

Manual técnico

Estructura principal

- Paquetes
 - Cubo
 - Cubo
 - GUI
 - Front
 - Mapeo
 - Conexion
 - Consulta
 - Dimension
 - Entidad
 - ForeignKey
 - Modelo
 - ModeloEstrella
 - Olap
 - Sql
 - QueryDim

Descripción

- Paquetes
 - **Cubo:** es el paquete en el que se encontrarán todas las clases encargadas de la manipulación de datos, tomando como preliminar un modelo de estrella.
 - **Cubo:** es la clase que contiene los métodos necesarios para poblar una

estructura de datos, la cual muestra la interacción entre 2 dimensiones y las métricas de sus intersecciones.

- Atributos

- dimensiones: atributo que contiene las dimensiones necesarias para armar el cubo.
- jerarquias: estructura que contiene el desglose de jerarquías posibles a partir de la tabla de hechos.
- filas: entero que maneja una posición en el cubo respecto de las filas.
- columnas: entero que maneja una posición en el cubo respecto de las columnas.

- Métodos

- addDimensionJerarquia: método que agrega una dimensión al cubo.
- getDimension: método que devuelve la dimensión en la cual se encuentre.
- getJerarquia: de igual manera el método getDimension devuelve la jerarquía en la cual se encuentre la posición actual de filas y columnas de el cubo.
- getModeloEstrella: este método devuelve una instancia del tipo modelo estrella un paso anterior a armar el cubo.

- **GUI:** en este paquete se encuentran las clases para la construcción de la interfaz gráfica y la interacción entre métodos.

- **Front:** una clase que hereda de JFrame, en la cual se configura una interfaz de “drag and drop” para las tablas set de conexión, usuario, etc. Todo lo anterior, de interacción con usuario.

- Métodos

- generarOlap: se encarga de crear de una forma gráfica las tablas en las cuales se seleccionan los atributos para luego formar el modelo de estrella.
- generarGerarquias: este método muestra gráficamente las posibles jerarquías que pueden existir luego de seleccionar la tabla de hechos o facts.

- **Mapeo:** en el paquete de mapeo se encuentran todas las clases para la manipulación de datos y la implementación de SQL en PostgreSQL, las cuales sirven como un medio de transformar los datos para su utilización en el cubo.

- **Conexion:** esta clase, como su nombre lo indica, es la encargada de realizar la conexión con la base de datos, la cual contiene los drivers y archivos .jar necesarios para establecer los parámetros necesarios para realizar la conexión a través de un DriverManager.

- Atributos

- login: atributo privado del tipo String el cual tiene valores para el login en la base de datos.
- password: atributo privado de tipo String que contiene los valores de la contraseña para el usuario de la base de datos.
- url: atributo privado de tipo String que contiene la dirección de la base de datos.
- driver: atributo privado de tipo String que contiene un constante nombre del driver utilizado para la conexión a la base de datos.

- Métodos

- getConexion: obtiene un objeto del tipo connexion, el cual está establecido con los valores para la conexión para la base de datos.
- getUrl: devuelve un objeto tipo String con el url de la base de datos.
- setUrl: método que recibe de parámetro un objeto String con el cual se establece la url de la base de datos.

- **Consulta:** con esta clase se obtienen los valores necesarios de la base de datos para agregar valores a estructuras que se utilizarán para formar dimensiones y métricas.

- Metodos

- getUsr: método que obtiene los usuarios en una lista de arreglos.
- getRol: mediante un parámetro de usuario devuelve el rol del usuario determinado en el parámetro.
- addUsr: método que agrega un usuario .
- getProducto: método que devuelve un objeto del tipo map<String,String> con los productos.

- **Dimension:** esta clase es una estructura de datos para guardar el orden de las jerarquías de los campos que son llaves y el nombre de la dimensión que se esta modelando.

- Atributos

- campos: nombre de los campos de la dimensión el cual se encuentra en un arreglo.
- jerarquias: lista de jerarquías posibles en un objeto del tipo arreglo de Strings.
- camposllave: nombre de los campos asociados de la dimensión el cual se encuentra en un arreglo.
- nombreDim: atributo de tipo String, el cual contiene el nombre de la dimensión
- Métodos
 - crearDimencio: método por el cual se crea la dimensión y se agrega al modelo.
 - crearDimencioTiempo: este método crea la dimensión del tiempo con sus jerarquías.
- **Entidad:** la clase entidad es una estructura de datos que se utiliza para cargar a memoria datos necesarios, como por ejemplo: nombres de tablas, nombres de campos y relaciones con otras entidades.
 - Atributos
 - fk: arreglo de String que contiene campos llaves foráneas.
 - pk: arreglo de String que contiene campos llaves primarias.
 - atributos: arreglo de String que contiene campos que componen a la entidad.
 - nombre: atributo de tipo String que contiene el nombre de la entidad.
- **Modelo:** esta clase utiliza instancias de las clases Sql y entidades para armar un modelo de los datos y sus relaciones en la base de datos.
 - Atributos
 - dominioOriginal: objeto que contiene al dominio original.
 - EntidadHechos: atributo de tipo entidad, que corresponde a la entidad de hechos.
- **ModeloEstrella:** relaciona la tabla de hechos con las dimensiones y llenándola con las métricas necesarias dinámicamente mediante parámetros ingresados por el usuario.
 - Atributos
 - nombreFactTable: atributo String que contiene el nombre de la tabla de hechos
 - fks: atributo de tipo arreglo de String que tiene el registro

de las llaves foráneas para la tabla de hechos.

- dimX1: arreglo que contiene los nombres de las dimensiones:
- metricas: arreglo de String que contiene los nombre de las métricas.
- Métodos
 - setMetricas: metodo encargado de agregar las métricas a utilizar.
- **Olap:** olap utiliza a la clase modelo estrella para organizar los datos en en función de las dimensiones.
 - Métodos
 - generarTablaDeHechos: método que genera la tabla de hechos a partir de un String con el nombre de la tabla de hechos que el usuario seleccionó.
 - GetDimencionesPosibles: devuelve una lista de las dimensiones alcanzables a partir de la tabla de hechos.
 - getModeloEstrella: es un método que devuelve un objeto del tipo modelo estrella.
 - setDimencionTiempo: método que tiene como entrada el campo que será denominado como la dimensión tiempo y genera la dimensión tiempo.
- **Sql:** la clase sql es la encargada de todo el manejo y manipulación de la base de datos brindando métodos con los cuales de una manera sencilla se puede manipular las bases de datos mediante operaciones simples de sql.
 - Atributos
 - DB: un objeto del tipo Conexion con la conexión de la base de datos.
 - Métodos
 - consulta: metodo que recibe como parámetro un String con la sentencia Sql que se manda a ejecutar y devuelve un resultSet con el resultado si es necesario. De lo contrario solo ejecuta la sentencia.
 - ejecutar: este método ejecuta la sentencia Sql establecida en la consulta.
 - getCampo: obtiene el campo de el resultado de la consulta ejecutada con el método ejecutar.
- **QueryDim:** se encarga de los querys necesarios para obtener las

dimensiones tanto de tablas, como de vistas.

- Atributos
 - atributo: arreglo de atributos.
 - fk: arreglo con llaves primarias obtenidas.
 - camposLLaveOriginal: String con el nombre del campo llave original.
- Métodos
 - getAtributo: método que retorna los atributos de una tabla.
 - getNombre : método que retorna un String con el nombre de la entidad.
 - getEntidad: método que retorna la entidad.