

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Curso: Manejo e Implementación de Archivos



Proyecto 2: Manual Técnico

Erik Vladimir Girón Márquez

Carnet # 200313492

10/11/06

Sección A+

Introducción

Ercle es una aplicación para almacenamiento de libros en una biblioteca, guardando también datos de sus autores y títulos, y permitiendo la visualización de las portadas de los libros.

El programa implementa un diseño híbrido, implementando las funciones de bajo nivel bajo el lenguaje C utilizando un paradigma procedimental y las funciones de más alto nivel y de interfaz gráfica fueron implementadas en el lenguaje C++, permitiendo un alto nivel de abstracción.

Se utilizó entonces para el desarrollo del proyecto, la plataforma GNU/LINUX. utilizando el compilador GCC. con las APIs de QT bajo xorg 7.1 para el desarrollo de la interfaz gráfica, así como para manejar estructuras de datos mas avanzadas que estas APIs ya contienen por defecto. Además de que el proyecto está desarrollado en el IDE Kdevelop utilizando CVS (local) para el control de versiones.

En este documento, el programador se podrá guiar a través del diseño de las tablas base de datos y de cada una de las estructuras y funciones que componen al proyecto escrito en C, así como las clases en C++. Se recomienda que para ampliar la información dada aquí, se consulte el manual en el directorio doc de la distribución, generado desde la documentación del código de la aplicación con la herramienta Doxygen, y localizado en la distribución oficial, además de revisar el código fuente si se tiene alguna duda sobre la implantación del código.

Este programa se distribuye exclusivamente bajo la licencia GNU GPL v2. por lo que se podrá modificar siempre y cuando se dé crédito al autor, y así mismo se publique el código fuente modificado.

Tablas de la base de datos:

A continuación se presentan las entidades de la base de datos utilizadas por la aplicación, el (*) indica llaves primarias.

Autor

Campo	Tipo
*id	int
nombre	char*
fecha_nac	char*
lugar	char*
dir	char*

Titulo

Campo	Tipo
*id	int
nombre	char*
fecha	char*
path_portada	char*

Libro

Campo	Tipo
*id	int
autor	int
titulo	int
edicion	int
cantidad	int

Usuario

Campo	Tipo
*id	int
nombre	char*
tipo	int
password	char*
email	char*

TipoUsuario

Campo	Tipo
id	int
descripcion	char*

Prestamo

Campo	Tipo
libro	int
usuario	int
fecha_prestamo	char*
fecha_entrega	char*

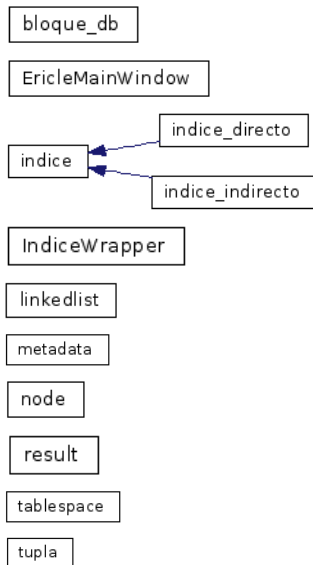
Contenido

Campo	Tipo
linea	int
texto	char*

Diseño de la aplicación del cliente.

A continuación se provee una descripción breve del funcionamiento interno, describiendo cada archivo, así como las estructura de datos mas relevantes de la aplicación. Si desea mayor información, vea la documentación completa en la distribución del CD.

Representación gráfica de la clase



Esta lista de herencias esta ordenada rigurosamente, pero no completa, por orden alfabético:

- [bloque_db](#)
- [EricleMainWindow](#)
- [indice](#)
 - [indice_directo](#)
- [indice_indirecto](#)
- [IndiceWrapper](#)
- [linkedlist](#)
- [metadata](#)
- [node](#)
- [result](#)
- [tablespace](#)
- [tupla](#)

Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

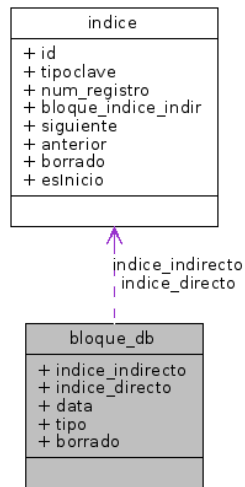
/home/kreig/projects/fs/ericle/src/bloque_db.cpp	Implementacion de bloque_db.h
/home/kreig/projects/fs/ericle/src/bloque_db.h [código]	Describe la clase bloque_db usada como unidad basica del archivo database.db
/home/kreig/projects/fs/ericle/src/database.h [código]	Objeto de base de datos
/home/kreig/projects/fs/ericle/src/dataoperations.h [código]	Coleccion de funciones para manipulacion de datos en los archivos
/home/kreig/projects/fs/ericle/src/ericle.cpp	Punto de entrada de la app
/home/kreig/projects/fs/ericle/src/ericleconfigdbdialog.log.cpp	Implementacion de ericleconfigdbdialog.h
/home/kreig/projects/fs/ericle/src/ericlemainwindow.h [código]	Ventana principal de la aplicacion
/home/kreig/projects/fs/ericle/src/indice.cpp	Implementacion de indice.h
/home/kreig/projects/fs/ericle/src/indice.h [código]	Contiene clase base de indice para archivo
/home/kreig/projects/fs/ericle/src/indice_directo.h [código]	Contiene la clase que describe el concepto de indice directo
/home/kreig/projects/fs/ericle/src/indice_indirecto.h [código]	Contiene la clase que describe el concepto de indice indirecto
/home/kreig/projects/fs/ericle/src/indicewrapper.cpp	Implementacion de indicewrapper.h
/home/kreig/projects/fs/ericle/src/logger.h [código]	Definicion de las funciones de bitacora
/home/kreig/projects/fs/ericle/src/metadata.c	Implementacion de metadata.h
/home/kreig/projects/fs/ericle/src/metadata.h [código]	Clase de almacenamiento de metadata
/home/kreig/projects/fs/ericle/src/modifylibro.cpp	Implementacion de modifylibro.cpp
/home/kreig/projects/fs/ericle/src/modifylibro.h [código]	Subclase de insertLibro que representa la ventana de modificacion de libro
/home/kreig/projects/fs/ericle/src/r_autor.h [código]	Clase de almacenamiento de autor en db
/home/kreig/projects/fs/ericle/src/r_contenido.h [código]	Clase de almacenamiento de contenido en db
/home/kreig/projects/fs/ericle/src/r_libro.h [código]	Clase de almacenamiento de libro en db
/home/kreig/projects/fs/ericle/src/r_prestamo.h [código]	Clase de almacenamiento de prestamo en db
/home/kreig/projects/fs/ericle/src/r_tipousuario.h [código]	Clase de almacenamiento de tipousuario en db
/home/kreig/projects/fs/ericle/src/r_titulo.h [código]	Clase de almacenamiento de titulo en db
/home/kreig/projects/fs/ericle/src/r_usuario.h [código]	Clase de almacenamiento de usuario en db
/home/kreig/projects/fs/ericle/src/registro.h [código]	Clase de almacenamiento de un registro, el cual es un bloque de referencia para el archivo datafile.db
/home/kreig/projects/fs/ericle/src/result.h [código]	Clase de result
/home/kreig/projects/fs/ericle/src/search_node.h [código]	Arbol de sintaxis abstracto para realizar las busquedas
/home/kreig/projects/fs/ericle/src/tablespace.cpp	Implementacion de tablespace.h

</home/kreig/projects/fs/ericle/src/tablespace.h> [\[código\]](#) Definición del struct tablespace
</home/kreig/projects/fs/ericle/src/tipos.h> [\[código\]](#) Definición de tipos y constantes
</home/kreig/projects/fs/ericle/src/tupla.h> [\[código\]](#) Representa una tupla

Referencia de la Estructura bloque_db

```
#include <bloque_db.h>
```

Diagrama de colaboración para bloque_db:



[\[leyenda\]](#)

Atributos públicos

```

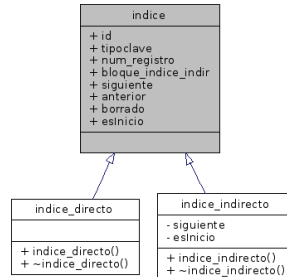
union {
  indice indice_indirect
0
  indice indice_directo
}
  
```

dato
 char [tipo](#)
 tipo de dato 0: idx_indir, 1 idx_dir, 2 tablespace
 bool [borrado](#)
 0 : si no esta borrado, 1 : si esta borrado

Referencia de la Estructura indice

```
#include <indice.h>
```

Diagrama de herencias de indice



Atributos públicos

unsigned int [id](#)
id de la clave

unsigned int [tipoclave](#)
tipo de la clave

unsigned int [num_registro](#)
numero de registro para indice directo

unsigned int [bloque_indice_indir](#)
numero de bloque de inicio de lista para indice directo

unsigned int [siguiente](#)
puntero a anterior

unsigned int [anterior](#)
puntero a siguiente

int **borrado**

char [esInicio](#)
es 1 si el indice esta borrado(usado como bandera de defrag) Define si es cabeza de lista 1 true. 0 false