

# Fundamentos de Bases de Datos

## Práctica 3

Miguel Lozano Alonso

Cristina Rodríguez de los Ríos Medina

<b>Objetivos</b>	<b>3</b>
<b>Algoritmos a implementar</b>	<b>3</b>
1. replaceExtensionByIdx	3
2. createTable	3
3. createIndex	3
4. printTree	3
5. findKey	3
6. addTableEntry	3
7. addIndexEntry	4

## Objetivos

En esta práctica se van a implementar una serie de funciones para crear una base de dato que recoge los datos de una serie de libros. Para ello, se va a hacer uso de un menú que la hará funcional, haciendo uso de funciones para crear la base de datos, borrar registros, insertar nuevos e imprimir los registros que pida el usuario.

## Algoritmos a implementar

### 1. replaceExtensionByIdx

Cambia la extensión “.dat” de los ficheros que contienen los datos a la extensión “.idx” para los ficheros que contienen los índices de ese fichero, por ello el nombre antes de la extensión se mantiene.

### 2. createTable

En el caso de que exista el fichero pasado como argumento, lo abre y comprueba que no se encuentre vacío. Si el fichero está vacío:

1. inicializa la cabecera que contiene el inicio del listado de registros borrados a -1
2. crea el fichero de índices haciendo una llamada a **createIndex**

Si el fichero no existe, lo crea y sigue los pasos para ficheros vacíos.

### 3. createIndex

Crea el fichero de índices e inicializa ambos punteros de la cabecera, nodo raíz y primer nodo borrado, a -1.

### 4. printTree

Imprime el árbol en el fichero proporcionado hasta la profundidad indicada. Para ello se ha creado la función **printnode**, que imprime el árbol a través de llamadas recursivas.

### 5. findKey

Lee la cabecera del archivo para encontrar el nodo raíz. Se sitúa en el nodo raíz y comienza a leer el registro. Compara la clave leída con la clave buscada y si es menor la buscada, lee el hijo izquierdo. y si la clave buscada es mayor, lee el hijo

derecho. Así hasta encontrar la clave o descubrir que la clave no pertenece a el archivo de índices.

## 6. addTableEntry

Esta función añade la información almacenada en una estructura tipo book en el fichero de datos que recibe. Para ello hace una llamada a la funcion **addTableIndex**.

El procedimiento es el siguiente:

1. Se verifica si la clave existe haciendo uso de **findKey**.
  - a. Si existe, devuelve *false* y muestra un mensaje de error.
2. Comprueba si hay algún registro borrado accediendo a la cabecera.
  - a. Si no hay registros borrados, se añade la estructura que se ha pasado a la función al final del fichero con el formato que se utiliza en la base de datos. Se debe tener en cuenta el offset, que es necesario para actualizar el índice.
  - b. Si hay registros borrados, se comprobará si hay alguno lo suficientemente grande y se utilizará el primer espacio disponible, actualizando posteriormente tanto la cadena de registros borrados como el tamaño del registro eliminado utilizado (si este no se ha utilizado por completo).
3. Actualiza el fichero de índices, haciendo uso de **addTableIndex**, que actualizará el índice agregando un nuevo nodo según las instrucciones descritas en el enunciado. Al igual que en el caso anteriormente descrito, se utilizarán los registros eliminados si es posible. se debe tener en cuenta que en el fichero índice todos los registros son del mismo tamaño.

## 7. addIndexEntry

Añade una entrada a la tabla de índices.