta de la clave 12640 (...0) no produce overflow.**
- c. El alta de la clave 30845 (...1) produce overflow, pero no es necesario duplicar la tabla.
- d. El alta de la clave 12640 (...0) produce overflow, pero no es necesario duplicar la tabla.
- e. Hay más de una opción válida.
- f. Todas las opciones anteriores son correctas.
- g. Ninguna de las opciones anteriores es correcta.

11. La saturación progresiva encadenada

- a. Se usa en caso de colisión
- b. Se usa en caso de colisión y saturación
- c. Se usa en caso en caso de saturación
- d. **Se puede usar en caso de saturación**
- e. Todas las anteriores
- f. Algunas de las anteriores
- g. Ninguna de las anteriores

13. La densidad de empaquetamiento

- a. Se modifica con cada alta o baja, utilizando hashing extensible
- b. Se modifica con cada alta, utilizando hashing extensible
- c. Se modifica con cada baja, utilizando hashing extensible
- d. No se modifica con altas o bajas, utilizando hashing extensible
- e. Todas las anteriores
- f. Algunas de las anteriores
- g. **Ninguna de las anteriores**

En hashing extensible NO se usa el concepto de densidad de empaquetamiento.

13. La densidad de empaquetamiento

- a. Se modifica con cada alta o baja, utilizando hashing extensible
- b. Se modifica con cada alta, utilizando hashing extensible
- c. Se modifica con cada baja, utilizando hashing extensible
- d. No se modifica con altas o bajas, utilizando hashing extensible
- e. Todas las anteriores
- f. Algunas de las anteriores
- g. Ninguna de las anteriores**