

Capítulo 3

Arquitectura

Este capítulo presenta la arquitectura con la que se desarrollo esta tesis. En esta arquitectura se muestra como están relacionadas las diferentes herramientas, módulos y trabajos involucrados.

3.1 Arquitectura General.

La arquitectura general del software es parte primordial para el diseño e implementación de un proyecto de programación, tal como esta tesis, ya que nos indica la estructura organizacional y el comportamiento asociado de un sistema. La arquitectura se muestra a continuación.

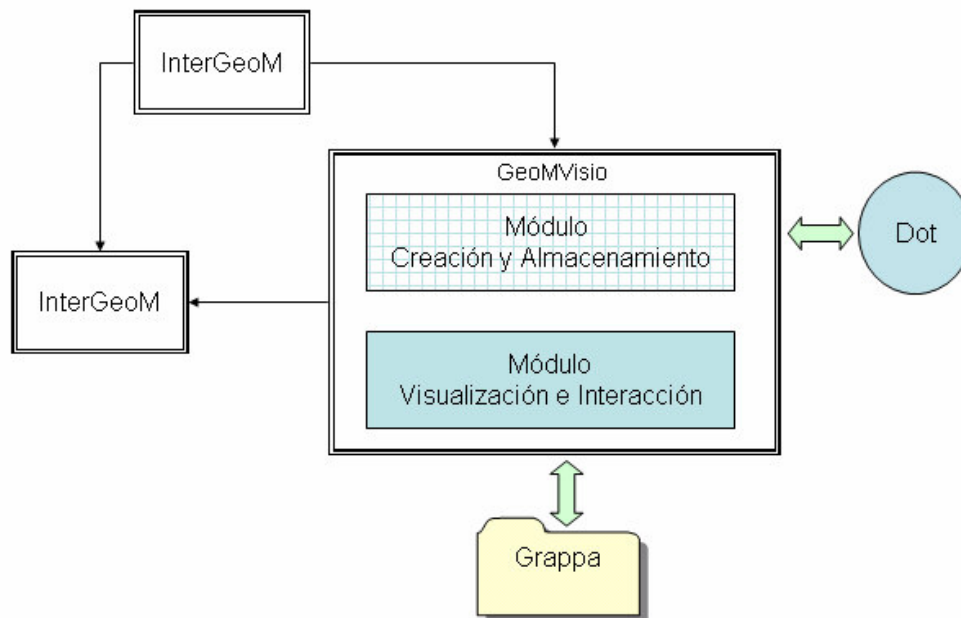


Figura 3.1

Arquitectura del GeoMVisio

En la arquitectura mostrada en la Figura 3.1 podemos apreciar como están relacionados diferentes elementos que en conjunto cumplen con el objetivo requerido. Los elementos involucrados que podemos observar son los ya mencionados Graphviz y Grappa, así como la Tesis Doctoral de Manuel Pech Palacio y la aplicación de Daniel Pérez (InterGeoM) sistemas externos que ayudan a esta tesis.

En el esquema podemos apreciar que la aplicación de GeoMVisio consta de dos módulos: el primero el Módulo de Creación y Almacenado y el segundo el Módulo de Visualización. Estos módulos se encargan de realizar las tareas principales de GeoMVisio.

El primer módulo realiza las funciones de leer, separar los grafos y procesar las subestructuras encontradas provenientes de los archivos de entrada de GeoMVisio. La otra función que realiza es la creación de archivos en formato Dot a cada uno de los grafos encontrados por Subdue para su posterior visualización con ayuda de Grappa.

Esta segunda función, como se aprecia en la arquitectura, es la que manda llamar al proceso de la aplicación Dot de Graphviz misma que regresa los archivos en formato Dot con las posiciones de cada elemento del grafo. Este módulo es una adaptación para GeoMVisio de un módulo ya existente creado por MPP usado en la Tesis Doctoral de Manuel Pech Palacio [Pech, 2002] el cual se explicará mas adelante en la sección 3.2 de este capítulo.

El segundo módulo se encarga de crear las imágenes para su visualización e interacción con los diferentes requerimientos de GeoMVisio. En la función de interacción con las imágenes también se hace uso de la aplicación Dot para convertir las imágenes visibles en el momento en archivos en formato de imagen.

Este segundo módulo es propio de GeoMVisio, sin embargo contiene una clase que extiende de otra del paquete de Grappa para la interacción con las imágenes y permite la creación de los hipergrafos. Como se mencionó aquí previamente es donde el visualizador Grappa entra en acción tal como se puede apreciar en la arquitectura. Se puede observar que ambas funciones están relacionadas con este paquete de visualización [Grappa, 2005].

También podemos observar que en la Tesis doctoral de Manuel Pech Palacio puede mandar llamar a GeoMVisio y a InterGeoM por separado, sin embargo GeoMVisio manda llamar a InterGeoM para que esta pueda realizar sus funciones, tanto como la Tesis doctoral de Manuel Pech Palacio y InterGeoM se explicarán a continuación en los siguientes puntos.

3.2 Antecedentes Tesis Doctoral de Manuel Pech Palacio

El Maestro Manuel Pech Palacio, como tesis doctoral desarrolla un proyecto que trabaja en dos contextos de prueba. El primero cotejar los cambios sufridos en la ciudad de Puebla a través del tiempo desde épocas coloniales. En este contexto se pretende brindar un sistema que incorpore y opere con información obtenida a partir de censos realizados en diferentes épocas de la

ciudad de Puebla. Esto permite conocer la evolución demográfica que ha sufrido en los últimos 200 años. El otro contexto de prueba es localizar rutas de evacuación de las localidades cercanas al volcán Popocatepetl. Para cumplir con ese objetivo hace uso de la minería de datos, más en específico hace uso del algoritmo Subdue para poder encontrar la información que le puede ser de interés para este propósito.

La tesis doctoral de Manuel Pech Palacio hasta ahora ya cumple con la mayoría de los requerimientos al cual fue expuesta. Cuenta con un conjunto de interfaces que realizan diferentes funciones como la aplicación de Subdue. Sin embargo, no cuenta con una manera dinámica para visualizar los resultados obtenidos por la minería de datos, que permita al usuario interpretar la importancia que estos representan que es lo que GeoMVisio pretende.

La tesis doctoral de Manuel Pech Palacio ya cuenta con un módulo que cumple con este requisito y es antecesor del utilizado por GeoMVisio, el módulo de Creación y Almacenamiento. Este funciona creando imágenes de los grafos en formato *jpg* por medio de la misma aplicación *Dot*. Una vez creadas las imágenes se almacenan en una carpeta para su posterior visualización e interpretación sin más interacción que la misma imagen. Estas imágenes muestran los grafos con toda la información incluso con subgrafos pero no lo muestra de manera comprimida [Pech, 2002].

En resumen podemos decir que funciona de la siguiente manera:

1. Primero, una eficiente aplicación implementada para la tesis doctoral de Manuel Pech Palacio crea los archivos que contiene los grafos encontrados por el sistema Subdue a partir de la base de datos creada para el proyecto de la tesis doctoral de Manuel Pech Palacio.

2. Una vez creados los archivos se procede a la extracción de los grafos de cada iteración para poder crear un archivo en lenguaje Dot por cada uno de ellos. Para esto se hace una lectura exhaustiva en el archivo de entrada localizando los componentes de cada grafo ya sea un grafo resultado o una subestructura predefinida y agregándolos a una plantilla en lenguaje Dot implementada en java, añadiendo así los vértices, arcos y subgrafos de cada subestructura.

3. Una vez creados los archivos en lenguaje Dot entra en acción la aplicación *Dot*. Todos y cada uno de los archivos son convertidos a imágenes al procesarlos por medio de una línea de comando directamente en consola que se muestra a continuación [Gansner, 2002]:

```
$ Dot -TJPEG graph1.Dot -o graph1.jpg
```

La línea anterior crea las imágenes y se eliminan todos los archivos del cual proceden. Las imágenes se guardan en una carpeta llamada datos para después de una manera “manual”, por así decirlo, el usuario pueda acceder a ésta para poder ver las imágenes creadas. El proceso se puede ver como lo muestra la siguiente imagen de la Figura 4.1.

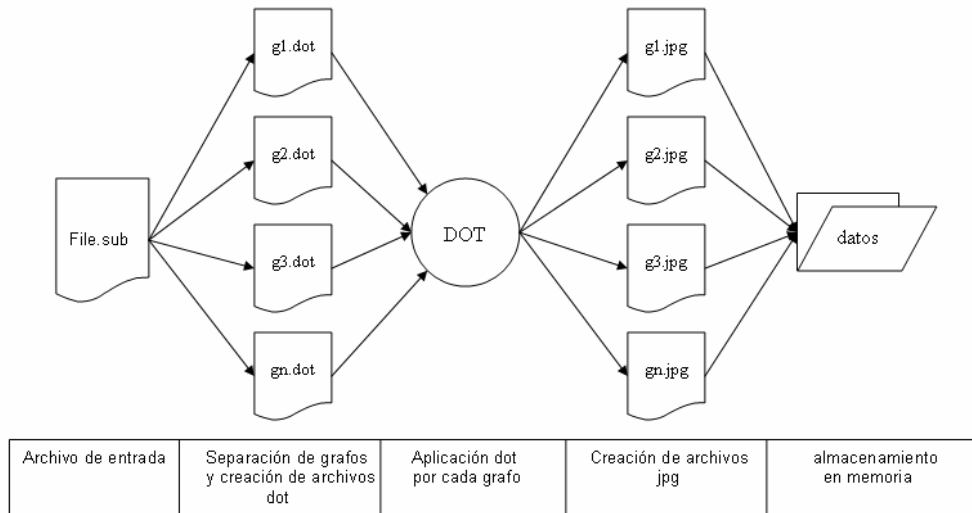


Figura 3.1

Proceso del módulo predecesor del módulo de creación y almacenado.

Se comprende que esta versión realiza la tarea que se le asigna, pero es obvia la necesidad de una nueva versión que a demás de crear las imágenes y mostrarlas de manera directa aporte interacción con las imágenes y sus componentes para una mejor interpretación.

La tesis doctoral de Manuel Pech Palacio requiere que se incorporen los módulos que él uso para la creación de los archivos Dot, pero con cambios realizados para la función GeoMVisio. Este módulo es el trabajo predecesor del módulo de creación y almacenado de GeoMVisio que se presenta en la arquitectura general.

3.2.1 Subdue y su integración a la tesis doctoral de Manuel Pech Palacio.

Con respecto a esta arquitectura Subdue es de gran importancia ya que este descubre los patrones a visualizar que pasa por todo los procedimientos de la tesis. Para propósitos de la tesis doctoral de Manuel Pech Palacio y GeoMVisio

los elementos de Subdue que aportan los patrones resultados son los que realmente importan ya que son la base de esta tesis. Estos componentes se explican a continuación.

3.2.1.1 Parámetros de Subdue

Los resultados arrojados por Subdue se obtienen a partir de un conjunto de parámetros predefinidos que dirigen la búsqueda de subestructuras interesantes [Subdue, 2005]. Estos parámetros pueden ser modificados por medio de comandos directamente en consola al momento de realizar la búsqueda según sean las necesidades del usuario. Las modificaciones de estos parámetros requieren que el usuario este familiarizado con el funcionamiento de Subdue.

Para mostrar dichos parámetros se muestran a continuación los usados para el siguiente ejemplo resultado de una búsqueda cualquiera en la Figura 2.3:

```
Parameters:
Input file..... house3.g
Predefined substructure file..... none
Output file..... none
Beam width..... 4
Evaluation method..... MDL
'e' edges directed..... true
Iterations..... 3
Limit..... 27
Minimum size of substructures... 1
Maximum size of substructures.. 30
Number of best substructures.... 3
Output level..... 2
Allow overlapping instances.....false
Prune..... .false
Threshold..... 0.000000
Value-based queue..... false
```

Figura 2.3
Muestra de parámetros que dirigen una minería de datos.

Todos y cada uno de estos parámetros son importantes para la búsqueda de patrones interesantes, sin embargo tres tienen más relevancia que otros con respecto al propósito de GeoMVisio, ya que estos afectan los archivos de salida, modificando la cantidad de subestructuras creadas con las que trabaja el software de la interfaz a implementar [Subdue, 2005]. Estos parámetros son:

El parámetro de “*Iterations*” (en español iteraciones) indica el número de veces que se evalúa el archivo de entrada para la búsqueda de patrones, encontrando un conjunto de mejores subestructuras para cada evaluación.

El parámetro de “Number of best Subestructuras” o en español “Número de Mejores Subestructuras” especifica el número de mejores subestructuras que debe reportar y desplegar Subdue por cada iteración. Si no se modifica este parámetro el número de mejores subestructuras a encontrar siempre será de tres

El parámetro de Predefined Substructure. Este parámetro permite al Subdue hacer una búsqueda orientada. Se buscan patrones exactos o inexactos a las subestructuras predefinidas, por ejemplo un caso sencillo un lápiz y un bolígrafo son muy parecidos pero no son iguales. Las subestructuras predefinidas dirigen la búsqueda para encontrar patrones muy similares a estas, aunque estas no sean 100% iguales [Cook, 1995].

Los parámetros anteriores son importantes para la tesis debido a que GeoMVisio intenta mostrar cada subestructura creada, así que estos parámetros son los que realmente afectan al funcionamiento del software de GeoMVisio.

3.2.1.2 Archivo de Entrada de Subdue

El archivo de entrada de Subdue esta formado por una o más subestructuras compuestas de vértices y arcos todos en formato de texto. Cada grafo tiene un prefijo ya sea "XP" que indica si es un ejemplo positivo, o en su caso negativo el prefijo "XN". En la tesis siempre se trabajan grafos positivos. Si no existe un prefijo para el grafo del archivo de entrada el sistema lo toma como positivo.

Cada vértice es definido como $v \langle \# \rangle \langle \text{etiqueta} \rangle$ donde el símbolo de $v \langle \# \rangle$ es un identificador único que representa a cada vértice del grafo. El símbolo de gato representa números de manera ascendente involucrado en el grafo, iniciando por el número uno. El símbolo $\langle \text{etiqueta} \rangle$ simboliza el nombre asociado. Si tienen espacios en blanco o cuenta con caracteres diferentes a números o letras entonces la cadena de caracteres debe ser encerrada entre comillas. La etiqueta representa al objeto que presenta el vértice.

Un arco es definido de tres maneras:

e $\langle v1 \rangle \langle v2 \rangle \langle \text{etiqueta} \rangle$

d $\langle v1 \rangle \langle v2 \rangle \langle \text{etiqueta} \rangle$

u $\langle v1 \rangle \langle v2 \rangle \langle \text{etiqueta} \rangle$

Los símbolos $\langle vn \rangle \langle vm \rangle$ son los identificadores de los vértices fuente y destino respectivamente ya creados con anterioridad y $\langle \text{etiqueta} \rangle$ funciona de igual

manera que la etiqueta de los vértices pero en este caso nos indica la relación que existe entre las dos vértices adyacentes [Subdue, 2005]. A cada arco se le tiene que especificar con un parámetro ya sea “e” o “d” para arcos dirigidos y con “u” si son no dirigidos.

Para poder tener un poco mas claro lo que se intenta explicar se muestra a continuación un el texto archivo mostrado en la Figura 3.4 y más adelante Figura 3.5 que representa dicho texto:

```
v 1 object
v 2 object
v 3 object
v 4 object
v 5 object
v 6 object
v 7 object
v 8 object
v 9 object
v 10 object
v 11 triangle
v 12 triangle
v 13 triangle
v 14 triangle
v 15 square
v 16 square
v 17 square
v 18 square
v 19 circle
v 20 rectangle

e 1 11 shape
e 2 12 shape
e 3 13 shape
e 4 14 shape
e 5 15 shape
e 6 16 shape
e 7 17 shape
e 8 18 shape
e 9 19 shape
e 10 20 shape
e 1 5 on
e 2 6 on
e 3 7 on
e 4 8 on
e 5 10 on
e 9 10 on
e 10 2 on
e 10 3 on
e 10 4 on
```

Figura 3.4

Estructura representada en la sintaxis con que trabaja Subdue.

Como podemos ver en este texto representa veinte vértices las cuales los primeros diez nos indican que son objetos y el resto representan una figura. El primer objeto es un triángulo y se presenta de las líneas 11 a la 14, los siguientes cuatro nos indican que son cuadrados y finalmente la línea 10 y 20 indican un círculo y un rectángulo respectivamente. En cuanto a los arcos podemos reconocer por la etiquetas *<on>* (en español sobre) que nos indica que objetos estén sobre cuales, y la etiquetas *<shape>* indica que forma tienen los diferentes objetos [Subdue, 2005]. Como no se cuenta con otro tipo de información la imagen queda de la siguiente forma que se muestra en la Figura 3.5:

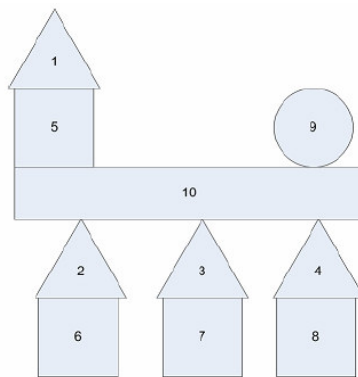


Figura 3.5

Estructura ejemplo con que trabaja Subdue

El formato del archivo de entrada es el mismo en las subestructuras descubiertas por Subdue. Estas estructuras son arrojadas directamente en consola mostrando todas y cada una de las instancias representadas en cada iteración.

3.3 Trabajos Relacionados Tesis de Daniel Pérez (InterGeoM)

GeoMVisio como se mencionó anteriormente tiene como fin aplicativo proporcionar ayuda a la tesis doctoral de Manuel Pech Palacio. Sin embargo, se desarrolla en tesis de licenciatura al mismo tiempo, que también pretende brindar ayuda a la tesis doctoral mencionada. El otro proyecto es el trabajo realizado por Daniel Pérez Armería, el cual tiene como objetivo mejorar una de las aplicaciones ya existentes en la tesis doctoral de Manuel Pech Palacio. La relación que existe entre esta tesis y la de Daniel Pérez, se encuentra al momento de evaluar los grafos visualizados y el almacenamiento en una base de datos aquellos que al usuario le parecieron interesantes y de importancia.

Entre los requerimientos de esta tesis esta el de permitir a un usuario agregar información a algún grafo que este considere interesante. La información que se incorpore tiene que ser almacenada junto con el grafo que se este comentando en una base de datos diseñada y creada previamente. Una vez almacenada la información InterGeoM debe presentarla en su aplicación de desarrollo.

InterGeoM como se puede observar en la arquitectura de esta tesis es mandada llamar por GeoMVisio si el usuario desea almacenar un grafo y por la tesis doctoral de Manuel Pech Palacio al momento de abrir la aplicación con la cual esta relacionada y mostrará todos los grafos ya almacenados según sea el caso.

En este capítulo se presentó la arquitectura general de GeoMVisio y como están integrados las diferentes herramientas para creación de archivos y de visualización y los sistemas relacionados a GeoMVisio. Esto es primordial para poder pasar al siguiente capítulo que es el análisis y diseño del software de GeoMVisio.