

# RECOPILATORIO PREGUNTAS TIPO TEST.pdf



**EasyIngeniero**



**Fundamentos de Bases de Datos**



**2º Grado en Ingeniería Informática**

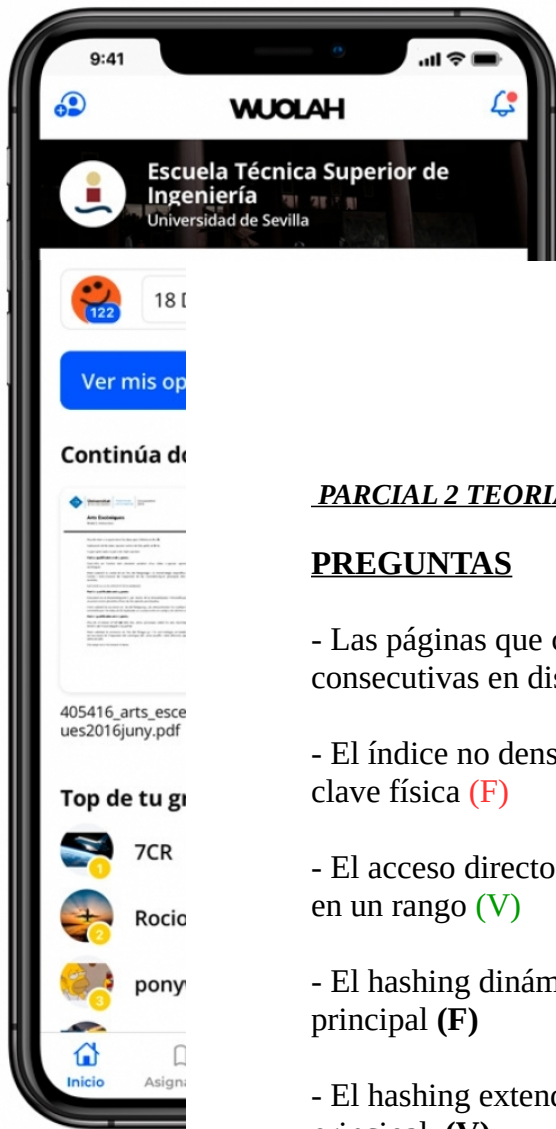


**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.





**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



## PARCIAL 2 TEORIA FBD TEMA 4 NIVEL INTERNO: PREGUNTAS

### PREGUNTAS

- Las páginas que componen un archivo almacenado no tienen por qué estar consecutivas en disco (V)
- El índice no denso mejora el barrido ordenado completo del fichero por la clave física (F)
- El acceso directo a registros no permite realizar la lectura secuencial de datos en un rango (V)
- El hashing dinámico es muy eficaz porque la tabla hash va en memoria principal (F)
- El hashing extendido es muy eficaz porque la tabla hash va en memoria principal. (V)
- Se pueden montar tantos índices densos como se necesiten (V)
- En un índice multinivel el índice de primer nivel (nodo hoja) puede ser denso o no denso (V)
- El acceso directo a registros garantiza siempre que encuentre una tupla con una sólo lectura de bloque (F)
- El acceso directo a bloques o cubos produce menos lecturas en disco que el acceso directo a registros (V)
- El índice denso es adecuado para consultas por rangos de valores del campo clave (V)
- Los bloques usados para almacenar los datos de la BD pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenen en ellos (F)
- El índice no denso permite realizar preguntas de tipo existencial sin acceder al fichero de datos (F)
- El hashing dinámico es el método de acceso que mejor distribuye los datos en disco y, por tanto, el que menos desperdicio ocasiona (V)

- Se pueden montar tantos índices no densos como sea necesario (F)
- Para montar un índice denso, los registros tienen que estar ordenados físicamente por algún campo (F)
- El índice denso ocupa el mismo tamaño que el propio fichero que indexa (F)
- En ficheros no ordenados físicamente, no se pueden montar índices no densos (V)
- El índice no denso es mucho menor que el denso cuando caben varios registros en un bloque (V)
- Las sentencias CREATE TABLE y CREATE INDEX de SQL generan nuevos conjuntos de páginas (archivos almacenados) en el nivel interno (V)
- El rendimiento de un índice no denso desciende considerablemente cuando se realizan inserciones o borrados (F)
- En acceso directo a registros, si se produce una colisión, habrá un hueco en el fichero maestro que nunca se va a aprovechar. (V)
- El índice denso no es rentable cuando se actualiza o se inserta con mucha frecuencia (V)
- El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente (F)
- En hashing dinámico hace falta una estimación del número de datos a insertar para dimensionar la tabla hash (V)
- El gestor de disco forma parte del SGBD (F)
- Lo normal es que cada archivo almacenado del nivel interno se almacene en un fichero físico separado (F)



**KEEP  
CALM  
AND  
ESTUDIA  
UN POQUITO**

- El objetivo principal de los mecanismos de indexación y métodos de acceso es: **localizar datos requeridos con el número mínimo de operaciones de lectura en disco**

- La sentencia CREATE TABLE provoca : **la creación de un nuevo archivo almacenado**

- El clúster

a. Acelera las consultas que involucran la reunión natural de las tablas que contiene

b. Perjudica la lectura individual de las tablas que contiene

c. Es una estructura inter-archivo

**d. Todo lo anterior es cierto**

- En un índice denso :

**El número de registros del índice es igual al número de registros de la tabla indexada.**

- Indica la afirmación verdadera

a. El índice no denso es el único mecanismo de indexación posible para los datos están ordenados físicamente

b. Un árbol B solo se puede montar sobre la clave física de un archivo

c. Un fichero no ordenado físicamente, se pueden montar índices no densos

**d. Se pueden montar tantos índices densos como se necesiten**

- La técnica de acceso directo

**a. No permite realizar la lectura secuencial de datos de un rango**

- En la indexación con árboles B

a. El orden de un árbol está determinado por el tamaño de página / bloque que se asigna a los nodos

b. Se puede montar un árbol B sobre cualquier campo

c. El orden de un árbol influye directamente en el número de niveles

**d. Todo lo anterior es cierto**

- El hashing dinámico, si el número de registros por bloque es 4 y tengo alrededor de 1000 registros, el número de bits necesario para la tabla hash es : **8**  
 **$1000/4=250$ ,  $2^8=256$ , satisface los 250**

- Un índice no denso :

**Exige que los registros estén ordenados físicamente**

- Las páginas que componen un archivo almacenado

a. Tienen que estar consecutivas en disco

b. Pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenen en ellas

c. Contienen siempre registros de una misma tabla

**d. Todo lo anterior es falso**

- El índice invertido

**Ayuda a distribuir mejor los datos en el espacio de almacenamiento**

- Cuando la cardinalidad del campo por el que se indexa una tabla es muy baja, el mejor mecanismo de indexación es:

**Un índice de mapa de bits**

- Con la consulta `select codpro, sum(cantidad) from ventas group by codpro`

**Se puede crear una vista, pero no será actualizable**

- Sean F y D las tablas procedentes de una entidad fuerte y una débil, respectivamente.

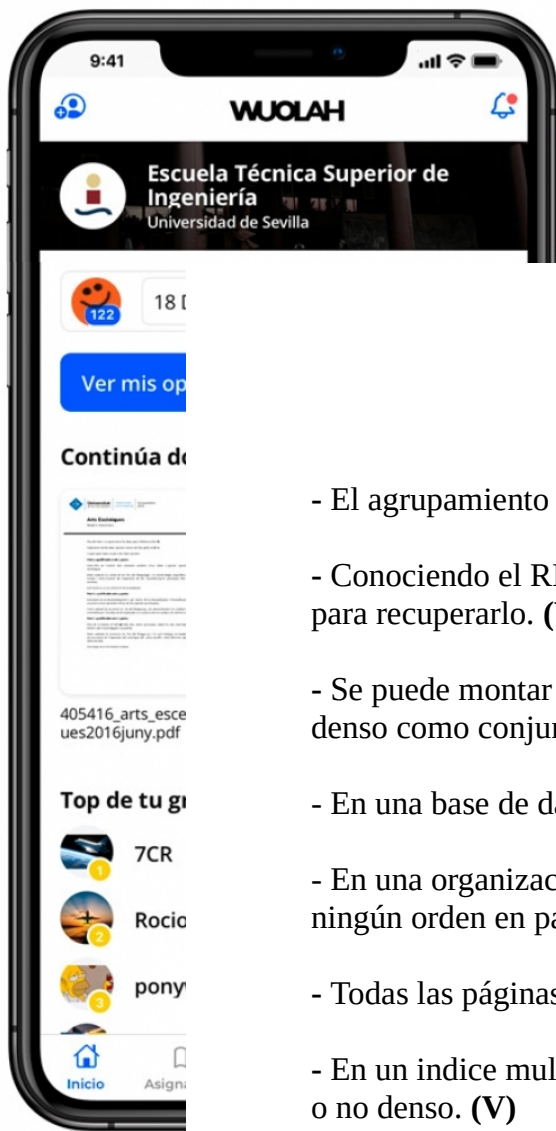
Las filas de D se recuperan siempre reunidas con las de F, y rara vez, por separado. La mejor opción sería:

**Almacenarlas conjuntamente en un clúster**

- Cuando se necesita acceder a la tabla Alumnos por rangos de NOTAS, el mejor mecanismo es:

**Un índice denso**

- Para búsquedas basadas en el valor de la clave, los árboles B **garantizan el acceso a un número fijo de páginas de base de datos.**



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



- El agrupamiento por defecto en el nivel interno es intra-archivos (V)
- Conociendo el RID de un registro no hace falta más que un acceso a disco para recuperarlo. (V)
- Se puede montar un árbol B sobre cualquier campo clave utilizando un índice denso como conjunto secuencia. (V)
- En una base de datos puede haber más de un índice primario (F)
- En una organización secuencial no es necesario que los registros mantengan ningún orden en particular. (V)
- Todas las páginas de una base de datos tienen la misma estructura. (F)
- En un índice multinivel el índice de primer nivel (nodo hoja) puede ser denso o no denso. (V)
- Los árboles B se montan en memoria para no tener que acceder a disco más que una vez para llegar al registro. (F)
- En un índice denso el número de elementos **es el mismo que el del archivo principal.**
- En el nivel interno de una BD hay que tener en cuenta también el nivel físico que gestiona el acceso a disco (V)
- Cuando la clave de un índice es compuesta (C1, C2) **resulta eficiente el uso del índice para buscar por C1 o por C2.**
- El número de accesos a disco que hacen falta para obtener una página **depende del tamaño de la página y del tamaño del bloque físico.**
- Las actualizaciones en los archivos almacenados no tienen por qué determinar la actualización de los índices no densos. (V)
- En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente la estructura. (V)
- La organización multilistas es independiente de las técnicas de acceso al archivo almacenado. (V)

- Conviene que estén relacionados el tamaño de los bloques físicos y de las páginas para mejorar el rendimiento de sistema de almacenamiento. **(V)**
- El orden de un árbol influye directamente en el número de niveles. **(V)**
- El orden de un árbol B fija el número de punteros que salen de un nodo. **(V)**
- En el agrupamiento inter-archivo se ubican en una página registros de distinto tipo. **(V)**
- El orden de un árbol está determinado por el tamaño de la página que se asigna a los nodos del árbol. **(V)**
- Se puede combinar una organización multilistas con un árbol B para gestionar los accesos en una estructura de datos jerárquica. **(V)**
- Puesto que es una variante de índice no denso, solo se puede montar un árbol B (sobre la clave física) de un archivo. **(F)**
- En el hashing extendido lo mejor es que la pseudollave se ajuste el tamaño del índice que se guarda en memoria **(V)**
- La organización multilistas puede servir para conectar ficheros y es la base de datos basados en grafos. **(V)**
- Un factor de bloqueo mayor a 1 implica tener más de un registro por página. **(V)**
- La organización IOT almacena las filas de una tabla en orden físico por clave principal **(V)****(aunque se pueden definir índices adicionales basados en otras claves)**
- Usando como función “hasing” la función módulo 8 y con 3 registros cubo, los registros con valor de clase {1, 4, 9, 0, 17, 25}, NO producen desbordamiento
- El administrador de la base de datos puede decidir la forma de agrupamiento en páginas de los archivos que corresponden a las tablas de una base de datos
- CREATE TABLE genera la inserción de varias filas en diversas tablas del catalogo



- Para conceder privilegios sobre un objeto, hay que ser necesariamente el dueño del objeto
- CREATE CLUSTER permite establecer un agrupamiento físico de registros que pertenece a distintas tablas en el nivel conceptual
- Todos los usuarios pueden consultar todas las vistas del catálogo, pero no modificar su contenido
- El índice del mapa de bits no puede crearse sobre un atributo numérico
- CREATE TABLE genera un conjunto de páginas asociado a un archivo a nivel interno **(V)**
- El nivel interno de una base de datos está enteramente gestionado por el S.O del ordenador **(F)**
- Nunca es recomendable crear un índice bitmap sobre una clase primaria
- La insercción a través de una vista se podrá realizar siempre y cuando se suministre el número adecuado de atributos y en las posiciones correctas
- La finalidad de un índice por clave invertida es alejar físicamente los registros cuyas claves son consecutivas.
- En las tablas organizadas por índice, las tuplas consultadas conforme al valor de la clave se recuperan mediante los RIDs del conjunto secuencia.
- El “hashing” extendido permite asignar más almacenamiento a determinados rangos de la clave y menos a otros.
- si sé cuantos registros va a tener el archivo almacenado y cuantos valores distintos de la clave, puedo dimensionar adecuadamente el acceso directo a cubos
- La clave de una tabla organizada por índice puede estar definida sobre cualquiera de sus campos.

- Las consultas basadas en OR sobre campos indizados mediante índices bitmaps, permiten obtener las direcciones de las tuplas que satisfacen la condición directamente de los índices.
- Se pueden crear tantos índices no densos sobre un archivo almacenado como se desee. **(F)**
- No se pueden resolver consultas basadas en AND sobre dos campos indizados mediante índices bitmaps, usando estos índices.
- En un clúster formado por las tablas proveedor y ventas, la consulta “select \* from proveedor p, ventas v where p.codpro=v.codpro” aumenta considerablemente su rendimiento.
- En el “hashing” extendido no se producen desbordamientos.
- El RID(Record Identifier)

**a) es un campo de un índice denso**

b) es un campo de un índice no denso

c) puede servir para identificar varios registros

d) se calcula mediante un algoritmo de direccionamiento

- El índice por clave invertida

a) Se puede montar sobre cualquier campo de la tabla

b) Ayuda a distribuir mejor los datos en el espacio de almacenamiento

c) Es útil para recorrer una tabla por el campo clave en orden inverso al establecido

**d) Todo lo anterior es cierto**

- Indical cuál de estas afirmaciones es verdadera:

a) El tamaño de los bloques físicos y de las páginas deben ser independientes.



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



- b) Cada archivo almacenado del nivel interno debe estar en un fichero físico separado
- c) El nivel interno de una base de datos está enteramente gestionado por el S.O. del ordenador
- d) Las páginas que componen un archivo almacenado no tienen por qué estar consecutivas en disco.**

- Las tablas organizadas por índice
- a) No admiten ningún tipo de índice
- b) No tienen llave primaria
- c) No pueden recuperarse de forma ordenada

**d) Se organizan como un árbol B cuyas hojas contienen las tuplas.**

- Indica cual de estas afirmaciones es falsa

- a) En el hashing extendido una mala elección en el tamaño de las páginas puede obligar a reorganizar completamente la estructura
- b) El hashing extendido es muy eficaz porque la tabla hash va en memoria principal
- c) El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente**
- d) La actualización de los archivos puede no influir en la actualización de los índices no densos

- El numero de niveles de un árbol B depende sólo del tamaño del fichero base **(F)**

- Se puede combinar una organización multilistas con un hashing extendido para gestionar los accesos en una estructura de datos jerárquica **(V)**

- La técnica de acceso directo garantiza siempre que encuentre una fila con una sola lectura de bloque **(F)**

- La técnica de acceso directo no necesita una estimación previa del número de registros **(F)**

- El objetivo principal de los mecanismos de indexación y métodos de acceso es garantizar que no se duplique la clave primaria. **(F)**

- El objetivo principal de los mecanismos de indexación y métodos de acceso es acceder a los datos de una tabla de forma ordenada. **FALSO.**

- En un índice denso el índice ocupa lo mismo que la tabla indexada. **(F)**

- En un índice denso el número de registros del índice es menor que el número de registros de la tabla indexada. **(F)**

- En el hashing dinámico lo mejor es que la pseudollave tenga muchos dígitos. **(F)**

- En el hashing dinámico hay que reservar de antemano un número fijo de bloques. **FALSO, FIJO NO, UNA ESTIMACIÓN.**

- En el hashing dinámico no hay que tener una estimación del número de registros a almacenar. **(F)**

- Un índice no denso es adecuado para consultas por rangos de valores del campo clave. **(F)**

- Con la consulta [SELECT codpro, sum(cantidad) from ventas GROUP BY codpro] no se puede crear una vista por estar agrupada. **(F)**

- Con la consulta [SELECT codpro, sum(cantidad) from ventas GROUP BY codpro] se puede crear una vista y será actualizable porque sólo usa una tabla. **(F)**