

INDICE DE CAMBIOS REALIZADOS POR LOS USUARIOS

VarLenBPlusTree:

Esta clase de árbol se almacena en un archivo en bloques de longitud fija. Para almacenar datos administrativos del mismo se utiliza el 1º bloque y los restantes contienen a los nodos propiamente dichos.

Este tipo de índice se utiliza para poder indexar las modificaciones realizadas por los usuarios al repositorio. Indexa a un archivo secuencial en bloques en el que se almacenan secuencialmente las modificaciones realizadas al repositorio por los usuarios; secuencialmente se pueden obtener las modificaciones de cada usuario. El árbol indexa la 1ª referencia de cada bloque del archivo mencionado. Este archivo está detallado en su estructura física y lógica más abajo.

Estructura Lógica de VarLenBPlusTree:

(nivel, id, padre, nClaves, espacioLibre, ((hijoIzquierdo, (clave, referencia)+) |
(hnoDerecho, hnoIzquierdo, (clave, referencia)+)))

Estructura Física de VarLenBPlusTree:

(nivel: entero, id: entero, padre: entero, nClaves: entero, espacioLibre: entero, (
(hijoIzquierdo: entero, (longClave: entero, clave: cadena, referencia: entero)+) |
(hnoDerecho: entero, hnoIzquierdo: entero, (longClave: entero, clave: cadena, referencia:
entero)+)))

UsersRegisterFile:

Este archivo es indexado por un índice B+ VarLenBPlusTree.

Esta estructurado en bloques que mantienen secuencialidad entanto almacenen referencias al archivo de log acerca de los cambios que un solo usuario en particular realiza sobre el repositorio. Es decir, vamos a tener al menos un bloque por usuario y podríamos tener mas si este usuario realiza tantos cambios como para no poder almacenar todas las referencias a los cambios de ese usuario en un solo bloque.

Estructura Lógica:

(id, anterior, siguiente, espacioLibre, cantReferencias, (referencia)+)

Estructura Física:

(id:entero, anterior: entero, siguiente: entero, espacioLibre: entero, cantReferencias: entero,
(referencia: entero)+)

INDICE DE LAS VERSIONES DE ARCHIVOS/DIRECTORIOS

FileDirBPlusTree:

Esta clase de árbol se almacena en un archivo de estructura tanto física como lógicamente igual que VarLenBPlusTree.

La diferencia de clases se hace porque esta clase tiene métodos distintos al realizar las búsquedas.

Este índice indexa un archivo secuencial en bloques en el que se almacenan la información administrativa de las diferentes versiones de los archivos/directorios almacenados en el repositorio. Se almacenan las primeras claves de cada bloque del archivo antes mencionado.

La estructura lógica y física del archivo en bloques se explica mas abajo.

Estructura Lógica:

(nivel, id, padre, nClaves, espacioLibre, ((hijoIzquierdo, (clave, referencia)+) |
(hnoDerecho, hnoIzquierdo, (clave, referencia)+)))

Estructura Física:

(nivel: entero, id: entero, padre: entero, nClaves: entero, espacioLibre: entero, (
(hijoIzquierdo: entero, (longClave: entero, clave: cadena, referencia: entero)+) |
(hnoDerecho: entero, hnoIzquierdo: entero, (longClave: entero, clave: cadena, referencia:
entero)+)))

FileVersionsFile:

Este archivo es un archivo secuencial en bloques en el que se almacena la informacion administrativa de las diferentes

versiones de los archivos pertenecientes al repositorio. Se va a tener al menos un bloque por cada archivo, y si hay mas de un bloque para alguno de los archivos estos van a estar encadenados de manera tal que se puedan acceder a todas las versiones de ese archivo secuencialmente. Además, cada versión contiene una referencia al offset en el archivo contenedor de versiones del cual se puede obtener la versión en cuestión.

Estructura Lógica:

(id, anterior, siguiente, espacioLibre, cantVersiones, (versionArchivo(numero, original, fecha, usuario, referencia, tipoArchivo('t'l'b'), tipoVersion("BORRADO"|"MODIFICACION")))+)

Estructura Física:

(id: entero, anterior: entero, siguiente: entero, espacioLibre: entero, cantVersiones: entero, (
version(numero: entero, original: entero, fecha: tm, longUsuario: entero, usuario: cadena, referencia:
entero, tipoArchivo: caracter, tipoVersion: entero))+)

Aclaraciones:

La referencia contenida en la versión es la que se usa para acceder al archivo contenedor de versiones de archivo. La estructura de este archivo contenedor se explica a continuación.

El tipo de versión se guarda como un entero, donde 1 = BORRADO y 0 = MODIFICACION.

Container:

Este archivo es un contenedor en el cual se almacenan las diferentes versiones de los archivos. Para los archivos binarios se almacenan los archivos completos dentro del contenedor cada vez que se los versiona. En cambio, para los archivos de texto se guarda el archivo completo cuando es la primera vez que se lo agrega al repositorio o cuando se trata de un "commit" luego de una versión de borrado.

Para todas las demás versiones se almacenan las diferencias.

Estructura Lógica:

((cantidadBytesCopiados, contenido)+)

Estructura Física:

((cantidadBytesCopiados entero, contenido: cadena de bytes)+)

Aclaracion:

Por contenido nos referimos al contenido completo de un archivo o su diff con la version última agregada al repositorio.

DirectoryVersionsFile:

Este archivo es un archivo secuencial en bloques en el que se almacena la información administrativa de las diferentes versiones de los directorios pertenecientes al repositorio. Se va a tener al menos un bloque por cada directorio, y si hay mas de un bloque para alguno de los directorios estos van a estar encadenados de manera tal que se puedan acceder a todas las versiones de ese directorio secuencialmente.

Estructura Lógica:

(id, anterior, siguiente, espacioLibre, cantidadVersiones, (versionDirectorio(numero, usuario, fecha, tipoVersion("BORRADO"|"MODIFICACION"), (nombreArchivo, tipoArchivo("t"|"b"|"d"), version)+))+)

Estructura Física:

(id: entero, anterior: entero, siguiente: entero, espacioLibre: entero, cantVersiones: entero, (versionDirectorio(numero: entero, longUsuario: entero, usuario: cadena, fecha: tm, cantArchivos: entero, (longNombreArchivo: entero, nombreArchivo: cadena, tipoArchivo: caracter, version: entero)+))+)

INDICE DE LAS MODIFICACIONES REALIZADAS AL REPOSITORIO EN UNA FECHA DADA

FixLenBPlusTree:

Esta clase de árbol se almacena en un archivo en bloques de longitud fija. Para almacenar datos administrativos del mismo se utiliza el 1º bloque y los restantes contienen a los nodos propiamente dichos. Este tipo de índice se utiliza para poder indexar las modificaciones realizadas por los usuarios al repositorio en una determinada fecha. Almacena como claves a las fechas de los diferentes días en el formato aaaa/mm/dd y como referencia almacena el offset de la primer línea en el archivo de log que contiene esa fecha.

Estructura Lógica:

```
(nivel, id, padre, nClaves, (  
    ( hijoIzquierdo, (clave, referencia)+ ) |  
    ( hnoDerecho, hnoIzquierdo, (clave, referencia)+ ) ) )
```

Estructura Física:

```
(nivel: entero, id: entero, padre: entero, nClaves: entero, (  
    ( hijoIzquierdo: entero, (clave: caracter[10], referencia: entero)+ ) |  
    ( hnoDerecho: entero, hnoIzquierdo: entero, (clave: caracter[10], referencia: entero)+ ) ) )
```