Rev. del Instituto de Investigación FIGMMG-UNMSM vol 21 n° 41, 2018: 53 - 60 ISSN-L:1561-0888

Uso de entidades geométricas para el análisis de entidades geográficas en los Sistemas de Información Geográfica usando el modelado de datos orientado a objetos

Geometrical entities for analysis of geographical entities in Geographic Information Systems using oriented object data modeling

Marcos Fidel Espinoza Pereyra*

Recibido: Octubre 2017 - Aprobado: Junio 2018

RESUMEN

El presente artículo propone una metodología (usando el modelado de datos orientado a objetos) en la utilización de entidades geométricas tales como *Punto, Polilínea y Polígono* de una manera abstracta con el fin de usar sus propiedades y algunas funciones para luego ser usados en los Sistemas de Información Geográfica (GIS en sus siglas en ingles) de una manera automatizada y entendible en la programación y/o automatización de algoritmos facilitando su aplicación para la representación de entidades geográficas. Al final del presente artículo se expondrán algunos ejemplos sencillos de la implementación de cada abstracción para su mejor entendimiento.

Palabras clave: Programación orientada a objetos; geometría computacional; topologías; datos espaciales; entidades geográficas; abstracción de geometrías.

ABSTRACT

This paper proposes a methodology (using Object-oriented data modeling) to use geometrical entities such as “Point” (Punto in Spanish), “Polyline” (Polilínea in Spanish) and “Polygon” (Polígono in Spanish) using an abstract way using their properties and some functions upon themselves, in order to use them in Geographic Information Systems (GIS) Programming in an easier way to represent geographical entities. The last part you can see some simple examples about performing these abstractions.

Key words: Object-oriented programming; computational geometry; topologies; spatial data;geographical entities; geometric abstraction.

---

* Ingeniero Geógrafo. Egresado de la maestría en Ciencias Ambientales, mención en Gestión y Ordenamiento Ambiental del Territorio de la Universidad Nacional Mayor de San Marcos. Lima, Perú. Editor del blog Ingeografos (www.ingeografos.com.pe) - mfespinozapereyra@gmail.com

I. INTRODUCCIÓN

En este artículo se presentan algoritmos para la implementación de abstracciones de entidades geométricas con el fin de representar entidades geográficas halladas en un espacio geográfico para su análisis en los Sistemas de Información Geográfica. Para ello se apoya en el modelado de datos orientada a objetos que ayuda en ver las propiedades de un objeto y el comportamiento del mismo ante una acción, todo esto de una forma sistémica interna (objeto y sus componentes) y externa (objeto y su entorno).

La abstracción de una geometría usando el modelado de datos orientada a objetos significa en crear una Variable que contenga una cantidad de otras Variables subordinadas (donde se guardaran algunas propiedades inherentes de la geometría y del objeto que se desea representar) y Funciones lógico-matemáticas (donde las variables subordinadas interactúan para devolver una magnitud de una propiedad de la geometría) necesarias para que pueda existir y se pueda manipular la información que se guarde en ella de una manera entendible y dinámica. Las herramientas usadas para los Sistemas de Información Geográfica, necesitan una manera de analizar el entorno geográfico y sus entidades (objetos y fenómenos) en forma de variables (geométricas y descripciones) que puedan ser entendidas de manera sencilla tanto para la propia herramienta como el usuario de la misma, además que tenga una manera más simple para leer y escribir información geométrica.

II. MARCO CONCEPTUAL

El modelado de datos orientada a objetos está basado en el paradigma de la programación orientado a objetos que da un enfoque diferente del mundo informático que implica la creación de modelos del mundo real y la creación de modelos que traten de explicar sus elementos y procesos (Joyanes, L., 1996).Con este tipo de enfoque se analiza a un objeto desde sus propios elementos y datos que lo conforman y eventos que es capaz de realizar el mismo objeto (Sánchez, J., 2010). Trabajos como el de Posada, N. & Sol, D. (S.F.); Chen, Y., Wang, Z. & Chen, Z. (2012); Albuquerque, K. (2002) y García, J. (1994) hablan del uso que se le puede dar a este paradigma en los trabajos de los Sistemas de Información Geográfica para el tratamiento de datos y por ende, ser capaz de ser un sistema que pueda gestionar datos geográficos georeferenciados; con este enfoque es posible relacionarlo con un espacio geográfico, conformado de objetos y fenómenos dinámicos (Tolaba, A., Caliusco, M. & Galli, M., 2013; Martínez-Rosales, M. & Levachkine, S., 2014) que pueden estar o no relacionados entre ellos y con el entorno mismo que pueden ser modelados para un mejor entendimiento. Estos objetos y fenómenos son capaces de ser representados por vectores (puntos, polilíneas y polígonos), para luego ser manipulados, analizados y dibujados para algún propósito que le dé el usuario final.

Estas entidades, objetos y fenómenos que están en un espacio geográfico pueden ser representados con geometrías o vectores (Vela, A., Fernández, M. & Castaño, S., 1997) que según su naturaleza y la escala de representación serian *Puntos* y la agrupación de *Puntos* que son denominados *Polilíneas* (agrupación abierta de *Puntos*) y *Polígonos* (agrupación cerrada de *Puntos*) y usando algoritmos lógicos-matemáticos en forma de funciones dentro de estas geometrías pueden dar propiedades o valores referentes sobre área, perímetro, longitud, y otras propiedades que el analista crea conveniente para su investigación. También permite usar dichas geometrías como variables para ser usadas en modelos matemáticos para realizar simulaciones del mundo real.

Ejemplos de una utilización del modelado de los elementos de un espacio geográfico por geometrías se encuentran en lo expuesto por Kumar, M., Bhatt, G. & Duffy, C. (2010) para un modelamiento hidrológico distribuido que apoyándose en las geometrías y algoritmos puede hacer que el proceso sea más dinámico y flexible para obtener mejores resultados o lo presentado por Liria, J. (2008) donde apoyándose del trazo de un taxón desde sus coordenadas primarias y utilizando algoritmos entre estas coordenadas, puede realizar un trabajo de estadística espacial para realizar estudios panbiogeográficos o el trabajo de Rodríguez, R. & Lazo, M. (2013) que mediante algoritmos sobre geometrías se determina la ruta más óptimo entre una red de caminos e impactando su beneficio en el tiempo y costo de un viaje.

III. METODOLOGIA

La metodología consiste en ir desde un objeto – geometría simple (*Punto*) hacia geometrías complejas (*Polilínea* y *Polígono*) que es un conjunto de *Puntos* relacionado entre sí. Supone también la jerarquización por agrupación de estas geometrías (Huang, M., Zhu, L., Liu, P., Wang, J. & Guo, L., 2015) donde los algoritmos presentes en geometrías (de mayor orden) como *Polilínea* y *Polígono* recaen en *Puntos* (geometrías de menor orden).

3.1 Abstracción de la geometría *Punto*

La abstracción de la geometría *Punto* se puede considerar el inicio de las demás abstracciones geometrías presentadas en el siguiente artículo. Si se crea la Variable *Punto*, se puede definir con la siguiente estructura:

```
Variable Punto

    Variable X, para la coordenada Longitud o Este, de carácter obligatorio;

    Variable Y, para coordenada Latitud o Norte, de carácter obligatorio;

    Variable Z, para el valor de la Altitud o de la Cota, de carácter opcional;

    Variable M, para un valor que se desee guardar, de carácter opcional;

    Variable id, para el valor de la identificación numérica, de carácter opcional;

    Variable Alias, para el valor de la identificación o nombre alterno, de carácter opcional;

Fin Variable Punto
```

Se observa que para definir la existencia de una variable de tipo *Punto*, se hace necesario que tenga los valores de tipo numérico real en la Variable *X* y en la Variable *Y* (se entiende, por lo tanto, que se trabaja en un espacio de coordenadas de dos dimensiones). De acuerdo a las exigencias de los algoritmos que se desee implementar, se puede agregar más variables dentro de la variable de tipo *Punto* como el caso de la Variable *Z* de tipo numérico real para guardar el valor de la Altitud (si se trabaja en un espacio en tres dimensiones), la Variable *M* (ESRI, 1998), variables requeridas para su identificación y algunas otras que sean necesarias.

La variable de tipo *Punto* puede quedar expresada en algoritmo:

NombreVariable = Punto(X, Y, Z, M, id, Alias)

Donde *NombreVariable* es alguna variable donde se guarda la información contenida en *Punto*.

3.2 Geometrías compuestas por más de un *Punto*. *Partes* de una geometría

Se define como *Polilínea* y *Polígono* a la sucesión continúa o no de Vértices (que en este caso serían variables de tipo *Punto*). Esta continuidad depende de la complejidad que puede ser una objeto o un fenómeno en el espacio geográfico como islas y lagos, una jurisdicción territorial partida en dos o más partes, entre otros casos. En la Figura 1 se puede observar esta complejidad mencionada donde se agrupan estas discontinuidades de Vértices formando *Partes* que conforman *Polilíneas* y *Polígonos* para un mejor procesamiento y análisis. En la misma Figura 1, se observa en "a)" y "b)" representan una *Polilínea* y un *Polígono* de una sola parte (vértices continuos). En "c)" se observa dos *Polilíneas*, la superior conformada por dos partes y la inferior de una sola parte. En "d)" se observa dos *Polígonos*, los cuales el *Polígono* superior tiene tres partes y el inferior solo una sola parte.

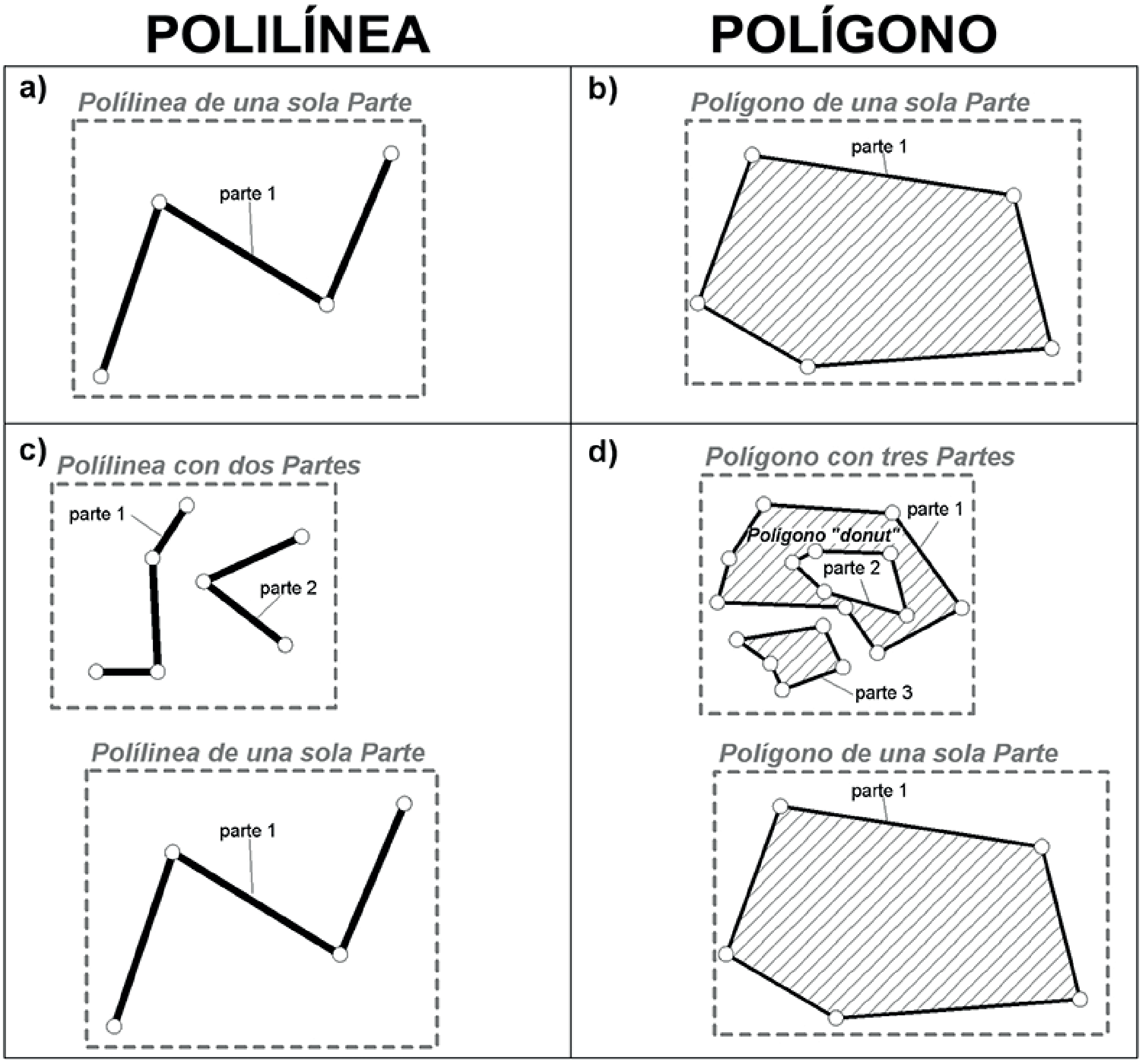

Figura 1. Representaciones de una Polilínea y un Polígono.
Fuente. Elaboración propia.

3.3 Abstracción *Parte* de una *Polilínea* y *Polígono*

La abstracción de una variable de tipo *Parte* de una *Polilínea* y *Polígono* puede definirse con la siguiente estructura:

Variable Parte

> *Variable Vértices, contenedor donde estarán almacenadas las variables de tipo Punto, de carácter obligatorio;*
>
> *Variable id, para el valor de la identificación numérica, de carácter opcional;*
>
> *Variable Alias, para el valor de la identificación o nombre alterno, de carácter opcional;*
>
> *Variable TipoBorde, para saber si el borde es exterior o interior, para el caso de polígonos de tipo Dona o Dount;*
>
> *Función CuentaVértices(), devuelve la cantidad de vértices en una variable Parte;*
>
> *Función ÁreaParte(), devuelve el área de una variable Parte;*[1]
>
> *Función PerímetroParte(), devuelve el perímetro de una variable Parte.*[2]

Fin Variable Parte

Se observa que a la misma variable se le puede agregar funciones, esto hace que la variable geométrica pueda dar valores intrínsecos a lo que representa en el mundo real como área, perímetro, entre otros.

3.4 Abstracción de la geometría Polilínea y Polígono

Conociendo que puede existir una polilínea o polígono de una o más partes, se define las abstracciones de tipo *Polilínea* y *Polígono* de la forma siguiente:

Variable Polilínea

> *Variable Partes, contenedor donde estarán almacenadas las variables de tipo Parte, de carácter obligatorio;*
>
> *Variable id, para el valor de la identificación numérica, de carácter opcional;*
>
> *Variable Alias, para el valor de la identificación o nombre alterno, de carácter opcional;*
>
> *Función CuentaPartes(), devuelve la cantidad de Partes en una variable Polilínea*
>
> *Función Longitud(), devuelve la suma de las longitudes de las partes que conforman la Polilínea.*

Fin Variable Polilínea

Variable Polígono

> *Variable Partes, contenedor donde estarán almacenadas las variables de tipo Parte, de carácter obligatorio;*
>
> *Variable id, para el valor de la identificación numérica, de carácter opcional;*
>
> *Variable Alias, para el valor de la identificación o nombre alterno, de carácter opcional;*
>
> *Función CuentaPartes(), devuelve la cantidad de Partes en una variable Polígono.*
>
> *Función Perímetro(), devuelve la suma de los perímetros de las partes que conforman el Polígono.*
>
> *Función Área(), devuelve la suma de las áreas de las Partes que conforman el Polígono.*

Fin Variable Polígono

Como en el caso de la variable *Parte*, se incluye las funciones *CuentaPartes*(), *Longitud(), Perímetro()* y *Área(),* que devuelve los valores indicados en cada uno. En el caso de la *Longitud(),* supone un sencillo cálculo de la distancia euclidiana entre dos puntos (como el teorema de Pitágoras), como también puede ser la distancia geodésica por la Ley de Cosenos u otro tipo de cálculo que puede ser un poco más complejo y exhaustivo haciendo que el algoritmo sea más largo y el rendimiento del cálculo baje (Antúnez, R. & Hernández, L., 2014). También deja abierta la idea de construir otros algoritmos para otros tipos de geometría específicos como *Línea*, *Rectángulo*, *Triangulo* y determinar su área y longitud (Davis, C., S.F.) y aplicando la Geometría Analítica para desarrollas más algoritmos de análisis de geometrías en dos dimensiones.

3.5 Geometrías y Partes de geometría vacía

Es bueno notar que como existen variables definidas (las que tienen algún valor en su interior), existen variables sin ningún valor en su interior y lo mismo se aplica a estas abstracciones que hacen referencia este artículo y las que llamaremos *Geometría Vacía* y *Parte Vacía*.

Una *Geometría Vacía* o *Parte Vacía* tiene la misma la misma estructura o definición de las tres geometrías comentadas anteriormente, la diferencia radica en que no tienen ningún valor almacenado[3].

Para el caso de una variable de tipo *Punto* Vacío:

NombreVariable=Punto(nulo,nulo,nulo,nulo,nulo,nulo)

Donde *NombreVariable* es una variable de tipo *Punto* pero sin ninguna información, pero en el transcurso del algoritmo se puede modificar o agregar la información

---

[1] Valido para *Parte* de *Polígono*.

[2] Valido para *Parte* de *Polígono*, para el caso de una Polilínea seria *LongitudParte()*.

[3] Cabe señalar que el valor numérico Cero (0) es distinto al vacío (nulo).

que aparece como *nulo*. Para el caso de una variable de tipo *Parte*, *Polilínea* y *Polígono*, se les consideran vacíos siempre y cuando en la función *CuentaVértices* (en el caso de una variable de tipo *Parte*) y en la función *CuentaPartes* (para variables de tipo *Polilínea* y *Polígono*) devuelvan un valor igual a cero (0), y un valor nulo o vacío en las variables *id* y *Alias* (en las variables subordinadas que conforman estas abstracciones de estas geometrías).

IV. EJEMPLOS Y RESULTADOS

Se presentan algunos sencillos ejemplos para un mejor entendimiento del uso de las abstracciones presentadas:

Ejemplo 1: Crear una variable de tipo *Polilínea*, con dos partes (dos variables de tipo *Parte*:

```
‘Definir variables de tipo Parte y Polilínea
Parte1 y Parte2, variables de tipo Parte {
        Variable id
        Variable Vértices
        Función CuentaVértices()
Función LongitudParte()
}

Polilínea_Ejemplo, variable de tipo Polilínea {
        Variable Partes
Variable id
        Función CuentaPartes()
Función Longitud ()
}

’usar una variable de tipo Punto para llenar los vértices de Parte1guardandolo en la variable Vértices
Vértice <- Punto(2,5,nulo,nulo,1)
Parte1 .Vértices (1) = Vértice ‘Vértice 1de Parte1

Vértice <-  Punto(5,5,nulo,nulo,2)
Parte1 .Vértices (2) <-  Vértice ‘Vértice 2 de Parte1

Vértice <- Punto(5,7,nulo,nulo,3)
Parte1 .Vértices (3) <-  Vértice ‘Vértice 3 de Parte1

Parte1.id <-  1 ‘asigno a la variable id el valor 1
Polilínea_Ejemplo .Partes(1) <-  Parte1 ‘guarda Parte1 a la variable Partes de Polilínea_Ejemplo

‘se repite lo anterior para Parte2
Vértice <-  Punto(0,1,nulo,nulo,1)
Parte2 .Vértices (1) <-  Vértice

Vértice <-  Punto(1,2,nulo,nulo,2)
Parte2 .Vértices (2) <-  Vértice

Vértice <-  Punto(2,3,nulo,nulo,3)
Parte2 .Vértices (3) <-  Vértice

Parte2.id <-  2
Polilínea_Ejemplo .Partes(2) <-  Parte2
```

Si se quiere observar los valores *X,Y* de las coordenadas de esta variable de tipo *Polilínea*:

```
NúmeroPartes <-  Polilínea_Ejemplo.CuentaPartes()
desde i <- 1 hasta NúmeroPartes
    NúmeroVértices <-  Polilínea_Ejemplo.Partes(i).CuentaVértices()
    desde j <- 1 hasta NúmeroVértices
        Vértice <- Polilínea_Ejemplo.Partes(i).Vértices(j)
        Imprime Vértice.X, Vértice.Y
    avanza j <-  j + 1
avanza i <-  i + 1
```

Se observa que es mucho más corto la impresión de los valores *X,Y* que la asignación de los mismos usando un bucle de control (conocido en la programación como un bucle *FOR…. NEXT*).

En la Figura 2 se observa que son dos variables de tipo *Parte* con tres y dos vértices respectivamente (cada vértice es una variable de tipo *Punto*). Solo se declara un solo vértice y se puede reutilizar para tantas veces se necesite.

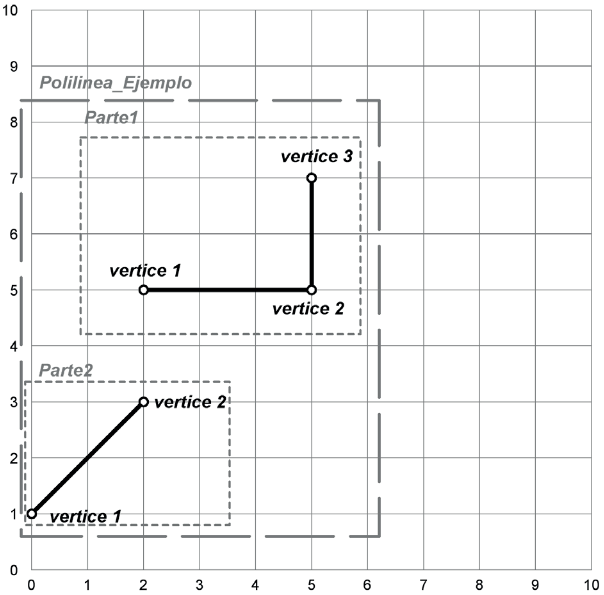

Figura 2. Grafico del Ejemplo 01
Fuente. Elaboración propia.

Ejemplo2: Calcular (de una manera muy simplificada) el área de un Polígono[4] del tipo Dona (Donut en ingles):

```
‘definición de las variables Parte1, Parte2 y Polígono_Ejemplo
Parte1 y Parte2, variables de tipo Parte {
        Variable id
        Variable Vértice
        Variable TipoBorde
        Función CuentaVértices()
        Función ÁreaParte()
        Función PerímetroParte()
}
```

---

[4] La solución general para el cálculo del área de un *Polígono* es usando determinantes de sus coordenadas.

```
Polígono_Ejemplo, de tipo Polígono {
        Variable Parte
        Función CuentaPartes()
        Función Área(){
                Área_Temporal = 0

                desde i = 1 hasta CuentaPartes()
                  Si Partes(i).TipoBorde = "Exterior"
                        Área_Temporal = Área_Temporal + Partes(i).Área()
                  Si Partes(i).TipoBorde = "Interior"
                        Área_Temporal = Área_Temporal - Partes(i).Área()
                avanza i = i + 1
                Retorna Área_Temporal
        }
}
'llenado de Parte1
Vértice = Punto(1,9,nulo,nulo,1)
Parte1.Vértices (1) = Vértice

Vértice = Punto(8,9,nulo,nulo,2)
Parte1.Vértices (2) = Vértice

Vértice = Punto(8,2,nulo,nulo,3)
Parte1.Vértices (3) = Vértice

Vértice = Punto(1,2,nulo,nulo,3)
Parte1.Vértices (4) = Vértice
Parte1.id = 1
Parte1.TipoBorde = "Exterior"
Polígono_Ejemplo.Partes(1) = Parte1

'llenado de Parte2
Vértice <- Punto(2,7,nulo,nulo,1)
Parte2. Vértices (1) <- Vértice

Vértice <- Punto(7,7,nulo,nulo,2)
Parte2.Vértices (2) <- Vértice

Vértice <- Punto(7,5,nulo,nulo,3)
Parte2.Vértices (3) <- Vértice

Vértice <- Punto(2,5,nulo,nulo,3)
Parte2.Vértices (4) <- Vértice

Parte2.id <- 2
Parte2.TipoBorde <- "Interior"
Polígono_Ejemplo.Partes(2) <- Parte2

Imprime Polígono_Ejemplo.Área() 'se ejecuta la función Área de Polígono_Ejemplo y se muestra el resultado
```

Del algoritmo presentado se observa que invocando la función *Polígono_Ejemplo.Área()* se puede obtener el área total de la variable de tipo *Polígono* que sería la suma aritmética de las *Partes* que conforman este *Polígono*. En la Figura 3 se observa que la variable *Parte1* es el borde exterior del *Polígono* de tipo *Dona (o Donut* en ingles) y la variable *Parte2* es el borde Interior, que para calcular el área, se hace la resta de ambas áreas de estas *Partes*.

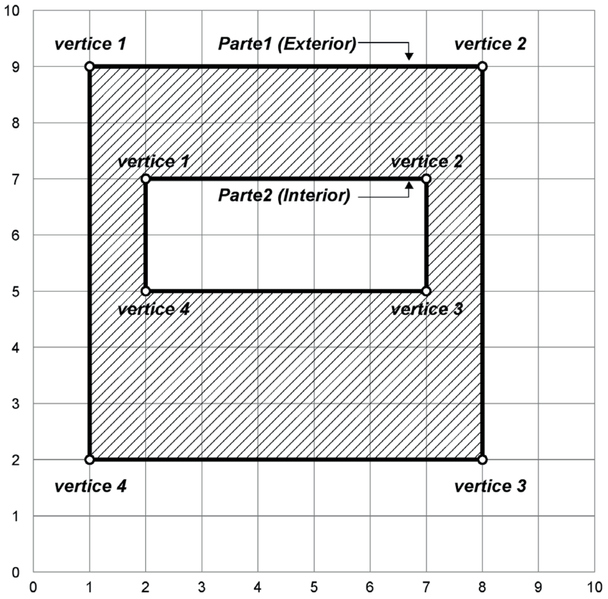

Figura 3. Grafico del Ejemplo 02
Fuente. Elaboración propia.

V. CONCLUSIONES

- El uso de abstracciones de entidades geométricas para representar entidades geográficas usando el modelado de datos orientado a objetos son de ayuda para un mejor entendimiento de algunas propiedades inherentes como la ubicación, extensión de los mismos y su interrelación con otros similares de una manera sistémica desde el mismo objeto y su relación con su entorno.

- La creación, diseño de estas abstracciones como su interrelación es de una manera sencilla, dando posibilidad de crear otras tantas como una variable de tipo *Rectángulo* (formado por dos variables de tipo *Punto* en las dos esquinas de una diagonal), otro de tipo *Circulo* (formado por una variable de tipo *Punto* y una variable para el radio), y otros más de acuerdo a la necesidad del especialista.

- La utilización de Funciones dentro de las mismas variables *Parte*, *Polilínea* y *Polígono* son el resultado de la interrelación lógico-matemática de las variables subordinadas obteniéndose valores de Perímetro, Longitud, Área, pudiéndose implementarse más funciones para hallar el Centroide de una geometría (que sería una variable de tipo *Punto*) o el rectángulo envolvente de una geometría (rectángulo formando por los vértices más externos de una geometría). La implementación de estas funciones debe estar acorde con el tipo de información que se tiene y el dominio de matemáticas, como por ejemplo la distancia por el teorema de Pitágoras entre dos puntos proyectados en un mapa UTM (Universal Transversal Mercator) o el cálculo de la distancia entre dos puntos geodésicos (longitud y latitud) utilizando la Ley de Cosenos, que aparte de la implementación se debe tener en cuenta el rendimiento del algoritmo al realizar cada cálculo.

VI. REFERENCIAS BIBLIOGRAFICAS

Albuquerque, K. (2002). *Modelagen de dados geográficos, Curso de especializaçao em geoprocessamento*. Recuperado de http://csr.ufmg.br/geoprocessamento/publicacoes/Modelagem%20de%20dados%20geografico.PDF

Antúnez, R. & Hernández, L. (2014). Propuesta de paradigma para análisis geométricos en SIG independiente del sistema de referencia. *Revista Geográfica Venezolana, 55*(1), 11-26. Recuperado de http://www.redalyc.org/pdf/3477/347732465002.pdf

Chen, Y., Wang, Z. & Chen, Z. (2012).Implementation of Object-oriented GIS data model with topological relations between spatial objects. *International Journal of Advanced Computer Science*, *2* (9), 334-338. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.428.8653&rep=rep1&type=pdf

Davis, C. (S.F.). Algoritmos *Geométricos em SIG*. Recuperado de http://andersonmedeiros.com/algoritmos-geometria-computacioal-sig/

ESRI (1998). *ESRI Shapefile Technical Description*. Recuperado de https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

García, J. (1994). Los sistemas de información geográfica orientados a objetos. *Perfiles actuales de la geografía cuantitativa en España*, 323-332. Recuperado de http://www.age-geografia.es/tig/1994_malaga/1994_028.pdf

Huang, M., Zhu, L., Liu, P., Wang, J. & Guo, L. (2015). A novel algorithm for agrregating the topological nodes in web GIS network management. *International Journal of Database Theory and Application*, *8*(1), 137-148. Recuperado de http://www.sersc.org/journals/IJDTA/vol8_no1/15.pdf

Joyanes, L. (1996). *Programación orientada a objetos*. Madrid, España: McGRAW-HILL

Kumar, M., Bhatt, G. & Duffy, C. (2010). An object-oriented shared data model for GIS and distributed hydrologic models. *International Journal of Geographical Information Science*, *24* (7), 1061-1079. Recuperado de http://people.duke.edu/~mk176/publications/KumarEtAl_ObjectOriented_IJGIS.pdf

Liria, J. (2008). Sistemas de información geográfica y análisis espaciales: un método combinado para realizar estudios panbiogeográficos. *Revista Mexicana de Biodiversidad*, *79*, 281-284. Recuperado de http://www.scielo.org.mx/pdf/rmbiodiv/v79n1/v79n1a24.pdf

Martínez-Rosales, M. & Levachkine, S. (2014). Modelo conceptual de entornos geográficos dinámicos. *Ingeniería Investigación y Tecnología*, *15*(2) ,163-174. Recuperado de http://www.scielo.org.mx/pdf/iit/v15n2/v15n2a1.pdf

Posada, N. & Sol, D. (S.F.). *Modelado de datos orientado a objetos para un sistema de información geográfica*. Recuperado de ict.udlap.mx/activities/GIS/html/files/ModeladoDeDatos.doc

Rodríguez, R. & Lazo, M. (2013). Modelo para la representación de redes y búsqueda de caminos óptimos en Sistemas de Información Geográfica. *Ingenieare*. *Revista chilena de ingeniería*, *21*(3), 394-407. Recuperado de http://www.scielo.cl/pdf/ingeniare/v21n3/art09.pdf

Sánchez, J. (2010). *Unidad 6: Programación Orientada a Objetos. Fundamentos de Programación 1º de ASI*. Recuperado de http://www.jorgesanchez.net/programacion/apuntes2009/fpr0609.pdf

Tolaba, A., Caliusco, M. & Galli, M. (2013). *Meta-ontología Geoespacial: Ontología para Representar la Semántica del Dominio Geoespacial*. Trabajo presentado en el Congreso Nacional de Ingeniería Informática – Sistemas de Información (CoNaIISI) de la Universidad Tecnológica Nacional, Santa Fe, Argentina. Recuperado de http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/1/176-473-1-DR.pdf

Vela, A., Fernández, M. & Castaño, S. (1997). La información geográfica y los S.I.G. *Ensayos: Revista de la Facultad de Educación de Albacete*. 12, 361-371. Recuperado de http://dialnet.unirioja.es/servlet/articulo?codigo=2291899