

**Apéndice A**

**Tabla comparativa de las características de un GIS**

<b>Características</b>	Pissinou	Ramesh	Shekhar	Parámetros para la Arquitectura
Soporte a la toma de decisiones	✓	✓	✓	✓
Administrar y compartir datos	✓		✓	✓
Grandes volúmenes de información	✓		✓	✓
Diferentes tipos de datos	✓		✓	✓
Objetos en 3D	✓			
Mantener Información temporal	✓			
Representación de relaciones espaciales		✓	✓	✓
Perspectiva		✓		
Operadores Espaciales		✓	✓	✓
Extender modelo a otras aplicaciones		✓		
Captura		✓	✓	✓
Almacenamiento		✓	✓	✓
Análisis Espacial		✓	✓	✓

**Apéndice B**

**Tabla comparativa de la clasificación de Consultas Espaciales**

<b>Clasificación de Consultas Espaciales según la relación entre los objetos</b>	Shekhar	Min-Soo	Open Gis	Parámetros para el modelo
Topológicas	Algunas	Algunas	✓	✓
Dirección	✓			
Métricas	✓		✓	✓
<b>Clasificación de Relaciones Espaciales Topológicas</b>				
Equals		✓	✓	✓
Disjoint	✓	✓	✓	✓
Intersects	✓	✓	✓	✓
Touches	✓	✓	✓	✓
Crosses			✓	✓
Within			✓	✓
Contains	✓	✓	✓	✓
Overlaps		✓	✓	✓
<b>Clasificación de Análisis Espacial y Métricas</b>				
Buffer		✓	✓	
ConvexHull			✓	
Intersection	✓		✓	✓
Difference			✓	
Análisis de Red	✓	✓		
Distancia (métrica)	✓		✓	

**Tabla comparativa de la clasificación de Consultas Espaciales  
en Software Comercial**

<b>Clasificación de Consultas Espaciales según la relación entre los objetos</b>	Arc/View	GeoMedia
Topológicas	Algunas	Algunas
Dirección		
Métricas	✓	
<b>Clasificación de Relaciones Espaciales Topológicas</b>		
Equals		✓
Disjoint		
Intersects	✓	
Touches		✓
Crosses		
Within	✓	✓
Contains	✓	✓
Overlaps		✓
Adjacent to ó Meet	✓	✓
Nearby	✓	
<b>Clasificación de Análisis Espacial</b>		
Buffer	✓	✓
ConvexHull		
Intersection	✓	
Difference		
Análisis de Red	✓	
Distancia (métricas)	✓	✓

## **Apéndice C**

### **Métodos Básicos de la clase Geometría**

*getDimension(g Geometry): Integer* Regresa un entero que corresponde a la dimensión de la geometría, la cual es menor o igual a la dimensión del espacio de coordenadas.

*getGeometryType(g Geometry): String* Regresa el nombre del tipo de geometría de la cual es instancia la geometría.

*getSRID(g Geometry): Integer* Regresa el tipo de Sistema de Referencia Espacial de la geometría.

*IsEmpty(g Geometry): Integer* Regresa un entero, con valor 1 si la geometría corresponde a el conjunto vacío, 0 para falso y -1 en caso de que no exista o null.

*IsSimple(g Geometry): Integer* Regresa un entero, con valor 1 si la geometría es simple, 0 para falso y -1 en caso de que no exista o null.

*Boundary(g Geometry): Geometry* Regresa una geometría que representa los límites de la geometría dada.

### **Métodos para obtener las relaciones Espaciales entre objetos geométricos**

*Equals(g Geometry): Integer* Regresa un entero con valor 1 si la relación “igual” es verdadera, 0 para falso y -1 en caso de que no exista o null.

*Disjoint(g Geometry): Integer* Regresa un entero con valor 1 si la relación “no toca a” es verdadera, 0 para falso y -1 en caso de que no exista o null.

*Intersects(g Geometry): Integer* Regresa un entero con valor 1 si la relación “intersección” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

*Touches(g Geometry): Integer* Regresa un entero con valor 1 si la relación “toca a” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

*Crosses(g Geometry): Integer* Regresa un entero con valor 1 si la relación “cruza a” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

*Within(g Geometry): Integer* Regresa un entero con valor 1 si la relación “esta dentro de” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

*Contains(g Geometry): Integer* Regresa un entero con valor 1 si la relación “contiene a” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

*Overlaps(g Geometry): Integer* Regresa un entero con valor 1 si la relación “cubre parte de” es verdadera, 0 para falso y  $-1$  en caso de que no exista o null.

### **Métodos de la clase Punto**

*getX(): Double* Regresa el valor de la coordenada X del punto.

*getY(): Double* Regresa el valor de la coordenada y del punto.

### **Métodos de la clase MultiPunto**

*getNumPoint(): Integer* Regresa el número de puntos que tiene este multipunto.

*getPointN(n Integer): Point* Regresa el punto indicado de este multipunto.

### **Métodos de la clase Polilínea**

*getStartPoint(): Point* Regresa el punto inicial de esta Polilínea.

*getEndPoint(): Point* Regresa el punto final de esta Polilínea.

*getNumPoints(): Point* Regresa el número de puntos de esta Polilínea.

*getPointN(n Integer): Point* Regresa el punto indicado de esta Polilínea.

*IsClosed(): Integer* Regresa un entero con valor 1 si la Polilínea es cerrada, 0 para falso. (punto inicial = punto final)

*IsRing(): Integer* Regresa un entero con valor 1 si la Polilínea es un anillo, 0 para falso. (punto inicial = punto final y esta Polilínea es simple)

### **Métodos de la clase MultiPolilínea**

*getNumLinesString(): Integer* Regresa el número de Polilíneas que tiene esta MultiLineString.

*getLineStringN(n Integer): LineString* Regresa la Polilínea indicada de esta MultiPolilínea.

### **Métodos de la clase Polígono**

*getExteriorRing(): LineString* Regresa el anillo exterior de este Polígono.

### **Métodos de la clase MultiPolígono**

*getNumPolygons(): Integer* Regresa el número de Polígonos que tiene este MultiPolígono.

*getPolygonN(n Integer): Polygon* Regresa el Polígono indicado de este MultiPolígono.