SISTEMA DE AYUDA A LA TOMA DE DECISIONES POR LA CONTAMINACIÓN DEL AIRE

31/10/2019

Práctica 2

Andrés Gavín Murillo 716358

Eduardo Gimeno Soriano 721615

Sergio Álvarez Peiro 740241

Grupo 2-6

Sistemas de Información

Ingeniería Informática

Universidad de Zaragoza

ÍNDICE

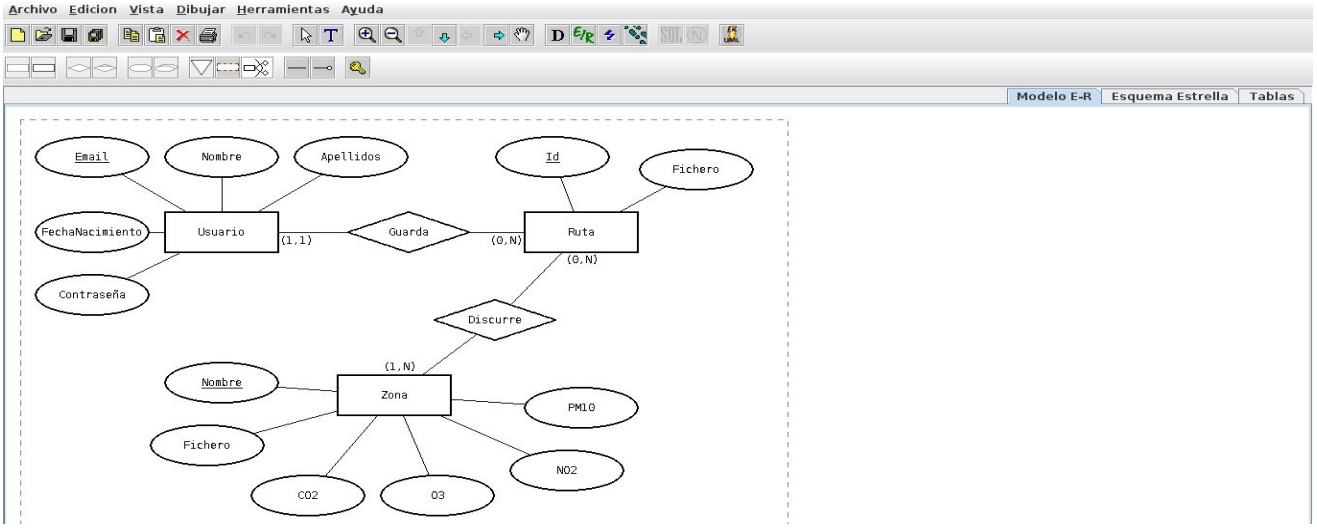
Herramientas utilizadas

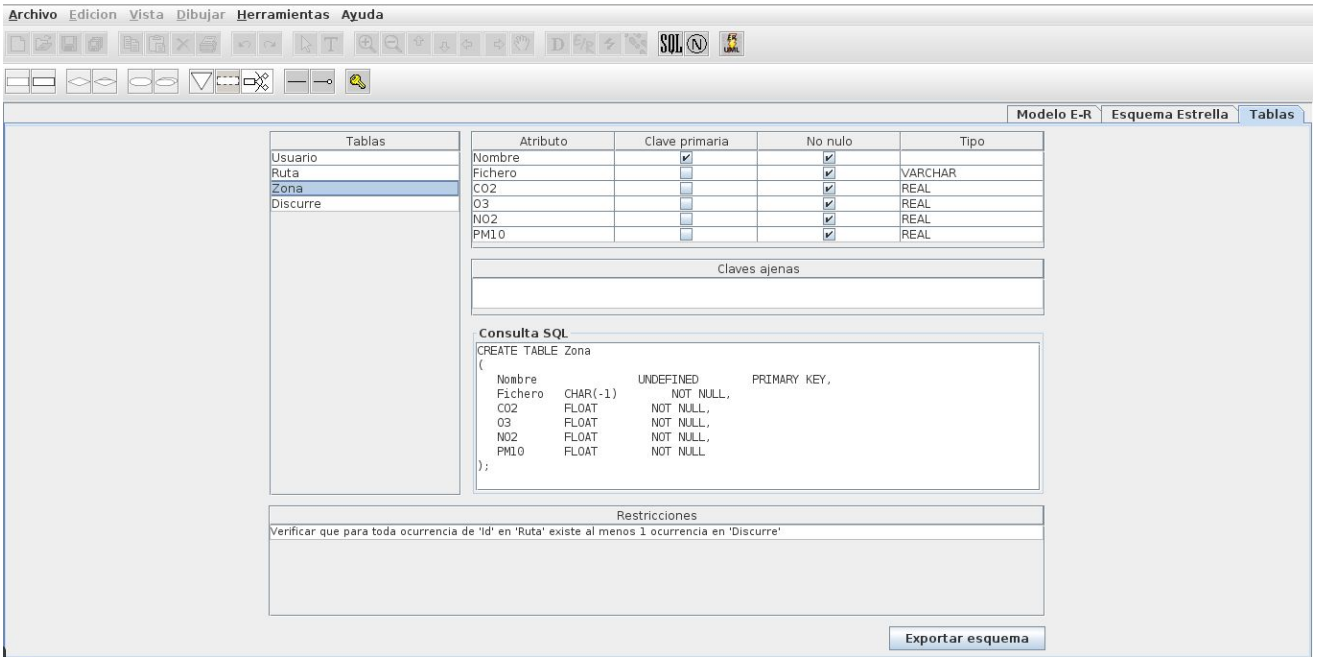
Base de datos

Para la creación del esquema entidad-relación, obtención del modelo relacional y del fichero *sql* de creación de tablas se ha utilizado la aplicación *DBDap* proporcionada en la asignatura Bases de datos 2.

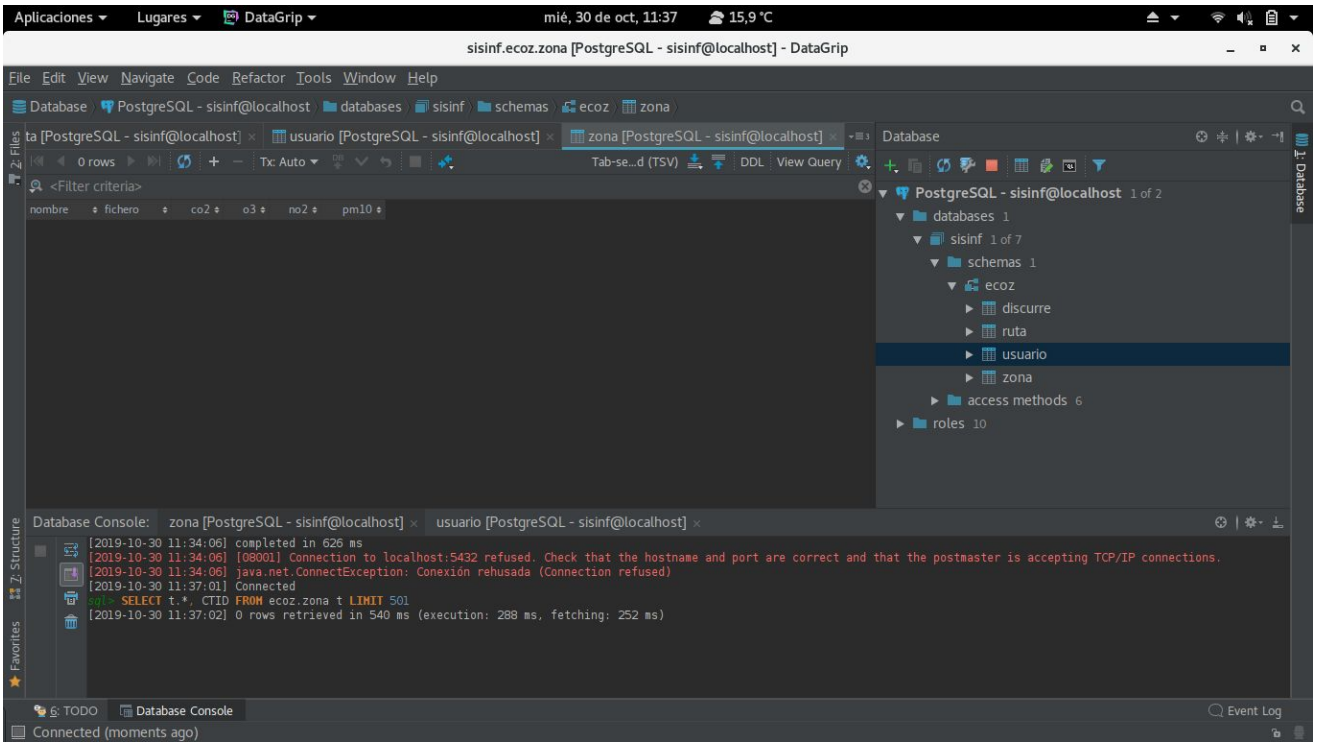
Esta aplicación permite crear el esquema entidad-relación de forma gráfica, normalizarlo y obtener a partir de él el modelo relacional en un archivo *.txt* y el código *sql* para la creación de las tablas en un archivo *.sql*.

Cada elemento del esquema tiene un menú propio donde establecer sus características, por ejemplo, para un atributo de una entidad se puede establecer su dominio, obligatoriedad, si es multivaluado, etc; además de opciones gráficas. También se pueden establecer restricciones sobre el esquema.





Para la monitorización de la base de datos una vez desplegada se ha utilizado *DataGrip*, un *IDE* proporcionado por *JetBrains*, que permite ver el contenido de todas las tablas, añadir y eliminar entradas una por una o incluso importar datos desde un fichero *.csv* y exportarlos a este tipo de ficheros.



Capa de persistencia de datos

Para poder trabajar con ficheros en formato *KML* se ha utilizado una *API* externa *open source* [1] creada por *Micromata GmbH* [2], una corporación alemana de software.

El principal uso que se le ha dado a esta librería en esta práctica ha sido utilizar el tipo de dato *KML* para poder manipular los ficheros con facilidad, así como operaciones que permiten volcar el contenido en ficheros *.txt* para facilitar su guardado en la base de datos.

Metodología y distribución de trabajo

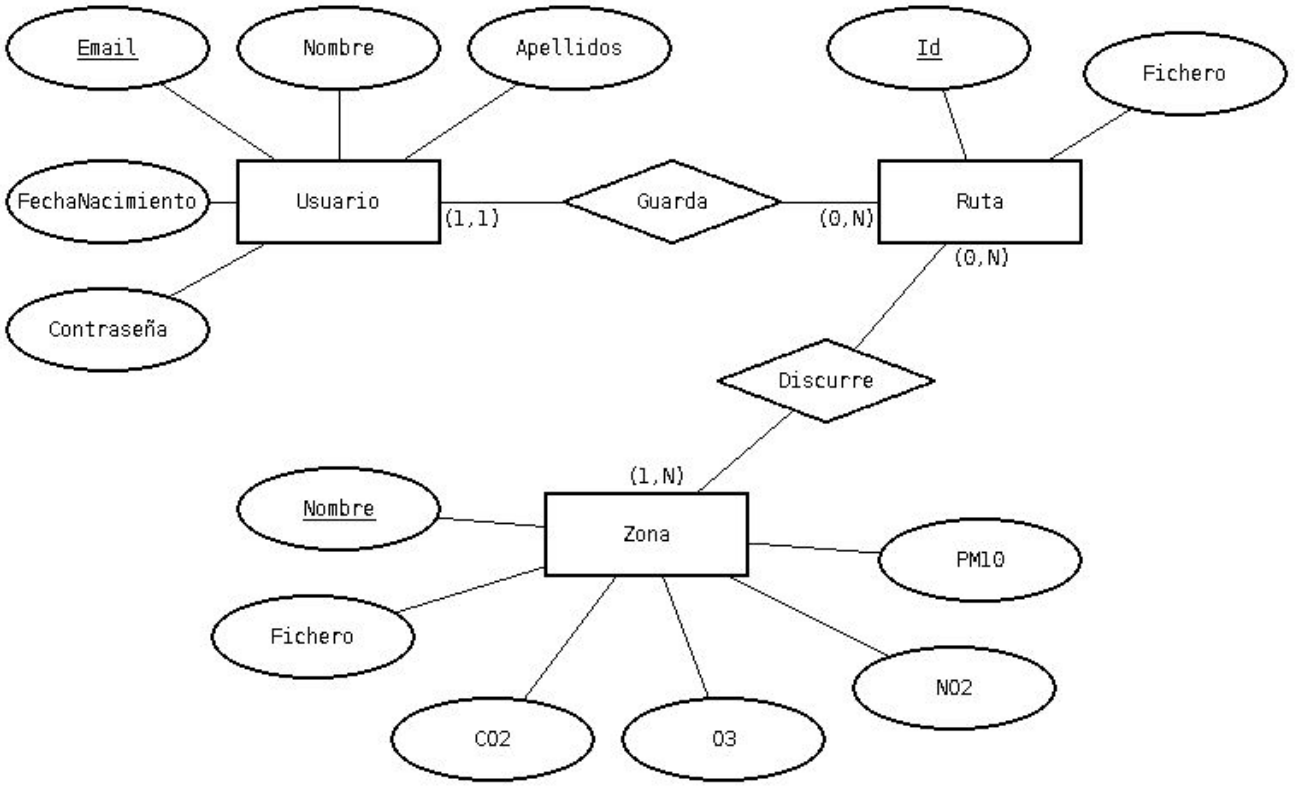
Para la realización del modelo entidad relación y el modelo relacional se hizo entre todos los integrantes del grupo con la ayuda de la herramienta ya mencionada. Una vez se tuvo un modelo, se realizó también en un proyecto de *Eclipse* el código necesario para la conexión con la base de datos. Luego cada integrante desarrolló cada clase correspondiente a las tablas de la base de datos:

- rutaRepository.java y discurreRepository.java - Sergio
- usuarioRepository.java - Andrés
- zonaRepository.java - Eduardo

En los ficheros se utiliza *JDBC* para poder hacer consultas a la base de datos *PostgreSQL*. Las clases necesarias para guardar información de rutas y por qué zonas pasan se realizan por la misma persona ya son más simples y no es necesario mucho código. Para el guardado de ficheros *KML* en la base de datos se usa una *API* externa que nos facilita un tipo de dato fichero *KML*, pudiéndose convertir a un tipo de fichero para poder guardarlo en la base de datos.

Modelo Entidad-Relación

Modelo entidad-relación creado con la aplicación *DBDap*:



Modelo Relacional

Modelo relacional obtenido a partir de la aplicación *DBDap*:

ESQUEMA RELACIONAL

Dominios:

Esquemas de relacion:

Usuario = (Email: VARCHAR, clave primaria; Nombre: VARCHAR;
Apellidos: VARCHAR; FechaNacimiento: FECHA;
Contraseña: VARCHAR, NO NULO);

Ruta = (Id: ENTERO, clave primaria; Fichero: VARCHAR, NO NULO;
Usuario_Email: VARCHAR, NO NULO);
clave ajena (Usuario_Email) referencia a Usuario(Email)

Zona = (Nombre: VARCHAR, clave primaria; Fichero: VARCHAR, NO NULO;
CO2: REAL, NO NULO; O3: REAL, NO NULO;
NO2: REAL, NO NULO; PM10: REAL, NO NULO);

Discurre = (Ruta_Id: ENTERO; Zona_Nombre: VARCHAR);
clave primaria (Ruta_Id,Zona_Nombre);
clave ajena (Ruta_Id) referencia a Ruta(Id)
clave ajena (Zona_Nombre) referencia a Zona(Nombre)

Verificar que para toda ocurrencia de 'Id' en 'Ruta' existe al menos 1 ocurrencia en 'Discurre'

Despliegue

Para el despliegue de la base de datos se ha utilizado *PostgreSQL* sobre un contenedor *Docker*, el cual se ejecuta sobre el puerto 5432. El contenedor *Docker* ha sido creado utilizando el *script crearBD.sh* (incluido en la entrega), que automatiza la creación de la base de datos a partir del fichero *SQL* “postgresql/crear_bd.sql”.

El nombre del contenedor es *sisinf-postgresql*, y para ponerlo en ejecución basta con lanzar el comando “sudo docker start sisinf-postgresql”.

Se ha decidido empaquetar la aplicación con *Docker* ya que permite lanzarla y detenerla de manera cómoda, además de poder compartirla y ejecutarla en diferentes máquinas que tengan *Docker* instalado.

Dificultades encontradas

En cuanto al uso de la *API* externa utilizada para gestionar ficheros *KML*, se han encontrado dificultades a la hora de guardar y rescatar los ficheros de la base de datos, ya que esta los ficheros se guardan como *byteA*, por tanto, para poder guardarlos y rescatarlos se necesita un tipo de dato *InputStream* de Java usando *JDBC*.

La *API* no presenta ningún método que permite la conversión de *KML* a *InputStream*, sí presenta métodos para obtener un fichero *.txt* de un *KML* y viceversa.

Por tanto, para guardar un fichero *KML* por ejemplo, se ha tenido que obtener primero el fichero *.txt* y a partir de este fichero el *InputStream* necesario. Los ficheros *.txt* generados se borran desde el método donde se crean y usan para evitar problemas de escalabilidad.

Esfuerzos invertidos

A continuación se muestra la tabla de esfuerzos:

Eduardo		
Fecha	Tarea	Duración
18/10/2019	Elección de herramientas a utilizar, instalación y creación del esquema E/R	2h
25/10/2019	Creación de la capa de persistencia de datos	2h
28/10/2019	Realización capa de persistencia tabla zona	1,5h
29/10/2019	Realización capa de persistencia tabla zona	3,5h
30/10/2019	Memoria	1h
Andrés		
Fecha	Tarea	Duración
18/10/2019	Elección de herramientas a utilizar, instalación y creación del esquema E/R	2h
19/10/2019	Creación, despliegue y configuración del servidor para la BD Postgresql	1,5h
25/10/2019	Creación de la capa de persistencia de datos	2h
29/10/2019	Realización capa de persistencia tabla usuario	3,5h
30/10/2019	Memoria	1h
Sergio		
Fecha	Tarea	Duración
18/10/2019	Elección de herramientas a utilizar, instalación y creación del esquema E/R	2h
25/10/2019	Creación de la capa de persistencia de datos	2h
30/10/2019	Realización capa de persistencia tablas discurre y ruta	2h
30/10/2019	Memoria	30 min

Bibliografía y referencias

[1] *JavaApiForKML official site*:
https://github.com/micromata/javaapiforkml

[2] *Micromata official site*:
https://www.micromata.com/

[3] *Docker official site*:
https://www.docker.com/

[4] *DataGrip official site*:
https://www.jetbrains.com/datagrip/