

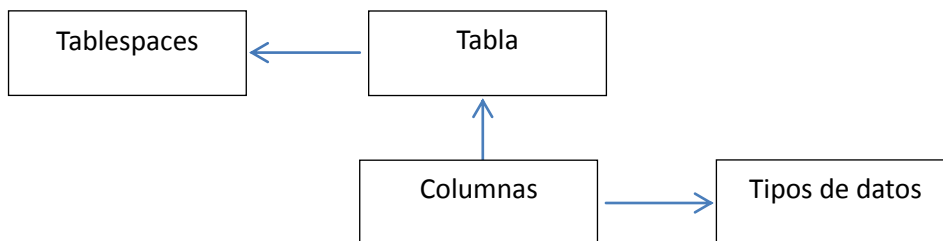
EIF211: Diseño e Implementación de Bases de Datos
Universidad Nacional
Heredia, Costa Rica
Proyecto I I ciclo 2014

Objetivo

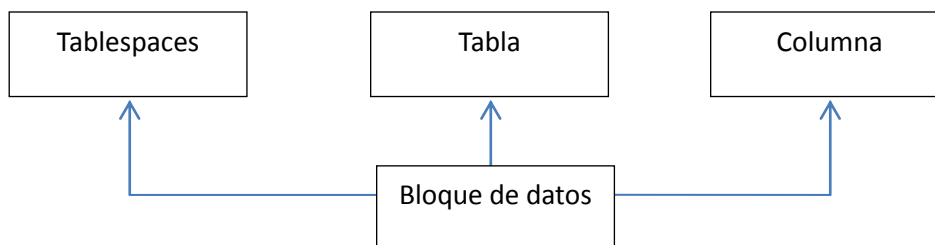
Diseñar e implementar un modelo de base de datos basada en el uso de un metadato y funciones de creación y manipulación de datos. El modelo tiene características similares a las estructuras físicas y lógicas de un Sistema Gestor de Bases de Datos (SGBD).

Requisitos de diseño

El metadato que representa al diccionario de datos debe ser un modelo de datos relacional que tenga el siguiente conjunto de relaciones, y almacenado en un tablespaces (datafiles) “sys01.dbf”. Las relaciones del modelo se implementan con sus estructuras correspondientes.



El almacenamiento de datos en la base de datos debe cumplir con el siguiente modelo lógico



Los bloques de datos son almacenados físicamente en el datafile del tablespace al que pertenece la tabla. Cada bloque de dato almacena los valores de las columnas de una tabla y su estructura es:

Struct bloque_dato

```
{      tablespace      tipo_identificador_tablespace,
      tabla           tipo_identificador_tabla,
      columna         tipo_identificador_columna
      valor           tipo_valor_columna
      next            loninteger;
      status          integer; }
```

Requisitos de implementación

El sistema debe ser desarrollado en C++ con el paradigma orientado a objetos. Las clases y objetos principales corresponden a los privilegios que puede tener cualquier usuario del modelo y representan los procesos básicos de un SGBD, tales como:

Creador	// crea tabla en el diccionario de datos
Insertor	// inserta registros en las tablas de usuario
Actualizador	// actualiza registros de las tablas de usuarios
Borrador	// borra registros de las tabla de usuario
Seleccionador	// proyecta los registros

Adicional a esto se cuenta con otras clases u objetos especializados como:

Lector	// Extraer los bloques de datos del datafile
Escritor	// Escribe bloque de datos en el datafile
Grabador	// Actualiza los datos del bloque
Validador	// verificar la valides de cada bloque
Borrador	// borrar en forma lógica los bloques
Compresor	// Borra en forma física los bloques borrador lógicamente

Se debe desarrollar una interfaz gráfica o de consola, que permita acceder y utilizar las funciones mencionadas y otras funciones que el grupo considere necesarias. El grupo de estudiantes debe proporcionar las soluciones al problema, de forma inédita, no se aceptan copias en ningún apartado del proyecto. Se evalúa el derecho a la propiedad intelectual de la solución. Cualquier omisión a lo mencionado será motivo de aplicación del reglamento universitario correspondiente.

Entregables

Documento que indique:

- Integrantes del grupo (máximo 3 estudiantes)
- Objetivos del proyecto
- UML para el metadato y la clase administrador, y otras clases necesarias
- Implementación en C++, uno o varios ejecutables
- Mini Manual de usuario
- Conclusiones sobre el trabajo

Fechas de Entrega

28 de marzo 2014 entrega parcial (20%) (primer revisión y aclaraciones respectivas)

13 de abril 2014 Entrega final (80%)