

## Desarrollo personalizado de aplicaciones SIG 3D

*M. de la Calle Alonso<sup>(1)</sup>, D. Gómez-Deck<sup>(1)</sup>, V. Olaya Ferrero<sup>(2)</sup>*

<sup>(1)</sup> IGO SOFTWARE. C/Santiago Caldera 4 Cáceres. [mdelacalle@igosoftware.es](mailto:mdelacalle@igosoftware.es)

<sup>(2)</sup> Universidad de Extremadura, [volaya@unex.es](mailto:volaya@unex.es)

### RESUMEN

*En los últimos años la proliferación de aplicaciones 3D en SIG ha sido enorme, desde la aparición de Google Earth el usuario está familiarizado con entornos 3D. Por otra parte los ordenadores con aceleración 3D son comunes en la actualidad y el acceso a banda ancha es prácticamente generalizado, además cada vez hay mayor cantidad de datos públicos que pueden ser utilizados por clientes SIG que sean capas de recibir datos de internet. Hay varias librerías apropiadas para la realización de este tipo de aplicaciones. IGO SOFTWARE ha comenzado a desarrollar aplicaciones 3D con la librería Nasa World Wind SDK para java. Decidimos usar estas librerías por su robustez, sencillez, cantidad de ejemplo, estar hechas en java (nos permite su unión a muchas librerías SIG) y su uso de caché local.*

*Las aplicaciones desarrolladas en 3D no sólo son visualmente más atractivas, también nos ofrecen más información que el SIG clásico en 2D. Gracias a la integración de librerías como SEXTANTE ahora es posible también realizar análisis.*

*En un futuro se espera poder desarrollar también algoritmos de análisis en 3D usando dicha plataforma. Por otra parte, llevamos desde hace algunos años desarrollando aplicaciones para el tratamiento de nubes de puntos proveniente de Láser Escáner y LIDAR, esta plataforma es ideal para mostrar nubes de puntos, por lo que es perfecta para visualizar nubes de puntos georreferenciadas.*

*Mostraremos ejemplos de aplicaciones programadas con estas librerías y las posibilidades que vemos de cara a un futuro a este tipo de desarrollos*

**Key words:** 3D, LIDAR, SEXTANTE, NASA WORLD WIND, LÁSER ESCÁNER, JAVA, JOGI, EUCLID

## INTRODUCCIÓN

Uno de los próximos pasos en la computación debería ser la generalización del uso del 3D para todas las aplicaciones que lo permitan. El mundo SIG no es ajeno a ello, y lo que comenzó por el uso de globos 3D donde representamos información 2D debe evolucionar a que los SIG sean capaces de representar información de manera.

Nuestros planes futuros pasan por la elaboración de librerías genéricas para el manejo de geometrías en 3D y por su visualización en productos SIG tales como Nasa World Wind.

## Euclid

Es un Framework para manejar geometrías multidimensional y multiprecisión. Es decir, podemos trabajar con él indistintamente típicamente de las dimensiones en que tengamos los datos 2D – 3D y de su precisión. Euclid se encarga de gestionar todo de manera que siempre los cálculos sean correctos.

Se han desarrollado multitud de optimizaciones que nos permiten por ejemplo construir un octGrid con 127 millones de puntos con color. Estas optimizaciones se han hecho en dos sentidos:

1. Optimizar el uso de la memoria. Uso apropiado de las Clases de Java. (Clases Buffer)
2. Aprovechamiento de la arquitectura de múltiples núcleos de los procesadores de última generación usando las posibilidades de programación multihilo de Java.

Un ejemplo de cómo funciona Euclid son las clases de generalización de una bola n-dimensional, esta sería la clase *n-Ball*, su especialización a 2D sería *Disk* y la especialización a 3D sería *Ball*.

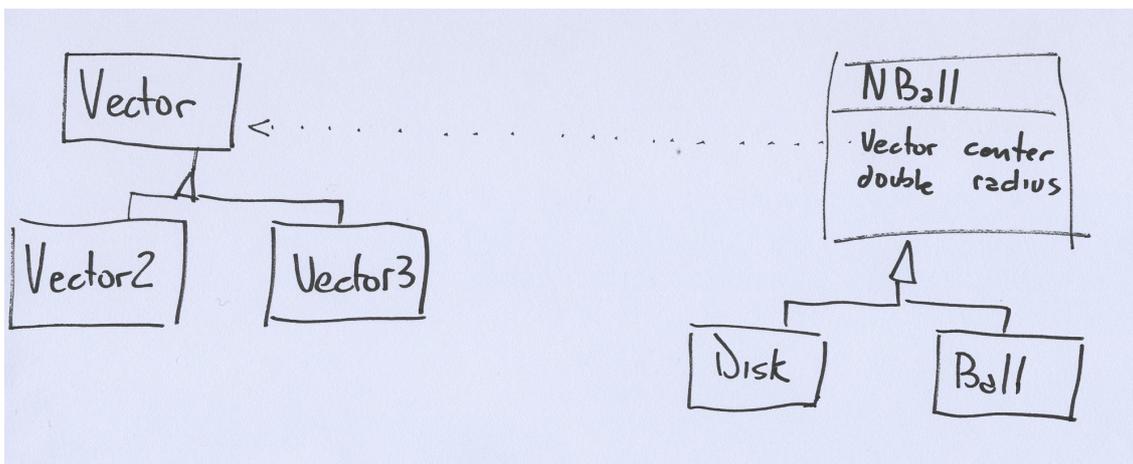


Figura 1: Euclid

Un método que funcionaría en las n-dimensiones podría ser:

```

public double distanceToBoundary(final VectorT point)
{
    return center.distance(point) - radius;
}
  
```

Este método devolvería para cualquier dimensión la distancia al borde.

Hay muchos ejemplos de algoritmos u operaciones que pueden ser multidimensionales, EUCLID nos abstrae de ello y nos permite trabajar con uno u otra.

### **S3xtante**

Sextante (Sistema Extremeño de Análisis Territorial, <http://sextantegis.blogspot.com/>). La primera aplicación de Euclid ha sido hacer que funcionen en n-dimensiones los algoritmos de Sextante que fueran n-dimensionales. Haciendo cambios muy pequeños en los algoritmos, es decir, simplemente implementando Euclid y trabajando con operaciones de nuestro framework de geometrías estos pasan a poder ser usados en 3D sin ningún tipo especial de cambio.

Un ejemplo sencillo en este sentido lo encontramos en el clásico buffer o zona de influencia. Si tomamos, por ejemplo, un segmento lineal (una línea con tan sólo 2 nodos, uno inicial y otro final), el buffer calculado para una distancia dada sería un rectángulo de longitud la misma que el segmento y de altura el doble que la distancia de influencia. Llevando este concepto a 3D el resultado sería un cilindro de radio dicha distancia y altura la longitud del segmento. De igual modo puede extenderse para otras figuras más complejas, siendo siempre el resultado una nueva figura tridimensional.

Cuando un algoritmo no es multidimensional, el Euclid no es suficiente para solucionar la nueva tarea. Este es el caso, por ejemplo, de los algoritmos de interpolación, que en lugar de generar capas raster (2D), ahora han de generar volúmenes raster (sustituir píxeles por voxels). El uso de Euclid soluciona la entrada de datos, puesto que estos son vectoriales, pero los resultados, que no son de tipo vectorial, no pueden gestionarse de igual modo. El algoritmo de interpolación, por su parte, al no estar implementado en la librería, debe modificarse igualmente.

### **Aplicaciones personalizadas. Framework 3D.**

Las aplicaciones SIG 3D que vamos a programar pueden ser muy distintas, pero su arquitectura básica es la siguiente:

**GeoTools** → Acceso a datos SIG  
**Proj4** → Manejo de proyecciones  
**Euclid** → Geometrías  
**S3xtante** → Análisis  
**Nasa World Wind** → Visualización

Según los requerimientos pueden incluirse más elementos a la arquitectura pero estas 5 librerías suelen ser el mínimo común entre todas ellas.



## Sig Histórico

Este desarrollo es bastante simple pero tiene el interés de introducir la variable temporal en el SIG.

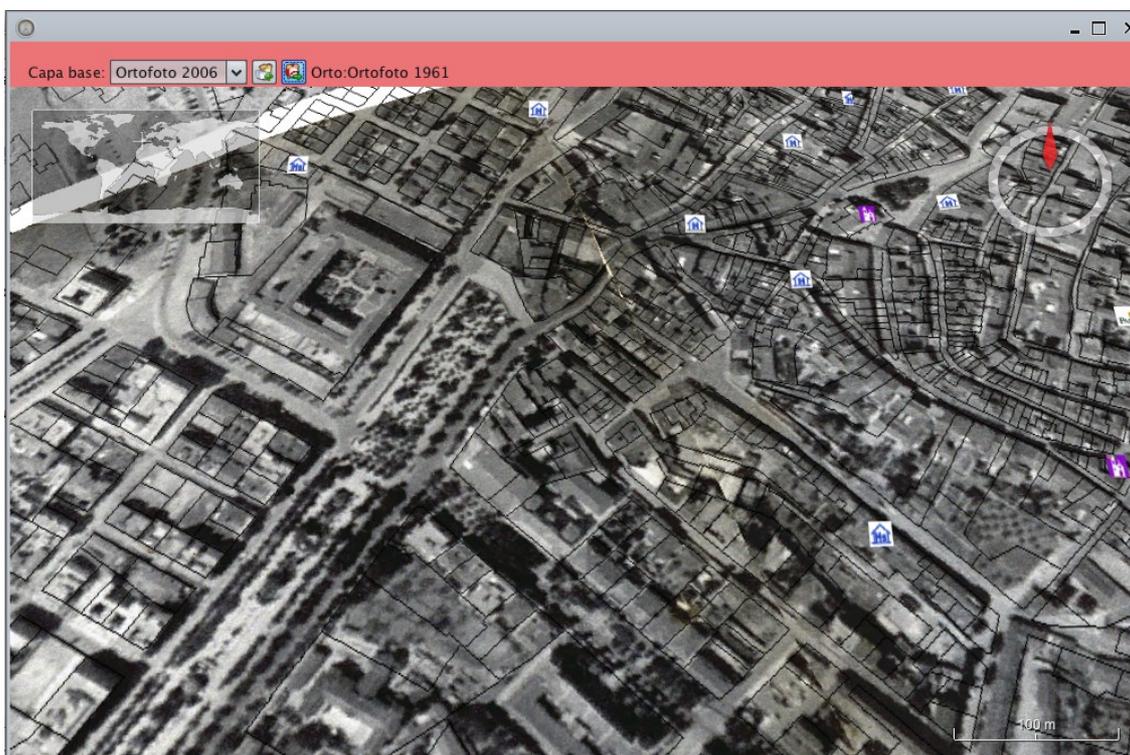


Figura 4: SIG Histórico

## Visor LIDAR 3D

No existen visores LIDAR en 3D dentro del panorama OpenSource siendo nubes de puntos 3D, con este visualizador podremos ver estas nubes y diseñar herramientas para trabajar con esa nube y visualizarlas en 3D en el acto. Este visor es experimental totalmente y para él se han utilizado numerosas optimizaciones, ya que los datos LIDAR suelen ser muy pesados.

Se utilizan técnicas como LOD (Level of Detail) y se llevan al límite las capacidades de visualización de Nasa World Wind y JOGL.

## CONCLUSIONES

Creemos que el uso de 3D está especialmente indicado para muchas actividades que realizan los usuarios SIG en la actualidad. Dentro del panorama open source, como se puede ver en esta comunicación, tenemos herramientas suficientes para construir una arquitectura que nos permite hacer aplicaciones SIG personalizadas. Por otra parte, aunque existe estas librerías, vimos también que cuando hemos querido empezar a desarrollar herramientas de análisis, no teníamos todo lo que necesitábamos por lo que ha habido que implementar librerías para ello.