



**Centro Universitario de la Defensa
en la Escuela Naval Militar**

TRABAJO FIN DE GRADO

*Análisis de los sistemas de indexado geoespacial para el
Conocimiento del Entorno Marítimo*

Grado en Ingeniería Mecánica

ALUMNO: Víctor Alonso Aller

DIRECTORES: Miguel Rodelgo Lacruz

CURSO ACADÉMICO: 2020-2021

Universida_deVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Análisis de los sistemas de indexado geoespacial para el
Conocimiento del Entorno Marítimo*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

Universida_{de}Vigo

RESUMEN

Este trabajo nace de la adaptación de la Armada Española a las nuevas tecnologías en su proyecto de Armada 4.0. Particularmente, se estudia la aplicación de un sistema de indexado geoespacial que ayude al centro de vigilancia marítima de la Armada, el COVAM, facilitando el desempeño de sus funciones de análisis de datos, con el objetivo de mejorar el Conocimiento del Entorno Marítimo.

Se ha realizado un estudio de los métodos de indexado geoespacial, así como de las características del *Big Data* y las particularidades del entorno marítimo como fuente de información, escogiendo uno de los sistemas para su prueba con datos reales. Estos sistemas dividen la superficie terrestre en celdas y asignan un índice jerárquico, agilizando el análisis de datos.

El sistema elegido es Uber H3, tras comparar sus características con las del resto de herramientas analizadas. Mediante el sistema gestor de datos MySQL y la aplicación de este sistema de indexado se han analizado datos reales, realizando consultas espaciales como la búsqueda de buques en una zona determinada o la detección de zonas habituales.

Finalmente se ha mostrado la mejora en cuanto al procesamiento de datos espaciales y la velocidad de respuesta frente a un sistema tradicional, agilizando el proceso de consulta a grandes bases de datos como realiza el COVAM a diario.

PALABRAS CLAVE

Indexado geoespacial, H3, *Big Data*, COVAM, MySQL

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mis compañeros de la promoción 421. Durante estos años he tenido la oportunidad y la suerte de compartir con ellos experiencias que me han cambiado para siempre. Me llevo conmigo muchos amigos, e incluso algunos a los que puedo considerar hermanos.

Agradecer a mi tutor, Miguel, por mostrar el interés y la dedicación que tanto me han ayudado a lo largo del desarrollo de este trabajo. Su guía y cercanía han logrado que haya podido desarrollar todo este trabajo, superar los obstáculos que han surgido y dar lo mejor de mí.

Mi más sincera gratitud al CC Juan Saúl Sabín Sánchez y al CC Faustino Moreno Pérez, por todo el apoyo recibido, así como la disposición y trato. Sus conocimientos me han ampliado la visión de la Armada y el futuro que me espera.

A Yu, por su paciencia como amigo y todos sus consejos.

Concluyendo, quiero agradecer especialmente a mi familia por su apoyo incondicional en todos estos años de escuela, ellos me han enseñado los valores fundamentales de la vida y a saber cómo vivirla. A mi abuelo Antonio, que hace poco más de un año que nos dejó, pero sigue y seguirá siendo un ejemplo de sacrificio y trabajo diario. Y por supuesto, a Andrea, por su apoyo diario y no menos importante, por aguantarme.

A todos ellos, muchas gracias.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	6
1 Introducción y objetivos	7
1.1 Contexto y motivación	7
1.2 Objetivos	10
1.3 Estructura de la memoria	10
2 Estado del arte	13
2.1 Big Data	13
2.1.1 Definición	13
2.1.2 Big Data en el Conocimiento del Entorno Marítimo.....	15
2.1.3 Datos AIS.....	17
2.2 El COVAM	18
2.2.1 Misión y funciones	19
2.2.2 Gestión y análisis de datos en el COVAM	19
2.2.3 Interés de la Armada en las nuevas tecnologías: Armada 4.0	20
2.3 El indexado geoespacial.....	21
2.3.1 Definición	21
2.3.2 Historia	22
2.3.3 La información geográfica.....	23
2.4 Métodos de indexación espacial	24
2.4.1 Análisis de datos espaciales.....	24
2.4.2 Grid spatial index.....	25
2.4.3 Z-order curve	26
2.4.4 HHCCode	27
2.4.5 Quadtree.....	27
2.4.6 Octree.....	29
2.4.7 R-tree	29
2.5 Análisis de los sistemas de indexado geoespacial	31
2.5.1 Uber H3	31
2.5.2 Google S2	34
2.5.3 Geohash	37
2.5.4 German Naval Grid.....	40

2.5.5 UTM	43
2.5.6 Warp 10	47
2.6 Spatial databases	48
2.6.1 Database Management Systems: MySQL	50
2.6.2 Spatial databases: IBM Informix	51
2.6.3 Spatial databases: Amazon Aurora	52
2.6.4 Spatial databases: Oracle Spatial	53
2.7 Tabla comparativa final	55
3 Desarrollo del TFG.....	57
3.1 Instalación de herramientas y software necesario.....	57
3.1.1 Sistema operativo: Linux Ubuntu	57
3.1.2 Lenguaje de programación: C.....	58
3.1.3 Sistema de indexación espacial: H3	58
3.1.4 SGDB: MySQL	59
3.1.5 Visualización geoespacial: Google Earth y Kepler.gl	61
3.2 Análisis de datos geoespaciales	61
3.2.1 Buques dentro de un área determinada.....	61
3.2.2 Detección de zonas habituales y buques estáticos	76
3.2.3 Detección de encuentros	82
3.3 Visualización de índices generados	84
3.3.1 Visualización: Área de búsqueda.....	87
3.3.2 Visualización: Zona habitual de un buque.....	87
3.3.3 Visualización: Zonas según velocidad media.....	89
4 Resultados	91
4.1 Resumen de resultados.....	91
5 Conclusiones y líneas futuras	95
5.1 Revisión de los objetivos	95
5.2 Líneas futuras	96
6 Bibliografía.....	97
Anexo I: etiqueta10.c	103
Anexo II: radio_exp3.c.....	105
Anexo III: Consulta MySQL.....	107
Anexo IV: Extracto cells8.kml.....	108

ÍNDICE DE FIGURAS

Figura 1-1. Captura Marine Traffic en el entorno español. [4]	8
Figura 1-2. COVAM. [6].....	9
Figura 2-1. Aplicaciones Big Data y fuentes. [10].....	15
Figura 2-2. Esquema general de datos manejados por un buque. [11].....	16
Figura 2-3. Esquema Vigilancia Marítima en el COVAM. [14].....	19
Figura 2-4. Zonas y eventos de interés en el COVAM. [14].....	20
Figura 2-5. Geolocalizador. [20]	22
Figura 2-6. Modelo de datos vectorial. [21]	24
Figura 2-7. Grid Indexing. [25]	25
Figura 2-8. Grid triangular generado por bisección recursiva. [26].....	26
Figura 2-9. Z-ordering. [25]	27
Figura 2-10. Ejemplo árbol <i>quadtree</i> . [28].....	28
Figura 2-11. Ejemplo elementos máximos m en las celdas (m= 2). [25].....	28
Figura 2-12. Ejemplo R-Tree espacial y esquemáticamente. [29]	29
Figura 2-13. Ejemplo de sobrecarga en R-tree. [29]	30
Figura 2-14. Ejemplo de mallado de Celdas H3. [31].....	31
Figura 2-15. Distancia entre centros de celdas según su geometría. [31]	32
Figura 2-16. Icosaedro terrestre. [31]	32
Figura 2-17. Ejemplo de subdivisiones del mallado de H3. [31]	32
Figura 2-18. Esfera de S2 y el cubo adscrito que envuelve la Tierra, con la curva de Hilvert en cada uno de los planos del cubo. [35]	34
Figura 2-19. Creación ID de la celda S2. [37].....	35
Figura 2-20. Curva de Hilvert sobre el globo. [36]	36
Figura 2-21. Ejemplo jerarquización mixta S2. [38].....	37
Figura 2-22. Niveles 1 y 2 Geohash. [41]	38
Figura 2-23. Curva de orden Z recorriendo las celdas. Imagen de [42].....	39
Figura 2-24. Captura de [44] en Marín, Pontevedra.....	39
Figura 2-25. Ejemplo: Atlántico, Caribe y Mediterráneo mallado mediante el sistema alemán. [46]	40
Figura 2-26. Ejemplo: Grid AE y todos los cuadrantes que lo conforman. [46].....	41
Figura 2-27. Capturas de búsqueda en [45] con coordenadas Marín, Pontevedra.	42
Figura 2-28. Proyección UTM. [48].....	43
Figura 2-29. 60 divisiones de Este a Oeste en UTM. [50]	43
Figura 2-30. Sistema de coordenadas UTM completo. [47]	44
Figura 2-31. Ejes verticales y horizontales para la designación métrica. [48].....	45

Figura 2-32. Coordenadas UTM de Marín, Pontevedra. [51]	45
Figura 2-33. Ejemplo correspondencia coordenadas UTM – 100.000 m2. [52].....	46
Figura 2-34. Coordenadas de Marín en MGRS/USNG/NATO UTM. [51].....	47
Figura 2-35. HHCode en nivel 2 y 4. [53].....	47
Figura 2-36. Ejemplo análisis espacial IBM Informix. [59]	52
Figura 2-37. Análisis espacial de la Z curve en Aurora. [62]	53
Figura 2-38. Características y ejemplo de la base de datos Oracle Spatial. [65]	54
Figura 3-1. Ejemplo de tabla en MySQL	59
Figura 3-2. Descripción tabla <i>aismediterraneo</i>	60
Figura 3-3. Descripción tabla <i>histórico</i>	60
Figura 3-4. Resultado consulta MySQL a puntos dentro de un área determinada.	62
Figura 3-5. Descripción tabla <i>etiquetas</i>	63
Figura 3-6. Vecinos para $k=1$ y vértices de la etiqueta correspondiente al área del experimento....	63
Figura 3-7. Vecinos de un salto ($k=1$) de la celda correspondiente a 36°N, 1°E.....	64
Figura 3-8. Resultado de análisis con resolución de nivel 10.	65
Figura 3-9. Resultado de análisis con nivel de resolución 9.	65
Figura 3-10. Resultado de análisis con nivel de resolución 8.	66
Figura 3-11. Resultados con índice en columnas etiqueta10, 9 y 8.	67
Figura 3-12. Resultados de <i>radio_exp2.c</i> (izq.) y <i>radio_exp3.c</i> (dcha.) para nivel de resolución 10.	68
Figura 3-13. Comparación temporal de análisis en un área determinada.....	69
Figura 3-14. Tiempo de análisis, aleatorios, según k saltos.	69
Figura 3-15. Ejemplo de identificadores de etiqueta.....	70
Figura 3-16. Celdas <i>children</i> de un nivel de resolución inmediatamente mayor. [33]	71
Figura 3-17. Menor error posible.	73
Figura 3-18. Gráfica error generado según k para un radio de búsqueda 7 km.	73
Figura 3-19. Gráfica error generado según el nivel de resolución	75
Figura 3-20. Mayor error posible.	76
Figura 3-21. Ejemplo de detección de zona habitual.	77
Figura 3-22. Ejemplo de consulta de buques estáticos.....	78
Figura 3-23. 25 buques que más repiten celda.	79
Figura 3-24. Ocupación <i>Welvaart</i> en misma celda cumpliendo el requisito de más de dos horas. .79	
Figura 3-25. Zonas habituales <i>Sea Falcon</i>	80
Figura 3-26. Gráfico detección de buques en la misma zona.....	81
Figura 3-27. Zonas habituales <i>Sea Falcon</i> . (2)	81
Figura 3-28. Zona habitual de buques con velocidad diferente de cero.....	82

Figura 3-29. Ejemplo encuentros (limitado 25 resultados).	83
Figura 3-30. Gráfico temporal según número de encuentros detectados.	84
Figura 3-31. Captura Google Earth de <i>cells8.kml</i>	85
Figura 3-32. Zoom en Google Earth, celdas ocupadas en las inmediaciones de Cabo Palos, Murcia.	85
Figura 3-33. Captura Kepler.gl con <i>cuenta10.csv</i>	86
Figura 3-34. Exploración de buques en una zona (37°N, 1°E).	87
Figura 3-35. Celdas según número de registros, inmediaciones de Cartagena.	88
Figura 3-36. Zona habitual del <i>Sea Falcon</i> , con las celdas más ocupadas en rojo oscuro (izq.) y aplicando altura (dcha.).....	89
Figura 3-37. Captura Kepler.gl, <i>avgsog8.csv</i>	90
Figura 3-38. Celdas según velocidad media (<i>avg_sog</i>), inmediaciones de Cartagena.....	90
Figura 4-1. Resumen resultados obtenidos.....	92
Figura A3-1. Análisis con etiquetas truncadas.	107

ÍNDICE DE TABLAS

Tabla 2-1. Tabla comparativa de datos antes y después del Big Data. [9].....	14
Tabla 2-2. Tipo de barco según código. [13].....	17
Tabla 2-3. Niveles de resolución H3. [32]	33
Tabla 2-4. Niveles de precisión Geohash. [41]	38
Tabla 2-5. Correspondencia latitud a zonas UTM. [48].....	44
Tabla 2-6. Escalas disponibles en Warp 10. [54]	48
Tabla 2-7. Ejemplo de operaciones algebraicas en bases de datos espaciales [55].....	49
Tabla 2-8. Tabla ejemplo de base de datos de un supermercado.	51
Tabla 2-9. Comparativa final de los sistemas de indexado analizados.	55
Tabla 3-1. Valores de error tomando el k óptimo para diferentes radios de búsqueda.	74

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto y motivación

La sociedad y el mundo actual avanza a pasos agigantados gracias a las nuevas tecnologías que los empujan. La información que circula por todo el globo cada vez requiere de mayor espacio de almacenamiento y velocidad. Esto hace que en todos los ámbitos sociales se quiera acceder a esa información rápida y fácilmente. Internet y las grandes redes motivan este crecimiento que cada día es mayor.

Es por tanto una necesidad almacenar esta información en grandes bases de datos y que su empleo facilite el trabajo a empresas, organizaciones, o incluso a los mismos gobiernos y estados. Esta información puede ser de cualquier tipo y de gran importancia para objetos como la gestión de toma de decisiones, estrategias comerciales, análisis de datos, etc., abarcando todos los ámbitos sociales, científicos y técnicos.

Dado esto, viendo el infinito abanico de opciones que ofrece el avance tecnológico, la gestión de grandes bases de datos informatizados es importante que las personas seamos capaces de gestionar todos los elementos que esta información proporciona, categorizarlos y priorizar según sean los intereses particulares. La inteligencia artificial tiene que ser capaz de que los resultados que se obtengan sean los correctos, además de que se encuentren rápidamente y con la menor ambigüedad posible.

Particularmente, una de las aplicaciones que ofrecen las nuevas tecnologías y esta capacidad de informatización de datos es la de analizar de datos espaciales y geográficos. Nos encontramos con una necesidad inminente de ser capaces de controlar todo el entorno geográfico, ser capaces de localizar espacialmente un elemento con el menor error posible y analizar datos o características que nos interesen.

La Armada, como importante entidad que es, no debe quedarse rezagada en esta carrera y es por eso que también se suma a la necesidad de implementar esta inteligencia artificial en su proyecto de Armada 4.0. La gran cantidad de datos que se puede llegar a manejar y, particularmente en el entorno marítimo, priman la precisión y la velocidad a la hora de tomar cualquier decisión que influya en éste. El entorno marítimo, como muchos de los escenarios actuales, cambia constantemente. En un mundo globalizado dónde la conexión entre países a través de los océanos y mares es esencial, es de vital importancia mantener un estricto control por parte de los estados o entidades pertinentes de las actividades que tienen lugar. La Armada no se queda atrás y este control en las zonas de interés españolas es uno de sus principales cometidos.

Cómo se especifica en las líneas generales de la Armada [1], en el área de inteligencia se quiere centrar el esfuerzo en la inteligencia naval a nivel táctico. Aprovechando los nuevos medios, se hace especial referencia al COVAM (Centro de Operaciones de Vigilancia Marítima) y a los sistemas de

mando y control, clasificados o no, que a su vez han de coordinarse con otros centros de información de seguridad marítima y ayudarse de la fusión de información nacional e internacional recurriendo a la Inteligencia Artificial (IA), ya nombrada, y el *Big Data* para sacar el máximo rendimiento de todos los datos posibles. De este trabajo se vale la FAM (Fuerza de Acción Marítima) para crear una imagen concisa de los espacios de interés nacional conocido como CEM (Conocimiento del Entorno Marítimo). Todo ello desemboca en la creación de un proyecto de investigación y análisis con el objetivo de mejorar Conocimiento del Entorno Marítimo que se convierte en la obtención de una *Recognized Maritime Picture* (RMP) como producto para los buques de la Armada.

La razón que motiva este desarrollo tecnológico, aparte de la propia innovación, es la necesidad de controlar todo el espacio marítimo que concierne tanto a España como a sus aliados de la OTAN (Organización del Tratado del Atlántico Norte) y la UE (Unión Europea). Actualmente, según el Ministerio para la Transición Ecológica y el Reto Demográfico [2], haciendo referencia a la Conferencia de las Naciones Unidas sobre Comercio y Transporte (UNCTAD), más del 90% de comercio es transportado vía marítima. Traducido a cifras reales, la demanda mundial de transporte marítimo creció en 2008 un 4,3% con relación al año anterior, hasta alcanzar los 7.745 millones de toneladas.

La importancia estratégica del comercio marítimo se refleja en los datos del sector del transporte, Según Lloyd's Register Fairplay (LRF), a 1 de enero de 2010, la flota mercante mundial estaba compuesta por 53.948 buques. La importancia comercial es enorme en este ámbito, mientras que el transporte de personas se reduce a cruceros y ferris de pequeñas distancias, el tráfico marítimo de mercantes supera con creces al resto de redes comerciales y siempre es interés principal de las naciones con costa su control y regulación.

Desde la perspectiva de la Armada, a todo el ámbito comercial se le suma la necesidad de mantener la seguridad de nuestras costas y nuestros intereses. España se encuentra en un lugar crucial para el tránsito marítimo (Figura 1-1), más de 200.000 buques al año navegan por las líneas de comunicaciones marítimas españolas y la seguridad se ve afectada por actividades ilegales de todo tipo incluyendo terrorismo, ciber delincuencia, tráfico de armas, drogas o personas, expolio del patrimonio acuático, inmigración ilegal, pesca ilegal, piratería, etc [3].

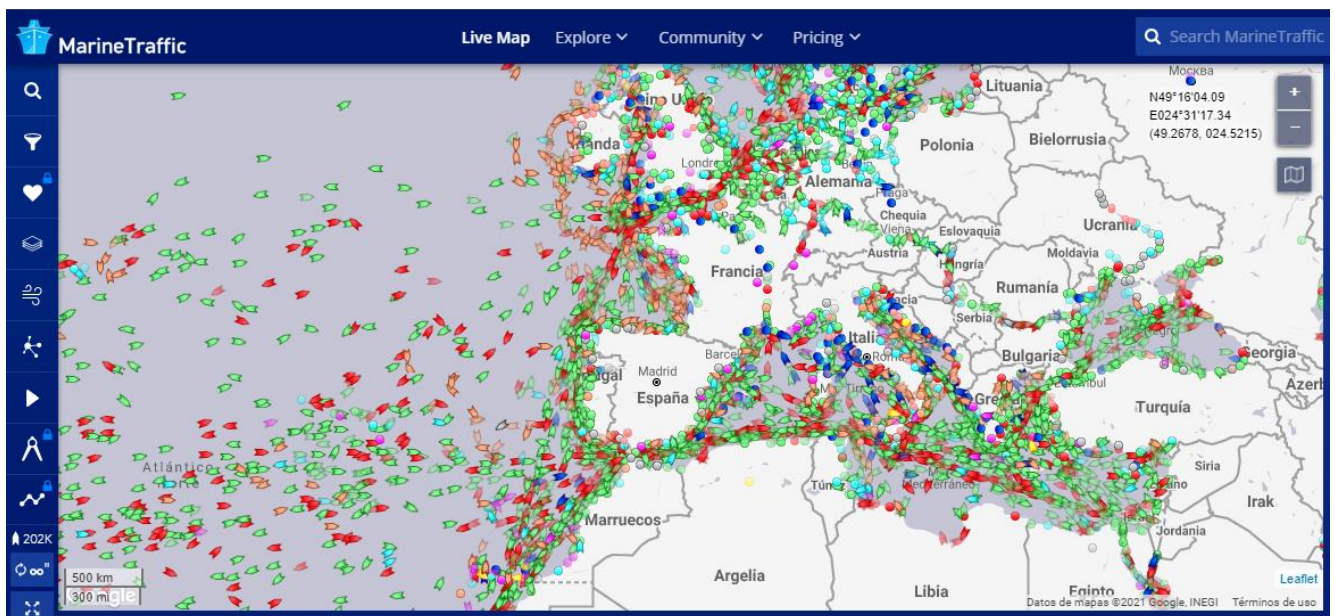


Figura 1-1. Captura Marine Traffic en el entorno español. [4]

Para llevar a cabo ese control de manera eficaz y precisa es necesaria una intensa y demandante vigilancia que tiene lugar en el COVAM (Figura 1-2), suponiendo un esfuerzo muy grande para sus miembros.

Es cuando se propone un Proyecto de Investigación (I+D+i) a desarrollar por el CUD de la ENM para la aplicación de la Inteligencia Artificial en el COVAM en las líneas de actuación del Plan de Transformación Digital de la Armada [5] con el objetivo de aplicar todas las nuevas técnicas de IA que se han descrito. En resumen, se tienen tres aplicaciones directas a corto y medio plazo, se quiere tratar todo el flujo de datos de los buques (*Big Data*), analizar esta información (*Data Analytics*) y detectar escenarios particulares de interés o cualquier anomalía explotando la IA (*Neural Networks* y *Machine Learning*). Estos conceptos están estrechamente relacionados con este trabajo, el cual se enmarca en este proyecto, siendo esto la motivación principal del TFG.

Con esto se quiere automatizar y facilitar el manejo de información, haciendo que sea más fácil y rápido reaccionar ante cualquier situación. Uno de los conceptos que se está tratando en este proyecto de innovación es el de la indexación geoespacial, mediante la aplicación de uno de los múltiples sistemas de indexado se pretende localizar más rápido a buques, son capaces de agilizar el procesado de información geográfica. Gestionan estos datos construyendo estructuras espaciales, que varían de unos a otros, y hacen que el manejo de cualquier posición geográfica sea más sencillo y más rápido. Estos sistemas se utilizan en muchas de las aplicaciones que usamos día a día que utilizan información geográfica como *Google Maps* o *Uber*. Básicamente se trata de analizar los datos de las posiciones de los buques, ordenar estos datos y movimientos de manera jerárquica y mediante diversas características o funcionalidades de estos sistemas simplificar el trabajo y el análisis de la gran cantidad de datos como trayectorias de buques, lugares más transitados, puntos de fondeo más usados, etc. Los sistemas de indexado geoespacial generan unos índices que permiten realizar estas consultas mucho más rápido, soportando mejor los datos en 2D que los índices tradicionales. Almacena las coordenadas geográficas de los datos y los ordena de tal forma que el acceso y las operaciones sean más sencillas y ágiles.



Figura 1-2. COVAM. [6]

1.2 Objetivos

El objetivo general de este TFG (Trabajo de Fin de Grado) es el de estudiar, comparar, escoger y evaluar uno de los múltiples sistemas de indexado geoespacial que existen actualmente. Así como comprobar la mejora de análisis que puede ofrecer y como puede afectar al Conocimiento del Entorno Marítimo. Como se ha visto en la introducción, el COVAM requiere de una adaptación al avance tecnológico y este concepto es una pieza muy interesante con respecto al manejo de información geográfica y análisis de datos.

Para ello, en el estado del arte se presentarán las características generales del *Big Data*, las particularidades de los datos en el entorno marítimo y la relación existente con las actividades llevadas a cabo en el COVAM. Se analizarán los distintos sistemas de indexado que existen, las librerías informáticas que los utilizan, identificando sus características y cualidades, así como sus desventajas y ventajas que pueden ofrecer. Se estudiarán paralelamente los motores de gestión de bases de datos que ofrecen un análisis de datos espaciales y se realizará una comparativa que ayude a seleccionar la herramienta que más se adecue a los requisitos de este trabajo.

Por otro lado, como un segundo paso u objetivos secundarios, se pondrá a prueba la herramienta seleccionada mediante diferentes experimentos relacionados con la gestión de datos espaciales en un entorno marítimo. Se utilizarán datos AIS reales para que las pruebas sean lo más próximas posibles al proyecto real. A continuación, se analizarán los resultados y se profundizará en las necesidades particulares del COVAM enlazándolas con los experimentos realizados. Se evaluará cualquier error que se pueda producir en los análisis, así como las mejoras conseguidas. Por último, como apoyo al resto de objetivos, se evaluará la posibilidad de representar los índices generados visualmente.

1.3 Estructura de la memoria

Tras contextualizar el trabajo y habiendo establecido los objetivos que se desean cumplir, se describe la estructura de la memoria con el fin de establecer una organización más clara. El documento se dividirá en cinco capítulos, la bibliografía y anexos.

- Capítulo 1: Introducción y objetivos. En este primer capítulo se introduce el proyecto, se contextualiza el propio trabajo definiendo la importancia de las nuevas tecnologías en el CEM y se introduce por primera vez la figura del COVAM.
- Capítulo 2: Estado del arte. En el segundo capítulo se introducen los conceptos teóricos, así como una revisión del estado del arte sobre los propios sistemas de indexado geoespacial, detalladamente se trata lo siguiente:
 - En primer lugar, una introducción al *Big Data*, haciendo especial referencia al flujo de datos en el ámbito marítimo.
 - En segundo lugar, se describe el trabajo del COVAM, así como el interés de la Armada en el *Big Data*.
 - En tercer lugar, se define el concepto general de sistema de indexado geoespacial e información geográfica.
 - En cuarto lugar, se analizan los diferentes métodos de indexado espacial que utilizan las herramientas de análisis.
 - En quinto lugar, se describen las herramientas o sistemas de indexado existentes de interés para este proyecto.
 - En sexto lugar, se analiza la importancia de las *Spatial databases*, se introduce el sistema de gestión de bases de datos MySQL y las bases de datos espaciales que incluyen su propio sistema de indexado.
 - En séptimo y último lugar, se realiza una tabla comparativa con el objeto de elegir la herramienta a utilizar durante el desarrollo de este trabajo.

- Capítulo 3: Desarrollo del TFG. Tercer capítulo dónde se describe el desarrollo del trabajo. Se detalla el software necesario para realizar las pruebas y experimentos de análisis. Se introduce la gestión de bases de datos MySQL y la posibilidad de visualización de los resultados, para dar paso a una serie de experimentos para probar la eficiencia del sistema elegido, así como la comprobación y comparación de los resultados obtenidos.
- Capítulo 4: Resultados. Cuarto capítulo en el que se exponen brevemente los resultados obtenidos durante los experimentos.
- Capítulo 5: Conclusiones y líneas futuras. Conclusión del trabajo y presentación de posibles líneas futuras de investigación para continuar con el proyecto general.
- Finalmente, se adjuntan las referencias bibliográficas y sitios web consultados, así como los anexos que apoyan el desarrollo del trabajo.

2 ESTADO DEL ARTE

El siguiente capítulo tratará de estudiar cómo afectan las nuevas tecnologías al análisis de datos espaciales. Se explicarán conceptos clave para contextualizar la indexación geoespacial, como el *Big Data* y las grandes bases de datos, así como las características de la información geográfica y las necesidades de la Armada y concretamente del COVAM en este ámbito. Se relacionarán los distintos sistemas de indexado geoespacial y librerías como *Uber H3*, *Google S2* y otros con el objetivo, de seleccionar el que más se adapte a los requisitos.

2.1 Big Data

Antes de adentrarse en el concepto de los sistemas de indexado geoespacial es necesario conocer un poco más acerca del *Big Data*, es decir, de la cantidad de información que se maneja hoy en día y que es parte crucial para el desarrollo de este proyecto. Se explicará brevemente para poner un poco más en contexto y poder comprender el resto de temas tratados.

2.1.1 Definición

El término *Big Data* se refiere simple y llanamente a datos que son tan grandes, rápidos o complejos que aumentan la dificultad o hacen imposible procesarlos con métodos tradicionales. Sin embargo, acceder y almacenar grandes cantidades de información para la analítica ha existido desde hace mucho tiempo. Pero el concepto de *Big Data* cobró impulso a principios de la década de los 2000 cuando el analista, Doug Laney, articuló la definición actual de grandes datos como las tres V [7]:

- Volumen: las organizaciones recopilan datos de diversas fuentes, como transacciones comerciales, dispositivos inteligentes (IO), equipo industrial, vídeos, medios sociales, etc...
- Velocidad: los datos llegan a las organizaciones a una velocidad sin precedentes y deben ser manejados de manera oportuna. Las necesidades actuales requieren que estos datos sean manejados en tiempo casi real.
- Variedad: los datos se presentan en todo tipo de formatos, desde datos numéricos estructurados en bases de datos tradicionales hasta documentos de texto no estructurados, correos electrónicos, vídeos, audios, datos de teletipo y transacciones financieras. [7]

Actualmente, según [8] los analistas tienen que lidiar con conjuntos de datos de un tamaño que crece rápidamente y pueden ir desde los 30-50 Terabytes a varios Petabytes. La naturaleza de los datos es muy compleja, pero se debe a la carencia de estructura de los datos generados hoy en día en las nuevas tecnologías como las páginas webs, los sensores en diferentes dispositivos, la máquina actual inteligente, los vehículos, las búsquedas de internet o las redes sociales. Es cuando surge el problema de cómo tratar tanta cantidad de información y seguir siendo eficiente. Un ser humano no es capaz por sí solo de hacerlo. Pero gracias a las nuevas tecnologías se pueden combinar todos estos datos con datos estructurados,

como una base de datos relacional, que ayude a analizarlos. Se obtiene reducción de costes de almacenamiento, mayor velocidad para la toma de decisiones y nuevos productos y servicios. En la Tabla 2-1 se muestra una comparativa general de la estructura tradicional de datos a las necesidades actuales de los flujos de información.

Enfoque tradicional	Nuevo enfoque o paradigma
Gigabytes a Terabytes	Petabytes a Exabytes
Almacenamiento centralizado	Almacenamiento distribuido
Datos estructurados	Cualquier tipo de datos
Volumen limitado (agregación)	Cualquier volumen
Alta transformación de la fuente de datos	Cualquier fuente de datos
Análisis histórico	Análisis complejos (alertas, machine learning, realtime)

Tabla 2-1. Tabla comparativa de datos antes y después del Big Data. [9]

Algunos ejemplos prácticos de uso del *Big Data* pueden ser: [7]

- Comercio, empresas o viajes y hoteles: la analítica del *Big Data* da a estas empresas la posibilidad de recopilar datos de clientes, aplicar la analítica e identificar de inmediato problemas potenciales antes de que sea demasiado tarde. Es decir, hacen posible la anticipación a la hora de mejorar los servicios y, conocer los requisitos que buscan los clientes.
- Salud: los registros de pacientes, información de seguros o las citas con los médicos, pueden ser difíciles de gestionar. Ésa es la razón por la que la tecnología analítica del *big data* es tan importante para la atención a la salud. Mediante el análisis de grandes cantidades de información (estructurada o no) de manera rápida, los hospitales, centros de salud o farmacias pueden proporcionar diagnósticos u opciones de tratamiento que mejoran el rendimiento y en definitiva pueden incluso llegar a salvar vidas.
- Gobierno: todos los estados manejan mucha información en cada una de sus funciones de gestión. Un ejemplo puede ser la justicia y todos los informes judiciales y policiales, cuyo análisis a la hora de resolver crímenes o programar procesos y vistas mucho más rápido.

En la Figura 2-1 se muestra un ejemplo de la estructura del *Big Data* y su integración en las aplicaciones. Las fuentes de datos nutren a estas bases de datos, en las plataformas *Big Data*, que lo almacenan de diferentes formas para hacerlo accesible según las necesidades. Las diferentes aplicaciones como la generación de informes o análisis estadísticos históricos son los que utilizan estos datos.



Figura 2-1. Aplicaciones Big Data y fuentes. [10]

Para comprender bien como se mueve toda esta información se debe ver como se estructura la información con el fin de gestionarla correctamente según los fines que una organización o empresa tenga. Se puede decir que la información puede ser externa o interna y estructurada o sin estructura. Cada información posee unas características, pero hoy en día tratamos con todas al mismo tiempo, es donde el análisis del *Big Data* cobra importancia. Se extraen las siguientes clases de información:

- Interna:
 - Estructurada: transacciones, GPS, sensores. Se conoce como *Operational Data*
 - No estructurada: emails, contratos, informes, información privada, *Web logs*. Se conoce como "*Dark data*".
- Externa:
 - Estructurada: pagos móviles, *Bussiness Information*, investigaciones de mercado, en general datos públicos o con fines comerciales.
 - No estructurada: blogs, redes sociales, webs... [9]

2.1.2 Big Data en el Conocimiento del Entorno Marítimo

Tras haber introducido brevemente el *Big Data* y sus características fundamentales, así como algunas aplicaciones, en este apartado se relaciona este concepto con un tema principal de este trabajo, el Conocimiento del Entorno Marítimo. Aquí es donde es necesario profundizar, pues una de las necesidades de la Armada y que además es objetivo directo de este proyecto, es mejorar el rendimiento a la hora de analizar estos datos. Primero, se debe explicar cómo cambia este concepto y que información y datos almacena al tratarse del entorno marítimo.

Un buque, en el mundo actual, debe manejar muchos datos para cumplir sus funciones. Las nuevas tecnologías hacen que los buques utilicen muchos sensores que hacen mucho más fácil el desempeño de sus actividades en la mar, las comunicaciones y en general la navegación. Se puede diferenciar cuatro grupos de información que debe tratar un barco (Figura 2-2) [11] :

- Datos de navegación: AIS (*Automatic Identification System*), posición GPS, rumbo, velocidad (sobre tierra y sobre el agua), tiempo, rutas, profundidad o radar.
- Datos meteorológicos: viento, humedad, previsión de temporales, lluvia o el estado de la mar.
- Datos del resto de la flota: comunicación con resto de barcos y que información que poseen ellos.
- Datos propios del barco: sensores de seguridad, estado de las maquinas, almacenamiento o el combustible.

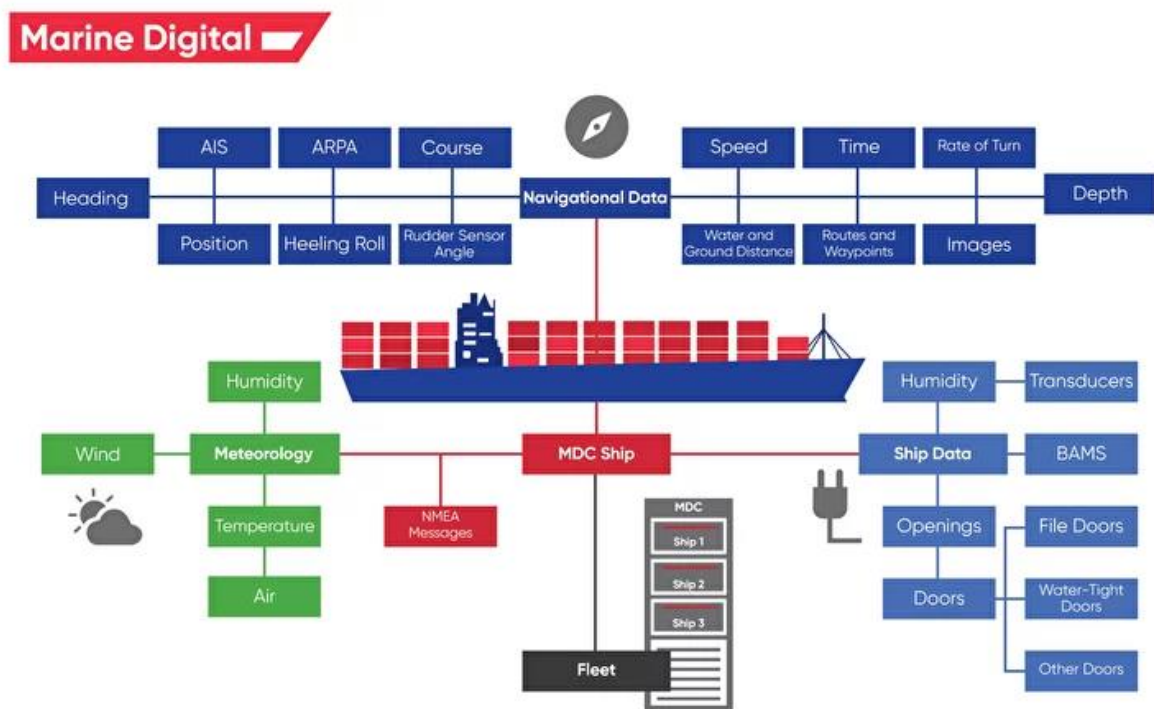


Figura 2-2. Esquema general de datos manejados por un buque. [11]

Si se habla de la marina mercante, mientras que el operador o capitán de un barco debe encargarse de apartados como la planificación de la navegación, donde abastecerse o consumir la mínima cantidad de combustible posible, el propietario se preocupa por la gestión del buque como la limpieza, planificación anual, seguridad o reparaciones. Gracias al análisis del Big Data se pueden facilitar estas tareas.

Como se nombró previamente, el Big Data tiene unas propiedades que lo hacen diferente al resto de datos que se ha manejado anteriormente. Su volumen, velocidad y variedad lo hacen complicado de analizar por sistemas tradicionales. A estas “tres V’s” hay que añadir dos más a la vez que completar las características que presentan en cuanto se refiere al Big Data Marítimo. El volumen viene definido por la cantidad de datos que se maneja como barcos, rutas marítimas, puertos, fondeaderos, dispositivos de separación de tráfico (DST), o eventos que ocurran en la mar. La velocidad a la que cambian estos datos viene determinada por el entorno cambiante que es la mar, posición de buques que cambia constantemente y que es necesario mantener en un tiempo real, rutas más rápidas, imposibilidad de transitar por una zona por razones meteorológicas, averías o incidencias en la mar que provocan que los barcos cambien su itinerario. La variedad es muy grande como se ha explicado previamente, los datos pueden ser de cualquier tipo (Figura 2-2). La veracidad, es una característica que se tiene en cuenta, depende de la precisión de los nuevos sensores y sistemas que disponen a bordo los barcos, esta precisión se puede ver afectada por razones meteorológicas, eléctricas o mala calibración, además de la información recopilada de otros barcos que puede poner en duda la nuestra propia. Por último, el valor, que viene definido por el tipo de dato y el uso que se le va a dar. Mientras que los datos meteorológicos

son la base de la predicción a la hora de tomar cualquier decisión, los datos históricos de tránsito del resto de barcos muestran información importante, pero ante una situación determinada pueden ser de menor relevancia, aunque indiquen una posible anomalía. [12]

2.1.3 Datos AIS

Uno de las principales fuentes de información y de datos que disponen los buques son los datos AIS. Como se ha resumido en 2.1.2, es una ayuda para conocer las características más importantes de buques en la mar. El AIS (*Automatic Identification System*) es un sistema que permite la comunicación e información a los buques sobre su posición, rumbo o destino entre otros datos. Es un sistema autónomo y automático de seguimiento que se interconecta entre todos los equipos AIS de los buques. Actualmente, la International Maritime Organization (IMO) requiere que todos los barcos de transporte de pasajeros y cualquier buque con un tonelaje mayor de 300 Tm lleven un transpondedor y receptor AIS con el objetivo de facilitar a las distintas organizaciones y estados el control del tráfico marítimo.

Datos transmitidos por el sistema AIS [13]:

- Información estática (cada 6 minutos):
 - a. *MMSI number*: Maritime Mobile Service Identify, es un número de 9 dígitos que identifica la estación transmisora, sea buque o servicio costero. Los tres primeros dígitos corresponden al país al que pertenece o dónde está registrado el buque y las seis restantes lo identifican. Para las estaciones costeras los dos primeros dígitos son 00, las tres siguientes indican el estado, y el resto la estación en concreto.
 - b. *IMO number*: International Maritime Organization number, es un número de 7 dígitos que identifica al buque, al propietario y a la naviera o empresa que lo emplea.
 - c. *Nombre y Distintivo de llamada (Name and Callsign)*
 - d. *Eslora y manga*
 - e. *Tipo de barcos*: define el tipo de barcos, se trata de un número de 2 dígitos desde el 10 al 99 (Tabla 2-2).

<i>Shiptype</i>	<i>Code</i>	<i>Shiptype</i>	<i>Code</i>
Reserved for future use	10-19	Pleasure craft	37
Wing in ground	20-29	Reserved	38,39
SAR	51	High-speed craft	40-49
Fishing vessel	30	Passenger ship	60-69
Tug	31,32,52	Cargo	70-79
Special craft	33-35, 50,53-59	Tanker	80-89
Sailing vessel	36	Other	90-99

Tabla 2-2. Tipo de barco según código. [13]

- f. *Lugar donde se sitúa la antena transmisora*
- Información dinámica:
 - a. Posición GPS e indicación de la precisión
 - b. Referencia horaria de la posición en UTC
 - c. *Course Over Ground (COG)*, rumbo del buque sobre el fondo marino.
 - d. *Speed Over Ground (SOG)*, velocidad del buque sobre el fondo, es decir, la velocidad real y absoluta que no depende del mar.
 - Información detallada del viaje (cada 6 minutos cuando sea requerido):
 - a. Calado del buque

- b. Tipo de cargamento
- c. Destino y ETA (*Estimated Time of Arrival*)
- d. Ruta con *waypoints*
- Mensaje corto de seguridad:
 - Si es necesario se podrá enviar un mensaje de estilo libre a una o varias direcciones por algún caso relacionado con la seguridad marítima.

Todos estos datos, multiplicados por todas las transmisiones que se realizan a diario por cada uno de los buques, crean un flujo de información enorme que cumple todas las características del *Big Data*.

2.2 El COVAM

Si se relaciona el *Big Data* y el Conocimiento del Entorno Marítimo se tiene una gran cantidad de información que cambia constantemente. La Armada, como entidad nacional encargada de proteger los intereses nacionales en la Mar, además de otras responsabilidades, se encuentra con la necesidad de explotar este flujo de datos. Continuando con la exposición del estado del arte del proyecto, se debe introducir al encargado de estas funciones, el COVAM. Explicar más a fondo su trabajo, ya introducido en 1.1, con objeto de enlazar ambos conceptos al completo al final de este apartado.

Se concreta en este apartado la motivación de este proyecto de actualización, para así dar un entendimiento superior sobre la situación actual del Conocimiento del Entorno Marítimo y la importancia de éste en las Operaciones Marítimas de Vigilancia y Seguridad. La situación de España en la geografía mundial hace de nuestro estado un punto de influencia marítima importantísimo. La trayectoria histórica también ha acompañado a este hecho y hace de España un país con una cultura marinera y naval muy antigua. Ya no solo por el carácter marítimo de la geografía española, de nuestras costas, y del buen clima del que se dispone que hace que España sea característica por sus playas y litoral, sino que España posee flotas pesqueras y mercantes muy importantes y gran parte de las mercancías que llegan a nuestro país lo hacen por la mar. Todo ello hace del mar un medio de vital importancia para la sociedad española, de ahí la importancia de su gestión y control.

Para controlar todo el tráfico marítimo, así como gestionar el entorno y las zonas de responsabilidad del país, España cuenta con organizaciones civiles y militares que ayudan a realizar la correcta gestión de los recursos para el control del entorno marítimo nacional. En concreto el COVAM se encuadra dentro de estas organizaciones, dentro de la propia Armada, y se encarga del mando, control y seguridad de las unidades desplegadas, así como de la gestión de toda la información que estas proporcionen sobre la situación de sus respectivas misiones en cuanto a seguridad y conocimiento del entorno se refiere. Se apoya de otras organizaciones nacionales como el Centro de Coordinación para la Vigilancia Marítima (CECORVIGMAR) de la Guardia Civil, la División Adjunta de Vigilancia Aduanera (DAVA) o el Centro Nacional de Coordinación de Salvamento (CNSC) de Salvamento Marítimo. Siendo de vital importancia la coordinación y el apoyo mutuo para lograr que todos los esfuerzos conjuntos sean lo más eficientes posible.

El COVAM o Centro de Operaciones y Vigilancia de Acción Marítima es el único centro a nivel nacional desde el cual su personal monitoriza 24/7 los 365 días del año, todo movimiento tanto militar como civil en el entorno marítimo nacional, no solo dentro de aguas jurisdiccionales españolas, sino también en aquellas con intereses nacionales como por ejemplo el Golfo de Guinea y el Océano Índico. Emplazado dentro del Cuartel General de la Fuerza de Acción Marítima en Cartagena, orgánicamente depende del ALMART y dentro de la Estructura Operativa del Estado Mayor de la Defensa (EMAD) depende directamente del Mando Operativo Marítimo (MOM) encuadrado dentro del Mando de Operaciones (MOPS). Por último, funcionalmente, se encuentra en la Sección de Operaciones de la FAM y operativamente realiza las funciones de apoyo y asistencia a MOM en todo lo referente a Operaciones de Vigilancia y Seguridad Marítima. [14]

2.2.1 Misión y funciones

El CMOM (Centro del Mando Operativo Marítimo) tiene como misión planear y ejecuta las operaciones de vigilancia y seguridad marítima que ayuden a contribuir a garantizar la seguridad de los espacios marítimos de interés. Esto se basa en los dos pilares fundamentales de la Seguridad Marítima: mantener un cierto nivel CEM para detectar riesgos y amenazas y llevar a cabo Operaciones de Seguridad Marítima (MSO).

Si especificamos, el COVAM realiza cuatro cometidos principales [14]:

- Establecer y mantener un permanente CEM en el Área de Operaciones Marítimas.
- Ejercer el Mando y Control de sus unidades a través de él.
- Colaborar con otras agencias marítimas nacionales en la proyección de la Acción del Estado en la Mar
- Funcionar como punto de contacto permanente entre la Comunidad Marítima y la Armada.

Existen varias tareas y procedimientos que les ayudan a cumplir esos cometidos u objetivos. Esto se base en un ciclo continuo para garantizar la obtención de una *Recognized Maritime Picture* (RMP) clara y concisa.

2.2.2 Gestión y análisis de datos en el COVAM

Para lograr los objetivos que se le presentan, el COVAM tiene que lidiar con una cantidad ingente de información cada día. Para ello, como se verá durante del desarrollo del arte de este TFG, se ayuda de la gestión de grandes bases datos donde se maneja y fluye toda la información respectiva a sus cometidos. Se recopila toda la información necesaria con el objetivo de mantenerla disponible de manera sencilla y facilitar la planificación, gestión y evaluación de los datos. Entre la gran cantidad de datos se pueden destacar los proporcionados por las propias unidades de la Armada, el seguimiento de todos los buques dentro de las zonas de interés con sus datos AIS y estadísticas como zonas de mayor tráfico o puertos más concurridos; todo ello en orden cronológico par así ser capaz de mantener un control a tiempo real. Además, estas bases de datos se nutren del resto de información que puede intercambiarse con otras organizaciones como la OTAN o la UE, así como también la propia Comunidad Marítima. En la Figura 2-3 se esquematiza y se resume el proceso de la Vigilancia Marítima.



Figura 2-3. Esquema Vigilancia Marítima en el COVAM. [14]

El objetivo es lograr mediante este proceso un RMP que permita a la Armada desempeñar sus cometidos de la manera más eficiente pero también mantener el control en las zonas de interés nacionales. De esta manera se consigue que cada estructura nacional pueda desempeñar su labor sin conflictos, como controlar el tráfico de drogas, la inmigración, el contrabando o el salvamento y rescate en la mar. En el caso de que se produzca cualquier incidencia en la mar, desde el COVAM se logra contactar con cualquier centro de rescate o de operaciones para poner en marcha las medidas pertinentes.



Figura 2-4. Zonas y eventos de interés en el COVAM. [14]

En Figura 2-4. Zonas y eventos de interés en el COVAM. se detalla esquemáticamente el entorno en el que el COVAM desarrolla su cometido abarcando desde el Atlántico pasando por el Mediterráneo y llegando hasta el Índico. También se indica ejemplos de eventos de interés o situaciones que el COVAM busca detectar para efectuar las respuestas pertinentes. Estos sucesos pueden pasar en cualquier momento, desde un buque realizando actividades sospechosas, detección de una patera, actividad GO-FAST de embarcaciones cruzando rápidamente el Estrecho de Gibraltar o incluso la alerta de presencia de embarcaciones sospechosas de ser piratas en el Golfo de Guinea.

Por tanto, debido al gigantesco ámbito geográfico marítimo y la cantidad de información sobre movimientos y posicionamiento de buques en la mar que puede llegar a recibir el COVAM, es necesario un análisis geoespacial. Gracias a los avances tecnológicos en telecomunicaciones, así como en la gestión de grandes bases de datos, es posible disponer de estos datos y localizarlos de forma casi en tiempo real. Es dónde los sistemas de indexado que hemos analizado entran a jugar y pueden facilitar el cometido y apoyar a la toma de decisiones mucho más rápidamente mediante su aplicación en los sistemas y flujos de información del COVAM.

2.2.3 Interés de la Armada en las nuevas tecnologías: Armada 4.0

Concluyendo, la Armada, como parte del Ministerio de Defensa y de las Fuerzas Armadas, es la encargada, como se ha visto, de asegurar los intereses de España en el ámbito marítimo. El concepto tratado del *Big Data* tiene un puesto clave en las nuevas tecnologías y en los objetivos tecnológicos del concepto de Armada 4.0

Este nuevo concepto nace de la necesidad de mantenerse en constante actualización, el *Internet of things* (IO), el *Machine Learning* (ML) y la *Inteligencia Artificial* (IA), están estrictamente relacionados con el Big Data ya que son parte de esta Industria 4.0 y son deberes que está potenciando la Armada. Una Armada cada vez más digital y tecnológica, con objeto de reducir el esfuerzo de los operadores y hacer mucho más fácil sus tareas.

Tal y como se expone en [15], el proyecto de I+D+i (Investigación, desarrollo e innovación) de dos años de duración, conocido como *Soprene* de la empresa Indra y adjudicado a la compañía por el Ministerio de Defensa a través de la Dirección General de Armamento y Material (DGAM), representa el paso más importante dado hasta la fecha para introducir el concepto de Industria 4.0 en las Fuerzas Armadas. Los desarrollos derivados de esta investigación serán pieza clave del Futuro Sistema Integrado de Sostenimiento en la Armada 4.0. Indra aplicará estas técnicas sobre los millones de datos que los equipos sensorizados de los buques registran mientras navegan para estudiar las ventajas que puede aportar su análisis. Aquí es donde entra el concepto de *Big Data*.

Se trata de una información almacenada en el Centro de Supervisión y Análisis de Datos Monitorizados de la Armada (CESADAR), que se encuentra en Cartagena. Estas técnicas imitarán la capacidad y estrategias que sigue el cerebro para razonar y tomar decisiones. Pero con la ventaja de que se añadirá a ellas el enorme potencial computacional que aportan los ordenadores más avanzados. De este modo se pueden detectar relaciones entre diferentes elementos que para un operador humano pasaría desapercibido. Es decir, establecer relaciones que hagan que el análisis de todos los datos sea más fácil, al igual que su manejo por parte del operador.

Enlazando con lo visto hasta ahora, el COVAM requiere de este avance tecnológico, gestión de datos AIS y detección de escenarios de interés para desempeñar sus funciones. La cuestión es ser capaz de interconectar todos los buques de la Armada, que a su vez producen y reciben datos. Información que se pone en común, se almacena y se analiza posteriormente para poder tomar decisiones que mejoren la operatividad.

2.3 El indexado geoespacial

Como se ha visto hasta ahora, la gestión de datos es una de las principales cuestiones de interés del COVAM. Sin embargo, las posiciones de buques en la mar es un flujo de información de datos espaciales. La capacidad de analizar estos datos añade una complejidad más a la cantidad de datos, y es la de gestionar datos en dos dimensiones en la superficie terrestre. Este es el concepto clave de este trabajo, la necesidad de la Armada de facilitar este procesado es una cuestión muy interesante. En este apartado se definirá las características de indexado geoespacial, los datos espaciales, así como la gestión de la información geográfica.

2.3.1 Definición

A grandes rasgos, la indexación espacial es una de las funcionalidades más importantes de las bases de datos espaciales. Según [16], los índices consiguen que las búsquedas espaciales en un gran número de datos sean eficientes. Sin indexación, la búsqueda se realizaría de manera secuencial teniendo que buscar en todos los registros de la base de datos, lo cual aumenta el tiempo y por tanto retrasa la búsqueda. La indexación organiza los datos en una estructura de árbol que es recorrida rápidamente en la búsqueda de un registro, según nuestras prioridades o según como jerarquicemos dicho árbol o esquema.

En primer lugar, un índice no es más que una estructura de datos, según la RAE (Real Academia Española) indexar significa “*registrar ordenadamente datos e informaciones, para elaborar su índice*”, estos datos se toman de campos particulares de una tabla y son almacenados de tal forma que el acceso es mucho más rápido. Todo ello con el objetivo de que al realizar cualquier consulta por el campo indexado se mejore el rendimiento y la velocidad en la respuesta. El empleo de estos sistemas es útil

cuando la base de datos crece significativamente, cómo se explica en [17]. La indexación tiene muchas utilidades y cada vez más con la cantidad de datos que se manejan hoy en día. Por ejemplo, cuando hablamos de indexar en Internet o en un buscador como Google, lo que se busca es que los resultados que se obtengan tengan la mayor relación posible con los términos de la búsqueda en particular. Esta relación, se produce internamente en los buscadores, los cuales han asignado previamente un determinado contenido de las páginas Web a diferentes términos o palabras, a la vez que se jerarquiza (indexado), dando prioridad a unos resultados sobre otros. Esta jerarquización es la que diferencia los diferentes algoritmos de cada uno de los motores de búsqueda y, por lo tanto, dando unos resultados más acertados que otros. [18]

La diferencia entre un índice tradicional y un índice espacial es la capacidad de indexar posiciones espaciales o geográficas, que tienen dos dimensiones. La idea de estos sistemas es crear estructuras que asignen a estas posiciones o datos bidimensionales, una única dimensión, y así facilitar la utilización de la información. El indexado geoespacial no es más que una de las aplicaciones del indexado espacial, el manejo de todos los datos que espaciales es prioridad para todas las organizaciones que requieran de una gestión espacial. Lo que se busca es crear formas 2D que dividan la superficie terrestre, es decir, indexar la superficie terrestre en celdas o polígonos. Estos polígonos engloban cada uno de ellos unas determinadas coordenadas [19]. Las coordenadas pueden ir desde los 90° Sur a los 90° Norte en la latitud y desde los 180° Este a los 180° Oeste, cubriendo la superficie completamente. La resolución de estas celdas o polígonos será la que determine la cantidad de pares de coordenadas que se encuentran en su interior, siendo más o menos precisa cuanto más o menos resolución se tenga, es decir más o menos celdas. Existen múltiples sistemas que basan este indexado en diferentes estructuras que se estudiarán a lo largo de este capítulo.

2.3.2 Historia

Los primeros sistemas de información donde “almacenar” datos comenzaron con los libros y los mapas. Seguidamente, al aumentar la cantidad de información a almacenar y a ordenar, aparecieron las bibliotecas, librerías y cartografía, dividiéndolos por separado. A partir de 1990, se comienza a usar otro modelo añadiendo un geolocalizador, es decir, se le añaden unos datos al objeto como el nombre, el tipo o la posición en la Tierra. Se trata de bibliotecas digitales completados con sistemas de información geográfica o referencia espacial. A partir del 2000, los indicadores geoespaciales pueden buscar espacialmente y semánticamente, pudiendo buscar un valor alfanumérico cualquiera que interese. [20]

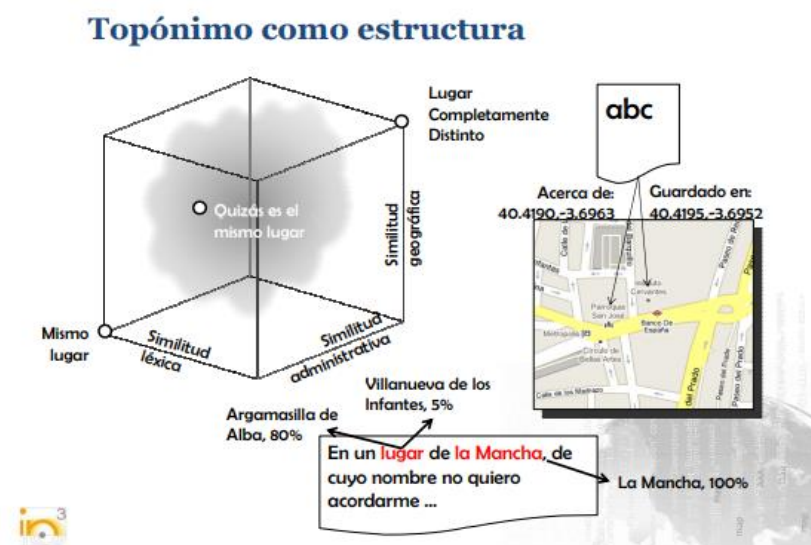


Figura 2-5. Geolocalizador. [20]

En Figura 2-5 se esquematiza la estructura de un objeto geoespacial, se puede decir que abarca tres dimensiones. La léxica, es decir la información que puede almacenar el objeto (en el caso de un buque, velocidad, destino, etc.), la administrativa (el tipo objeto que es, una posición o una superficie completa, un polígono), y la más importante, la geográfica, que indica la posición en la superficie terrestre. Este concepto sirve de cómo de complejo puede ser gestionar estos datos, lo importante es centrarse como localizar estos datos a través de sus posiciones espaciales. Esto provoca que pueda haber resultados que sean los buscados, parecidos o completamente distintos, en cada una de las dimensiones. Por ejemplo, “*un lugar de la Mancha*” solo indica todo lo que abarca la similitud geográfica, mientras que Argomasillas de Alba tiene un 80% de posibilidades de ser el lugar concreto, Villanueva de los Infantes solo un 5%.

2.3.3 La información geográfica

Como se ha explicado, desde los inicios del ser humano se han utilizado mapas y cartas para representar la superficie terrestre. Ya sea para desplazarse, organizar territorios, o tomar cualquier decisión. Sin embargo, a lo largo de las últimas décadas, debido al avance tecnológico, estos sistemas quedan cada vez más arcaicos. Actualmente, como se sabe, existen múltiples modelos de cartografía digital y navegadores que nos ayudan en nuestra vida cotidiana o profesional, como es en el caso de los buques y las ECDIS (*Electronic Charts Display and Information Systems*). Estos sistemas han revolucionado la cartografía, la navegación, la planificación y el análisis espacial geográfico.

Como se dice en [21], la información geográfica tiene varias características, dependiendo la información que interese será más importante una u otra a la hora de buscar. Sus cualidades son:

- Multidimensional: se sabe que mediante dos coordenadas se puede obtener cualquier posición en la Tierra (x e y, o latitud y longitud)
- Dependiendo de la resolución geográfica, esta puede ser muy detallada, muy genérica muy voluminosa, etc.
- Puede ser representada en varios formatos digitales, que influye en análisis y resultados
- Puede ser proyectada en una superficie plana pero también es posible que lo sea en otras superficies, al igual que las cartas y las distintas proyecciones.
- El tiempo de análisis de la información geográfica es muy grande si se carece de una indexación.

Sobre esto último, se ejemplifica en [21], la localización espacial de los atentados del 11S en New York tras el primer ataque fue un elemento crucial para dar la alerta y comenzar la evacuación lo más rápido posible.

Por otro lado, cuando se ha nombrado la multidimensionalidad de la información geográfica se está mencionando a la localización exacta de los objetos de interés. Aun así, cada objeto posee unas propiedades determinadas, por lo que el tamaño varía según lo que se esté estudiando, como la longitud de una carretera, las dimensiones de un bosque o una masa de agua como un lago o un río, no son comparables a las dimensiones que puede presentar un barco, un coche o una persona. La representación de estos objetos se hace más fácil si se usa una resolución mayor o menos según convenga. Como ejemplo claro el modelo vectorial define tres formas que ayudan a aportar esa resolución. Mientras que un punto se localiza exactamente en el espacio en dos coordenadas X e Y, es más fácil y rápido encontrar el punto cuando pertenece a un conjunto de puntos, como una línea o un polígono. El modelo vectorial (Figura 2-6) es la solución para poder abarcar estos datos.

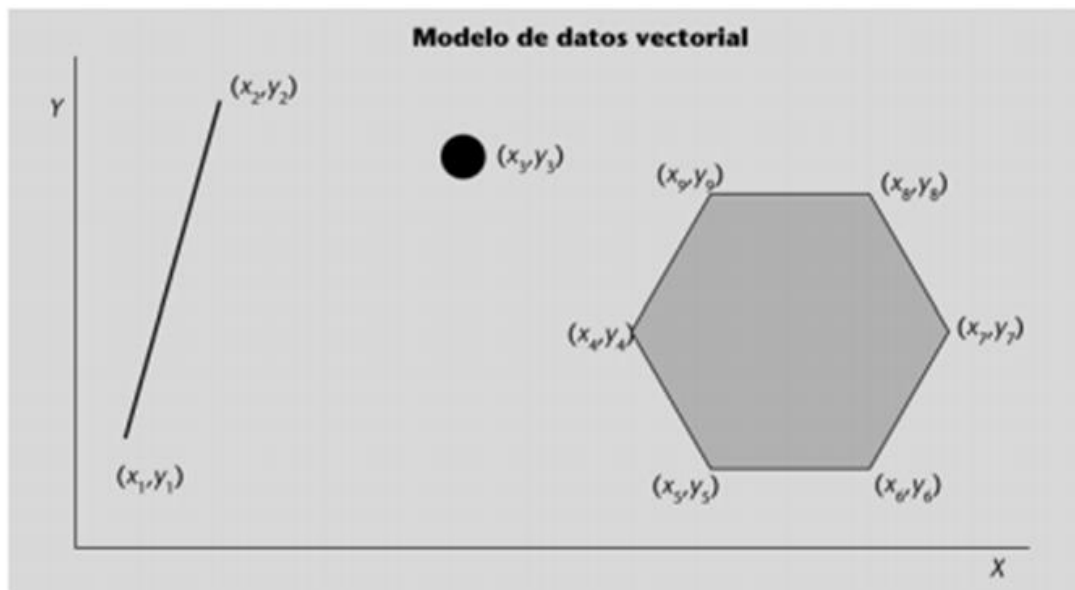


Figura 2-6. Modelo de datos vectorial. [21]

En resumen, en los últimos años muchas organizaciones ven necesario recopilar información geográfica, ubicarla y gestionarla. Se ve necesario la introducción de relaciones espaciales que creen similitudes semánticas entre conceptos geoespaciales o geográficos que resuelvan una consulta que combine ambos aspectos (textuales y espaciales). El objetivo es que esta información pueda ser consultada fácilmente por el usuario y pueda ayudar al propósito que se tenga. [22]

2.4 Métodos de indexación espacial

Tras definir las características de los datos espaciales en el apartado anterior se van a estudiar a continuación los diferentes tipos o métodos de indexación espacial que existen o que actualmente son más utilizados. Para ello, en primer lugar, se presentan las funciones de un método de indexado y el análisis de datos para posteriormente definir los métodos.

2.4.1 Análisis de datos espaciales

El indexado espacial no es más que uno de los pilares fundamentales de los sistemas de bases de datos espaciales. Se encarga, como se ha explicado, en jerarquizar y facilitar la gestión de todos los datos almacenados y organizar las relaciones que se presentan entre ellos. Lo que hace es optimizar las consultas espaciales o *spatial queries*. Según Lance McKee en 1997, el *Open Geospatial Consortium* (OGC) [23] estableció las especificaciones y estándares que diferencian las bases de datos espaciales y sus características de las relacionales, desarrollando las *OpenGIS Simple Features Specification*, determinando las características que permiten a distintos sistemas a comunicarse e intercambiar información basada en geometrías 2D. Esto incluye las líneas, puntos, curvas y polígonos que hemos visto antes.

Mediante las consultas, los sistemas de indexado acceden a la información de base de datos para hacer más rápido la gestión de esta información. Como ejemplo se pueden establecer las típicas consultas a bases de datos SQL (*Structured Query Language*), a las que añadimos la capacidad de las bases de datos espaciales de realizar otro tipo de consultas. Estas capacidades vienen determinadas por el OGC y son las siguientes [24]:

- *Spatial Measurements*: longitud de líneas, áreas, distancia entre geometrías.
- *Spatial Functions*: permite la modificación, creación o eliminado de cualquiera de las características de la base de datos.

- *Spatial Predicates*: capacidad de determinar si existe o no relaciones entre varias geometrías, como si dos polígonos se superponen o dos líneas se cortan en algún punto.
- *Geometry Constructors*: permite la creación de nuevas geometrías, asignando nuevos vértices o puntos dentro de estas.
- *Observer Functions*: consultas que únicamente observan los datos sin interactuar, como preguntar el centro de un círculo o el radio de éste.

Para ello las bases de datos se deben apoyar, como se ha nombrado, en los sistemas de indexado. Los sistemas de indexado espaciales son capaces de manejar estas consultas y establecer un análisis jerárquico y conciso. La mayor parte de las empresas y organizaciones que requieren de estos sistemas tienen su propia base de datos espacial a la que aplican uno de los sistemas o crean uno propio.

Existen distintos tipos de sistemas de indexado diferenciándose en varias características como el criterio de división del espacio, las figuras geométricas que utilicen o los procesos con los que determinan la solución las consultas que se realicen. La división del espacio y el método que se utilice para ello es la principal característica que diferencia a unos de otros, así como la precisión que pueden llegar a tener o las estructuras que pueden formar.

2.4.2 Grid spatial index

Este método es de lo más utilizados por las diferentes bases de datos espaciales. Dividen el espacio en superficies 2D, llamadas celdas, que son asignadas a un identificador que puede ser alfanumérico, decimal, binario, etc. Estas figuras geométricas pueden ser cuadradas, rectangulares o incluso hexagonales. Mediante este método aceleramos la búsqueda a una posición que se encuentre en una celda en concreto. Estos sistemas asocian las coordenadas correspondientes a una posición a la celda cuya área contenga ese lugar en concreto, estas coordenadas pueden ser cartesianas como la latitud y la longitud. Una de las propiedades es que estos métodos pueden ser creados antes de que se les asigne la información a indexar, creando todo el reticulado o divisiones previamente a las consultas [25].

Las ventajas de estos métodos son:

- Simples, fácil uso.
- Se puede crear una tabla que muestre el índice creado fácilmente.
- Si el sistema tiene mucha resolución, es decir, muchas celdas, no se tiene el problema de sobrecarga en una de ellas (Figura 2-7).

Las desventajas que pueden tener son:

- Puede que la escala utilizada en algunas regiones no sea la adecuada.
- Dependiendo de la resolución se pueden obtener resultados un poco ambiguos, si las celdas están muy sobrecargadas no se sabe con exactitud la posición de los lugar u objetos.

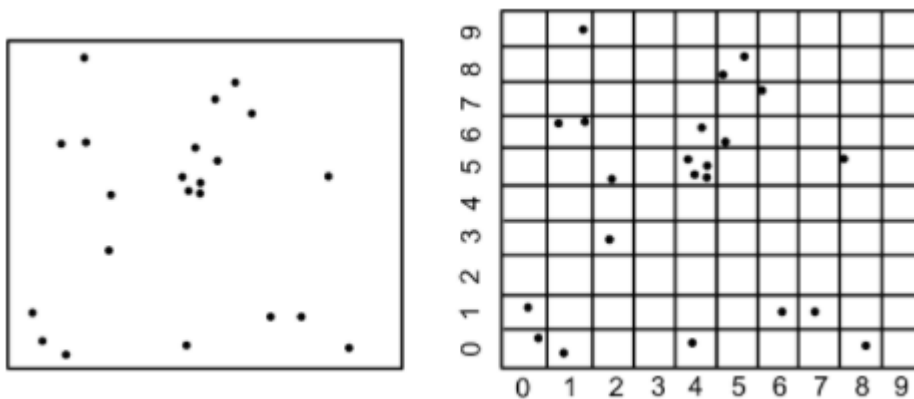


Figura 2-7. Grid Indexing. [25]

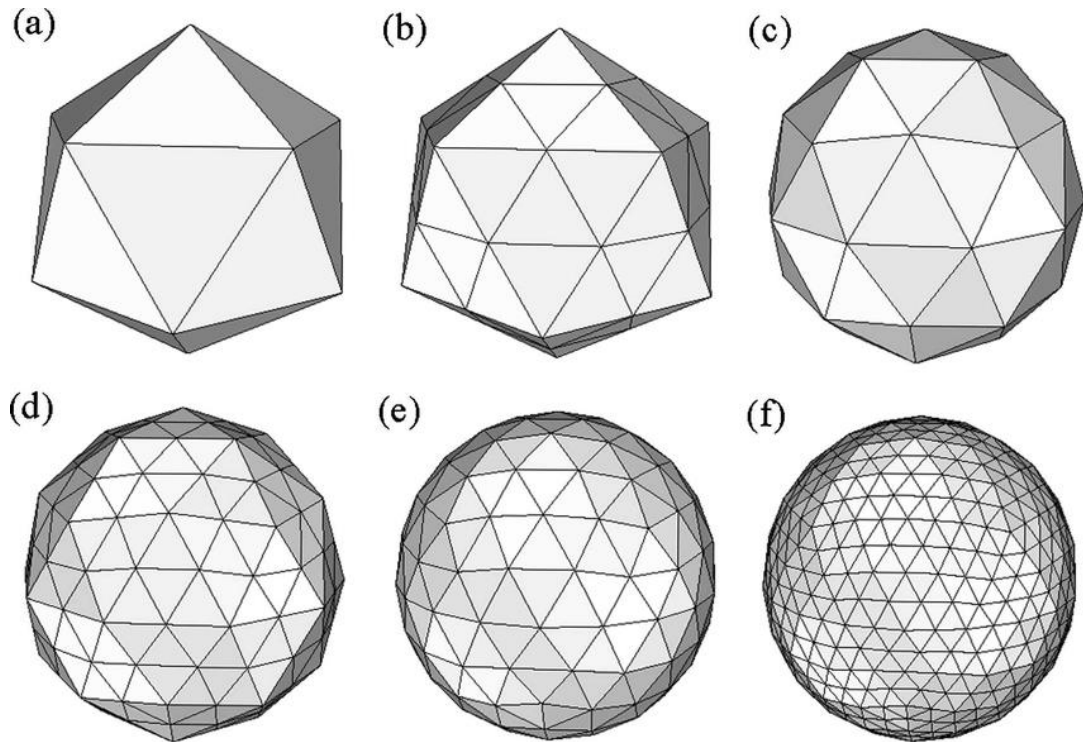


Figura 2-8. Grid triangular generado por bisección recursiva. [26]

En Figura 2-8 se muestra la generación de una rejilla triangular, mediante este método se pueden ir almacenando posiciones o datos espaciales en los diferentes triángulos con mayor o menor resolución. Este método es de los más empleados por las organizaciones que utilizan datos GPS.

2.4.3 Z-order curve

En un sistema cuadrículado las celdas vecinas no son siempre adyacentes, ya que una de estas puede tener hasta ocho vecinos, provocando que la posibilidad de error pueda ser mayor. Para ello estos sistemas intentan mediante una curva recorrer las celdas y seguir un tipo de orden que haga que la búsqueda sea más rápida. Este sistema busca que una celda con esquema bidimensional se asigne a un esquema de direccionamiento unidimensional, dejando las celdas vecinas cerca entre ellas. [25]

Para ello este sistema crea un ordenamiento representando las direcciones de las celdas con bits. La curva recorre las celdas de forma intercalada siguiendo un orden lineal ascendente en los bits uno a uno, de esta manera realiza su recorrido creando varias *zetas* [25]. Un sistema que utiliza estas curvas es el llamado *UB-tree* que genera los enlaces a cada uno de los datos asignándoles el valor según la curva, tal y como se muestra en Figura 2-9.

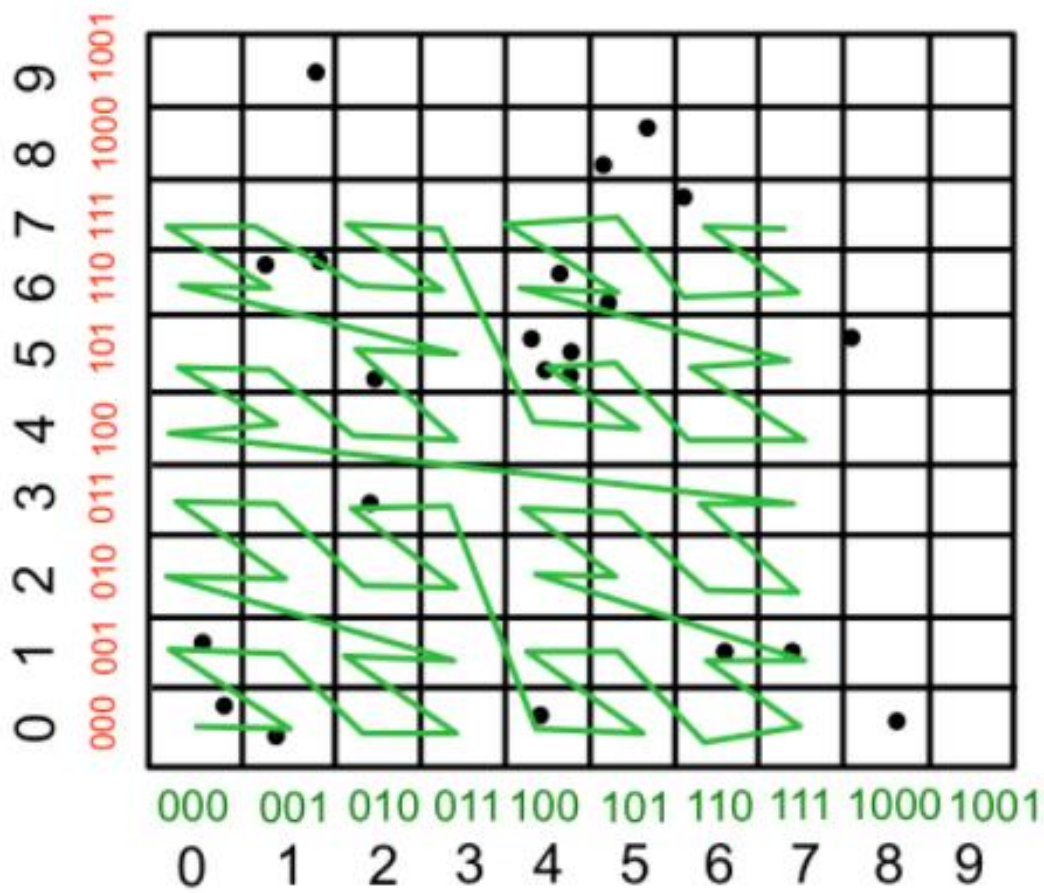


Figura 2-9. Z-ordering. [25]

2.4.4 HHCCode

Helical Hyperspatil Code, más conocido como HHCCode, es un sistema de indexado para bases de datos espacio-temporales muy grandes. Fue inventado en los años 90 por Herman Varma para el almacenamiento de información batimétrica en el *Canadian Hydrographic Service*. Este código está basado en la curva Z-order. Actualmente este sistema es el empleado en las bases de datos Oracle versión 4. Además, múltiples organizaciones como CuberWerx o Helical Systems Inc lo utilizan en sus bases de datos. [27]

2.4.5 Quadtree

Quadtree es otra de las alternativas que existen a la hora de analizar los datos jerárquicamente. Divide sucesivamente la imagen o espacio en cuadrantes y subcuadrantes de las mismas dimensiones hasta conseguir bloques de la resolución necesaria. Mediante nomenclatura binaria se nombran las celdas y se obtiene la región objetivo. Los nodos en el árbol representan las regiones que se forman del nodo padre. El orden de los nodos hijos corresponde a su posición con respecto al centro de la región inicial (NW, NE, SW, SE). Como se puede ver en Figura 2-10 se va dividiendo las zonas a estudiar en los cuadrantes necesarios sucesivamente, hasta dar con la solución, resultando en cada nodo en su división 4 nodos hijo, es decir, 4 regiones. [28]

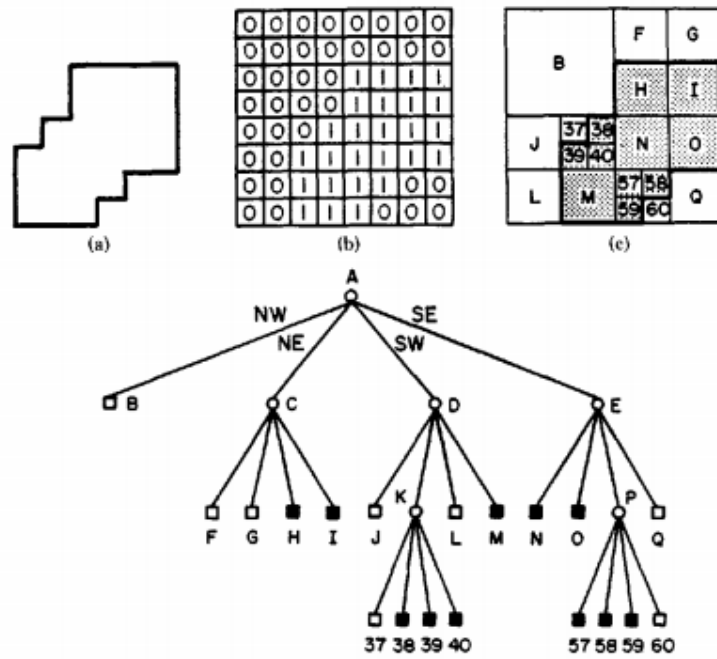


Figura 2-10. Ejemplo árbol *quadtree*. [28]

Antes de realizar el análisis es necesario determinar el número máximo de elementos que puede contener una de las celdas creadas. Para ello se fija un número m determinado de elementos (ejemplo en Figura 2-11), si se supera este número es necesario realizar una división en cuatro nuevos cuadrantes, creando a su vez por cada cuadrante nuevo un nuevo *leaf node* o nodo hijo.

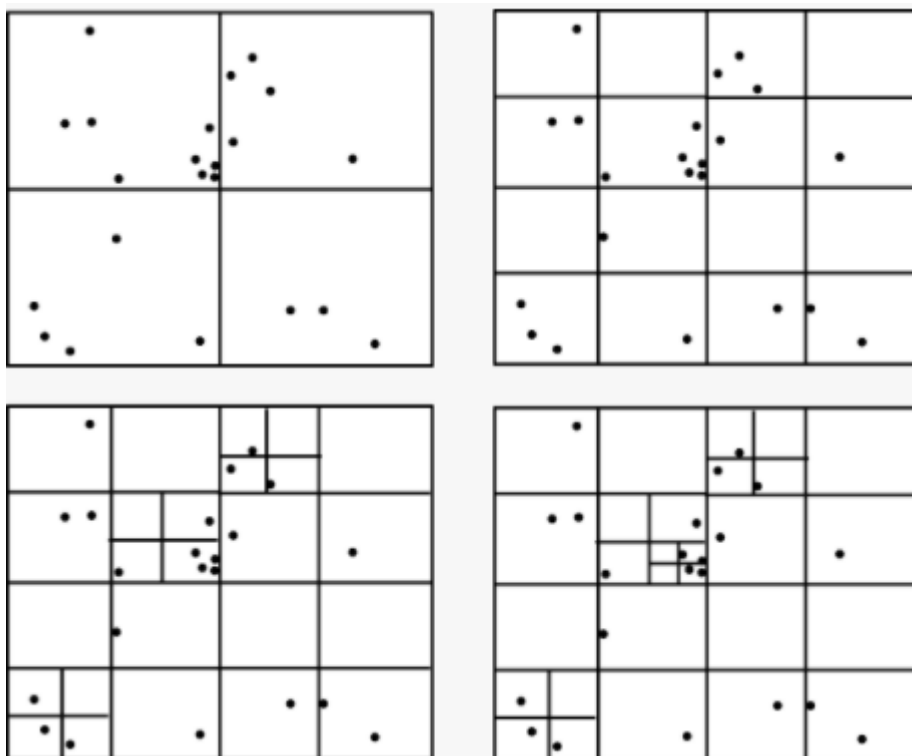


Figura 2-11. Ejemplo elementos máximos m en las celdas ($m= 2$). [25]

Este método presenta algunas ventajas como la adaptación de la descomposición espacial a la cantidad de datos, la capacidad de cada nodo de dividirse, y la facilidad a la hora de visualizar el esquema de direccionamiento de cada uno de los cuadrantes mediante el árbol. Sin embargo, cada vez que se presentan nuevos datos, es más complejo ya que se tiene que actualizar todo el sistema dependiendo de la nueva densidad de datos. Por esta característica quizás puede ser más lento que otros.

2.4.6 Octree

Muy parecido al anterior pero trasladado al espacio tridimensional, donde cada nodo tiene ocho hijos en esta ocasión, obviamente debido a la nueva dimensión. Estos sistemas son muy empleados, más que para el análisis geoespacial, en el uso de gráficos 3D.

2.4.7 R-tree

R-Tree, “R” de *rectangle*, es un tipo de estructuración de datos tridimensionales que sirve para crear sistemas de indexado. Este método es de los más utilizados y empleados a la hora de crear bases de datos espaciales. Básicamente asocia datos que se encuentran próximos en un árbol de nodos padre y *leaf nodes*, donde realmente se almacenan los objetos o datos. Tienen límites como cantidades mínimas y máximas en los nodos. El problema que tienen estos sistemas es la dificultad de asegurar que los nodos internos no se superpongan demasiado o no contengan espacio vacío. [25]

La estructura de R-Tree es muy similar a la de *B-Tree*, éste último sirve también para indexar información y datos, pero no multidimensional. Este tipo de árboles (B-Tree) almacenan datos puntuales. Los R-Tree pueden almacenar unas coordenadas geoespaciales (x, y) y estructurarlas jerárquicamente en forma de conjuntos, implementar mapas virtuales, e indexar toda esa información. Para ello utiliza formas geométricas rectangulares, mediante un sistema de nodos padre e hijo [29]. Fue propuesto y promovido por Antonin Guttman en 1984 con “*R-Trees: A Dynamic Index Structure for Spatial Searching*” [30], con el objetivo de ser capaz de almacenar datos espaciales como identificaciones de radiofrecuencia, control de tráfico y en general análisis de información geográfica.

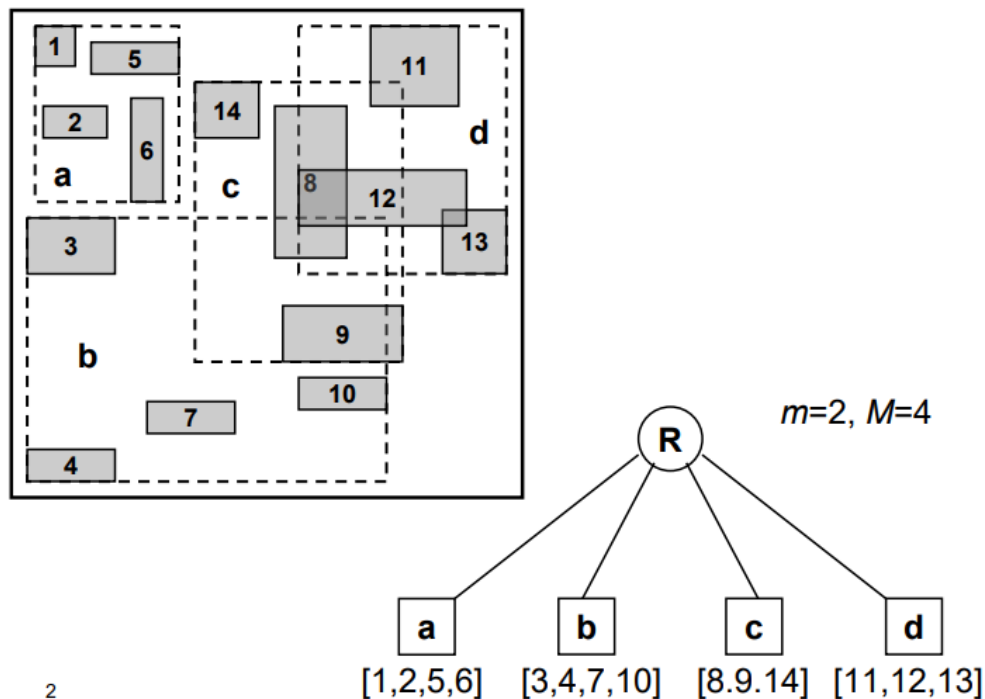


Figura 2-12. Ejemplo R-Tree espacial y esquemáticamente. [29]

Genéricamente un R-tree es un árbol de nodos. Existen dos tipos de nodos: *leaf nodes* y *non-leaf nodes*. Un *leaf node* corresponde a un *array* o matriz de *leaf entries* mientras que un *non-leaf node* agrupa *nodes entry*. Los *leaf nodes* se encuentran al mismo nivel en el árbol (a, b, c, d) (ejemplo en Figura 2-12) y contienen el resto de entradas de la base de datos, definiendo los rectángulos que poseen cada uno de ellos. Sin embargo, puede haber nodos que se encuentren en varios *leaf nodes* o solapes entre diferentes *leaf nodes*. Un *non-leaf node* (1, 2, 5...) almacena dos datos: la forma de identificar al nodo hijo, es decir cada una de las entradas, y los límites que abarcan el nodo inferior. Cada nodo puede tener como máximo M entradas, el *root* R como mínimo m entradas ($m = M/2$). Los nodos entrada se sitúan en cada *leaf node* siempre y cuando no contenga más de M entradas y la mayor parte de su área se encuentre dentro de ese *leaf node* (ejemplo en la imagen superior dónde 9 está en c y no en b). En el caso de que algún nodo más cumpla estos requisitos escoger siempre el que menor área tenga. Si se tiene otra entrada nueva y que no tenga espacio en ninguno de los nodos anteriores se crea un nuevo *leaf node* que agrupe los nodos necesarios para aliviar la sobrecarga y se crea un nivel superior [30].

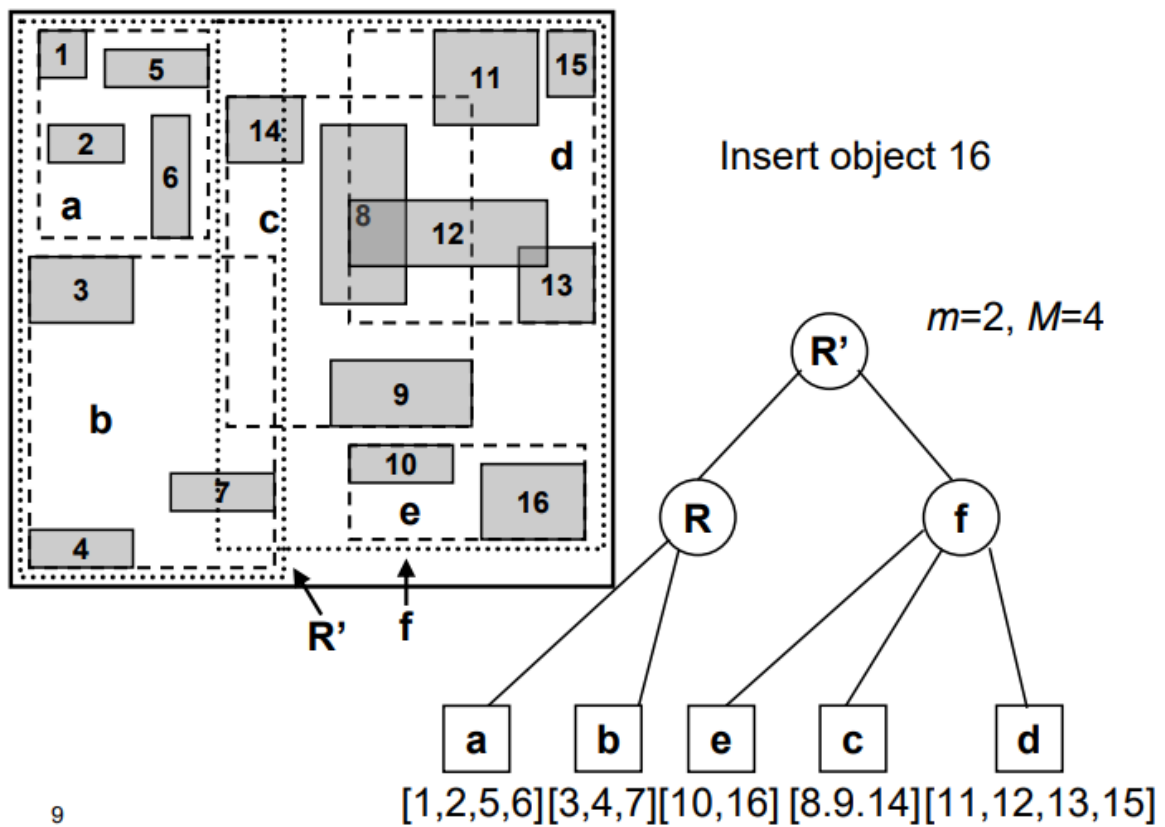


Figura 2-13. Ejemplo de sobrecarga en R-tree. [29]

De este modo M , tal y como se ve en el ejemplo de Figura 2-13, marca el límite de cada *leaf node* y ayuda al algoritmo de búsqueda a la hora de hacer sus consultas y encontrar los datos que se requieran, ahorrando tiempo de búsqueda al no hacerla en todos los nodos. Al insertar un nuevo dato, se genera un nuevo árbol cumpliendo las características de tamaño m . Definiendo mejor el método de búsqueda ésta empieza desde el nodo raíz R , cada *leaf node* contiene los nodos o rectángulos y por tanto el camino al nodo hijo correspondiente. Como ya se ha comentado, la búsqueda irá priorizando dentro de los nodos cuál es el camino que más área contiene sobre la localización o punto específico que se quiere buscar. Sin embargo, según el propio Guttman [30], no es posible garantizar un buen rendimiento en el peor de los casos en el caso de que la información se encuentre solapada y no sea tan preciso o rápido. Aun así, afirma que para la mayoría de aplicaciones obtiene unos buenos resultados y elimina la búsqueda en regiones irrelevantes.

2.5 Análisis de los sistemas de indexado geoespacial

Para poder cumplir con el objetivo de este proyecto se pasará previamente por analizar cuidadosamente las cualidades de cada uno de los sistemas que manejan las bases de datos geoespaciales a la hora de indexar y localizar los objetos o espacios geográficos que sean de interés. Estos sistemas pueden ser desde librerías informáticas hasta simples aplicaciones que utilizan los métodos mencionados en el apartado anterior para indexar los datos. Se analizan las cualidades que pueden presentar, como la resolución, la facilidad de implementar el código o algoritmo, los errores, etc.

2.5.1 Uber H3

H3 es el sistema de indexado espacial jerárquico que utiliza la empresa Uber con objetivo de optimizar su modelo de negocio. Como se sabe, Uber es una empresa de VTC (vehículos de transporte con conductor) que funciona mediante aplicación móvil. Uber es capaz de encontrar, cuando nos encontramos en una de las ciudades que opera, el mejor precio y al conductor más cercano en cuestión de segundos. Para ello Uber necesita de la gestión de una base de datos geoespacial, donde se almacenan todos los datos de las ciudades como carreteras, edificios, gestión del tráfico, clientes y conductores. Es un claro ejemplo de organización que se vale del *Big Data* y de la indexación para optimizar las soluciones a problemas como qué precio dar a cada cliente o como que conductor está más cerca o va a llegar más rápido, todo ello en tiempo real. También ha sido capaz de crear una filial llamada Uber Eats que, como su propio nombre indica, ofrece transporte de comida a domicilio. Esa filial también se vale del mismo sistema de gestión de datos que los VTC, dando a cada uno de los transportistas la información necesaria de los envíos más cercanos, proponiendo el que menos distancia le requiera entre recoger la comida y entregarla a su destino dependiendo de donde se encuentren.

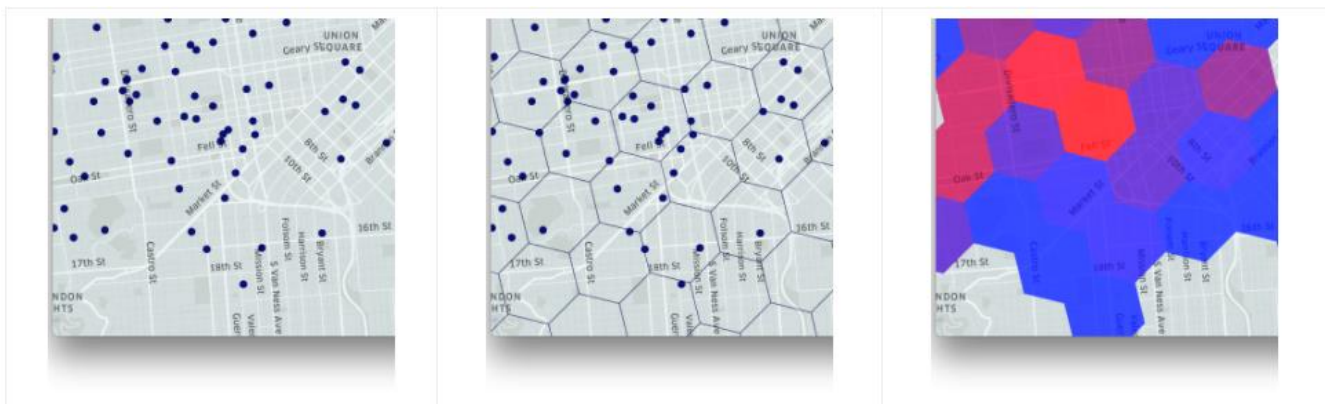


Figura 2-14. Ejemplo de mallado de Celdas H3. [31]

Como las ciudades son un entorno que también cambia a gran velocidad, son geográficamente diversas requieren de un análisis cuidadoso. Sin embargo, como se ha visto, el análisis de cada uno de los casos es muy lento, mientras que dividiendo en regiones es mucho más práctico. Para ello H3 divide la superficie en áreas o células hexagonales (ejemplo en Figura 2-14), desde esa forma geométrica basa el análisis de todos los datos. Según Uber es la forma que tienen de obtener menor error en un entorno como las grandes ciudades. Se puede pensar que los códigos postales o los barrios sería una manera más eficiente, de hecho, así es como se reparten el territorio los taxis tradicionales o los autobuses de una ciudad. Sin embargo, el éxito de Uber se debe a la velocidad a la que a un cliente le puede llegar su transporte, y es porque este tipo de áreas tienen una forma especial que no ayuda en su análisis. Por ello es preferible, según el conocimiento que dispongan del entorno urbano, asignar esas células hexagonales de formar jerárquica. [31]

H3 combina indexación geométrica y jerárquica, creando una malla global de hexágonos, lo que requiere la proyección en un mapa para transformar una localización en 3D en la Tierra a las 2D de un punto en la Tierra. Este proceso puede realizarse de muchas maneras, como con una proyección Mercator y un sistema de celdas cuadradas, pero no es lo más eficiente para los intereses de Uber, ya que un cuadrado puede llegar a tener hasta 8 cuadrados que lo rodeen (lados y vértices), haciendo que los resultados puedan variar mucho a la hora del análisis. [31]

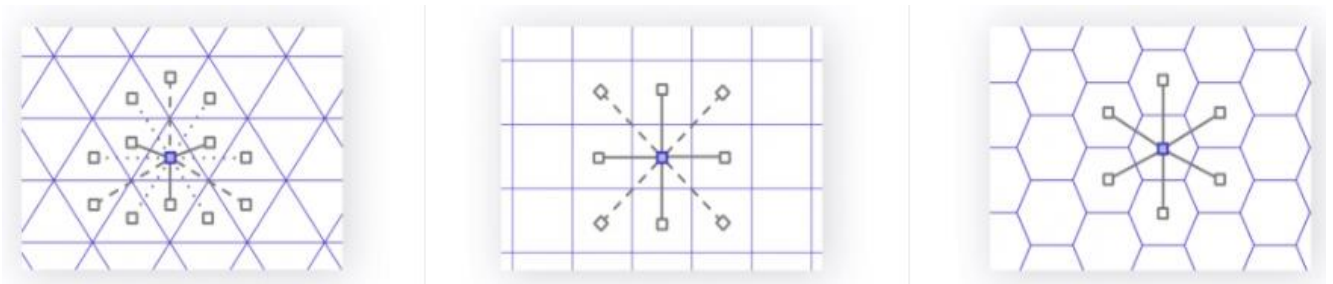


Figura 2-15. Distancia entre centros de celdas según su geometría. [31]

Como se ve en Figura 2-15, la distancia entre los centros de las células hexagonales vecinas es menor y es siempre la misma, a diferencia que con celdas cuadradas o triangulares. Por otro lado, como proyección en la superficie terrestre H3 utiliza la proyección gnómica transformando la esfera terrestre en un icosaedro (poliedro de 20 caras triangulares, Figura 2-16). Como para Uber la superficie ocupada por los océanos es de poco interés es donde se sitúan los vértices de las caras del icosaedro, en esos vértices se coloca un pentágono en cada uno para completar todo el mallado. Además, H3 posee la capacidad de cambiar su resolución (Figura 2-17), es decir el tamaño de celda, hasta en 16 niveles. Cada célula tiene el área equivalente a un séptimo del área de la célula padre.

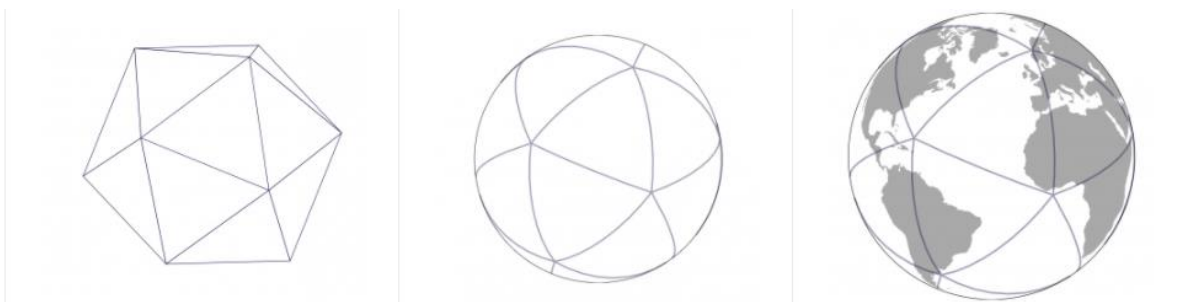


Figura 2-16. Icosaedro terrestre. [31]

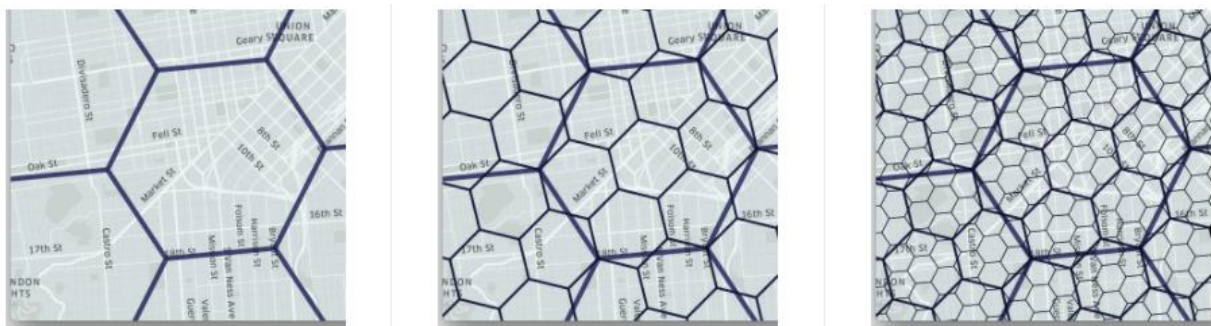


Figura 2-17. Ejemplo de subdivisiones del mallado de H3. [31]

Este sistema de indexado es de código abierto y público, y está disponible en GitHub, escrito en C y disponible en otros idiomas informáticos. La principal función de H3 es convertir pares de latitud y longitud en un índice de H3 de 64 bits, que dependerá de la resolución que se haya especificado. Su naturaleza jerárquica también puede variar dependiendo de la zona que interese analizar. [31]

En Tabla 2-3 se describen los diferentes niveles de resolución de la librería, así como las características principales de las celdas.

<i>H3 Resolution</i>	<i>Average Hexagon Area (km2)</i>	<i>Average Hexagon Edge Length (km)</i>	<i>Number of unique indexes</i>
0	4,250,546.8477000	1,107.712591000	122
1	607,220.9782429	418.676005500	842
2	86,745.8540347	158.244655800	5,882
3	12,392.2648621	59.810857940	41,162
4	1,770.3235517	22.606379400	288,122
5	252.9033645	8.544408276	2,016,842
6	36.1290521	3.229482772	14,117,882
7	5.1612932	1.220629759	98,825,162
8	0.7373276	0.461354684	691,776,122
9	0.1053325	0.174375668	4,842,432,842
10	0.0150475	0.065907807	33,897,029,882
11	0.0021496	0.024910561	237,279,209,162
12	0.0003071	0.009415526	1,660,954,464,122
13	0.0000439	0.003559893	11,626,681,248,842
14	0.0000063	0.001348575	81,386,768,741,882
15	0.0000009	0.000509713	569,707,381,193,162

Tabla 2-3. Niveles de resolución H3. [32]

Se puede concretar entonces que H3 es un sistema de tipo *grid* hexagonal, con capacidad organizar la información jerárquicamente que va desde las 122 celdas de un nivel 0 de resolución, hasta las 569 millones divisiones del nivel 15, comprendiendo un total de 16 niveles (Tabla 2-3). El índice de H3 es de tipo hexadecimal, dónde los dígitos determinaran el nivel de resolución y la posición exacta. Las celdas base, es decir de resolución 0, están numeradas desde 0 hasta 121 según la latitud de sus puntos centrales, siendo el 0 el extremo norte y el 121 el más al sur. Cada una de las celdas se divide en el siguiente nivel en aproximadamente 7 celdas, siendo la relación real en su lado [33]:

$$H3\ edge_{finer\ resolution} = H3\ edge_{low\ resolution} \cdot \sqrt[7]{1}$$

Por último, H3 posee funciones listas para su uso que facilitan el análisis de datos espaciales, determinando distancia entre celdas, celdas que se encuentran juntas, celdas padre e hijo, hasta incluso desde la misma librería, realizar una compresión de diferentes niveles de resolución de un área determinada. Todo esto da un plus a este sistema ya que facilita el trabajo y puede ayudar a la hora de comprobar su funcionamiento y su empleo en el ámbito de interés de éste TFG, el entorno marítimo.

Desde la propia documentación online [33] de H3 se puede ver todas las aplicaciones del sistema, así sobre como emplearlas, los resultados que se pueden obtener y toda la información detallada para su ejecución.

Algunas de estas funciones ofrecen la capacidad de realizar búsquedas jerárquicas, y dependiendo de la resolución que se quiera utilizar, buscar unas celdas u otras. El sistema además permite asignar un nivel de resolución mayor o menor dependiendo de la cantidad de datos espaciales que se tengan en cada zona. Para facilitar consultas, se pueden buscar celdas *truncadas*, es decir, que parte de la etiqueta que corresponde a una celda de determinado nivel de resolución corresponde también a la celda de tamaño superior, celda *padre*. Esto hace que buscando esa celda truncada obtengamos todas las correspondientes que abarca la celda padre y viceversa.

2.5.2 Google S2

Como no podía ser de otra forma, el gigante Google también tiene su propio sistema de indexación espacial llamado S2, que viene de la notación para unidades esféricas s^2 . Este sistema es único ya que ofrece la representación de los datos en una esfera tridimensional. Hace posible la construcción de una base de datos con baja distorsión en todas partes con una forma muy similar a la de la Tierra. Este sistema es usado en su famosa aplicación de navegación *Google Maps*. Al igual que en otros sistemas de indexación, los creadores de S2 afirman que:

“S2 realiza una rápida indexación espacial de los tres tipos de objetos (punto, línea, polígono), resuelve operaciones normales como medir distancias, calcular centros y todo ello de manera robusta y fiable, con una representación sencilla y rápida de coordenadas” [34].

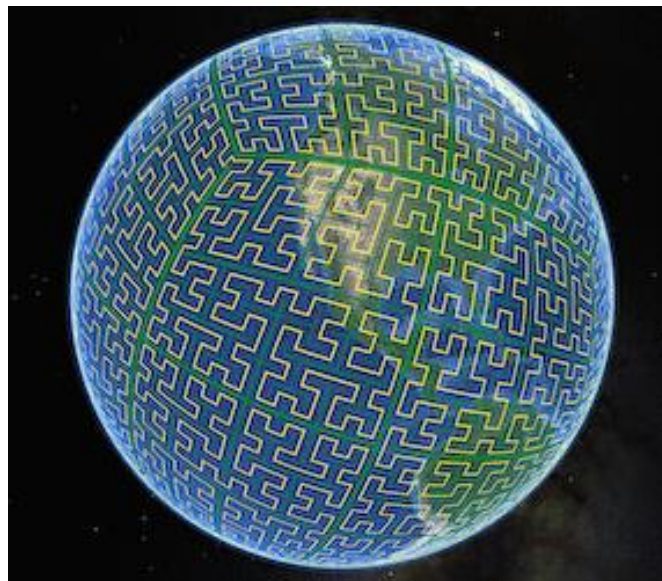


Figura 2-18. Esfera de S2 y el cubo adscrito que envuelve la Tierra, con la curva de Hilbert en cada uno de los planos del cubo. [35]

En Figura 2-18 se visualiza como divide el espacio terrestre este sistema. Utilizando la curva de Hilbert (parecido a la curva *Z-order*) recorre los diferentes puntos terrestres, para de esta forma, organizar los datos espaciales.

S2 trata de cubrir los problemas de resolución de los sistemas tradicionales de cartografía basados en proyecciones planas cómo la ya nombrada proyección Mercator. Está claro que nuestro planeta no es una esfera perfecta, pero según S2 Geometry [36] “con proyecciones esféricas, es posible aproximar toda la superficie de la Tierra con una distorsión máxima del 0,56%. Lo más importante es que las proyecciones esféricas preservan la topología correcta de la Tierra: no hay singularidades o discontinuidades con las que lidiar.”

Puede surgir la pregunta de por qué no utilizar una proyección elipsoidal que se aproxima más a la forma real de la Tierra, pero los cálculos aumentan debido a que trabajar con coordenadas esféricas es mucho más rápido y sencillo y tampoco se sabe cómo implementar la geometría elipsoidal en los algoritmos de búsqueda [36]. Al igual que Uber, se trata de utilizar una descomposición jerárquica en celdas o células, en este caso esféricas, representando en este caso las coordenadas en vectores 3D.

Este sistema utiliza una estructura de datos en forma de *quad tree* o árbol cuádruple que básicamente divide el espacio 2D consecutivamente en cuatro cuadrantes o regiones [28]. La construcción de la célula que utiliza S2 sigue unos pasos determinados, partiendo de una esfera de dimensiones $[(1,-1);(1,-1);(1,-1)]$ dentro de un cubo, proyectado un punto P , que será la localización que se quiere situar, en una cara del cubo, y con el correspondiente *quad tree* (árbol cuádruple) que contenga P [37].

Este punto sigue un proceso para terminar asignándole un valor o *ID* de 64 bits que determina la posición en el mallado de esa localización exacta (Figura 2-19).

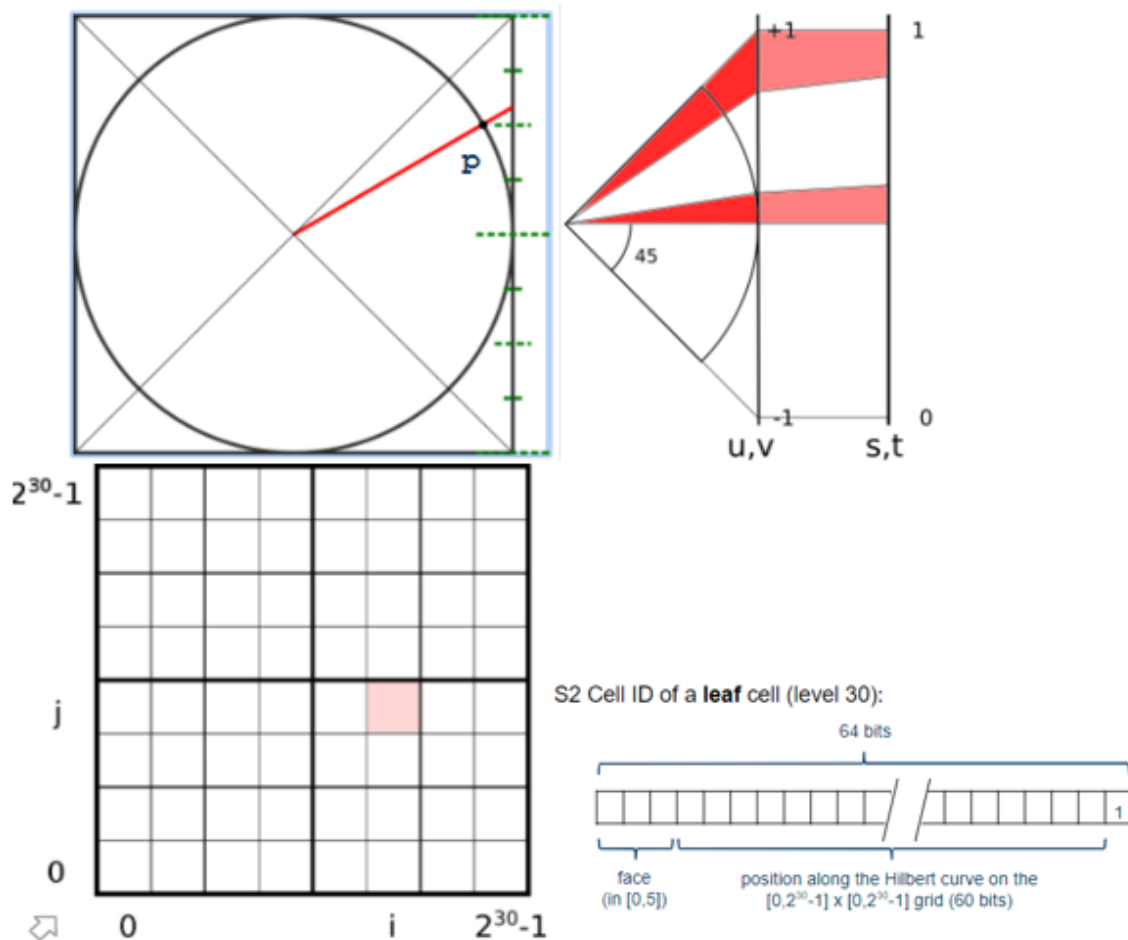


Figura 2-19. Creación ID de la celda S2. [37]

Este sistema está basado en el fenómeno de la *curva de Hilbert* (Figura 2-20), que básicamente es una curva que ocupa todo el espacio y gracias a la cual el ingeniero de Google Eric Veach, pudo construir esta librería. Básicamente la curva va recorriendo la cara del cubo designando el ID a cada celda. Este sistema posee 30 niveles, donde cada cm^2 puede ser representado, el nivel 30 es el que posee la celda más pequeña y el nivel 0 el más grande. Las áreas que abarcan las celdas van desde los $85.011.012 \text{ km}^2$ hasta los $0,48 \text{ cm}^2$. En definitiva, divide la superficie en 6 celdas en primer lugar, las cuales son las caras del cubo y mediante la curva de Hilbert es capaz de recorrer cada una de las celdas que cada vez se dividen en otras más pequeñas con el objetivo de situar la posición deseada. Mientras se va recorriendo estas celdas se va creando el ID de la celda.

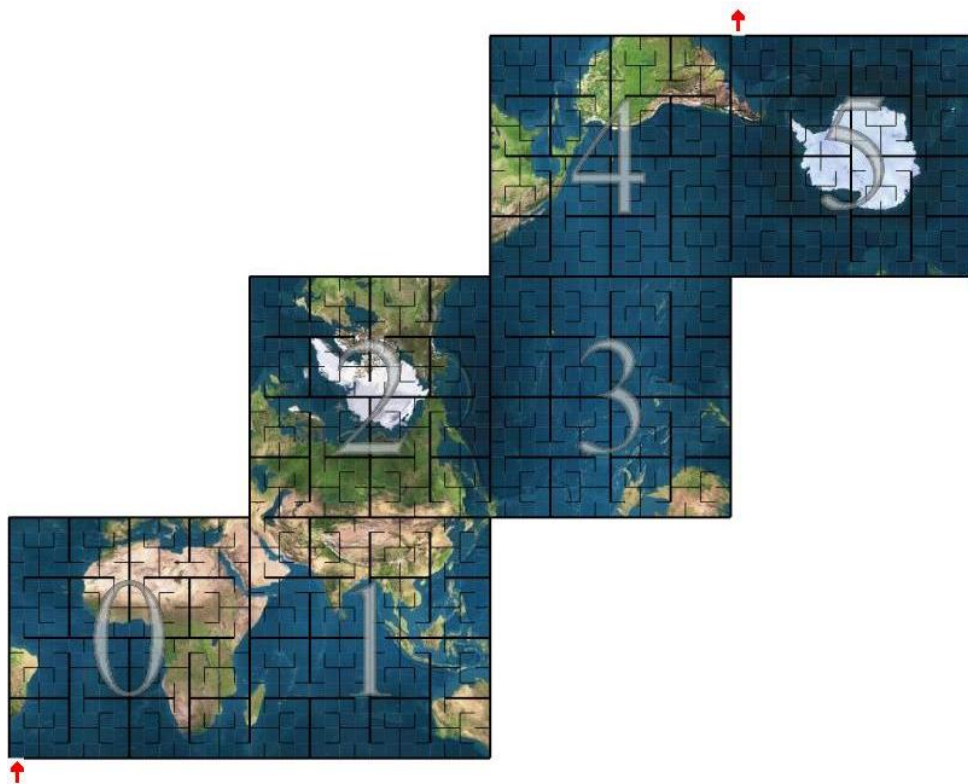


Figura 2-20. Curva de Hilvert sobre el globo. [36]

También se posee la capacidad de jerarquizar una región a diferentes niveles según convenga, especificando los niveles máximos y mínimos que se quieren utilizar. Como se puede ver en Figura 2-21. Este proceso, más conocido como *RegionCoverer*, permite aproximar cualquier región de interés, arbitraria, como uniones de celdas o células. Sirve para enfocar nuestra búsqueda a la hora de manejar toda la información sobre un territorio en concreto, cubriendo toda la región escogida. Este sistema está disponible en lenguajes como C++ y Java.

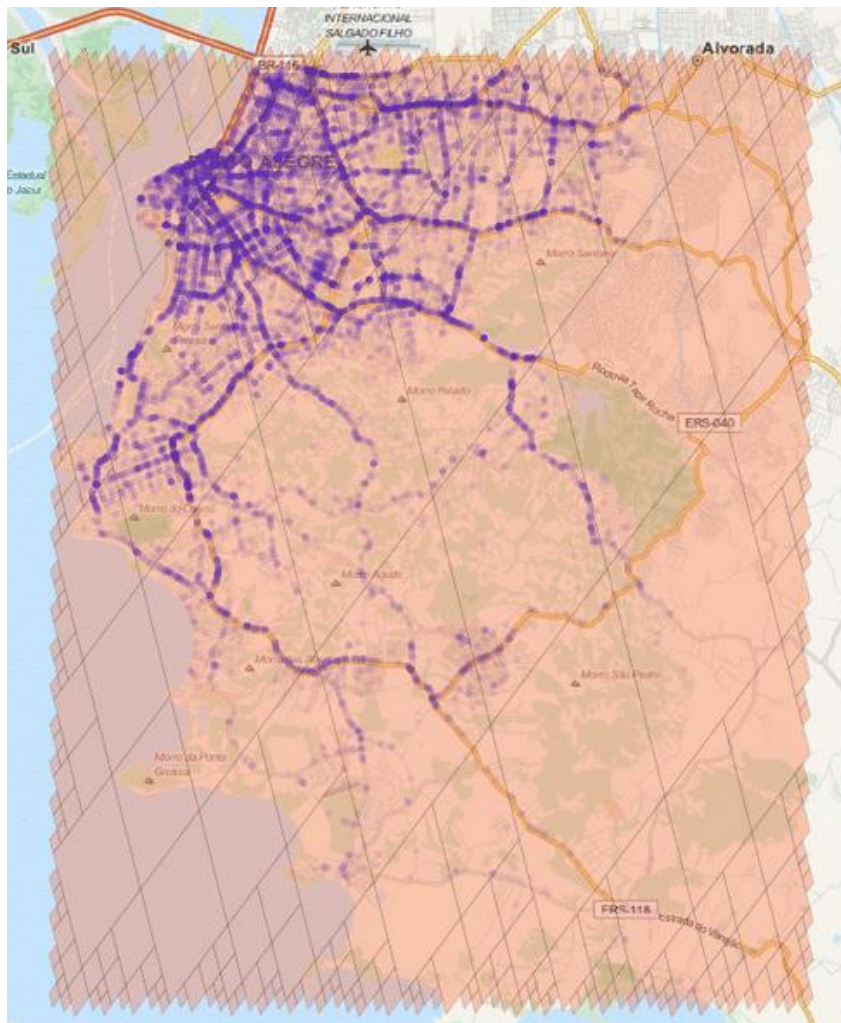


Figura 2-21. Ejemplo jerarquización mixta S2. [38]

2.5.3 Geohash

Geohash es otro sistema que utiliza la *Z curve* para su funcionamiento y a la vez el método de rejilla *grid*. Tiene sus orígenes en 2008, cuando Gustavo Niemeyer publicó en [39] el geocódigo creado por él mismo. El objetivo según su propio blog es proporcionar al usuario un índice que transforma latitud y longitud en un código mucho más sencillo y corto, proporcionando la capacidad de localizar puntos específicos en el espacio. Es de código completamente libre y principalmente en lenguaje Java. Originalmente tenía su objetivo orientado a un método de acortamiento de URL, pero finalmente funciona como un sistema de indexado geoespacial que mezcla el reticulado *grid* y la *Z curve*. Sus usos actualmente se centran en búsquedas de proximidad espacial, viajes digitales, o en redes sociales como aplicaciones de citas o *dating apps*.

Utiliza una nomenclatura alfanumérica de base-32 es decir, caracteres desde 0 al 9 y desde A a la Z (sin A, I, L, y O). Divide la tierra en 32 cuadrículas. Las cuales cada una se van dividiendo en 32 y así sucesivamente. La precisión va desde celdas de 5.000 km de lado hasta celdas de 4,77 km de lado. Sin embargo, las celdas no son siempre cuadradas [40]. A diferencia de otros sistemas, la curva que describe este sistema es una curva *Z* que varía dependiendo las formas de las celdas. Posee unos doce niveles de precisión o resolución. Siendo la primera división el primer carácter, siendo el número de letras y números que contenga el código el nivel de precisión al que se llega. En la Tabla 2-4 se describen los niveles de resolución, así como las características de las celdas y en la Figura 2-22 se ven como ejemplo los primeros niveles de precisión.

Nivel	Ancho celda	Longitud celda
1	5000 km	5000 km
2	1250 km	625 km
3	156 km	156 km
4	39,1 km	19,5 km
5	4,89 km	4,89 km
6	1,22 km	0,61 km
7	153 m	153 m
8	38,2 m	19,1 m
9	4,77 m	4,77 m
10	1,19 m	0,596 m
11	149 mm	149 mm
12	37,2 mm	18,6 mm

Tabla 2-4. Niveles de precisión Geohash. [41]

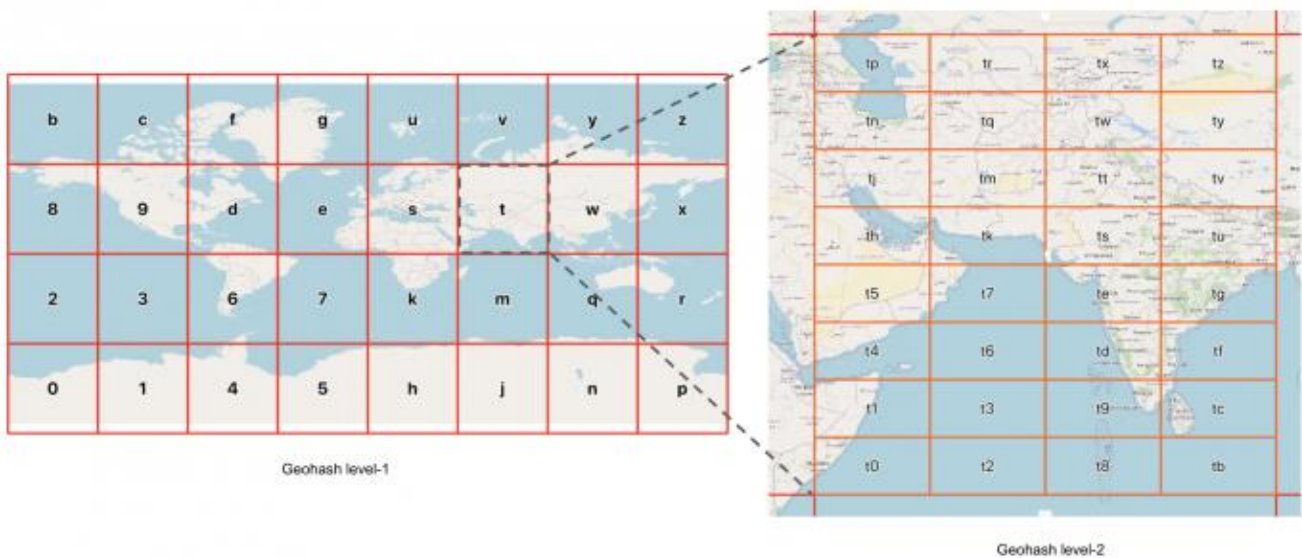


Figura 2-22. Niveles 1 y 2 Geohash. [41]

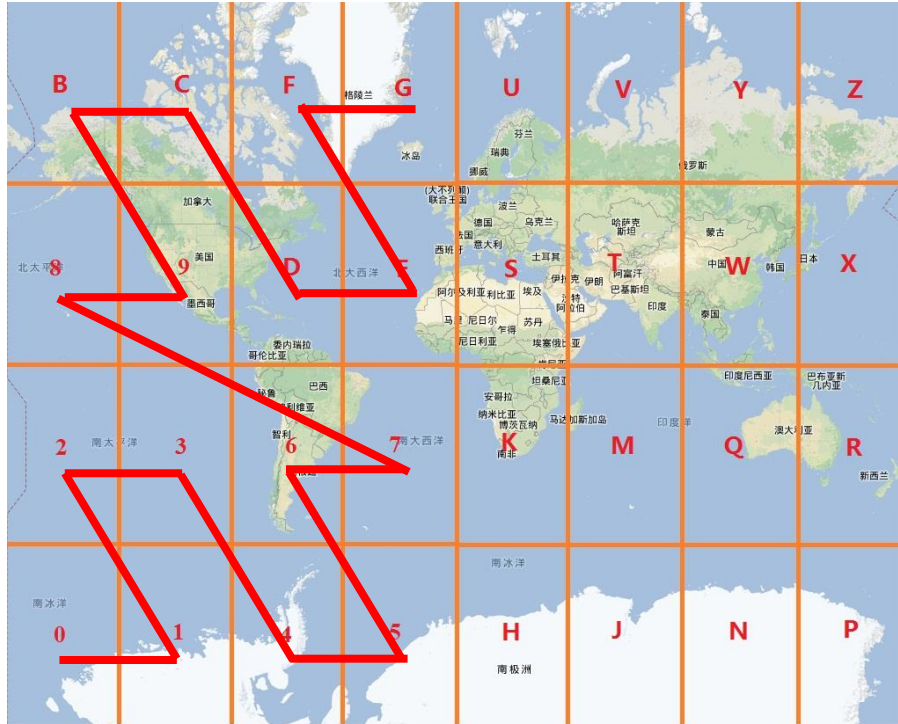


Figura 2-23. Curva de orden Z recorriendo las celdas. Imagen de [42].

Mediante un código binario de base 32 va nombrando cada una de las celdas y se obtiene una gran cantidad de resoluciones. Este sistema se encuentra disponible en Github, además de que existen algunas variantes cambiando la curva de orden Z (Figura 2-23) por una curva Hilbert. [42]

Existe un calculador online en [43] donde se puede obtener la correspondiente coordenada a una posición en concreto de latitud y longitud. Por ejemplo, las coordenadas de Marín serían ez9c4j0uqu3e en el máximo nivel de resolución, y según le se le resten dígitos reducimos la resolución. Como se ve en Figura 2-24 conforme se hace zoom la cuadrícula cambia (ez9c1, ez9c1v...).



Figura 2-24. Captura de [44] en Marín, Pontevedra.

2.5.4 German Naval Grid

Los últimos sistemas que se han analizado son muy modernos, pero desde siempre se ha requerido organizar el espacio de tal forma que se pueda tener un control más exhaustivo de éste. Durante la Segunda Guerra Mundial los alemanes plantearon un sistema rejilla que permitía simplificar y clasificar más fácilmente las posiciones en la mar. Esta malla o red de cuadrículas (Figura 2-25) consistía en 536 cuadrados que cubrían prácticamente toda la superficie marina. Aunque actualmente es difícil utilizar ese modelo de cuadrícula y transformarlo en coordenadas geográficas [45]. En un principio fue introducida por la *Luffwaffe* (fuerza aérea nazi) justo antes del comienzo de la guerra. La versión de la *Kriegsmarine* (Armada nazi) usado para localizar rápidamente, informar y vectorizar sus unidades (buques y submarinos), proporcionando una ventaja clara sobre el enemigo además de que codificaba sus posiciones.

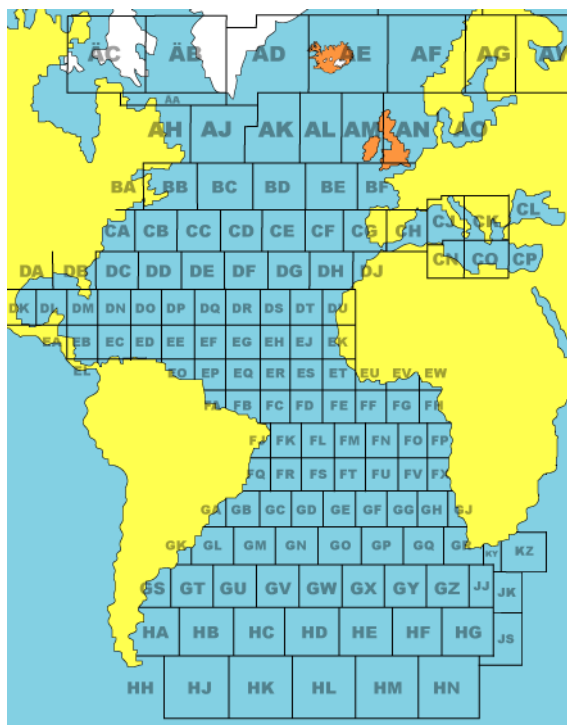


Figura 2-25. Ejemplo: Atlántico, Caribe y Mediterráneo mallado mediante el sistema alemán. [46]

Este sistema divide la superficie terrestre entera en cuadrados mediante una proyección Mercator, cada cuadrado es designado con 2 letras (AB, AE, BC...). Los cuadrados más grandes ocupan 486 MN cuadradas, estos cuadrados o *grids* son subdivididos en matrices de 9 cuadrantes más pequeños (matrices de 3x3). Cada uno de los nueve cuadrantes es dividido otra vez en 9 dando lugar a 81 cuadrados en el mismo sector alfabético. Los nuevos cuadrados reciben una designación numérica como se ve en Figura 2-26, desde 11 hasta 99

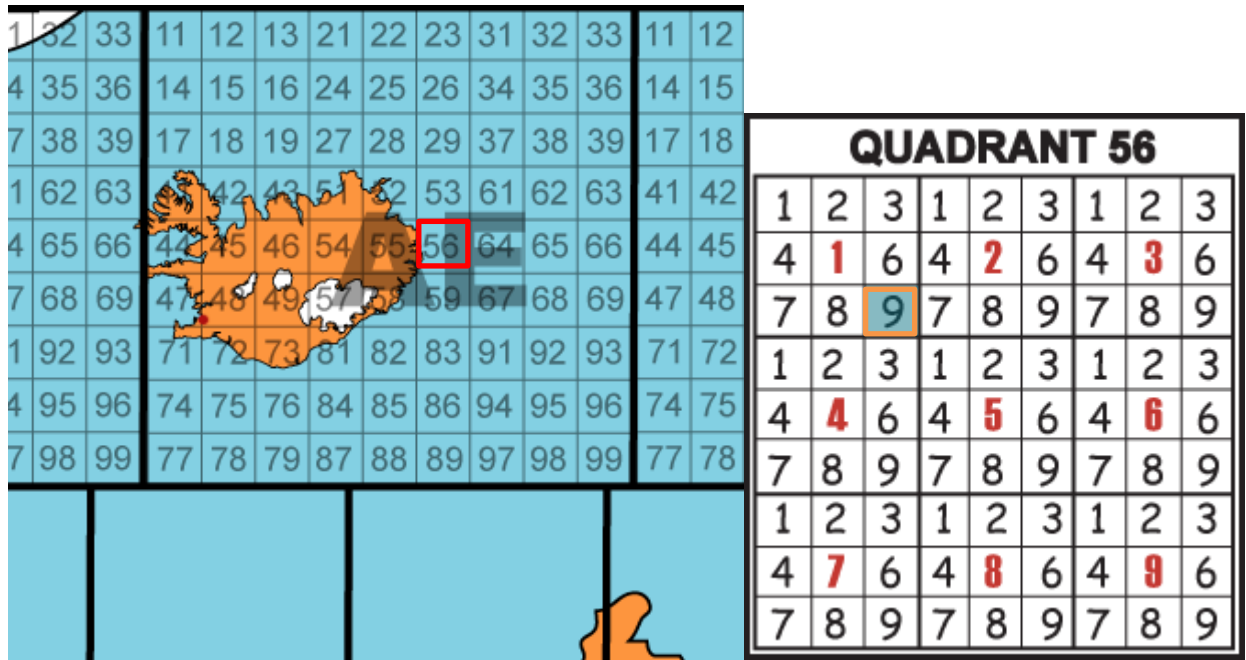


Figura 2-26. Ejemplo: Grid AE y todos los cuadrantes que lo conforman. [46]

El último paso vuelve a dividir cada uno de los cuadrantes en 9 que también son nombrados numéricamente de 1 a 9 cada cuadrado y en el centro el número de celda de izquierda a derecha, llegando a la máxima resolución. El índice de cualquier posición en la Tierra tendrá entonces 6 dígitos alfanuméricos con un área de 6 MN cuadradas (ej.: AE5619). Con el paso del tiempo los alemanes comenzaron a sospechar que los aliados estaban descifrando el código y dejó de utilizarse de la misma forma.

Actualmente Jan Kockrow y Chris Veness [45] han creado una herramienta que aproxima este sistema. Mediante su modelo matemático en Java pueden aproximar las coordenadas de los cuadrantes lo más aproximado posible. Su aplicación puede transformar una situación GPS de latitud y longitud en el cuadrante correspondiente y presentarlo en Open Street Map. Además, gratuitamente se puede descargar el cuadrado representado en KML para representarlo en Google Maps o Google Earth. Los cuadrados pueden resultar de tres tipos: regulares, irregulares parciales o irregulares poligonales, ya que la exactitud no es máxima. El código se encuentra disponible en el mismo sitio web de los creadores, que han aplicado esta vieja técnica para crear un sistema de indexado geoespacial moderno, escrito en PHP.

En la web de *Naval Grid Calculator* [45], se puede calcular el cuadrante correspondiente a una localización cualquiera. Por ejemplo, si colocamos las coordenadas de Marín (42°23'34"N, 008°41'56"W) obtenemos que el cuadrante correspondiente es CG2357, además podemos obtener información de su centro, vértices, y longitudes teniendo en cuenta la proyección Mercator (*Mean width*) cómo se ve en Figura 2-27.

Square CG2357

Clipboard KML

Coordinates:

Center
42°21'00"N, 008°38'00"W

Top left
42°24'00"N, 008°42'00"W

Top right
42°24'00"N, 008°34'00"W

Bottom left
42°18'00"N, 008°42'00"W

Bottom right
42°18'00"N, 008°34'00"W

Size:

Height
6 NM

Mean width
5.92 NM

Max width
5.92 NM

Min width
5.91 NM



Figura 2-27. Capturas de búsqueda en [45] con coordenadas Marín, Pontevedra.

2.5.5 UTM

UTM o *Universal Transverse Mercator* es un sistema parecido al anterior, a modo de rejilla y basado en la proyección cilíndrica transversal Mercator (Figura 2-28), que divide principalmente la superficie terrestre en 60 zonas. A diferencia de la proyección Mercator normal que es tangente el Ecuador, esta se hace secante a un meridiano. Este sistema cambia las coordenadas típicas de latitud y longitud en un sistema expresado en metros [47]. Esta proyección es cilíndrica, es decir, proyección del globo en una superficie, transversa, coincide con el eje del Ecuador, y conforme, mantiene valor de los ángulos.

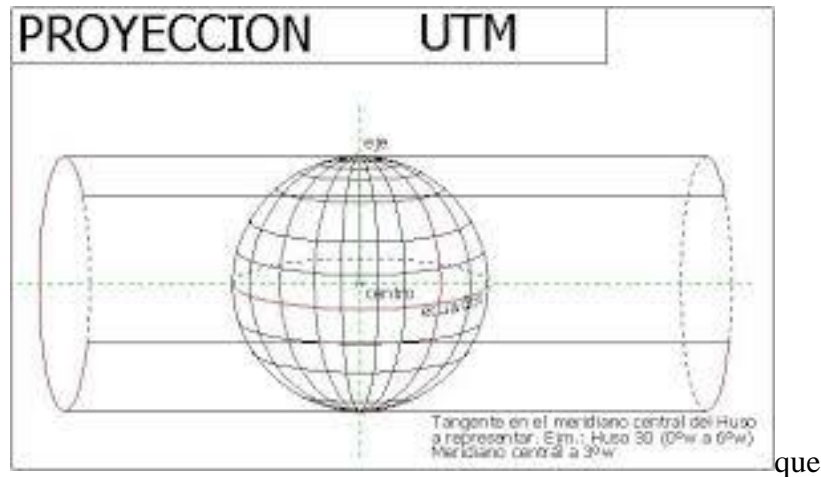


Figura 2-28. Proyección UTM. [48]

Según la *National Oceanic Atmospheric Administration* (NOAA) de los Estados Unidos de América el sistema fue desarrollado y puesto en marcha en los años 40 por el Cuerpo de Ingenieros del Ejército de los Estados Unidos. Sin embargo, algunas investigaciones apuntan a que los alemanes durante la Segunda Guerra Mundial podrían haber desarrollado un sistema muy similar, a parte del *Naval Grid*. Esto puede ser una confusión ya que según el cartógrafo Manfred Ferdinand Buchroithner en su libro “*Geodetic grids in authoritative maps—new findings about the origin of the UTM Grid. Cartography and Geographic Information Science*” este sistema indicaba las posiciones mediante letras seguidas de números, al igual que el sistema que hemos examinado anteriormente. [49]

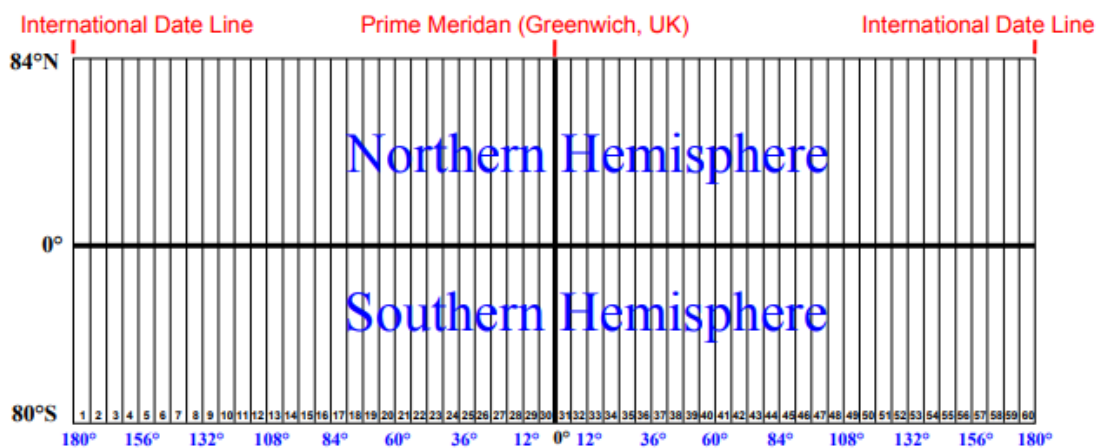


Figura 2-29. 60 divisiones de Este a Oeste en UTM. [50]

Como se ha nombrado antes, este sistema utiliza la proyección transversal de Mercator, las divisiones se realizan cada 6° creando 60 zonas de proyección o husos nombrados numéricamente (Figura 2-29). Cada una de las franjas es proyectada en un plano separadamente reduciendo la distorsión que se obtiene, estos planos son tangentes en cada uno de los usos al meridiano central. Esta proyección tiene límites al norte y al sur, concretamente en las regiones polares 80°S y 84°N, donde se utiliza el

sistema de coordenadas UPS o Universal Polar Stereographic [47]. Mediante esta proyección los meridianos y paralelos son líneas rectas y gracias a ello el sistema de coordenadas pasa de ser esférico a rectangular, facilitando la representación de cualquier punto o rumbo. Las distancias se miden fácilmente, entre distancias pequeñas son líneas rectas, pero si la distancia es muy grande las líneas se curvan.

Por otro lado, cada huso se divide en bandas de 8° de latitud. Las 20 bandas que se crean a lo largo de todo el huso se nombran con letras desde la C hasta la X con la I y la O omitidas para evitar confusiones. De esta manera una cuadrícula cualquiera quedaría designada con un primer carácter numérico indicando el huso y un segundo alfabético indicando la zona. En la Tabla 2-5 se describen las coordenadas UTM y el rango de latitud y longitud que abarca.

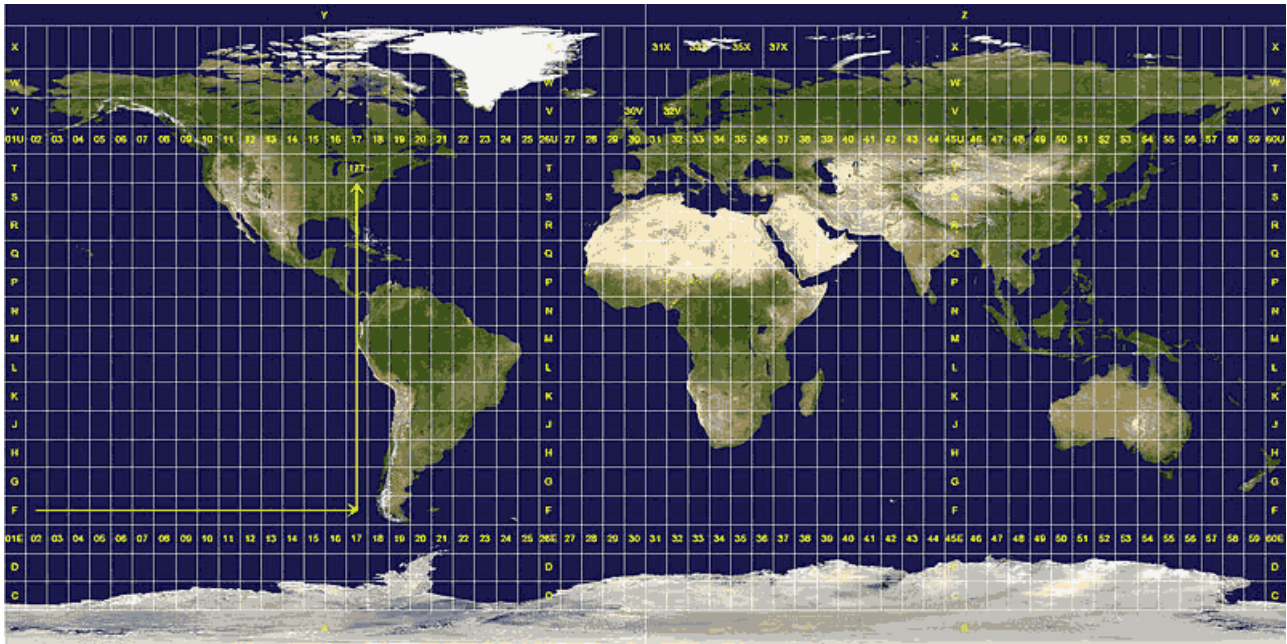


Figura 2-30. Sistema de coordenadas UTM completo. [47]

Norte/Sur	UTM	Latitud (°)	Norte/Sur	UTM	Latitud (°)
S	C	78-80	N	N	0-8
S	D	64-72	N	P	8-16
S	E	56-64	N	Q	16-24
S	F	48-56	N	R	24-32
S	G	40-48	N	S	32-40
S	H	32-40	N	T	40-48
S	J	24-32	N	U	48-56
S	K	16-24	N	V	56-64
S	L	8-16	N	W	64-72
S	M	0-8	N	X	72-84

Tabla 2-5. Correspondencia latitud a zonas UTM. [48]

Como se ha mencionado antes el sistema utiliza el sistema métrico para referirse a un punto en concreto dentro de cada celda formada. En cada huso el origen se sitúa en el punto donde corte ese meridiano central en particular con el Ecuador. Ahora, como se puede ver en Figura 2-30, en el eje vertical la distancia comienza desde el paralelo 80° S hasta el Ecuador (0m – 10000m) y continúa de nuevo desde el Ecuador al norte hasta el paralelo 84° N (0m – 10000m). En el eje horizontal el origen son los 500m situándose el 0 al Oeste (Figura 2-31), tomando valores únicamente entre 0 y 1000m. Las distancias en este sistema de coordenadas tienen como máxima resolución 1 metro cuadrado.

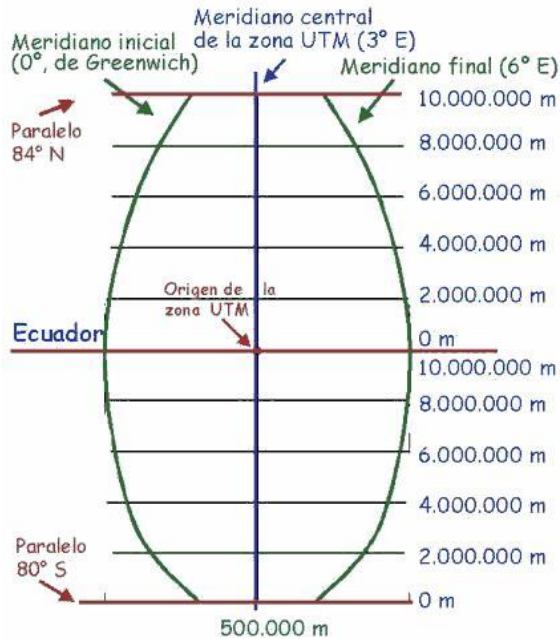


Figura 2-31. Ejes verticales y horizontales para la designación métrica. [48]

Por último, la notación se completa con las distancias añadiendo a los anteriores que indican la celda añadimos las distancias sobre los ejes vertical y horizontal formadas por 4,6 o 8 dígitos como máximo. La primera mitad referido al Este y la segunda al Norte o Sur. En [51] podemos calcular una posición cualquiera y obtener la correspondiente en coordenadas UTM. Por ejemplo, las coordenadas de Marín en UTM serían 29T 524715,3 4693276,8 (Figura 2-32).

Segundos

Latitud Grados ° : 42 Minutos ' : 23 Segundos " : 29 Hemisferio: N/+ ▾

Longitud Grados ° : 8 Minutos ' : 41 Segundos " : 59 Hemisferio: W/- ▾

Convertir los GG, MM, SS

Ingrese las coordenadas Estándar UTM

Zona 29 Hemisferio: N ▾

Este (UTMX): Norte (UTMY):

Convertir estándar UTM

Ingrese las coordenadas NATO UTM

Zona Long.: Zona Lat.: Dígrafo:

Figura 2-32. Coordenadas UTM de Marín, Pontevedra. [51]

2.5.5.1 MGRS

El MGRS o Military Grid Reference System es un sistema creado por el Ejército de los Estados Unidos para facilitar el posicionamiento y envío de coordenadas para soldados y marines americanos. Este sistema deriva del UTM utilizando su sistema de coordenadas y proyección, sin embargo, el MGRS divide cada región formada por las celdas de 6° x 8° en cuadrados de 100.000 metros cuadrados, recibiendo una designación de dos letras para distinguirse del resto. La primera letra es la A y empieza en el oeste aumentando hacia el este, corresponde además con el primer dígito de la coordenada UTM. La segunda letra comienza con la A en el sur y aumenta conforme más al norte se encuentre la posición, correspondiendo a su vez con el segundo dígito de coordenadas UTM (Figura 2-33). Gracias a este sistema de designación se simplifican las coordenadas UTM eliminando los primeros números. [50]

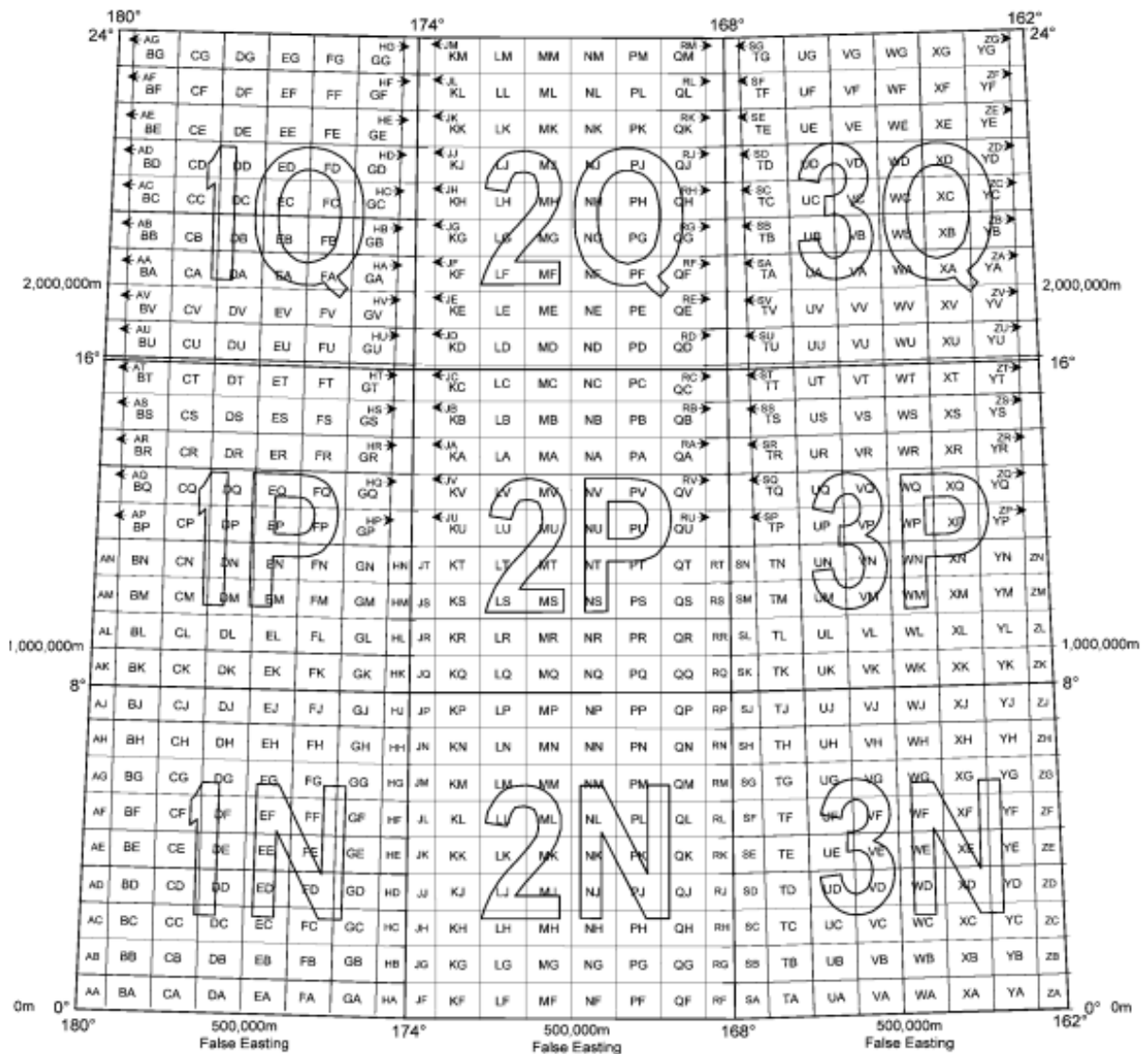


Figura 2-33. Ejemplo correspondencia coordenadas UTM – 100.000 m2. [52]

A la hora de posicionar un punto específico en este sistema primero se determina en cuál de los cuadrantes de UTM se encuentra, y luego cambiarlo por la nueva celda alfabética. De nuevo en [51] se puede obtener la posición correspondiente al MGRS. Vemos que se elimina el primer dígito numérico del este y los dos primeros del norte de la nomenclatura UTM. Siendo la posición de Marín 29T NG 24715,3 93276,8 (Figura 2-34).

Zona Long.: Zona Lat.: Dígrafo:

Este (UTMX):

Norte (UTMY):

[Convertir las coordenadas NATO UTM](#)

Figura 2-34. Coordenadas de Marín en MGRS/USNG/NATO UTM. [51]

2.5.6 Warp 10

Este sistema utiliza el método HHCode mediante la curva Z (Figura 2-35). Warp 10 trata de analizar datos de naturaleza espacio-temporal. Tiene 16 niveles de resolución, desde 2 a 32. Por ejemplo, en el nivel 2 está dividido en 16 celdas, en el nivel 4 en 256 y así sucesivamente. Siendo el nivel con más resolución el 32 con 18 billones de billones de celdas. Se utiliza una nomenclatura hexadecimal, gracias a la cual usando P dígitos tienes el nombre de una celda en el nivel 2P, fácilmente convertible a la longitud.

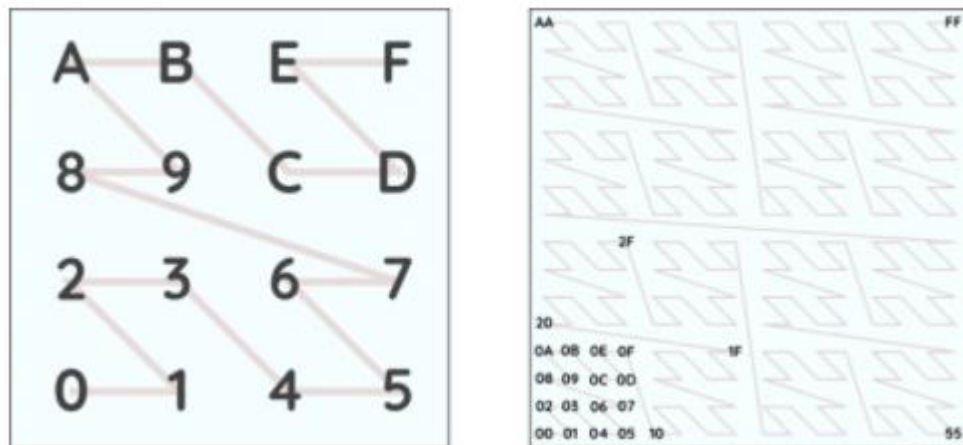


Figura 2-35. HHCode en nivel 2 y 4. [53]

El código ofrece escalas a los diferentes niveles de resolución que van desde los 10.000 km a 1 cm, tal y como se ve en Tabla 2-6. Estas medidas corresponden al ancho de cada celda a la altura del Ecuador. El tamaño de cada celda es la mitad y no cambia con la latitud.

Resolution (HHCode level)	Scale	Celdas para cobertura global
2	10000 km	16
4	2500 km	256
6	625 km	4,096
8	156 km	65,536
10	39 km	1,048,576
12	10 km	16,777,216
14	2,5 km	268,435,456
16	600m	4,294,967,296
18	150 m	68,719,476,736
20	40 m	1,099,511,627,776
22	10 m	17,592,186,044,416
24	2,5 cm	281,474,976,710,656
26	60 cm	4,503,599,627,370,496
28	15 cm	72,057,594,037,927,936
30	4 cm	1,152,921,504,606,846,976
32	1 cm	18,446,744,073,709,551,616

Tabla 2-6. Escalas disponibles en Warp 10. [54]

El objetivo que busca además Warp 10 es trabajar con áreas, como el resto de sistemas que se han analizado. Buscar aproximar lo máximo posible la realidad de cualquier área geográfica y el *geodata* a las celdas HH mediante la gestión de resoluciones e indexado es la meta. Para ello Warp 10 utiliza en su código GEOSHAPE que asocia a cada zona una resolución específica dependiendo de los datos que se quieran obtener. La resolución está almacenada en forma de prefijo en las celdas, por lo que GEOSHAPE solo puede contener celdas hasta nivel 32 y por tanto representables en 64 bits. Warp 10 ofrece la capacidad de combinar de múltiples GEOSHAPES con el objetivo de ajustar más los resultados. [53]

Warp 10 asocia coordenadas de latitud y longitud a estas celdas. A la hora de indexar, en vez de asociar una etiqueta a cada uno de los datos, se prefiere que se asocie un atributo que ayude a caracterizar los objetos y jerarquizar la búsqueda. [53]

Como características de uso, Warp 10 se encuentra disponible en GitHub para lenguajes como R y Python.

2.6 Spatial databases

Tras comprender un poco toda la información geográfica y cómo funciona la indexación, ya no solo en cómo el geoespacio si no en todas las aplicaciones que existen hoy en día, se puede apreciar que se encuentra completamente relacionado con el *Big Data*.

Después de haber analizado los diferentes sistemas de indexado y para comprender mejor dónde funcionan estos sistemas se procede en el siguiente apartado a explicar las bases de datos espaciales, describiendo a continuación las diferencias entre una base de datos espacial y una relacional. Tras ello se presentan algunos sistemas que juntan bases de datos, sistemas de indexado y análisis geoespacial que también son de interés de este proyecto.

Para poder analizar todos los datos que se puede manejar es necesario aplicar un criterio, un orden, con objeto de jerarquizarlos y lograr unos resultados más eficientes. Sin embargo, los objetivos de estos sistemas son muy variados, ofrecen características que son muy útiles para múltiples organizaciones. En particular, los datos espaciales se almacenan en unas bases de datos espacial o *Spatial database*, es donde el indexado trabaja a la hora de gestionar todos estos datos, que normalmente son georreferenciados, es decir, referenciados a las coordenadas terrestres.

Una base de datos espacial permite el almacenamiento de geometrías, es decir de polígonos, en forma de archivo cartográfico de tal forma que un usuario pueda analizar todos los datos con lenguajes como SQL. Si se quiere una definición más específica, según explica Ralf Hartmut Güting de la Universidad de Hagen, Alemania [55], una base de datos espacial es:

1. Una base de datos espacial es un sistema de bases de datos
2. Ofrece *spatial data types* (SDT) en su almacenamiento de datos y modelo de consulta.
3. Ofrece el soporte necesario para el manejo de estos datos, como su indexado geoespacial y algoritmos eficientes para poder asociarlos espacialmente

Explicando concretamente esta definición parte por parte. En primer lugar, una base de datos espacial no solo almacena datos geográficos o localizaciones, sino que en sus datos también se encuentran valores alfanuméricos no espaciales, dando un plus adicional a sus capacidades. En segundo lugar, los SDT no son más que los puntos, líneas y polígonos que hemos nombrado previamente dando propiedades de modelización de estructuras geométricas a la vez que sus relaciones (intersecciones, líneas) o propiedades (como el área, polígono). En tercer y último lugar el sistema debe de ser capaz de analizar desde una gran cantidad de objetos dentro de un área en particular sin tener que escanear a todo el conjunto restante; es donde el indexado espacial es completamente obligatorio. Debe de ser capaz de conectar objetos de diferentes clases dentro de una relación espacial, filtrando las características de esos objetos que sean de más interés para la aplicación en concreto. [55]

Los SDT son también llamados *spatial algebras* ya que fundamentan las relaciones entre puntos, líneas y regiones con operaciones. Esta álgebra es más conocida como ROSE (*RObust Spatial Extension*).

Operation	Solution
<i>line x line</i>	<i>point (intersection)</i>
<i>region x region</i>	<i>region (intersection)</i>
<i>geo x region</i>	<i>bool (inside)</i>
<i>ext1 x ext2</i>	<i>bool (intersects, meets)</i>

Tabla 2-7. Ejemplo de operaciones algebraicas en bases de datos espaciales [55].

Los SDT interactúan entre ellos de tal forma que como se ve en Tabla 2-7 las operaciones entre ellos resultan otro elemento espacial. Mientras que la multiplicación de dos líneas resulta la intersección de ellas en forma de punto, la de dos regiones lo mismo, pero en forma de otra región. Por otro lado, si juntamos un *geo* (conjunto de líneas, puntos y regiones) con una región en particular se obtiene como resultado un conjunto de objetos *bool* (booleano) de *geo* que se encuentran dentro de la región en

particular. Al igual que al unir dos *ext* (conjunto de líneas y regiones) obtenemos diferentes intersecciones y zonas en común. Esto es solo un ejemplo de las operaciones que se pueden realizar, las funciones del programa que se utilice cambiarán según cada uno de ellos

Estos datos guardan unas relaciones que son las que sirven al operador y las que interesan a la hora de trabajar con la base de datos. Se nombra en [55], “entre las operaciones que ofrecen las álgebras espaciales, *spatial relationships* son las operaciones más importantes”. De forma general, tenemos tres tipos de relaciones:

- *Topological relationships*: como objetos adyacentes, unos dentro de otros o sin relación alguna
- *Direction relationships*: Encima, debajo, al sureste, al norte...
- *Metric relationships*: distancia <100, >1000...

Existen una gran cantidad de sistemas gestores de bases de datos relacionales, según sus capacidades pueden dotar de diferentes funciones y diferentes tipos de análisis. Antes de ver que sistemas de indexado pueden presentar algunas bases de datos espaciales o simplemente se aplican a bases de datos únicamente relacionales, se va a analizar una de estas en profundidad.

En los siguientes apartados se describen diferentes sistemas de gestión de datos. Además, se abre el abanico para aquellos que también utilizan un método de indexado geoespacial en sus *databases*. Como se ha comentado, la principal diferencia entre estos sistemas y las bases de datos únicamente relacionales es la capacidad de analizar datos en dos dimensiones mucho más rápido.

2.6.1 Database Management Systems: MySQL

Un sistema de gestión de base de datos DBMS o SGBD es un conjunto de programas que permite el uso y modificación por parte de los usuarios, asegura los datos y facilita el manejo. Hay básicamente cuatro modelos de bases de datos: relacionales, jerárquicos, en red y relacional con objetos. Lo más comunes son los de tipo relacional, que organizan las bases de datos en forma de tablas con filas de registros (tuplas) y columnas que son equivalentes a los atributos y características de cada uno de los registros [56].

Particularmente, MySQL es un sistema de base de datos relacional *open-source*. Cómo indica su propio nombre utiliza el lenguaje SQL (*Structured Query Language*) con el que un usuario puede comunicarse con el propio sistema, este lenguaje es seguramente el más utilizado en el mercado en cuanto a SGBD [56]. En la propia documentación de MySQL [57] se dice que su servidor es de los más rápidos, flexibles y sencillos del mercado. Mediante el lenguaje SQL, el cual desde 1986 es un estándar definido por ANSI (American National Standards Institute), se permite la modificación y gestión de todas las tablas de una forma sencilla y polivalente. MySQL se originó como un sistema de consultas rápidas a bajo nivel (ISAM), pero debido a la falta de velocidad se pensó en mejorar el sistema apareciendo el actual. Se encuentra escrito en C y C++, trabaja con diferentes plataformas, y utiliza B-tree como sistema de indexado para sus datos relacionales. [58]

Soporta diferentes tipos de datos: FLOAT, DOUBLE, CHAR, VARCHAR, BINARY, VARBINARY, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, SET, ENUM.

DB: SUPERMERCADO - TABLE: CUENTA 1				
ID	Producto	Tipo	Cantidad	Precio unitario
01	Manzana	Frutería	6	1,24
02	Ensaladilla rusa	Comida preparada	2	4
03	Zumo naranja	Bebidas	2	2
04	Cerveza premium	Bebidas	12	0,89
05	Cereales	Desayuno	2	1,52

Tabla 2-8. Tabla ejemplo de base de datos de un supermercado.

Mediante los comandos SQL se puede manejar las bases de datos, acceder a esos datos, y utilizarlos de la manera que más convenga. Mediante diferentes comandos podemos acceder a las tablas y realizar las consultas que necesitemos. Algunos comandos de consulta y modificación, referidos a la Tabla 2-8, pueden ser:

- `SELECT * FROM CUENTA 1`: selecciona todos los registros y los atributos de la tabla, es decir con (*) seleccionamos toda la tabla.
- `SELECT tipo FROM CUENTA 1 WHERE price = 1.24`: seleccionaríamos el atributo tipo de todos los registros que tengan un precio de 1,24. En este caso “Frutería”.
- `INSERT INTO CUENTA 1 VALUES (Cereales, Desayuno, 2, 1.52)`: introduciríamos la nueva fila como un nuevo registro
- Existen otros comandos como `DROP` (elimina registros), `ORDER` (ordena según criterio), o `LIMIT` (que indica un límite de la consulta).

Para realizar consultas espaciales se necesita la utilización de dos campos, pues se trata de datos en dos dimensiones. Por ejemplo, para ver la distancia que existe entre las posiciones de dos buques, se tendrían que realizar las operaciones pertinentes para ver la diferencia, primero con un campo como la latitud y luego complementar con la longitud. Esto se traduce en un incremento del tiempo de análisis y consulta.

Aunque esta base de datos no es espacial, permite la creación de estructuras internas como índices que facilitan la consulta de datos, ordenando los valores de cada uno de los campos y permitiendo un acceso más ágil. Sin embargo, la agilidad a la hora de trabajar con datos en dos dimensiones como son las posiciones de buques en la mar requiere de un sistema de indexado como los que se han visto, de forma externa. No obstante, posee las características necesarias para la gestión de datos de manera sencilla y efectiva.

2.6.2 Spatial databases: IBM Informix

La empresa IBM utiliza en su software IBM Informix, en el que gestiona gran cantidad de datos para finanzas, análisis de mercado en tiempo real, gestión de datos espaciales y temporales de forma flexible e implementación en el IoT o *Internet of Things*. Utiliza el *R-Tree* en la indexación de datos geoespaciales. Tiene diversas funcionalidades extras orientadas a la gestión de empresas o negocios aplicando consultas en SQL y jerarquizando según se desee. El sistema en particular se llama *Spatial Datablade*. Este sistema utiliza los árboles R para decidir si utilizar celdas más grandes o más pequeñas ya que dependen de cada una de las búsquedas que se realizan.

Este sistema, a diferencia de MySQL, permite el acceso directamente a datos espaciales ya que la propia base de datos posee un sistema de indexado geoespacial como es *R-tree*.

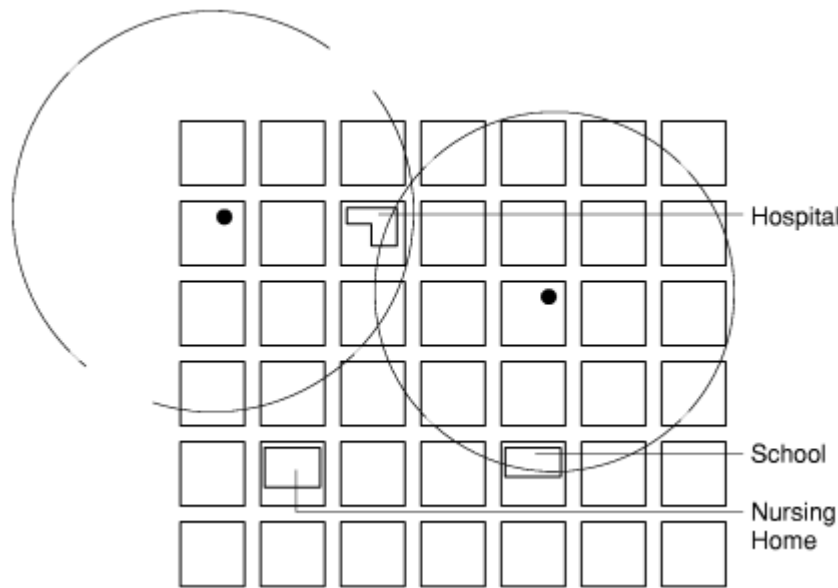


Figura 2-36. Ejemplo análisis espacial IBM Informix. [59]

Este sistema debido a que está dentro de un paquete de productos de los que dispone IBM no es de código libre. Sin embargo, en el sitio web de IBM [60] se puede encontrar toda la información de cómo realizar cualquier consulta al sistema, acceder a la base de datos, y realizar diversas funciones como se ha nombrado previamente. Dota de la capacidad de realizar consultas como búsqueda de vecinos en un radio o indexar según cualquier etiqueta (como un índice de peligrosidad, por ejemplo). Al utilizar R-tree como sistema de indexado no hay tamaño de celda ya que dependerá de los límites establecidos en cada índice para el número de nodos y número de entradas de las que se disponga. Esto es común para todos los sistemas que aplican R-tree de manera espacial, sin aplicar un sistema de reticulado terrestre o *grid* (ejemplo en Figura 2-36).

2.6.3 Spatial databases: Amazon Aurora

Amazon Aurora es un sistema de gestión de bases de datos de todo tipo. Amazon utiliza este sistema para analizar todo tipo de datos para su empresa, análisis de mercado, aprovisionamiento, detección de errores, asignación de almacenamiento automático, etc. Este sistema concretamente no utiliza *R-tree* siempre, sino que usa *B-tree* cuando es requerido [61]. Como se había comentado anteriormente *B-tree* por sí solo no analiza datos multidimensionales. Para ello, Aurora utiliza la *Z curve* para analizar los puntos espaciales. Como se ha explicado, la curva va recorriendo las celdas en orden binario. El B-tree organiza esos puntos según esta numeración, es decir en el orden que da Z, proponiendo el camino más rápido a la hora de encontrar uno de los nodos. No se va a explicar este sistema ya que la definición es prácticamente igual que la de *R-tree*. [62]

Sin embargo, como explica Sirish Chandrasekaran como Product Manager de Aurora en [62], no es suficiente con implementar esta curva ya que a la hora de realizar una búsqueda en un área determinada se van a analizar también el resto de orden inferior. Como se ven en la Figura 2-37, siendo el área verde el área objetivo y los puntos con el símbolo de peligro los redundantes. Para solucionar esto Aurora realiza las consultas en consultas más pequeñas reduciendo el número de valores que tiene que recorrer Z. Además, en los datos de direccionamiento del B-tree, el nodo almacena a que nivel se encuentra en el árbol y además la dirección Z.

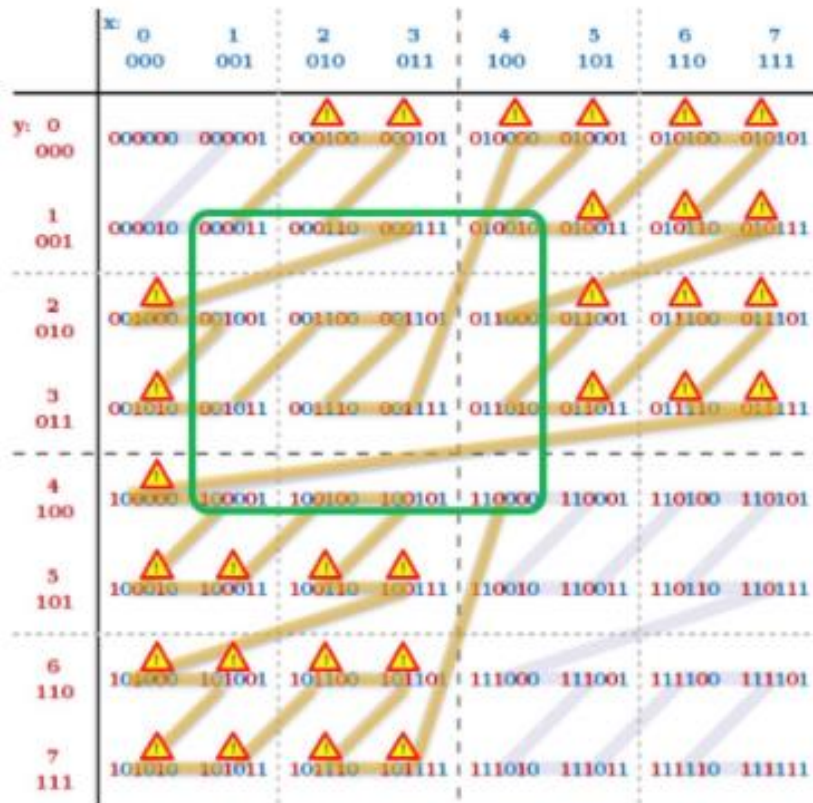


Figura 2-37. Análisis espacial de la Z curve en Aurora. [62]

2.6.4 Spatial databases: Oracle Spatial

Oracle Spatial es un sistema de manejo de base de datos espaciales. Está ideado para almacenar información geográfica y resolver consultas con respecto a cualquier referencia espacial. Mediante HHCCode representan figuras (puntos, líneas o polígonos) del espacio Euclídeo incluyendo además profundidad, elevación o tiempo. El HHCCode es generado mediante la descomposición recursiva del espacio en cuadrados, mediante un sistema de gestión de base de datos Oracle PL/SQL se almacenan los datos en tablas de bases de datos. [63]

Según la propia Oracle en [64] las características que ofrece esta herramienta para empresas son tales como modelo de datos en 3-D, *point clouds*, soporte de servicio web Geoespacial, motor de ruteo de localizaciones más rápido, gran capacidad de carga de pedidos de consulta, sistemas de Geoespacial ubicación mediante su aplicación *GeoRaster*, etc. Esta aplicación está orientada a su uso en grandes empresas y está disponible en lenguajes estándar como SQL y Java. Aunque el sistema inicial utiliza el HHCCode, también se puede implementar un *R-tree* estándar. Previamente dividía el espacio y, por tanto, los datos, en cubos 3D que permitían almacenamiento de hasta petabytes. Sin embargo, ahora cambiando a *R-tree* consigue una velocidad de indexación de 20 a 30 veces superior. Puede determinar regiones, además establece relaciones entre proximidad de objetos o localizaciones.

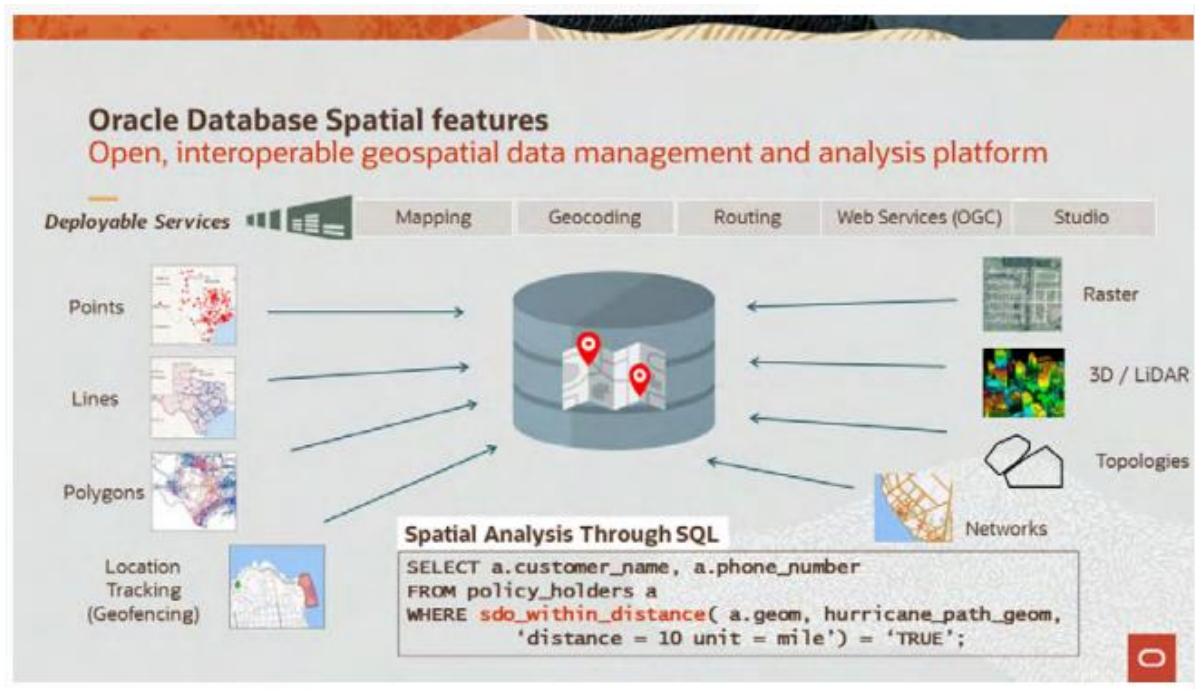


Figura 2-38. Características y ejemplo de la base de datos Oracle Spatial. [65]

En la Figura 2-38 se puede ver como con una sentencia SQL se puede realizar un análisis espacial, aplicar topologías, polígonos, puntos. Ofrece la capacidad de almacenar los datos y realizar las consultas a la vez ya que integra indexado, análisis y almacenamiento de datos espaciales.

2.7 Tabla comparativa final

Nombre	Tipo de sistema de indexado	Open Source	Lenguaje	Niveles de precisión	Forma de celda	Index representation	Sistema de proyección	Funciones analíticas extra	Renombre (subjetivo)
<i>Uber H3</i>	Hierarchical Hexagon grid	Sí	C, Java, Python.	16	Hexagonal	64-bit Hexadecimal	Gnomónica (icosaedro)	Sí	Sí
<i>Google S2</i>	Quadtree Hilvert curve, grid	Sí	C++, Java	30	Rectangular	64-bit binario	Cuadrática (caras de un cubo)	Sí	Sí
<i>IBM Informix</i>	R-tree	No	C, C++, Java.	Sin límite	Rectangular	Alfanumérico (nodos)	Ninguna	Sí	No
<i>Amazon Aurora</i>	B-tree Z-curve	No	Java, PHP, Python	Sin límite	Rectangular	Alfanumérico (nodos)	Ninguna	No	No
<i>Geohash</i>	Z-curve grid	Sí	Java	12	Rectangular	Alfanumérico base-32	Mercator	No	Sí
<i>Naval Grid</i>	Grid	Sí	PHP	3	Rectangular	Alfanumérico	Mercator	No	No
<i>Oracle Spatial</i>	HHCCode R-tree	Sí	Java	Sin límite	Rectangular	Numérico	Ninguna	Sí	No
<i>Warp 10</i>	HHCCode Z-order curve	Sí	R, Python	16	Rectangular	Alfanumérico	Mercator	No	Sí

Tabla 2-9. Comparativa final de los sistemas de indexado analizados.

Mediante una comparativa final en la Tabla 2-9 se quiere reflejar los puntos fuertes y débiles más importantes de las diferentes herramientas que hemos analizado, según las características y objetivos propios de este proyecto. Se valoran las siguientes características:

- Tipo de sistema de indexado: se especifica sobre que sistemas tipo están basados las diferentes herramientas de indexado, tras haber analizado cada una con anterioridad da una idea de sus características principales.
- *Open Source*: se indican cuáles de los sistemas presentan un código de acceso totalmente libre y público.
- Lenguaje: importante aclarar en que lenguajes están escritos los sistemas para poder implementarlos.
- Niveles de precisión: en el caso de existir un reticulado de celdas terrestre, se detallan los niveles de precisión que existen, en el caso de los R-tree al no haber unas celdas fijas por ser un sistema de nodos no existen niveles de precisión más que la capacidad de los nodos.
- Forma de la celda: se evalúa la forma geométrica que engloba los datos espaciales.
- *Index representation*: se indican las diferentes representaciones de los datos espaciales en cada sistema.

- Sistema de proyección: se especifican las proyecciones geográficas que utilizan los diferentes sistemas, si es que la representación espacial se hace geográficamente, en caso contrario se indica la carencia de proyección.
- Funciones extras: se evalúan si las herramientas pueden presentar alguna característica extra de análisis importantes para este proyecto. Tales pueden ser el análisis de puntos cercanos a un centro de coordenadas, cálculo de la cercanía entre varios puntos y la proporción de librerías extra que aporten otras características
- Renombre: se evalúa subjetivamente valores como si la herramienta es usada por entidades importantes, si está en continuo desarrollo o si existe información más que suficiente para trabajar con ella.

Existe algún sistema que se ha analizado, pero se ha desestimado para esta comparativa final ya que no posee una implementación software. Este sistema es el MGRS (UTM), que únicamente ofrece el sistema de coordenadas y la formación de la rejilla.

Tras el completo análisis llevado a cabo, se tiene que a priori Uber H3 y Google S2 son las herramientas más completas y que más pueden adaptar su funcionamiento a las necesidades de este proyecto. Ambos permiten realizar consultas de manera jerárquica, truncando en el ID o etiqueta dependiendo el nivel de resolución que se quiera obtener. De este modo facilita la búsqueda de datos. Además, ofrecen un soporte y grandes empresas que las utilizan, son proyectos más que fiables y de reconocido valor. Aunque S2 puede ser más preciso en los espacios marítimos, ya que H3 está pensado para los transportes en tierra, no se requiere tanto nivel de precisión si no más velocidad de consulta y una jerarquización completa. Es por esto que se decide utilizar este sistema para el desarrollo de este trabajo.

En concreto, H3 permite realizar consultas sobre zonas cubiertas con sus celdas hexagonales muy parecidas a áreas circulares, tal y como se ha descrito en 2.5.1. Las funciones permiten relacionar distintas celdas, trabajar a diferentes niveles de resolución y realizar consultas como la distancia entre celdas. La sencillez y la precisión que ofrece es superior a la de S2 ya que la distancia entre centros de las celdas es siempre la misma y no varía como puede ser en los cuadrados el sistema de Google.

Otra razón más por la que se selecciona H3 es por la multitud de lenguajes en los que se encuentra disponible, además de las multitudes pruebas que existen alrededor al ser de código libre. Aunque S2 también es muy parecido en este aspecto, la orientación de la librería H3 está más orientada a trabajar con puntos que se mueven como son los VTC. En nuestro caso los barcos toman ese lugar, y la capacidad de asignar un valor jerárquico a las zonas y poder diferenciar entre la resolución a usar en el estrecho de Gibraltar, por ejemplo, y un punto cualquiera en el Océano Atlántico, da muchos puntos a este sistema.

Por otro lado, otros sistemas como Aurora e Informix están más orientados a gestionar bases de datos y aplicar este filtro de análisis espacial a la vez. Sin embargo, a pesar de que pueden ofrecer una interfaz mucho más sencilla, están orientados al mundo del negocio y a la gestión de venta de productos, con lo que el resto de sus funcionalidades no son necesarias en este trabajo.

Finalmente, se decide utilizar H3 de forma conjunta con un sistema gestor de bases de datos como es MySQL. La librería H3 contiene las funciones necesarias para comprobar su funcionamiento en el entorno marítimo y realizar diferentes experimentos en el desarrollo de este TFG. MySQL gestiona por su parte todos los datos que se necesiten durante el proyecto, además de facilitar el acceso a datos gracias a la creación de índices en los distintos campos como se ha mencionado en 2.6.1.

3 DESARROLLO DEL TFG

Después de haber analizado las diferentes herramientas disponibles y tras haber comprendido mejor cuales son las capacidades y ventajas que proporciona cada una de ellas se va a evaluar el sistema elegido. En este caso analizaremos el funcionamiento y aplicación del sistema H3 viendo cómo responde a las pruebas que el proyecto requiere, calificando sus capacidades y las mejoras que ofrece. Previamente se presenta el software que se va a emplear. Se explicará el procedimiento de instalación y funcionamiento básico del código. Se almacenarán los datos proporcionados por el COVAM en una base de datos MySQL para realizar el análisis posteriormente y analizar la respuesta a diferentes consultas.

Como se ha mencionado en el estado del arte, el COVAM recibe datos de múltiples fuentes y su gestión es de crucial importancia. Para el desarrollo de este trabajo se decide trabajar con datos AIS, que son más que suficientes para poder comprobar el funcionamiento del sistema elegido. En especial se necesita sobre todo las posiciones geográficas (latitudes y longitudes) de cada uno de los registros AIS. Más adelante también se requiere de datos como la fecha y hora del registro o las velocidades registradas por el sistema AIS para poder realizar diferentes experimentos.

3.1 Instalación de herramientas y software necesario

3.1.1 Sistema operativo: Linux Ubuntu

Para realizar todo el desarrollo del trabajo de la manera más eficiente se ha decidido emplear el sistema operativo Ubuntu de Linux en su versión 20.04.1 debido a las características que presenta. Sin entrar en las especificaciones técnicas, algunas de las ventajas que ofrece el sistema son [66]:

- Software libre: instalación totalmente libre y gratuita, completamente flexible y además en continua actualización por los desarrolladores. A pesar de no ser el más popular para uso en ordenadores personales, es el más utilizado como sistema operativo en servidores, Android o supercomputación. Posee una gran comunidad, que, en su mayoría, está compuesta por programadores e informáticos que dan buena fama de su empleo.
- Sencillez de empleo: a la hora de hablar la facilidad de empleo es de los sistemas más intuitivos, las aplicaciones de serie proporcionan las mismas facilidades que cualquier otro sistema. Además, instalarlo es tan fácil como descargarlo libremente de la página web oficial.
- Soporte para múltiples lenguajes de programación: es una opción muy asequible para programar, soporta la implementación de varios lenguajes como C, C++, PHP, HTML, CSS, Java, Python...
- Rendimiento y recursos: el uso de la terminal es muy intuitivo y fácil, se emplea en la enseñanza en muchos grados y módulos informáticos, sin ir más lejos aquí en la Escuela Naval en las asignaturas de Informática y Redes de Ordenadores.

Es por ello que para este TFG se decide utilizar este sistema, debido a su facilidad y las características nombradas que nos proporcionan un entorno muy cómodo para trabajar. Debido a que a priori los cálculos no requerirán mucha capacidad es factible la instalación de este sistema operativo en una máquina virtual en un equipo básico.

3.1.2 Lenguaje de programación: C

Debido a que el sistema de indexado elegido para los experimentos es H3 se elige como lenguaje de programación C. Como se ha visto previamente, H3 se encuentra disponible en varios idiomas, pero está desarrollado inicialmente en C y es donde más se ha comprobado el correcto funcionamiento del sistema. Se disponen de varios ejemplos escritos en este lenguaje y ayudaran a comprender fácilmente las funciones de las que dispone el sistema.

Además, este lenguaje es el más familiar para el alumnado ya que es el impartido en las diferentes asignaturas del ámbito informático. A la hora de programar y realizar las diferentes consultas, MySQL permite utilizar todos los datos desde C y trabajar con ellos mediante sus librerías y diferentes sentencias, un punto más a favor para utilizar este lenguaje.

3.1.3 Sistema de indexación espacial: H3

Tras el análisis de los diferentes sistemas se importa H3 desde su repositorio en GitHub [67]:

```
git clone https://github.com/uber/h3.git
```

Siguiendo los pasos que se especifican de forma sencilla en la web se descarga fácilmente todos los ficheros para utilizar el sistema.

Después de clonar el contenido se siguen los pasos y el sistema está listo para su uso. Para comprobar el funcionamiento la web propone diferentes formas, se puede acceder directamente a la función *geoToH3* que transforma las coordenadas de latitud y longitud en la etiqueta correspondiente a la celda H3 y según la resolución escogida. También se puede realizar desde un archivo en C compilando el propio ejemplo que trae la instalación H3 desde la terminal, es importante añadir las librerías matemática (*lm*) y del propio H3 (*h3*). Este ejemplo se detallada a continuación:

```
gcc examples/index.c -o example -lh3 -lm
./example
The index is: 8a2a1072b59ffff
Boundary vertex #0: 40.690059, -74.044152
Boundary vertex #1: 40.689908, -74.045062
Boundary vertex #2: 40.689271, -74.045341
Boundary vertex #3: 40.688785, -74.044711
Boundary vertex #4: 40.688936, -74.043801
Boundary vertex #5: 40.689573, -74.043521
Center coordinates: 40.689422, -74.04443
```

Como se puede observar se obtiene el índice correspondiente a la localización del archivo *index.c*. Además, indica la ubicación de los vértices de esa celda y el centro. En los ficheros que hemos instalado de H3 vienen diferentes ejemplos en C empleando diferentes funciones que pueden servir para utilizar en los experimentos. A la hora de utilizar el sistema en las diferentes pruebas que se realicen se explicará las funciones que ofrece el sistema, así como un detallado análisis del rendimiento.

3.1.4 SGDB: MySQL

Como se ha explicado en el estado del arte, se necesitará un sistema de almacenamiento que asocie todos los datos que vamos manejar y analizar. Para ello se elige el sistema de gestión de bases de datos relacional MySQL, debido también a su facilidad de empleo y la sencillez a la hora de acceder a ella desde la propia terminal del sistema operativo Ubuntu. Desde la propia terminal se instala el servidor de MySQL, y se crea un servidor donde realizaremos los primeros ejemplos antes de analizar todo el grueso de datos que proporcione el COVAM.

```
sudo apt install mysql-server
```

Se crea un servicio de MySQL en el ordenador personal desde la propia terminal de Ubuntu. Se crea *barcos* como primera base de datos a modo de ejemplo, creando la primera tabla *BARCOS* con algunos datos como ejemplo. Esta tabla solo se utiliza para familiarizarse con el entorno de trabajo y las sentencias de acceso a la base de datos, en ella solo introducimos algunos buques con los campos de *id*, *nombre*, *origen*, *destino*, *latitud* y *longitud* y *etiqueta* generada por el sistema, que se explica más adelante. La creación de esta tabla se realiza mediante un archivo *.sql* que contiene la propia información a excepción de la etiqueta, y mediante la sentencia *source* se importa el archivo y se insertan los datos en la tabla. En la Figura 3-1 se ven las sentencias en MySQL para poder visualizar desde la terminal la descripción de la tabla, así como todos los registros de los diferentes campos. En la columna Type se pueden ver los tipos de variable de cada campo, el id es una forma de ordenar todos los registros.

```
mysql> describe BARCOS;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int    | NO   | PRI | NULL    | auto_increment |
| NOMBRE | varchar(30) | YES |     | NULL    |               |
| ORIGEN | varchar(30) | YES |     | NULL    |               |
| DESTINO | varchar(30) | YES |     | NULL    |               |
| LATITUD | float  | YES |     | NULL    |               |
| LONGITUD | float  | YES |     | NULL    |               |
| ETIQUETA | varchar(30) | YES |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> SELECT * FROM BARCOS;
+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | ORIGEN | DESTINO | LATITUD | LONGITUD | ETIQUETA |
+-----+-----+-----+-----+-----+-----+
| 1 | ELCANO | ESPAÑA | EEUU    | 42.4578 | 56.4748 | 8f2c5b776cc920d |
| 2 | F101   | ESPAÑA | EEUU    | 38.4548 | 45.4877 | 8f2c1c810560d1a |
| 3 | F103   | ESPAÑA | AFRICA  | 47.4578 | 165.475 | 8f16d47a1144621 |
| 4 | F104   | ESPAÑA | SUECIA  | 75.88   | -89.45  | 8f0254b2a49938b |
| 5 | F105   | ESPAÑA | DINAMARCA | 12.154 | 64.475 | 8f6341acdc002b2 |
| 6 | TORALLA | ESPAÑA | CHINA   | 47.4578 | 111     | 8f2463d0da64844 |
| 7 | FORMENTOR | ESPAÑA | HAWAI  | -47.4   | -87.88 | 8fcf437714d8589 |
| 8 | HESPÉRIDES | ESPAÑA | ANTARTIDA | 47.4578 | 165.475 | 8f16d47a1144621 |
| 9 | FUROR   | ESPAÑA | MÁLAGA  | 45.74   | 112.78 | 8f2441840663cea |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

Figura 3-1. Ejemplo de tabla en MySQL

Tras la familiarización con el uso de MySQL en la consola, se reciben dos volcados de bases de datos del COVAM para poder trabajar con datos reales. Se reciben dos archivos, *dump_aismediterraneo.zip* y *dump_historico.zip*, que directamente contienen los datos en formato *.sql*, preparados para su descarga en el propio servidor del TFG. Se crea una base de datos nueva *tfg* dentro del propio servidor y se ejecutan los dos archivos para crear las tablas correspondientes *aismediterraneo* e *historico*. El primer archivo almacena numerosos registros AIS por buque de los campos que se ven en Figura 3-2, como son *mmsi*, *imo*, *sog*, *longitude*, *latitude*, *cog*, *msgtimestamp*, *source*, *callsign* y *name*, en conjunto un total de 178 MB sobre el movimiento de buques en el Mediterráneo en el día 11 de diciembre de 2019. Esta tabla será la que será utilizada para realizar los experimentos y el análisis.

```
mysql> describe aismediterraneo;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | bigint    | YES  |     | NULL    |       |
| mmsi      | int       | YES  |     | NULL    |       |
| imo       | int       | YES  |     | NULL    |       |
| sog       | int       | YES  |     | NULL    |       |
| longitude  | double    | YES  |     | NULL    |       |
| latitude   | double    | YES  |     | NULL    |       |
| cog       | double    | YES  |     | NULL    |       |
| msgtimestamp | varchar(25) | YES  | UNI | NULL    |       |
| source     | varchar(25) | YES  |     | NULL    |       |
| callsign   | varchar(25) | YES  |     | NULL    |       |
| name      | varchar(25) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.30 sec)
```

Figura 3-2. Descripción tabla *aismediterraneo*.

```
mysql> describe historico;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int       | NO   | PRI | NULL    |       |
| msgid      | varchar(45) | YES  |     | NULL    |       |
| repeatindicator | varchar(45) | YES  |     | NULL    |       |
| mmsi      | int       | YES  |     | NULL    |       |
| imo       | varchar(15) | YES  |     | NULL    |       |
| callsign   | varchar(15) | YES  |     | NULL    |       |
| name      | varchar(45) | YES  |     | NULL    |       |
| navstate  | int       | YES  |     | NULL    |       |
| rot       | int       | YES  |     | NULL    |       |
| sog       | int       | YES  |     | NULL    |       |
| posacc    | int       | YES  |     | NULL    |       |
| longitude  | float     | YES  |     | NULL    |       |
| latitude   | float     | YES  |     | NULL    |       |
| position  | varchar(45) | YES  |     | NULL    |       |
| cog       | float     | YES  |     | NULL    |       |
| trueheading | varchar(45) | YES  |     | NULL    |       |
| shiptype  | varchar(45) | YES  |     | NULL    |       |
| dimensiona | int       | YES  |     | NULL    |       |
| dimensionb | int       | YES  |     | NULL    |       |
| dimensionc | int       | YES  |     | NULL    |       |
| dimensiond | int       | YES  |     | NULL    |       |
| eta       | varchar(45) | YES  |     | NULL    |       |
| draught   | float     | YES  |     | NULL    |       |
| destination | varchar(45) | YES  |     | NULL    |       |
| country   | varchar(45) | YES  |     | NULL    |       |
| msgtimestamp | timestamp | YES  |     | NULL    |       |
| bmn_src   | varchar(15) | YES  |     | NULL    |       |
| source     | varchar(15) | YES  |     | NULL    |       |
| grupo     | varchar(15) | YES  |     | NULL    |       |
| propietario | varchar(15) | YES  |     | NULL    |       |
| formatalfa | varchar(15) | YES  |     | NULL    |       |
| operacion | varchar(15) | YES  |     | NULL    |       |
| categorizacion | varchar(15) | YES  |     | NULL    |       |
| vulnerabilityassessment | varchar(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
34 rows in set (0.00 sec)
```

Figura 3-3. Descripción tabla *histórico*.

3.1.5 Visualización geoespacial: Google Earth y Keepler.gl

Existen múltiples herramientas disponibles en la red que pueden permitir visualizar un mapa terrestre con datos geográficos y figuras geométricas como puntos, líneas y polígonos. La propia documentación de Uber H3 indica una serie de aplicaciones con las que ellos mismos trabajan. Sin embargo, mediante librerías extra de H3 se pueden formar archivos KML para su posterior representación. Mediante algunas funciones del sistema se pueden generar estos archivos que almacenan la información necesaria para poder representar los resultados como puntos geográficos o las mismas celdas que genere H3.

Google Earth es una de las aplicaciones que fácilmente soporta este formato de archivo y representa la información contenida. Es de uso sencillo y el único inconveniente es generar archivos que no sean tan grandes como para que el propio servidor no soporte su manejo.

Otro sistema de visualización es Keepler.gl, nombrado anteriormente, es una de las herramientas que la propia documentación de H3 nombra. En su web [68] se explica que el sistema de análisis espacial posee de serie la capacidad de representar los índices o celdas generados por H3. Esta herramienta es muy utilizada para la representación de datos geoespaciales y su análisis por su capacidad de aplicar diferentes capas que facilitan estas operaciones.

Estas herramientas pueden ser utilizadas en algunos casos para ofrecer ese elemento visual y demostrativo que ayude a comprender el desarrollo del trabajo.

3.2 Análisis de datos geoespaciales

Ya se tiene el software necesario, así como los datos para comenzar a realizar los diferentes experimentos. Primero se realizan diferentes ejemplos para comprobar el correcto funcionamiento del sistema, comprender el sistema de asignación de etiquetas y familiarizarse con las secuencias básicas de MySQL que serán necesarias para el resto del desarrollo. A continuación, se emplearán estas funciones para realizar diferentes propuestas de análisis, viendo y analizando como el sistema actúa sobre el rendimiento del sistema.

3.2.1 Buques dentro de un área determinada

En este primer experimento se evalúa como responde el sistema ante la consulta a la tabla *aismediterraneo* sobre que buques se encuentran dentro de un área circular determinada. Para ello primero se realiza la consulta sin ningún índice ni etiqueta. Se puede hacer directamente desde la propia terminal y ejecutando la secuencia en MySQL, o desde un programa en C. El área de interés que se implementa es la de un círculo de radio r sobre un centro con coordenadas de latitud y longitud. En este caso, como *aismediterraneo* contiene posiciones en el entorno del Mar Mediterráneo, se establecen las coordenadas P (37 N, 1° E) al suroeste de Cartagena, como centro de coordenadas del área del interés, que en este caso será un círculo de radio 7 Km. El radio se escoge de este orden ya que el diámetro (14 kilómetros) se aproxima a la distancia de 12 millas náuticas coincidiendo con el alcance visual medio en el horizonte en alta mar. De esta manera se asegura que se está buscando en un área donde todo buque puede ver a otro que también se encuentre dentro de esa zona por encontrarse en el alcance visual y así, siendo más interesante la búsqueda de buques que estén en contacto cercano en esa área. Se realiza un análisis sin utilizar el sistema para posteriormente compararlo con los diferentes análisis utilizando H3.

1. Análisis sin sistema de indexado

Para esta primera consulta se utiliza la fórmula que calcula la distancia entre dos puntos de la superficie terrestre, y así compararlos uno a uno si se encuentran a la distancia requerida. Además, se ordenan de menor distancia al punto a mayor. Se realiza la consulta en la tabla *aismediterraneo* desde la propia consola escribiendo la siguiente sentencia:

```

SELECT name, latitude, longitude, (6371 * acos (cos (radians (37))
* cos (radians(latitude) )
* cos (radians (longitude) - radians (1)) + sin (radians (37)) *
sin(radians(latitude)))) AS distance
FROM aismediterraneo HAVING distance < 7 ORDER BY distance;

```

Obteniendo los resultados que se muestran en Figura 3-4, donde se presentan los nombres de los buques que han registrado una posición que se encuentra a menos de 7 Km del centro de coordenadas que hemos insertado y sus respectivas coordenadas. En este caso uno de los buques presenta varios registros ya que ha estado reportando posiciones en la zona en varias ocasiones. El tiempo empleado en este primer proceso es de 10,46 seg. Se ejecuta dos veces más el mismo proceso obteniendo unos tiempos de 9,27 y 9,51 seg.

name	latitude	longitude	distance
JALADI	37.0268821716309	1.01106667518616	3.14651781465754
STOLT SYCAMORE	37.0296859741211	0.989400029182434	3.4324740737227604
CLIPPER VENUS	36.947696685791	1.01875162124634	6.049722234456824
NORDLOTUS	36.9738883972168	1.06009495258331	6.07619698744873
CLIPPER VENUS	36.9493179321289	1.02579832077026	6.083751972569857
CLIPPER VENUS	36.9460716247559	1.01166999340057	6.085517578507139
CLIPPER VENUS	36.9495697021484	1.02691495418549	6.096044746098219
CLIPPER VENUS	36.9455184936523	1.00929999351501	6.114142709562201
CLIPPER VENUS	36.9447250366211	1.00573170185089	6.167350881902161
CLIPPER VENUS	36.943904876709	1.00216662883759	6.240462139133009
CLIPPER VENUS	36.9434814453125	1.00037837028503	6.284666433365508
CLIPPER VENUS	36.9522438049316	1.03868162631989	6.325028171121686
CLIPPER VENUS	36.9527587890625	1.04096162319183	6.390142911515976
CLIPPER VENUS	36.9418678283691	0.99321836233139	6.492017938767378
CLIPPER VENUS	36.9404907226562	0.987208306789398	6.714001194804437
CLIPPER VENUS	36.939697265625	0.983628332614899	6.861286339152194

16 rows in set (10.46 sec)

Figura 3-4. Resultado consulta MySQL a puntos dentro de un área determinada.

2. Análisis H3 sin índice

A continuación, se comprueba cómo reaccionaría el sistema a la hora de realizar esta consulta mediante una resolución de nivel 10 en primer lugar, para luego pasar a comprobarlo con un nivel 9 y 8. Como se ha visto en la Tabla 2-3 en 2.5.1, en un nivel de resolución 10 se tiene un lado de hexágono de aproximadamente 66 metros, que a priori parece una resolución bastante considerable ya que la eslora de un buque mercante medio suele encontrarse entre 150 y 200 metros. Sin embargo, es evidente que, al realizar el análisis por celdas se está cometiendo un error de cálculo que depende de la resolución utilizada y el área que se abarca.

En este experimento, en primer lugar, se analizan todos los registros y se almacena la etiqueta formada en el proceso en una nueva tabla llamada *etiquetas*. Para este proceso hay que acceder a la base de datos desde C y convertir cada una de las posiciones geográficas en etiquetas H3 mediante las funciones que ofrece esta librería. Este proceso es sencillo, pero dura un tiempo considerable debido a la gran cantidad de información de la tabla *aismediterraneo*. El programa *etiqueta10.c* analiza todas las latitudes y longitudes de los registros y genera la etiqueta de resolución 10. Además, almacenamos también un ID que a priori utilizaremos para analizar etiqueta a etiqueta, es decir, sin nombrar la propia

etiqueta como índice de la tabla lo que agilizará la búsqueda y el nombre del buque. Para ahorrar tiempo, en el mismo programa se insertarán también los valores de las etiquetas de nivel 8 y 9, que serán con los que realizaremos el experimento (ver Anexo I: etiqueta10.c).

```
mysql> describe etiquetas;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| nombre    | varchar(25)   | YES  |     | NULL    |                |
| etiqueta10 | varchar(30)  | YES  |     | NULL    |                |
| etiqueta9  | varchar(30)  | YES  |     | NULL    |                |
| etiqueta8  | varchar(30)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Figura 3-5. Descripción tabla *etiquetas*.

Ahora se obtiene el índice correspondiente al centro de coordenadas que se quiere examinar mediante la propia función de H3. Para ello se escribe un pequeño programa en C que utilice las funciones de H3, produciendo la etiqueta y además los vecinos, es decir, las celdas que se encuentran a su alrededor. El objetivo es aproximar el radio que se quiere analizar a un número de saltos, es decir de cantidad de vecinos que analizamos. En la Figura 3-6 se muestra la ejecución de *radio_exp1.c*, dónde se obtiene la etiqueta correspondiente al punto de interés. Como se ve, con un nivel de resolución 10 el centro de coordenadas corresponde prácticamente con las coordenadas del punto de estudio. El objetivo es encontrar un nivel de saltos k (Figura 3-7) que es necesario para analizar el área en cuestión. Cuanto mayor sea este número, mayor es el nº de celdas que se encuentran en el rango. Uno de los inconvenientes de H3 es que este valor solo puede ser representado con este número de saltos ya que la librería no contiene ninguna función que haga corresponder el número óptimo de saltos directamente para una distancia dada.

```
victor@victor-VirtualBox:~$ gcc radio_exp1.c -lh3 -lm -lmysqlclient -o radio
victor@victor-VirtualBox:~$ ./radio
Inserte centro de coordenadas del área de interés,
latitud,
longitud,
resolución
37
1
10
Índice: 8a382cab2077fff
Vertices #0: 37.000187, 0.999077
Vertices #1: 36.999495, 0.998927
Vertices #2: 36.999063, 0.999560
Vertices #3: 36.999324, 1.000344
Vertices #4: 37.000017, 1.000495
Vertices #5: 37.000448, 0.999861
Centro de coordenadas: 36.999756, 0.999711
Vecinos:
8a382cab2077fff
8a382cab202ffff
8a382cab200ffff
8a382cab2057fff
8a382cab2047fff
8a382cab2067fff
8a382cab215ffff
```

Figura 3-6. Vecinos para $k=1$ y vértices de la etiqueta correspondiente al área del experimento.

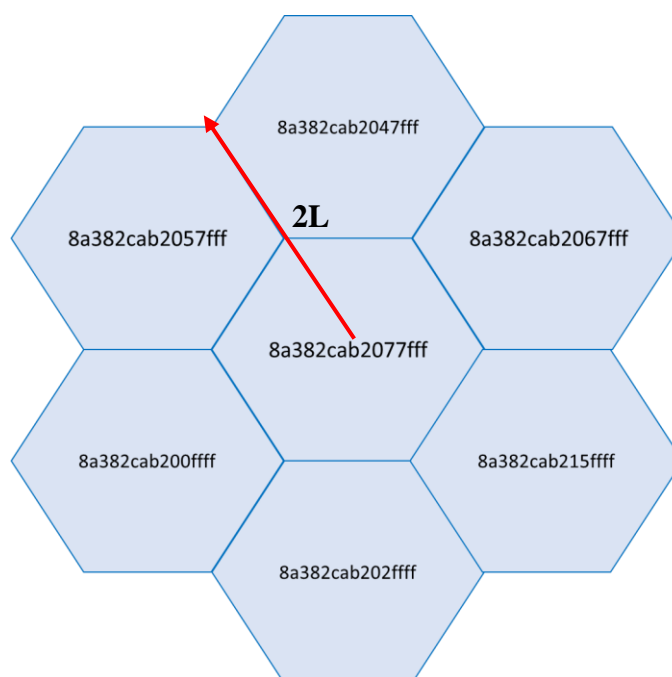


Figura 3-7. Vecinos de un salto ($k=1$) de la celda correspondiente a 36°N , 1°E .

Para calcular k se tiene que el radio efectivo que se está analizando se puede determinar como la distancia desde el origen o centro de coordenadas ($8a382cab2077fff$ en el ejemplo) al vértice común externo de los vecinos, como se puede ver en la Figura 3-7. Esto asegura que las celdas que se analicen y se encuentren más lejos del centro de coordenadas sean las únicas que pueden estar fuera del área original (el círculo real), y por tanto que sean posiciones que no se quieran tener en cuenta, pero maximizando el área examinada, que a priori no es problema. Como son hexágonos regulares esta distancia es igual a dos veces el lado de la celda. Es decir que para k igual a 1, el radio analizado sería aproximadamente de 132 metros. Para k igual a 2 sería de 264 metros y así consecutivamente.

Para el radio de 7 Km se tiene entonces un k de valor aproximado 54. Se ejecuta el mismo programa anterior, pero en este caso introducimos el valor necesario, y así se obtienen todas las celdas que componen el área de estudio. Los resultados se introducen en nueva tabla llamada *vecinos1*. Tras este proceso, mediante una única consulta MySQL se comparan ambas columnas de las dos tablas y devuelve dónde coinciden las etiquetas de ese nivel de resolución con los vecinos creados. Esta consulta compara ambas consultas y devuelve como resultado los registros donde coinciden etiquetas.

```
SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta10 =
etiqueta;
```

La Figura 3-8 muestra el resultado del análisis. Se puede ver la secuencia que compara ambas columnas e imprime por pantalla los barcos cuya etiqueta a nivel de resolución 10 es igual a la etiqueta generada por los vecinos del punto P . El tiempo de ejecución ronda los 18 seg (18'53, 18'65 y 18'58 en diferentes pruebas), siendo superior debido a la cantidad de etiquetas que se han generado como vecinos. Además, se puede observar que solo se obtienen 12 de los 16 resultados que hay, esto se debe al error producido por las diferencias entre realizar la búsqueda en un círculo con latitud y longitud al hacerlo con el sistema de celdas hexagonales. Este error producido se estudia más adelante. Las funciones H3 que ayudan a generar este sistema de vecinos de celdas no deja de ser una aproximación a un círculo real, se pueden llegar a analizar zonas que no son de interés en ese momento y dejar otras que sí lo son.


```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta10 = etiqueta;
```

id	nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
3020262	JALADI	8a382cab0bafffff	89382cab0bbffff	88382cab0bfffff	8a382cab0bafffff
3042328	STOLT SYCAMORE	8a382cab47a7ffff	89382cab47bffff	88382cab47fffff	8a382cab47a7ffff
2968290	CLIPPER VENUS	8a382ca92127ffff	89382ca9213ffff	88382ca929fffff	8a382ca92127ffff
2968329	CLIPPER VENUS	8a382ca92c37ffff	89382ca92c3ffff	88382ca92dfffff	8a382ca92c37ffff
2968343	CLIPPER VENUS	8a382ca92a17ffff	89382ca92a3ffff	88382ca92bfffff	8a382ca92a17ffff
2968352	CLIPPER VENUS	8a382ca92b8ffff	89382ca92bbffff	88382ca92bfffff	8a382ca92b8ffff
2968387	CLIPPER VENUS	8a382ca92db7fff	89382ca92dbffff	88382c065bfffff	8a382ca92db7fff
2968390	CLIPPER VENUS	8a382ca92c2ffff	89382ca92c3ffff	88382ca92dfffff	8a382ca92c2ffff
2189037	NORDLOTUS	8a382ca91b9ffff	89382ca91bbffff	88382ca91bfffff	8a382ca91b9ffff
2968241	CLIPPER VENUS	8a382c065a87fff	89382c065abffff	88382c065bfffff	8a382c065a87fff
2968261	CLIPPER VENUS	8a382ca93d47fff	89382ca93d7ffff	88382ca93dfffff	8a382ca93d47fff
2968278	CLIPPER VENUS	8a382c065af7fff	89382c065afffff	88382c065bfffff	8a382c065af7fff

12 rows in set (18.53 sec)

Figura 3-8. Resultado de análisis con resolución de nivel 10.

Se realiza el mismo procedimiento, en este caso con unas resoluciones de 8 y 9. Se tiene para 7 Km unos valores de k igual a 8 y 21 respectivamente. Con nivel de resolución 9 se consigue un tiempo de 7'28 segundos, y con el nivel inmediatamente inferior un tiempo de 7'08 segundos (Figura 3-9, Figura 3-10). En estos casos se está reduciendo el tiempo de respuesta, sin embargo, cabe la posibilidad de que alguno de los resultados comparta celda, pero en este experimento no es inconveniente.

```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta9 = etiqueta;
```

id	nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
2189037	NORDLOTUS	8a382ca91b9ffff	89382ca91bbffff	88382ca91bfffff	89382ca91bbffff
2968241	CLIPPER VENUS	8a382c065a87fff	89382c065abffff	88382c065bfffff	89382c065abffff
2968251	CLIPPER VENUS	8a382ca938b7fff	89382ca938bffff	88382ca939fffff	89382ca938bffff
2968261	CLIPPER VENUS	8a382ca93d47fff	89382ca93d7ffff	88382ca93dfffff	89382ca93d7ffff
2968264	CLIPPER VENUS	8a382c06510ffff	89382c06513ffff	88382c0651fffff	89382c06513ffff
2968278	CLIPPER VENUS	8a382c065af7fff	89382c065afffff	88382c065bfffff	89382c065afffff
2968290	CLIPPER VENUS	8a382ca92127fff	89382ca9213ffff	88382ca929fffff	89382ca9213ffff
2968300	CLIPPER VENUS	8a382c065c57fff	89382c065c7ffff	88382c065dfffff	89382c065c7ffff
2968329	CLIPPER VENUS	8a382ca92c37fff	89382ca92c3ffff	88382ca92dfffff	89382ca92c3ffff
2968343	CLIPPER VENUS	8a382ca92a17fff	89382ca92a3ffff	88382ca92bfffff	89382ca92a3ffff
2968352	CLIPPER VENUS	8a382ca92b8ffff	89382ca92bbffff	88382ca92bfffff	89382ca92bbffff
2968365	CLIPPER VENUS	8a382c065c17fff	89382c065c3ffff	88382c065dfffff	89382c065c3ffff
2968387	CLIPPER VENUS	8a382ca92db7fff	89382ca92dbffff	88382c065bfffff	89382ca92dbffff
2968390	CLIPPER VENUS	8a382ca92c2ffff	89382ca92c3ffff	88382ca92dfffff	89382ca92c3ffff
3020262	JALADI	8a382cab0bafffff	89382cab0bbffff	88382cab0bfffff	89382cab0bbffff
3042328	STOLT SYCAMORE	8a382cab47a7ffff	89382cab47bffff	88382cab47fffff	89382cab47bffff

16 rows in set (7.28 sec)

Figura 3-9. Resultado de análisis con nivel de resolución 9.

```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta8 = etiqueta;
```

id	nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
2189037	NORDLOTUS	8a382ca91b9ffff	89382ca91bbffff	88382ca91bfffff	88382ca91bfffff
2968241	CLIPPER VENUS	8a382c065a87fff	89382c065abffff	88382c065bfffff	88382c065bfffff
2968251	CLIPPER VENUS	8a382ca938b7fff	89382ca938bffff	88382ca939fffff	88382ca939fffff
2968261	CLIPPER VENUS	8a382ca93d47fff	89382ca93d7ffff	88382ca93dfffff	88382ca93dfffff
2968264	CLIPPER VENUS	8a382c06510ffff	89382c06513ffff	88382c0651fffff	88382c0651fffff
2968278	CLIPPER VENUS	8a382c065af7fff	89382c065afffff	88382c065bfffff	88382c065bfffff
2968290	CLIPPER VENUS	8a382ca92127fff	89382ca9213ffff	88382ca929fffff	88382ca929fffff
2968300	CLIPPER VENUS	8a382c065c57fff	89382c065c7ffff	88382c065dfffff	88382c065dfffff
2968329	CLIPPER VENUS	8a382ca92c37fff	89382ca92c3ffff	88382ca92dfffff	88382ca92dfffff
2968343	CLIPPER VENUS	8a382ca92a17fff	89382ca92a3ffff	88382ca92bfffff	88382ca92bfffff
2968352	CLIPPER VENUS	8a382ca92b8ffff	89382ca92bbffff	88382ca92bfffff	88382ca92bfffff
2968365	CLIPPER VENUS	8a382c065c17fff	89382c065c3ffff	88382c065dfffff	88382c065dfffff
2968387	CLIPPER VENUS	8a382ca92db7fff	89382ca92dbffff	88382c065bfffff	88382c065bfffff
2968390	CLIPPER VENUS	8a382ca92c2ffff	89382ca92c3ffff	88382ca92dfffff	88382ca92dfffff
3020262	JALADI	8a382cab0baffff	89382cab0bbffff	88382cab0bfffff	88382cab0bfffff
3042328	STOLT SYCAMORE	8a382cab47a7fff	89382cab47bffff	88382cab47fffff	88382cab47fffff

16 rows in set (7.08 sec)

Figura 3-10. Resultado de análisis con nivel de resolución 8.

3. Análisis H3 con índice MySQL

A continuación, para optimizar la búsqueda se establece un índice en la tabla *etiquetas* mediante una sentencia en la propia consola en MySQL, haciendo que la columna de interés, ya sea etiqueta 8, 9 o 10, se convierta en un índice ordinario. El índice es una estructura interna de la base de datos, que ayuda a la hora de realizar cualquier búsqueda que implique el campo en el que se ha establecido el índice sea mucho más rápida. Estos índices son muy útiles en los campos que siguen alguna estructura sintáctica, permitiendo mejorar el rendimiento de búsqueda. También se elimina el anterior índice *id*, que se había generado para recorrer la tabla en un orden determinado. Se realizan las mismas pruebas que se han realizado en el punto anterior, esta vez con el índice en cada una de las columnas.

```
ALTER TABLE etiquetas ADD INDEX etiqueta (etiqueta10);
```

Tras la generación del índice, se realiza la misma consulta realizada anteriormente obteniendo los resultados de la Figura 3-11, obteniendo los resultados de 0'48, 0'11 y 0'02 segundos. Estos resultados son muy satisfactorios ya que indican una mejora bastante significativa en cuanto al rendimiento en las consultas. Los tiempos van disminuyendo conforme el nivel de resolución es menor ya que el número de etiquetas objetivo de la consulta será menor al abarcar mayor área.

```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta10 = etiqueta;
```

nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
JALADI	8a382cab0bafffff	89382cab0bbfffff	88382cab0bfffff	8a382cab0bafffff
STOLT SYCAMORE	8a382cab47a7fff	89382cab47bfffff	88382cab47fffff	8a382cab47a7fff
CLIPPER VENUS	8a382ca92127fff	89382ca9213fffff	88382ca929fffff	8a382ca92127fff
CLIPPER VENUS	8a382ca92b8fffff	89382ca92bbfffff	88382ca92bfffff	8a382ca92b8fffff
CLIPPER VENUS	8a382ca92a17fff	89382ca92a3fffff	88382ca92bfffff	8a382ca92a17fff
CLIPPER VENUS	8a382ca92c2fffff	89382ca92c3fffff	88382ca92dfffff	8a382ca92c2fffff
CLIPPER VENUS	8a382ca92c37fff	89382ca92c3fffff	88382ca92dfffff	8a382ca92c37fff
CLIPPER VENUS	8a382ca92db7fff	89382ca92dbfffff	88382c065bfffff	8a382ca92db7fff
CLIPPER VENUS	8a382c065af7fff	89382c065afffffff	88382c065bfffff	8a382c065af7fff
CLIPPER VENUS	8a382c065a87fff	89382c065abfffff	88382c065bfffff	8a382c065a87fff
CLIPPER VENUS	8a382ca93d47fff	89382ca93d7fffff	88382ca93dfffff	8a382ca93d47fff
NORDLOTUS	8a382ca91b9fffff	89382ca91bbfffff	88382ca91bfffff	8a382ca91b9fffff

12 rows in set (0.48 sec)

```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta9 = etiqueta;
```

nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
JALADI	8a382cab0bafffff	89382cab0bbfffff	88382cab0bfffff	89382cab0bbfffff
STOLT SYCAMORE	8a382cab47a7fff	89382cab47bfffff	88382cab47fffff	89382cab47bfffff
CLIPPER VENUS	8a382ca92b8fffff	89382ca92bbfffff	88382ca92bfffff	89382ca92bbfffff
CLIPPER VENUS	8a382ca92a17fff	89382ca92a3fffff	88382ca92bfffff	89382ca92a3fffff
CLIPPER VENUS	8a382ca93d47fff	89382ca93d7fffff	88382ca93dfffff	89382ca93d7fffff
CLIPPER VENUS	8a382ca938b7fff	89382ca938bfffff	88382ca939fffff	89382ca938bfffff
CLIPPER VENUS	8a382c065c17fff	89382c065c3fffff	88382c065dfffff	89382c065c3fffff
CLIPPER VENUS	8a382c065c57fff	89382c065c7fffff	88382c065dfffff	89382c065c7fffff
CLIPPER VENUS	8a382c06510fffff	89382c06513fffff	88382c0651fffff	89382c06513fffff
CLIPPER VENUS	8a382c065a87fff	89382c065abfffff	88382c065bfffff	89382c065abfffff
CLIPPER VENUS	8a382c065af7fff	89382c065afffffff	88382c065bfffff	89382c065afffffff
CLIPPER VENUS	8a382ca92db7fff	89382ca92dbfffff	88382c065bfffff	89382ca92dbfffff
CLIPPER VENUS	8a382ca92c37fff	89382ca92c3fffff	88382ca92dfffff	89382ca92c3fffff
CLIPPER VENUS	8a382ca92c2fffff	89382ca92c3fffff	88382ca92dfffff	89382ca92c3fffff
CLIPPER VENUS	8a382ca92127fff	89382ca9213fffff	88382ca929fffff	89382ca9213fffff
NORDLOTUS	8a382ca91b9fffff	89382ca91bbfffff	88382ca91bfffff	89382ca91bbfffff

16 rows in set (0.11 sec)

```
mysql> SELECT * FROM etiquetas INNER JOIN vecinos1 WHERE etiqueta8 = etiqueta;
```

nombre	etiqueta10	etiqueta9	etiqueta8	etiqueta
JALADI	8a382cab0bafffff	89382cab0bbfffff	88382cab0bfffff	88382cab0bfffff
STOLT SYCAMORE	8a382cab47a7fff	89382cab47bfffff	88382cab47fffff	88382cab47fffff
CLIPPER VENUS	8a382ca92127fff	89382ca9213fffff	88382ca929fffff	88382ca929fffff
CLIPPER VENUS	8a382c065a87fff	89382c065abfffff	88382c065bfffff	88382c065bfffff
CLIPPER VENUS	8a382c065af7fff	89382c065afffffff	88382c065bfffff	88382c065bfffff
CLIPPER VENUS	8a382ca92db7fff	89382ca92dbfffff	88382c065bfffff	88382c065bfffff
CLIPPER VENUS	8a382ca92c37fff	89382ca92c3fffff	88382ca92dfffff	88382ca92dfffff
CLIPPER VENUS	8a382ca92c2fffff	89382ca92c3fffff	88382ca92dfffff	88382ca92dfffff
CLIPPER VENUS	8a382ca92a17fff	89382ca92a3fffff	88382ca92bfffff	88382ca92bfffff
CLIPPER VENUS	8a382ca92b8fffff	89382ca92bbfffff	88382ca92bfffff	88382ca92bfffff
CLIPPER VENUS	8a382c065c57fff	89382c065c7fffff	88382c065dfffff	88382c065dfffff
CLIPPER VENUS	8a382c065c17fff	89382c065c3fffff	88382c065dfffff	88382c065dfffff
CLIPPER VENUS	8a382c06510fffff	89382c06513fffff	88382c0651fffff	88382c0651fffff
CLIPPER VENUS	8a382ca93d47fff	89382ca93d7fffff	88382ca93dfffff	88382ca93dfffff
CLIPPER VENUS	8a382ca938b7fff	89382ca938bfffff	88382ca939fffff	88382ca939fffff
NORDLOTUS	8a382ca91b9fffff	89382ca91bbfffff	88382ca91bfffff	88382ca91bfffff

16 rows in set (0.02 sec)

Figura 3-11. Resultados con índice en columnas etiqueta10, 9 y 8.

Para comprobar que el funcionamiento es correcto y que se está aprovechando al máximo la velocidad de consulta que proporciona MySQL y el índice generado se realiza el mismo experimento de forma distinta. Se escribe otro código en C, *radio_exp2.c*, el cual analiza también todos los vecinos según el punto de interés, resolución requerida y número de saltos que aproxime el área. De esta manera, se ejecuta el programa y vemos como se obtienen los mismos resultados. Implementamos además una función como cronómetro que imprima el tiempo transcurrido únicamente en la consulta MySQL. En Figura 3-12 se muestra en primer lugar el resultado tras introducir coordenadas, nivel de resolución y número de saltos manualmente. En este caso se obtiene, para una resolución de nivel 10, un tiempo aproximado de 0'887 segundos.

```
victor@victor-VirtualBox:~$ ./radio2
Inserte centro de coordenadas del área de interés,
latitud,
longitud,
resolución
37
1
10
Índice: 8a382cab2077fff
Centro de coordenadas: 36.999756, 0.999711
Indique nº saltos=
54
Celdas ocupadas en el área:
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382c065a87fff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382c065af7fff
Nombre: NORDLOTUS
Etiqueta en área: 8a382ca91b9ffff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92127fff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92a17fff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92b8ffff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92c2ffff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92c37fff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca92db7fff
Nombre: CLIPPER VENUS
Etiqueta en área: 8a382ca93d47fff
Nombre: JALADI
Etiqueta en área: 8a382cab0baffff
Nombre: STOLT SYCAMORE
Etiqueta en área: 8a382cab47a7fff
Transcurrido: 887 miliseg

victor@victor-VirtualBox:~$ ./radio3
Coordenadas punto aleatorio:
Latitud: 36.111696, longitud: 24.220392
Inserte resolución: 10
Índice: 8a3f762498b7fff
Centro de coordenadas: 36.111791, 24.219706
Nº saltos aleatorio: 58
Celdas ocupadas en el área:
Nombre: VASILIKI
Etiqueta en área: 8a3f70510ae7fff
Nombre: SAGA
Etiqueta en área: 8a3f70510bb7fff
Nombre: ULTRA PANACHE
Etiqueta en área: 8a3f7051142ffff
Nombre: SAGA
Etiqueta en área: 8a3f7051198ffff
Nombre: VASILIKI
Etiqueta en área: 8a3f7051280ffff
Nombre: MV HALISDEMIR
Etiqueta en área: 8a3f70512a37fff
Nombre: VASILIKI
Etiqueta en área: 8a3f70512c57fff
Nombre: MV HALISDEMIR
Etiqueta en área: 8a3f705166effff
Nombre: VASILIKI
Etiqueta en área: 8a3f705a521ffff
Nombre: CHRISTOS XXXIV
Etiqueta en área: 8a3f76249d57fff
Nombre: SAGA
Etiqueta en área: 8a3f7624bcd7fff
Nombre: CHRISTOS XXXIV
Etiqueta en área: 8a3f7624d047fff
Nombre: NISSOS SERIFOS
Etiqueta en área: 8a3f7625999ffff
Transcurrido: 970 miliseg
```

Figura 3-12. Resultados de *radio_exp2.c* (izq.) y *radio_exp3.c* (dcha.) para nivel de resolución 10.

Por último, con el objeto de comprobar la versatilidad operando en cualquier punto del Mediterráneo, se modifica ligeramente el programa anterior, generando en este caso las coordenadas del punto de estudio, así como el número de saltos, aleatoriamente. Se acotan la selección de números aleatorios entre 30,25°N y 45,8° N para la latitud y entre 5,61° W y 36,22°E para la longitud, además de generar también un número de saltos aleatorios. Así lo único que se introduce es el nivel de resolución, que debe corresponder con la sentencia MySQL que lleva intrínseca el propio código C *radio_exp3.c* (ver Anexo II: *radio_exp3.c*) en este caso un nivel de resolución igual a 10. Como se ve en Figura 3-12, se obtiene un tiempo aproximado de 0,97 segundos. Estos resultados no son comparables, ya que el tiempo dependerá del radio de búsqueda, es decir, del número de saltos.

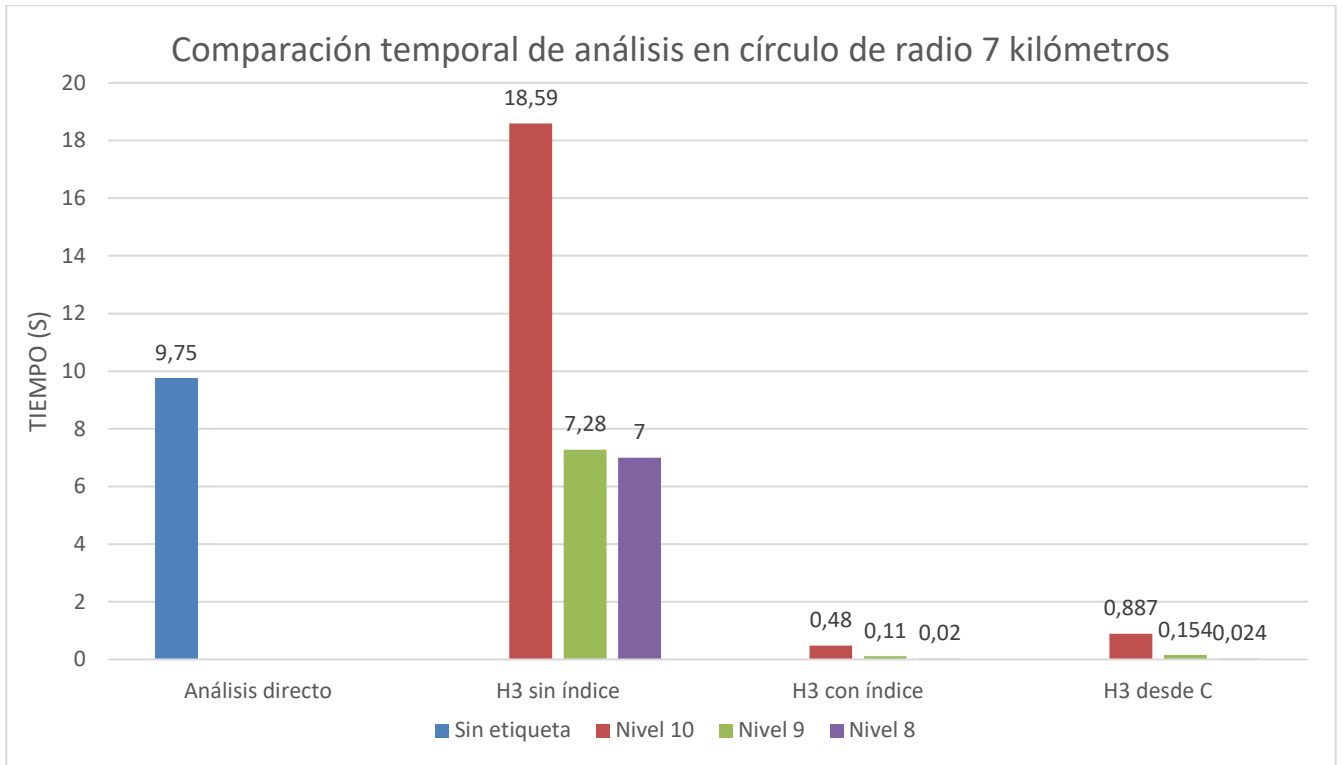


Figura 3-13. Comparación temporal de análisis en un área determinada.

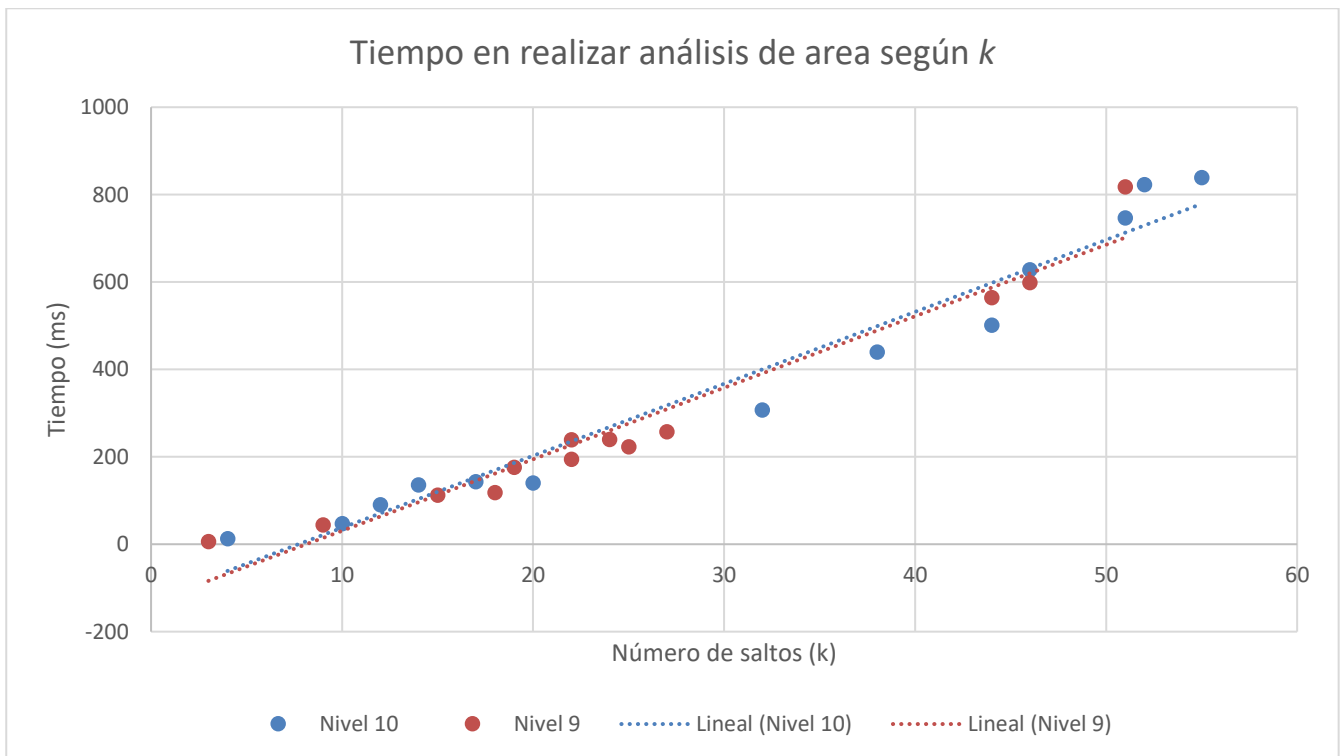


Figura 3-14. Tiempo de análisis, aleatorios, según k saltos.

En el gráfico de barras de la Figura 3-13 se comparan los tiempos transcurridos en el experimento según el método utilizando. Se puede ver como al aplicar el sistema H3 y crear el índice en la base de datos se obtiene una mejora de velocidad en los resultados en cuanto a tiempo de respuesta. El sistema proporciona una velocidad de obtención de resultados de hasta 20 veces superior a la realizada con las coordenadas geográficas, verificando la eficiencia del sistema. Como comprobación, se obtienen

también unos resultados similares al realizar la misma operación, pero implementando la sentencia MySQL desde C.

Por otro lado, en el gráfico de la Figura 3-14 se realizan diferentes pruebas con puntos geográficos y número de saltos totalmente aleatorios. De esta manera se puede ver que el tiempo transcurrido en la consulta no depende de la localización ni del número de barcos que se encuentren en la zona ni tampoco del nivel de resolución, sino del número de saltos, es decir, de la amplitud de área que se examine. Ya que cuanto mayor sea el rango de búsqueda de etiquetas, mayor tiempo tardará el sistema en comprobar las celdas que se encuentran dentro de éste.

4. Análisis con etiquetas truncadas

Una de las ventajas de Uber H3 es la capacidad jerárquica del sistema, esta ventaja es muy interesante para este proyecto como se ha comentado en el estado del arte. El Conocimiento del Entorno Marítimo puede requerir de un análisis más exacto en zonas de alto tráfico o de mayor interés como puede ser la Zona Económica Exclusiva ya que el interés es mucho mayor que en otras zonas. Se tiene que las etiquetas de un nivel determinado comparten los primeros valores menos el que determina el nivel de la celda (el segundo) y los dos último que corresponden al identificador dentro de ese nivel.

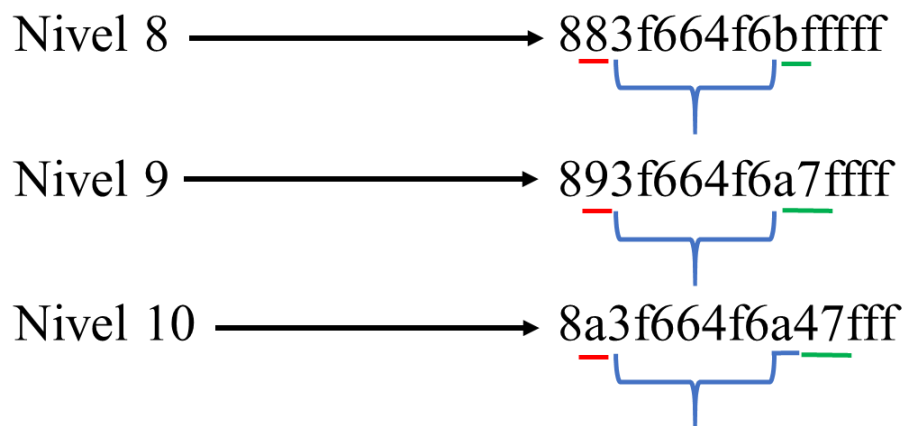


Figura 3-15. Ejemplo de identificadores de etiqueta.

En la Figura 3-15 se muestra como ejemplo las etiquetas correspondientes al mismo punto geográfico, pero a distintos niveles. Como se ve, el bit rojo corresponde al nivel de celda en hexadecimal, los siguientes dígitos son los que se corresponden hasta llegar a un nivel 8 de resolución, siendo los bits verdes los identificadores de esa celda dentro de su nivel. En cada nivel que se aumenta la resolución uno de los dígitos se hace permanente mientras que el último puede variar para identificar la celda en el siguiente nivel. La razón por la que las etiquetas no son iguales a excepción del último dígito es porque no siempre una celda de un nivel de resolución muy alto va a ser *celda hija* de la inmediatamente superior ya que la posición, al aumentar la precisión, puede no corresponder a las celdas hijas de la inicial. Tal y como se aprecia en Figura 3-16, un punto *P* puede estar contenido en una celda de menor resolución que no sea su celda *padre* ya que su sufijo correspondería a la celda *padre* adyacente, aunque su posición geográfica en realidad pertenezca la otra.

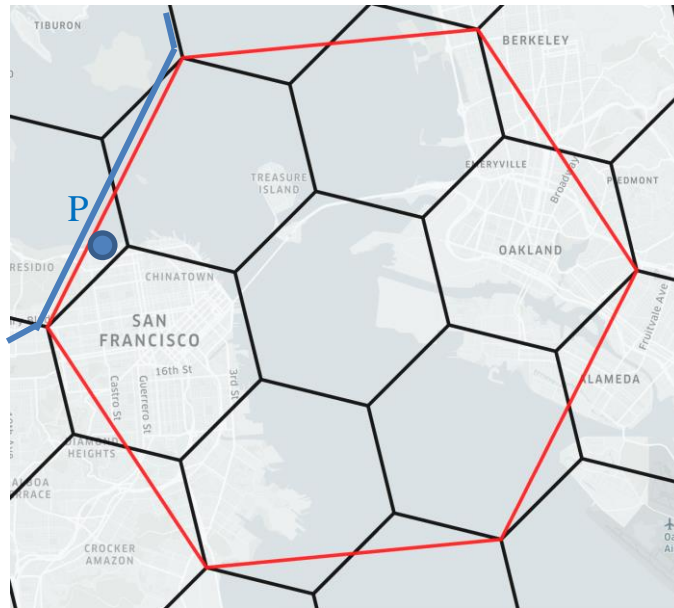


Figura 3-16. Celdas *children* de un nivel de resolución inmediatamente mayor. [33]

De esta manera se tiene que, en vez de realizar el cálculo de todos los niveles de etiquetas que se quieran emplear, se puede truncar la búsqueda hasta el nivel que se requiera en la consulta que se quiera realizar, cambiando únicamente el identificador del nivel en el que se quiere hacer el análisis. Tomando el ejemplo de búsqueda anterior, se puede realizar el mismo análisis determinando únicamente cuales son las celdas de un nivel cuyo análisis se parezca lo máximo posible al radio de búsqueda. En este caso se tiene que para un radio de 7 kilómetros se puede aproximar a las etiquetas de nivel 6 con un lado de 3 km, siendo el número de saltos necesarios igual a 1. Obteniendo las celdas de este nivel con estas características se vuelve a realizar la consulta en la base de datos, haciendo corresponder estas celdas con los inicios de las de nivel 10.

```
SELECT * FROM etiquetas WHERE etiqueta10 LIKE '8a382cab%' OR
etiqueta10 LIKE '8a382c16%' OR etiqueta10 LIKE '8a382c14%' OR
etiqueta10 LIKE '8a382ca9%' OR etiqueta10 LIKE '8a382ca8%' OR
etiqueta10 LIKE '8a382caa%' OR etiqueta10 LIKE '8a382cb9%';
```

En esta consulta aplicamos una máscara, haciendo que las etiquetas buscadas empiecen por un sufijo que vendrá determinado por las celdas de nivel 6 correspondientes a esa zona. Como se ha explicado se obtiene un error, aumentando el número de registros en el resultado, aunque el tiempo se reduce al mínimo (ver resultados en Figura A3-1).

3.2.1.1 Error en área de análisis

Debido a la naturaleza hexagonal del sistema se está cometiendo un error, ya que la distancia real que se desea explorar es un círculo, el cual está aproximando con un conjunto de celdas. Se puede aproximar este valor y acotar entre el peor de los casos, en el que mayor error se obtenga, y el mejor de ellos, con el menor error posible.

1. *Menor error posible*

El menor error tendrá lugar cuando el centro del área de interés, o punto P se situó justo en el centro de la celda que corresponda a sus coordenadas geográficas. Este error depende del valor k , ya que de este mismo depende el radio efectivo del hexágono producido. Como se ha visto anteriormente, este valor es aproximadamente dos veces la longitud del lado de la celda por cada aumento de k . Por tanto, se puede determinar la siguiente aproximación en cuanto al área que se está analizando.

$$\text{Área circular} = \pi \cdot r^2$$

$$\text{Área analizada efectiva} = \pi \cdot (k \cdot 2L)^2$$

Para aproximar lo máximo se iguala ambas áreas, haciendo que el radio sea igual al doble del lado por k . Como se puede ver en Figura 3-17, se apura al máximo el área que se analiza mediante esta fórmula, sin embargo, se comete un error ya que las zonas dentro del círculo, pero fuera del área total ocupada por las celdas quedan excluidas de la búsqueda. Si se quiere acotar este error de forma genérica, se puede determinar que el error será igual a la diferencia del área del círculo menos el área ocupada por todas las celdas. El número de celdas se puede definir como la suma de la celda inicial origen más seis veces el sumatorio de los valores que va tomando el número saltos k . Ya que en cada salto el número de vecinos o número de celdas nuevas es el anterior más seis.

$$\text{Área hexágono regular} = 6 \cdot \left(\frac{L^2 \cdot \sqrt{3}}{4} \right)$$

$$\text{Número de celdas totales} = 1 + 6 \cdot \sum_{i=1}^k i$$

$$\text{Área no analizada} = \pi \cdot r^2 - \left(1 + 6 \cdot \sum_{i=1}^k i \right) \cdot 6 \cdot \left(\frac{L^2 \cdot \sqrt{3}}{4} \right)$$

$$\% \text{ error sobre área real} = \frac{\text{Área no analizada}}{\text{Área real}}$$

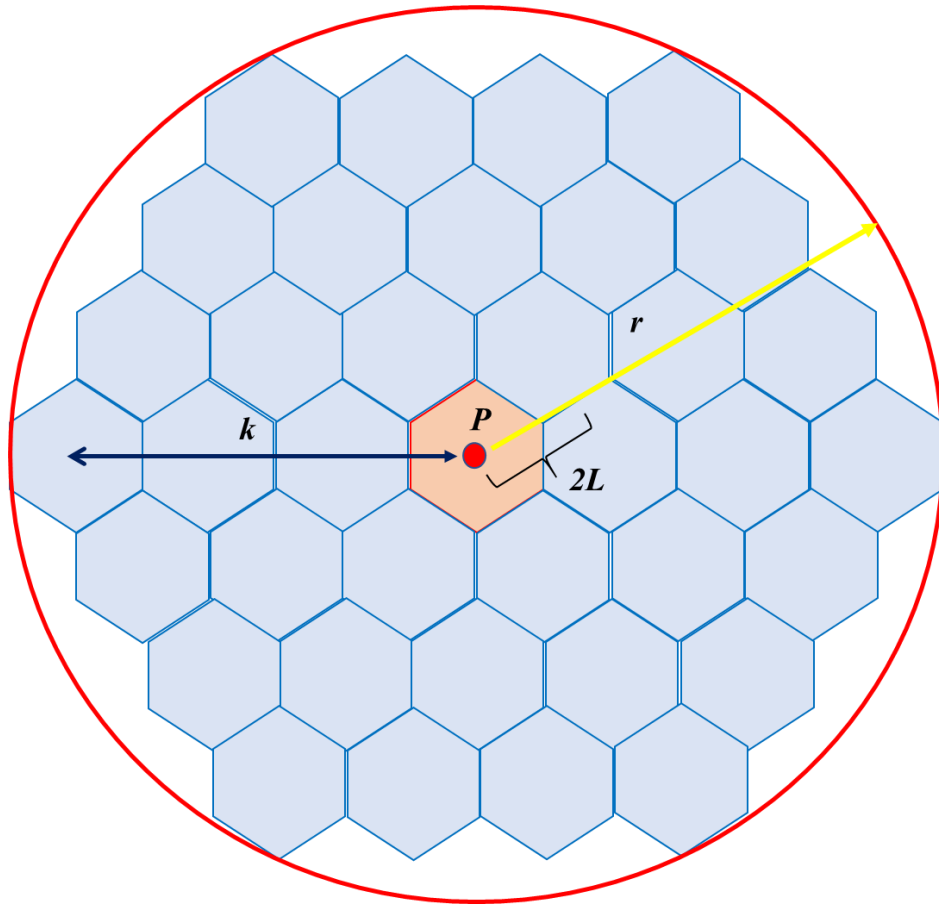


Figura 3-17. Menor error posible.

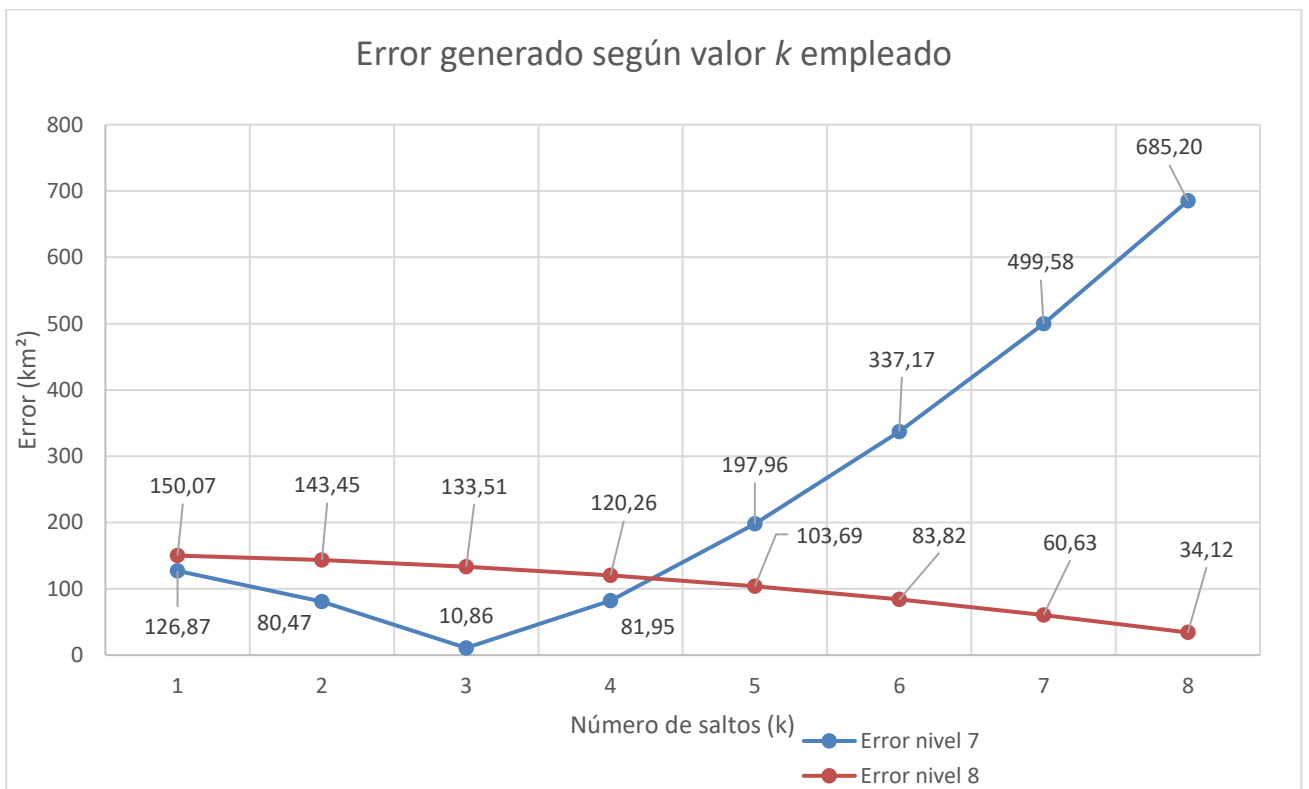


Figura 3-18. Gráfica error generado según k para un radio de búsqueda 7 km.

Como se puede ver en Figura 3-18, el error tiende a disminuir según se acerca al valor que debe de tomar k para que el área total generada sea lo más próxima al círculo real, en este caso de 7 km. Como el lado del nivel 7 corresponde a un lado mucho más grande que el del nivel 8, su error mínimo se alcanza en tres saltos. Sin embargo, el valor mínimo del siguiente nivel de resolución se alcanza en ocho saltos, debido de nuevo a un lado menor. Se concluye que esta gráfica solo aporta información de cómo varía el error según los valores k que se toman para este caso en concreto.

Por otro lado, si se toma siempre el valor óptimo del número de saltos se puede determinar cuál es el error producido y que nivel de celda es más interesante utilizar. En la Tabla 3-1 se toman diferentes radios y niveles de resolución y se obtiene el valor aproximado del error. Este error está referenciado con respecto al área total de análisis, es decir, el círculo de radio r . A su vez, en el gráfico de la Figura 3-19, se ve como el error tiende al nivel séptimo de resolución en radios de este orden por ser el que construye el círculo más próximo. De esta manera, se estudia como varía el error dependiendo del radio de estudio, tomando siempre el número de saltos óptimo.

El redondeo del valor óptimo del número de salto se toma al alza con el objetivo de abarcar el máximo espacio posible. Al final, el error provocado se producirá por barcos que se encuentren en el limbo fuera de nuestra área real o justo en las zonas reseñadas en el esquema anterior, donde el área de nuestras celdas no alcanza a analizar. No es un gran problema ya que, los radios analizados finalmente son aproximados a los reales, tratando de abarcar una zona lo más próxima a la real. Como se puede observar, los porcentajes de error se reducen al máximo al aproximarse al valor óptimo, siendo este valor el número de saltos que se utilizará en cada caso de análisis.

Ambas gráficas y la tabla se obtienen mediante las fórmulas descritas en la página anterior. Si calculamos el área total no analizada obtenemos un valor en kilómetros cuadrados que solo es representativo para en ese caso concreto. En cambio, al calcular el valor óptimo de k para diferentes radios y el error en porcentaje sobre el área real, se obtienen unos resultados más representativos para cada resolución.

Radio (r)	Nivel de resolución	k óptimo	Error (% sobre el área circular real)
5 km	7	2	65,49%
	8	5	6,45%
	9	14	36,03%
7 km	6	2	23,26%
	7	3	7,05%
	8	8	22,17%
10 km	6	2	39,60%
	7	5	12,01%
	8	11	30,23%

Tabla 3-1. Valores de error tomando el k óptimo para diferentes radios de búsqueda.

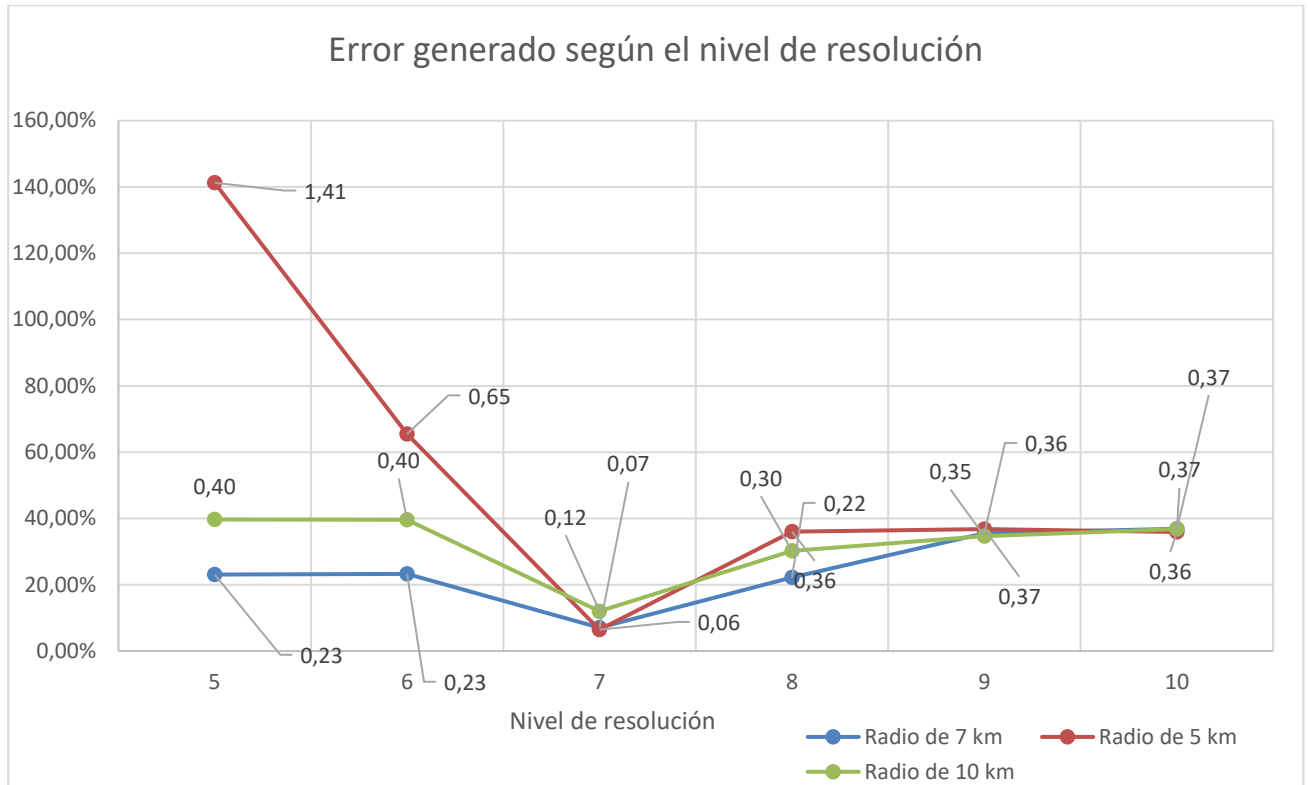


Figura 3-19. Gráfica error generado según el nivel de resolución

En esta gráfica (Figura 3-19), como se ha nombrado, se aprecia mejor como el nivel de resolución siete es el que más aproxima en estos casos, con su número óptimo de saltos, al valor real de un área de análisis. Se puede determinar además que un error de entre el 12 y el 7% sobre el área total no es un gran error, como se ha mencionado anteriormente, ya que se encontrarán en las zonas límite del círculo.

2. Mayor error posible

El mayor error tendrá lugar cuando el punto en cuestión se encuentre en uno de los vértices de la celda correspondiente a su latitud longitud. De esta manera el error es también proporcional al tamaño del lado, se siguen las mismas fórmulas que en el apartador anterior. Sin embargo, en este caso el área original de análisis se verá desplazada como se ve en Figura 3-20, siendo este error el peor de los casos ya que el sistema obvia una importante zona hacia el lado del vértice donde se sitúa el centro y además incluye en sus celdas otra zona que no se encuentra en el área de interés justo en el extremo opuesto del vértice.

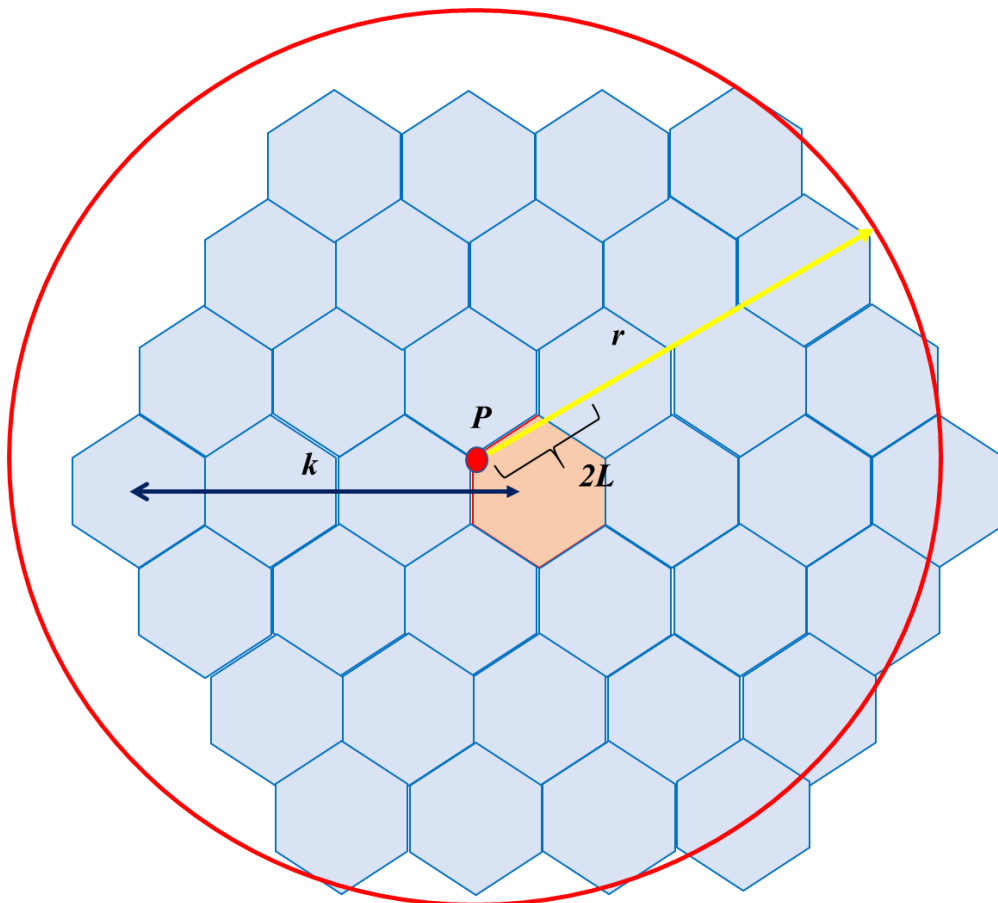


Figura 3-20. Mayor error posible.

Este error es mucho más complejo y difícil de analizar debido al desplazamiento del centro de coordenadas. Sin embargo, este error también dependerá de la resolución utilizada, la aproximación al error utilizado anteriormente corresponderá, aunque aumentando el porcentaje de error ligeramente, con los datos obtenidos en el caso anterior. El error será más pequeño cuanto mayor resolución se aplique y el redondeo en el valor óptimo de k no sea excesivo. Sin embargo, esto provocaría que las consultas sean más lentas.

H3 no deja de ser un sistema de análisis geoespacial y lo que puede aportar es velocidad a la hora de determinar la situación geográfica de los buques, sin embargo, si se suma esto a otras características de las propias bases de datos se puede facilitar la detección de estos comportamientos anómalos.

3.2.2 Detección de zonas habituales y buques estáticos

Para realizar este experimento se crea una nueva tabla que contendrá todos los datos que contiene la tabla *aismediterraneo*, pero en este caso cambiando latitud y longitud por sus correspondientes etiquetas. Esta nueva tabla será *medih* y con una sencilla sentencia de MySQL se toman los datos de la tabla original para construirla. Tras esto se vuelve a transformar las latitudes y longitudes en las etiquetas determinadas.

```
CREATE TABLE medih SELECT id, mmsi, imo, sog, cog, longitude,
latitude msgtimestamp, source, callsign, name FROM aismediterraneo;
ALTER TABLE medih ADD COLUMN (etiqueta10 VARCHAR (30), etiqueta9
VARCHAR (30), etiqueta8 VARCHAR (30));
```

Al mismo tiempo se convierte el valor de la columna *msgtimestamp*, el cual no es más que la fecha del registro, a unidades Unix Time. Unix Time es una unidad temporal que indica en segundos el tiempo transcurrido desde la medianoche UTC del 1 de enero de 1970. Este valor se utilizará a la hora de comparar los registros de un mismo buque, ver si han ocupado la misma celda en algún momento y si es el caso, ver si esto ha ocurrido en las últimas dos horas, determinando que se encuentra en una zona reciente. Mediante una sentencia a MySQL se compara la misma columna viendo en que registros se cumple la condición de que el tiempo transcurrido entre registros este dentro del rango de dos horas y además las etiquetas coincidan (Figura 3-21).

```
SELECT a.name, a.unixtime, a.etiqueta10, b.name, b.unixtime,
b.etiqueta10 FROM medih a inner JOIN medih b on
a.etiqueta10=b.etiqueta10 where ( a.unixtime-b.unixtime BETWEEN 1
AND 7200) limit 10;
```

name	unixtime	etiqueta10	name	unixtime	etiqueta10
LONGLINE BUOY 06 89%	1576027794	8a3f664f6a47fff	LONGLINE BUOY 06 89%	1576027499	8a3f664f6a47fff
LONGLINE BUOY 06 89%	1576056878	8a3f6641a08ffff	LONGLINE BUOY 06 89%	1576056059	8a3f6641a08ffff
LONGLINE BUOY 06 89%	1576039806	8a3f664f4ca7fff	LONGLINE BUOY 06 89%	1576039476	8a3f664f4ca7fff
LONGLINE BUOY 06 89%	1576050184	8a3f6641a71ffff	LONGLINE BUOY 06 89%	1576049452	8a3f6641a71ffff
LONGLINE BUOY 06 89%	1576038020	8a3f664f4c9ffff	LONGLINE BUOY 06 89%	1576037686	8a3f664f4c9ffff
LONGLINE BUOY 06 89%	1576039053	8a3f664f4c87fff	LONGLINE BUOY 06 89%	1576038692	8a3f664f4c87fff
LONGLINE BUOY 06 89%	1576053393	8a3f6641a727fff	LONGLINE BUOY 06 89%	1576052639	8a3f6641a727fff
PILOTS P	1576087464	8a184b50b82ffff	PILOTS P	1576086031	8a184b50b82ffff
PILOTS P	1576087464	8a184b50b82ffff	PILOTS P	1576082058	8a184b50b82ffff
PILOTS P	1576087464	8a184b50b82ffff	PILOTS P	1576086804	8a184b50b82ffff

10 rows in set (0.00 sec)

Figura 3-21. Ejemplo de detección de zona habitual.

Si se cambia la consulta anterior haciendo que el rango de tiempo en vez de encontrarse por debajo de las dos horas justo por encima estaremos consultando buques que han transmitido en la misma celda durante más de dicho tiempo. Esto puede indicar que estos buques se encuentran parados o todo el tiempo en la misma área, sin embargo, puede ocurrir que haya transmisiones intermedias en otras zonas.

```
SELECT a.name, a.msgtimestamp, a.unixtime, a.etiqueta10, b.name,
b.msgtimestamp, b.unixtime, b.etiqueta10 FROM medih a inner JOIN
medih b on a.etiqueta10=b.etiqueta10 where ( abs(a.unixtime-
b.unixtime) > 7200) limit 25;
```

name	msgtimestamp	unixtime	etiqueta10	name	msgtimestamp	unixtime	etiqueta10
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 03:21:59.791	1576030919	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 09:52:12.146	1576054332	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 08:42:09.931	1576050129	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 08:25:18.204	1576049118	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 02:01:03.948	1576026063	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 11:18:18.283	1576059498	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 00:14:13.3	1576019653	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 04:08:04.286	1576033684	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 07:25:59.563	1576045559	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 11:18:17.818	1576059497	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 06:07:23.889	1576040843	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 21:48:50.554	1576097330	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 03:49:35.632	1576032575	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 04:10:13.675	1576033813	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 02:16:56.275	1576027016	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 10:30:56.427	1576056656	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 06:06:10.783	1576040770	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 23:37:47.628	1576103867	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 06:45:32.94	1576043132	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 23:30:46.47	1576103446	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 04:06:49.814	1576093609	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 08:02:04.028	1576047724	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 03:55:48.362	1576032948	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 02:11:54.217	1576026714	8a184b50b82ffff
PILOTS P	2019-12-11 19:04:24.102	1576087464	8a184b50b82ffff	PILOTS P	2019-12-11 07:13:06.08	1576044786	8a184b50b82ffff

25 rows in set (0.00 sec)

Figura 3-22. Ejemplo de consulta de buques estáticos.

Mediante esta consulta se van comparando todas las etiquetas e imprime en pantalla cuando ambas etiquetas son iguales y el tiempo entre registros (*unixtime*) es mayor de 7200 segundos, el registro inicial que se está comparando (izq. de Figura 3-22) y el cumple el requisito en la derecha. Así se están indicando todos los registros que cumplen estos requisitos, sin atender a si son el mismo buque. Esta consulta compara cada registro con todos los que cumplen las condiciones, cuando termina de comparar un registro se pasa a comparar el siguiente y así consecutivamente. Es por ello que, pueden producirse combinaciones similares, pero en comparaciones distintas. No obstante, se comprueba el funcionamiento de la consulta en tiempo real.

Para realizar esta consulta de manera más eficiente, en primer lugar, se ve cuáles son las celdas más concurridas y además por el mismo buque en la base de datos disponible. Se realiza la siguiente consulta, en la que seleccionamos las 25 celdas más ocupadas por el mismo buque:

```
SELECT etiqueta10, name, count(*) AS c FROM medih GROUP BY
etiqueta10, name HAVING c>1 ORDER BY c DESC limit 25;
```

Cuyo resultado se muestra en Figura 3-23, mostrando la celda, el nombre del barco y la cantidad de veces *c* que se ha ocupado esa celda por él. Esta consulta tarda unos 19,21 segundos.

etiqueta10	name	c
8a39562b4b2ffff	WELVAART	2484
8a391e383adffff	SCOMBA	2418
8a3835346927ffff		2389
8a383491c3afffff	TARGA	2380
8a394260b287ffff	HIGH ENERGY	2314
8a39562b4a27ffff	JOAQUIN TORRES	2020
8a3f304e8837ffff	MFV RANIA	2015
8a3f304e8907ffff	MARE BLU	1983
8a1849499b97ffff	FACAL DIECIOCHO	1947
8a185ba1c437ffff	PAU CASALS	1884
8a3f304ed097ffff	THE MERCY BOYS	1868
8a1849499167ffff	FACAL DIECINUEVE	1859
8a3f304e8d77ffff	SANDOKAN	1814
8a3825a996c7ffff	HISPANIA DOS	1805
8a38246164a7ffff	SERTOSA VEINTICUATRO	1776
8a3951cecc4ffff	CASTAVI JET	1768
8a391a91a58ffff	N PRONTITUD	1709
8a3954a06097ffff	V.B.VIGOR	1689
8a38253352cffff	V.B.BRIO	1679
8a394265538ffff	NIRVANA FORMENTERA	1651
8a39562b4a27ffff	V.B. SUPLENTE	1646
8a39540f01a7ffff	VB XALOC	1626
8a391aba9807ffff	CLARA CAMPOAMOR	1592
8a184b50e607ffff	RANGER	1580
8a391aa26317ffff	GRANT DEL MAR	1567

25 rows in set (19.21 sec)

Figura 3-23. 25 buques que más repiten celda.

Como ejemplo se puede ver que el *Welvaart* tiene un total de 2484 registros en esa celda, se aplica una sentencia parecida a las anteriores con el objetivo de determinar si la diferencia entre esos registros es mayor de 7200 segundos o 2 horas. Mediante la siguiente sentencia, solo se imprimirán los valores que además tengan ese nombre y repitan esa etiqueta. Se limita la búsqueda a solo 10 valores que ya nos determinará que en efecto ese buque se ha situado durante más de 2 horas en ese mismo lugar.

```
SELECT a.name, a.unixtime, a.etiqueta10, b.name, b.unixtime,
b.etiqueta10 FROM medih a INNER JOIN medih b on
a.etiqueta10=b.etiqueta10 WHERE (abs(a.unixtime-b.unixtime) >7200
AND a.name = 'WELVAART' AND a.etiqueta10= '8a39562b4b2ffff') LIMIT
10;
```

name	unixtime	etiqueta10	name	unixtime	etiqueta10
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576082130	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576066920	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576033652	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576029616	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576077803	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576064543	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576074115	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576081277	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576071566	8a39562b4b2ffff
WELVAART	1576044835	8a39562b4b2ffff	WELVAART	1576028921	8a39562b4b2ffff

10 rows in set (0.00 sec)

Figura 3-24. Ocupación *Welvaart* en misma celda cumpliendo el requisito de más de dos horas.

Otra manera de relacionar un barco con una zona en concreto es seleccionar uno de ellos y contabilizar la cantidad de veces que ha ocupado una celda en todos sus registros. De esta manera, aunque no se contabilice el tiempo que ha transcurrido entre cada posicionamiento, se puede ver dónde ha permanecido el buque durante gran parte del tiempo. Mediante la siguiente sentencia MySQL se obtiene el resultado deseado.

```
SELECT etiqueta9, name, count(*) AS c FROM medih WHERE name = 'SEA FALCON' GROUP BY etiqueta9, name HAVING c>1 ORDER BY c DESC;
```

Tomando este primer ejemplo se obtiene el resultado mostrado en Figura 3-25. Realizando el mismo procedimiento con otros buques se construye un diagrama de barras (Figura 3-26) donde se compara visualmente los tiempos que se han obtenido para cada uno de los ejemplos tomados, los cuales presentan diferentes zonas habituales dependiendo del itinerario que han seguido durante el día. El tiempo varía según la cantidad de etiquetas diferentes que disponga en ese nivel de resolución, siendo diferente para cada uno de los casos. Sin embargo, todos los tiempos de consulta oscilan entre los 20 y los 30 segundos.

etiqueta9	name	c
893f304e8c3ffff	SEA FALCON	734
893946812bbffff	SEA FALCON	24
8939468adbffff	SEA FALCON	20
89394681217ffff	SEA FALCON	16
893946812abffff	SEA FALCON	4
8939468adabffff	SEA FALCON	4
89394681287ffff	SEA FALCON	3
893946813abffff	SEA FALCON	2
8939468ad1bffff	SEA FALCON	2
8939468adcbffff	SEA FALCON	2

10 rows in set (30.78 sec)

Figura 3-25. Zonas habituales *Sea Falcon*.

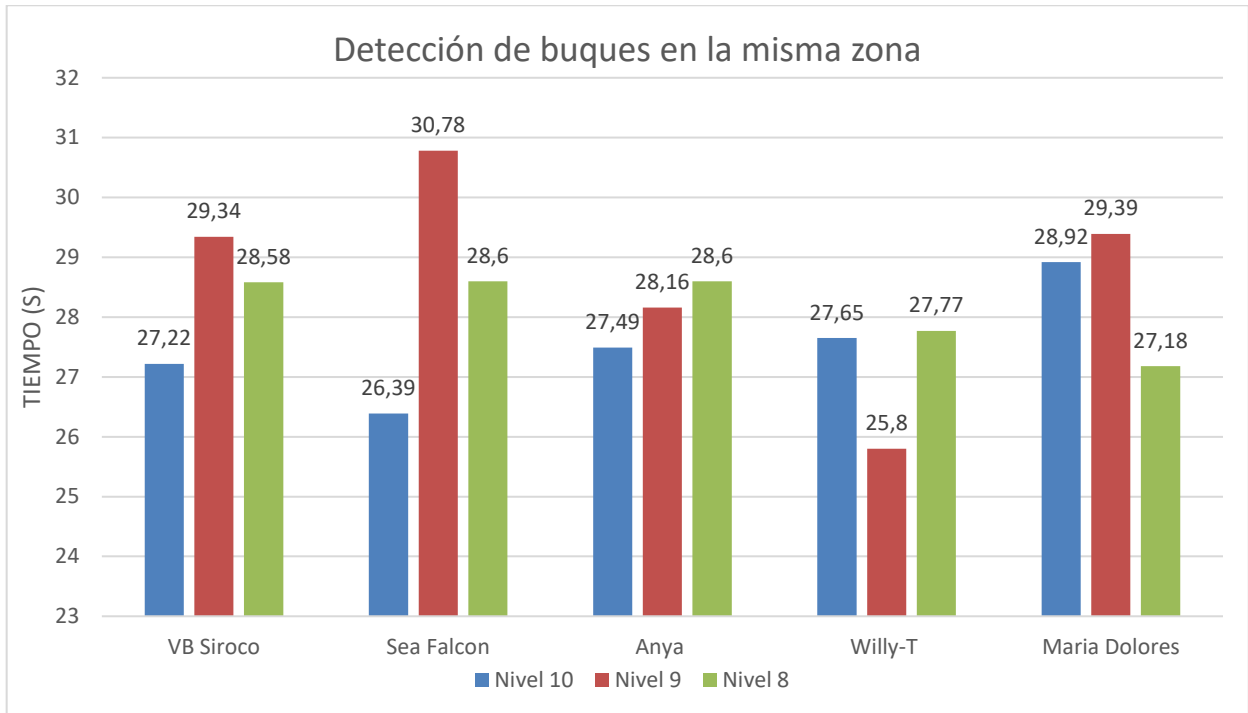


Figura 3-26. Gráfico detección de buques en la misma zona

Si se añade un índice al nombre de los barcos (*name*) igual que se ha hecho con las etiquetas se lidia con este problema y se obtienen resultados en el tiempo mínimo (Figura 3-27).

```

+-----+-----+-----+
| etiqueta9 | name | c |
+-----+-----+-----+
| 893f304e8c3ffff | SEA FALCON | 734 |
| 893946812bbffff | SEA FALCON | 24 |
| 8939468adbffff | SEA FALCON | 20 |
| 89394681217ffff | SEA FALCON | 16 |
| 8939468adabffff | SEA FALCON | 4 |
| 893946812abffff | SEA FALCON | 4 |
| 89394681287ffff | SEA FALCON | 3 |
| 8939468adcbffff | SEA FALCON | 2 |
| 8939468ad1bffff | SEA FALCON | 2 |
| 893946813abffff | SEA FALCON | 2 |
+-----+-----+-----+
10 rows in set (0.05 sec)
    
```

Figura 3-27. Zonas habituales Sea Falcon. (2)

Al realizar el análisis con celdas de mayor tamaño se está examinando una zona más amplia. La resolución de nivel 10 es demasiado precisa para estas consultas puesto que un barco se puede mover en una zona más amplia. Mientras que con una resolución de ese nivel se pueden detectar buques fondeados o atracados, con una resolución menos se quiere detectar esos buques que se encuentran en unas inmediaciones, es decir, en una zona más amplia.

Se realiza la misma consulta anterior, pero añadiendo un requisito y es que la velocidad sea diferente de cero y con un nivel 8 de resolución. De esta forma se amplía el rango de búsqueda en zona, pero se reduce a buques que están en movimiento (Figura 3-28).

```
SELECT etiqueta8, name, count(*) AS c FROM medih WHERE sog != 0
GROUP BY etiqueta8, name HAVING c>1 ORDER BY c DESC limit 10;
```

etiqueta8	name	c
88391e3a41fffff	BUYANGO SEGUNDO	821
883f73c55dfffff	KANELOS II	400
881843c031fffff	F/V IDURRE BERRIA	327
88184eba95fffff	PEDRINA BLANCA	251
88184ec0e5fffff	NA	207
88184eb1a7fffff	PEDRINA BLANCA	198
883951a2a9fffff	ES LLEVANT	171
88395286b9fffff	NA	156
8839540f01fffff	VB XEREA	152
8838253353fffff	V.B. GLACIAL	140

10 rows in set (13.62 sec)

Figura 3-28. Zona habitual de buques con velocidad diferente de cero.

3.2.3 Detección de encuentros

De la misma manera que el experimento anterior se puede detectar los casos en que los buques se han encontrado en la misma celda. Como la celda con la que se está trabajando es de nivel de resolución diez, este hecho indica una aproximación que puede llegar a ser muy peligrosa o incluso sospechosa en algunos casos. Detectar estas situaciones y tener constancia de ellas es uno de los propósitos del trabajo del COVAM. Mediante una consulta MySQL parecida a las anteriores se puede visualizar encuentros producidos. En este caso solo interesa que el encuentro en la misma celda se haya producido durante un periodo más corto de tiempo, en este caso dos minutos. Además, se añade una cláusula que indique que el nombre entre registros debe ser diferente ($a.name \neq b.name$).

```
SELECT a.name, a.unixtime, a.etiqueta10, b.name, b.unixtime,
b.etiqueta10 FROM medih a INNER JOIN medih b on
a.etiqueta10=b.etiqueta10 WHERE (abs(a.unixtime-b.unixtime) <120
AND a.name != b.name ) LIMIT 25;
```

name	unixtime	etiqueta10	name	unixtime	etiqueta10
PILOTS P	1576074031	8a184b503497fff	FINNSUN	1576073979	8a184b503497fff
PILOTS P	1576067465	8a184b51c087fff	NORDIC CYGNUS	1576067433	8a184b51c087fff
PILOTS P	1576067530	8a184b51c18ffff	GATIKA	1576067484	8a184b51c18ffff
PILOTS P	1576038811	8a184b51c72ffff	GATIKA	1576038801	8a184b51c72ffff
PILOTS P	1576066987	8a184b51c447fff	NORDIC CYGNUS	1576067079	8a184b51c447fff
PILOTS P	1576074763	8a184b50e51ffff	IBAIZABAL CATORCE	1576074655	8a184b50e51ffff
PILOTS P	1576102337	8a184b50e51ffff	ARKLOW RIVAL	1576102344	8a184b50e51ffff
PILOTS P	1576072735	8a184b51c46ffff	SESTAO KNUITSEN	1576072633	8a184b51c46ffff
PILOTS P	1576067373	8a184b51c087fff	NORDIC CYGNUS	1576067433	8a184b51c087fff
PILOTS P	1576083484	8a184b50e68ffff	IBAIZABAL CATORCE	1576083564	8a184b50e68ffff
PILOTS P	1576083484	8a184b50e68ffff	IBAIZABAL CATORCE	1576083416	8a184b50e68ffff
PILOTS P	1576066932	8a184b51c457fff	IBAIZABAL SEIS	1576066824	8a184b51c457fff
PILOTS P	1576066932	8a184b51c457fff	IBAIZABAL CATORCE	1576067027	8a184b51c457fff
PILOTS P	1576066932	8a184b51c457fff	IBAIZABAL CATORCE	1576066966	8a184b51c457fff
PILOTS P	1576062836	8a184b51c44ffff	IBAIZABAL SEIS	1576062730	8a184b51c44ffff
PILOTS P	1576062836	8a184b51c44ffff	IBAIZABAL CATORCE	1576062880	8a184b51c44ffff
PILOTS P	1576074699	8a184b50e51ffff	IBAIZABAL CATORCE	1576074655	8a184b50e51ffff
PILOTS P	1576073092	8a184b51c88ffff	IBAIZABAL ONCE	1576073005	8a184b51c88ffff
PILOTS P	1576074359	8a184b50e507fff	IBAIZABAL CATORCE	1576074391	8a184b50e507fff
PILOTS P	1576074359	8a184b50e507fff	IBAIZABAL CATORCE	1576074305	8a184b50e507fff
PILOTS P	1576067051	8a184b51c447fff	NORDIC CYGNUS	1576067079	8a184b51c447fff
PILOTS P	1576073020	8a184b51c12ffff	SESTAO KNUITSEN	1576073028	8a184b51c12ffff
PILOTS P	1576038679	8a184b51c71ffff	TOUR POWEROL	1576038684	8a184b51c71ffff
PILOTS P	1576038095	8a184b51c637fff	TOUR POWEROL	1576038142	8a184b51c637fff
PUNTA LAMINA	1576049832	8a3835c6b0e7fff	QUILATES	1576049815	8a3835c6b0e7fff

25 rows in set (38.11 sec)

Figura 3-29. Ejemplo encuentros (limitado 25 resultados).

En la Figura 3-30 se resume el tiempo tomado en detectar encuentros, ampliando el rango de encuentros producidos. Los tiempos registrados van aumentando linealmente cuanto más se amplía el abanico de encuentros que se pueden detectar. En este caso la generación de índices internos en MySQL para el nombre o el tiempo (*unixtime*) no mejora los resultados ya que la propia consulta recorre la tabla siempre comprobando que los nombres sean diferentes y realizando la resta entre los tiempos en los que ha registrado la transmisión.

Por otro lado, sería interesante comprobar si estos encuentros son espontáneos, si se producen en puerto o, lo más importante, si son encuentros que siempre se efectúan en la misma zona y con los mismos barcos. Sin embargo, para ello se necesita del análisis de una base de datos mucho más grande con datos que abarquen un periodo temporal más grande.

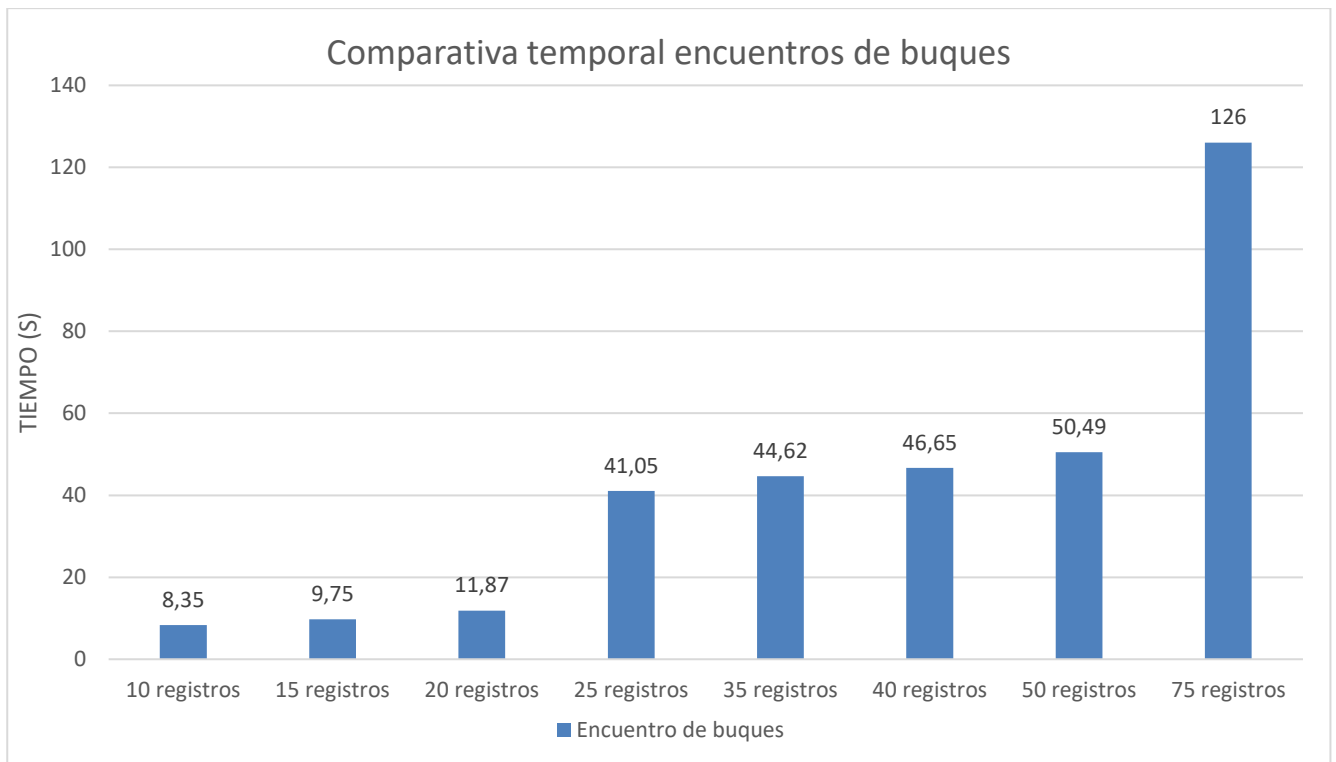


Figura 3-30. Gráfico temporal según número de encuentros detectados.

3.3 Visualización de índices generados

Tras realizar los experimentos, se comprueba el funcionamiento de los sistemas de visualización comentados al principio de este capítulo. Se realiza una consulta que cuente todas las veces que se repite cada celda y lo almacene en una nueva tabla.

```
INSERT INTO cuenta8 SELECT etiqueta8, count(*) AS c FROM medih GROUP BY etiqueta8 HAVING c>0 ORDER BY c DESC;
```

Teniendo esta tabla se extraen los datos en un archivo *.txt* desde C, recorriendo los datos e imprimiéndolos en el nuevo fichero. Este fichero es leído por la función *H3ToGeoBoundary*, ejecutada desde el propio terminal Linux, que convierte los índices contenidos en un archivo KML listo para ser integrado en Google Earth.

```
:~$ h3ToGeoBoundary --kml --kml-name "celdas8" --kml-description "h3 cells" < celdas8.txt > cells8.kml
```

Se realiza este procedimiento con un nivel de resolución 8 ya que con otras etiquetas genera problemas a la hora de procesarlo en Google Earth. Con esta resolución se asegura un tamaño menor de 100 MB de fichero KML, en este caso 76 MB (ver estructura del archivo en Anexo IV: Extracto *cells8.kml*).

Al implementar el archivo en Google Earth se obtienen las imágenes de Figura 3-31 y Figura 3-32. En ellas se pueden apreciar la acumulación de celdas en las zonas de mayor tránsito marítimo como el Estrecho de Gibraltar y los puertos internacionales. Además, se puede apreciar que en los puertos

españoles es dónde más registros existen y se pueden diferenciar las rutas más transitadas. En la segunda figura se aproxima a una zona costera y se ve como se han formado claramente las celdas.



Figura 3-31. Captura Google Earth de *cells8.kml*



Figura 3-32. Zoom en Google Earth, celdas ocupadas en las inmediaciones de Cabo Palos, Murcia.

Sin embargo, este sistema no es lo suficientemente eficiente ya que sufre demasiado al procesar un archivo tan grande. Además, habría que cambiar para cada una de las celdas para que mostrase la cantidad de veces que ha sido ocupada y asignar una característica nueva que asignase el valor jerárquico en función de esa carga (un color, por ejemplo).

Es por tanto que se decide realizar un procedimiento similar pero esta vez con *Kepler.gl* [69], herramienta online y de acceso libre, que permite la aplicación de H3 directamente. Para ello se necesita un archivo que almacene la tabla que se quiere representar. En este caso exportamos la misma tabla generada anteriormente. Mediante una sentencia MySQL exportamos la tabla y construimos un archivo que es fácilmente legible como CSV, formato que Kepler soporta, forzando a que se separen los distintos campos. Se extraen en este caso la cuenta realizada anteriormente, pero con celdas de nivel 10 para comprobar el rendimiento de la herramienta.

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/cuenta10' FIELDS
TERMINATED BY '|' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '\\\' LINES
TERMINATED BY '\n' FROM cuenta10;
```

Resultando la visualización de la Figura 3-33, dónde se observa, al igual que en la figura anterior, las rutas y las zonas con mayor actividad marítima. En este caso el tiempo consumido en procesar el archivo es mucho menor a la vez que la fluidez mejora considerablemente. El archivo generado (*cuenta10.csv*) ocupa tan solo 9 MB.



Figura 3-33. Captura Kepler.gl con *cuenta10.csv*.

3.3.1 Visualización: Área de búsqueda

Si en el mapa generado en Kepler se busca la posición del centro de coordenadas que se ha utilizado para el experimento de buques en una zona determinada (3.2.1) se pueden observar claramente los resultados obtenidos. Si esquematizamos el círculo de búsqueda real (Figura 3-34) con el hexágono formado por los saltos de celda se puede comprobar los registros que exactamente se encuentran dentro del área.



Figura 3-34. Exploración de buques en una zona (37°N, 1°E).

3.3.2 Visualización: Zona habitual de un buque

Por otro lado, esta herramienta proporciona la capacidad de aplicar una paleta de colores según el valor numérico que se le ha asignado a cada celda (en este caso la cuenta de las veces que se ha ocupado). Esta paleta de colores proporciona una visualización más clara de cuáles son las zonas más ocupadas (Figura 3-35), sin embargo, al ser un nivel de resolución muy alto, la acumulación de datos en puertos es muy grande cuando en las rutas son muy pequeñas ya que los buques no pasan siempre por las mismas celdas. La propia interfaz de Kepler también permite comparar dos celdas y sus valores respectivamente como se muestra.

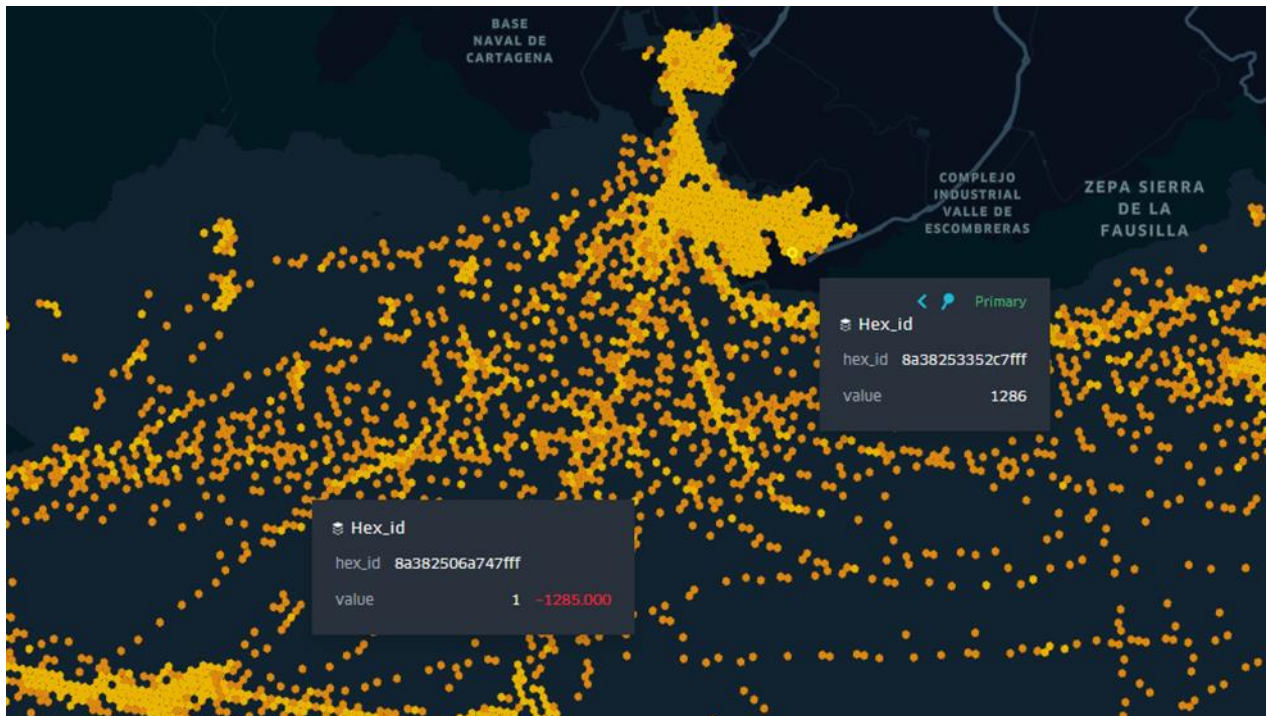


Figura 3-35. Celdas según número de registros, inmediaciones de Cartagena.

Extrayendo los resultados de la contabilización de celdas realizada en el experimento de 3.2.2 se puede visualizar las zonas más ocupadas por unos buques en concreto, combinando las consultas de exportación de datos a CSV y de zonas habituales. En esta ocasión un nivel de resolución 8.

```
SELECT etiqueta8, count(*) AS c INTO OUTFILE '/var/lib/mysql-
files/barcosestatics' FIELDS TERMINATED BY '|' OPTIONALLY ENCLOSED
BY '"' ESCAPED BY '\\\ ' LINES TERMINATED BY '\n' FROM medih WHERE
name = 'SEA FALCON' GROUP BY etiqueta8 HAVING c>1 ORDER BY c DESC;
```

En Figura 3-36 se pueden ver únicamente las celdas que ha ocupado este buque en concreto. Por otra parte, Kepler proporciona la capacidad de visualizar el mapa en 3D. Si a esa capacidad se le añade la aplicación de un valor *height* basado en la cuenta de veces que se ha ocupado la celda, hace más visual el análisis. Se aprecia, a modo de ejemplo, la permanencia del buque en las inmediaciones de Tarragona tras una navegación, dónde se puede ver como espera una entrada en puerto.

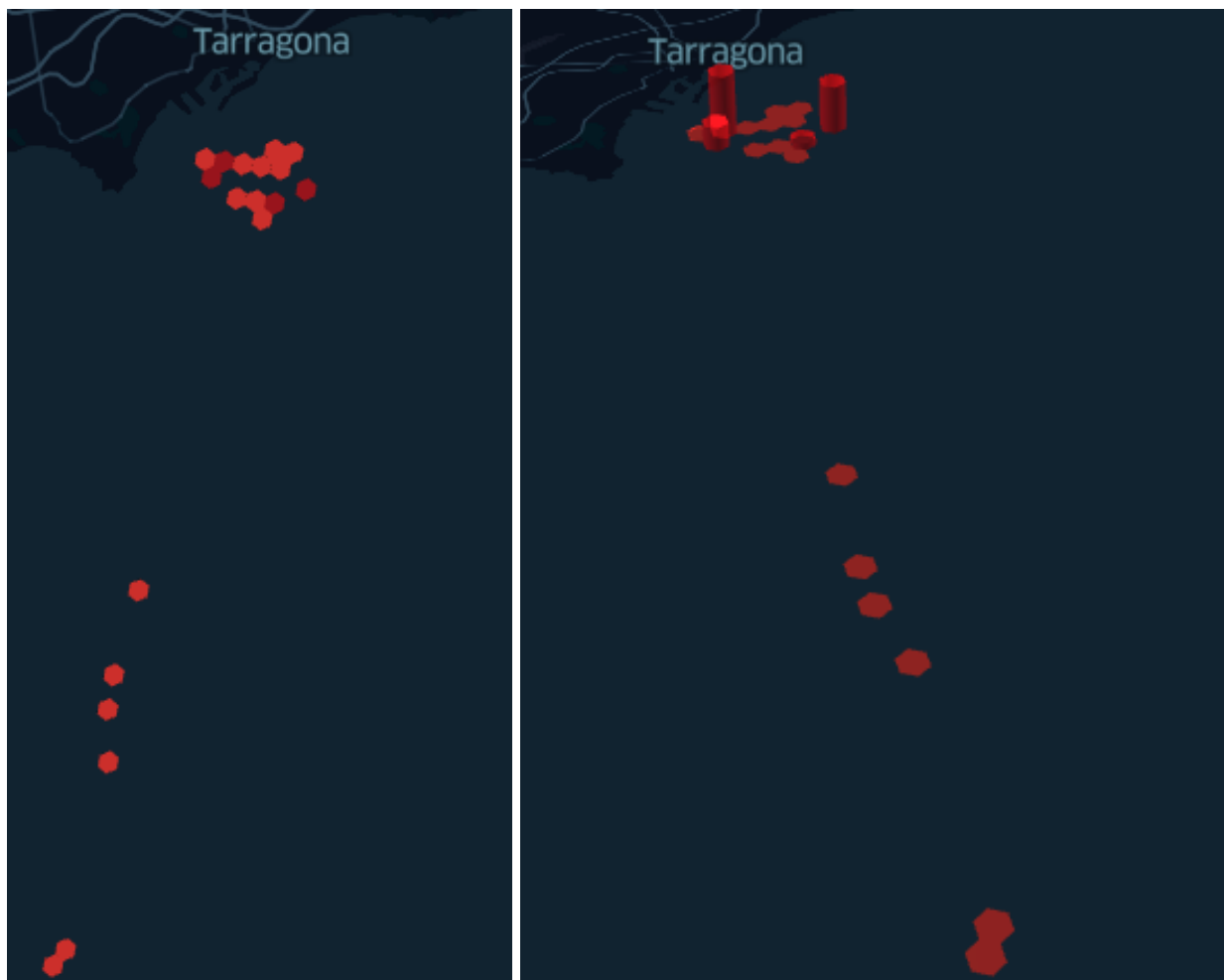


Figura 3-36. Zona habitual del *Sea Falcon*, con las celdas más ocupadas en rojo oscuro (izq.) y aplicando altura (dcha.).

3.3.3 Visualización: Zonas según velocidad media

De la misma manera que se han extraído los valores anteriores se pueden extraer en este caso las celdas con un valor que indique la velocidad media en la zona. Para ello se extrae, mediante una sentencia a la base de datos, las etiquetas y su valor medio en toda la tabla, de nuevo construyendo un archivo tipo CSV.

```
SELECT  etiqueta8,    AVG(sog)  INTO    OUTFILE    '/var/lib/mysql-  
files/avgsog8' FIELDS TERMINATED BY '|' OPTIONALLY ENCLOSED BY ''  
ESCAPED BY '\\\ ' LINES TERMINATED BY '\n' FROM medih GROUP BY  
etiqueta8;
```

Y de nuevo, empleando Kepler para su visualización, se obtiene una imagen mucho más clara de las rutas y de las zonas de estacionamiento o con velocidad cero, dónde buques permanecen estáticos (Figura 3-37 y Figura 3-38). Implementando el mapa 3D y la altura según el valor de velocidad media se aprecia mucho mejor las zonas de mayor velocidad y los puertos o fondeaderos.

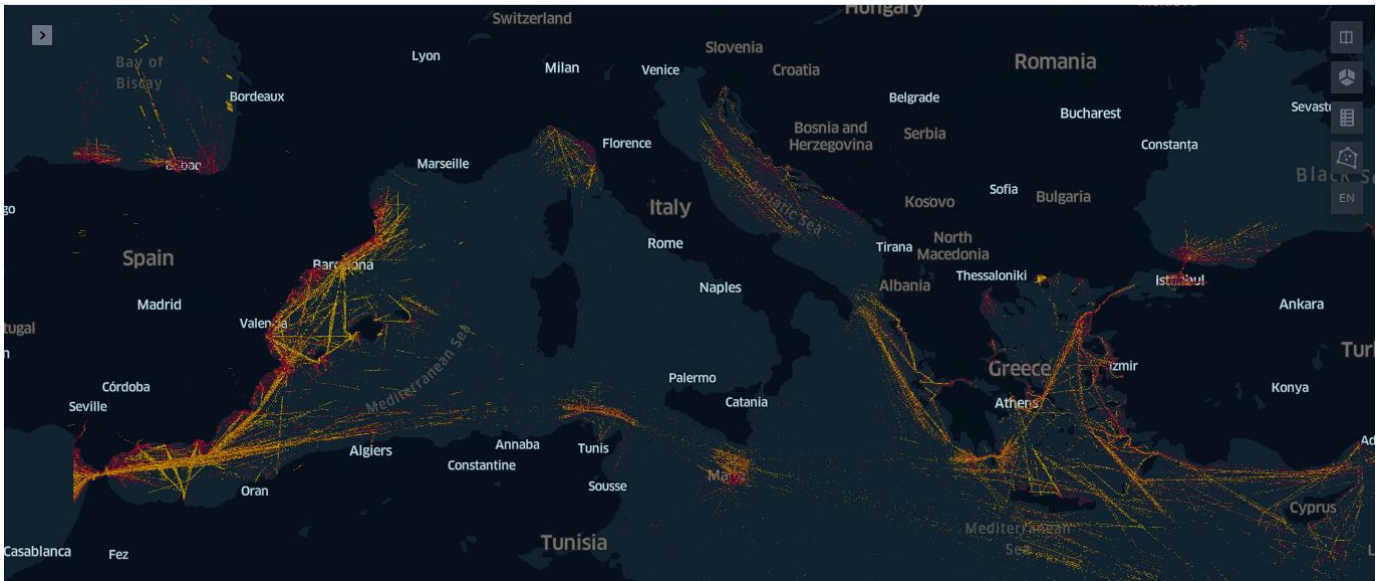


Figura 3-37. Captura Kepler.gl, *avgsog8.csv*

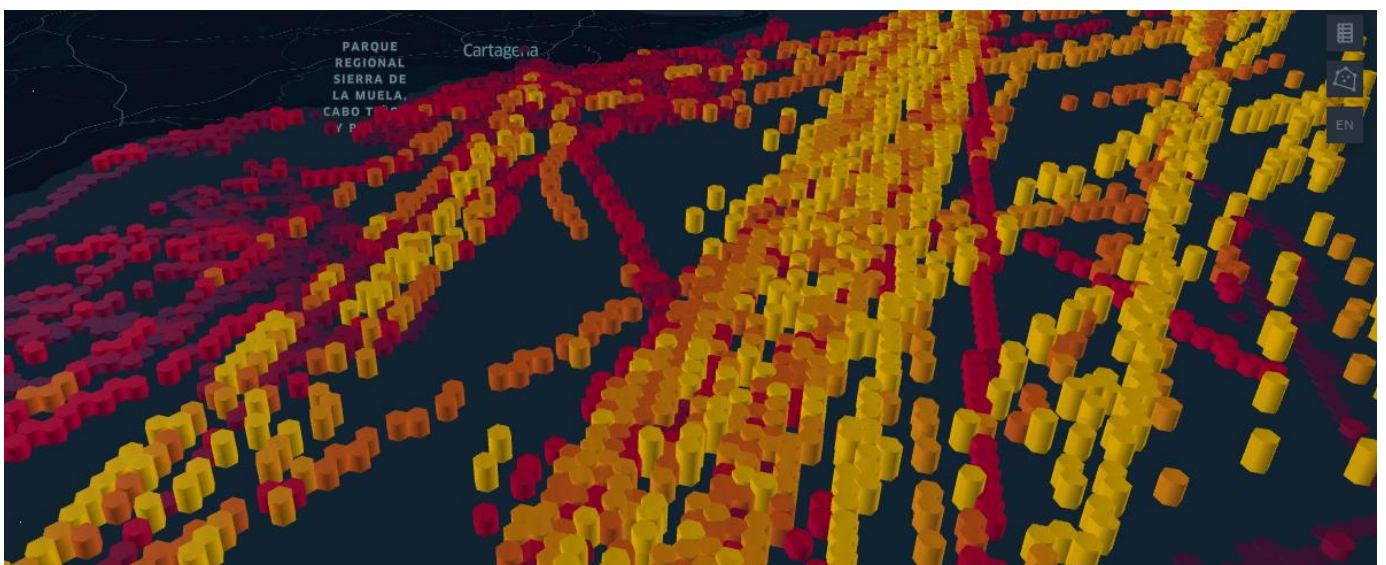
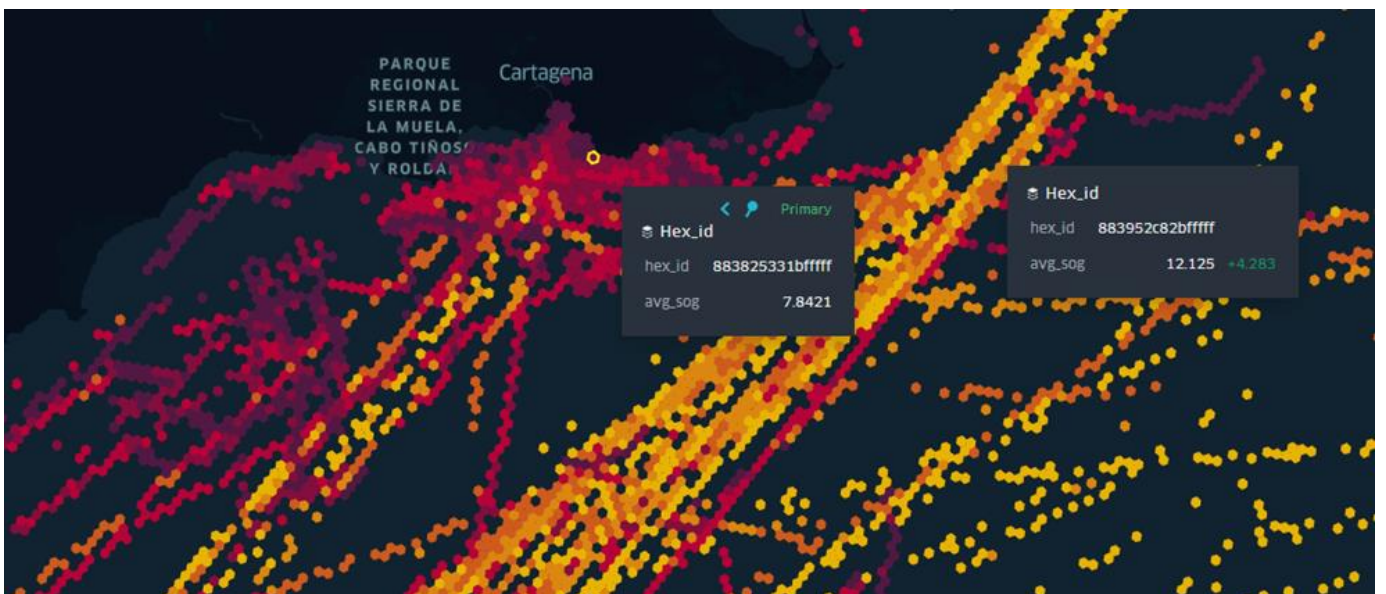


Figura 3-38. Celdas según velocidad media (*avg_sog*), inmediaciones de Cartagena

4 RESULTADOS

En este capítulo se resumen los resultados obtenidos de los diferentes análisis que se han realizado. Se ha comprobado el funcionamiento de la librería de H3 y la compatibilidad con la gestión de datos AIS.

Se presenta a continuación un resumen de todo el desarrollo del trabajo, con objetivo de extraer la información más importante resultante del proyecto.

4.1 Resumen de resultados

En primer lugar, se comprobó el correcto funcionamiento de la librería con los datos proporcionados por el COVAM. La capacidad de asignar una cadena de caracteres diferente para cada una de las posiciones, dependiendo de su latitud y longitud y con un amplio rango de resoluciones ha mejorado sustancialmente el tiempo necesario para realizar las consultas. Se ha contado, además de las posiciones geográficas de cada uno de los registros, con otros datos reseñables como la velocidad o la fecha y hora de la transmisión AIS. Estos datos han facilitado otras consultas que nos acercan a los casos reales de análisis que afronta la Armada en el COVAM.

En la Figura 4-1 se resumen los resultados obtenidos durante el trabajo, los diferentes experimentos, así como el margen de error medio que se ha determinado. Kepler se utiliza a modo de visualización del producto del análisis. Se ha podido comprobar que las librerías y la gestión de grandes cantidades de datos son totalmente compatibles, la resolución es más que suficiente para las necesidades de COVAM en cuanto a Conocimiento del Entorno Marítimo. Se resuelve que la aplicación de este sistema de indexado reduce el tiempo de ejecución de las consultas, así como la capacidad de detectar distintos comportamientos en tiempo real. El tiempo de análisis en búsqueda de barcos que cumplan diferentes características en sus registros se reduce hasta en 20 veces como se ha podido comprobar analizando un área determinada. El error obtenido siempre va a depender del nivel de celda utiliza, sin embargo, siempre se usará un número de saltos de celda que aproxime al máximo el área de análisis haciendo que este error no sea muy reseñable.

Por otro lado, se ha comprobado que la búsqueda de celdas por truncado no siempre mostrará los resultados deseados ya que la correspondencia no es perfecta a la hora del agrupamiento de celdas padre e hijo. Sin embargo, el error que podía producir la posición de los vértices en zonas oceánicas del primer nivel de resolución no es mayor inconveniente. Se ha comprobado que los niveles de resolución poseen la precisión necesaria para llevar a cabo las consultas a gran velocidad.

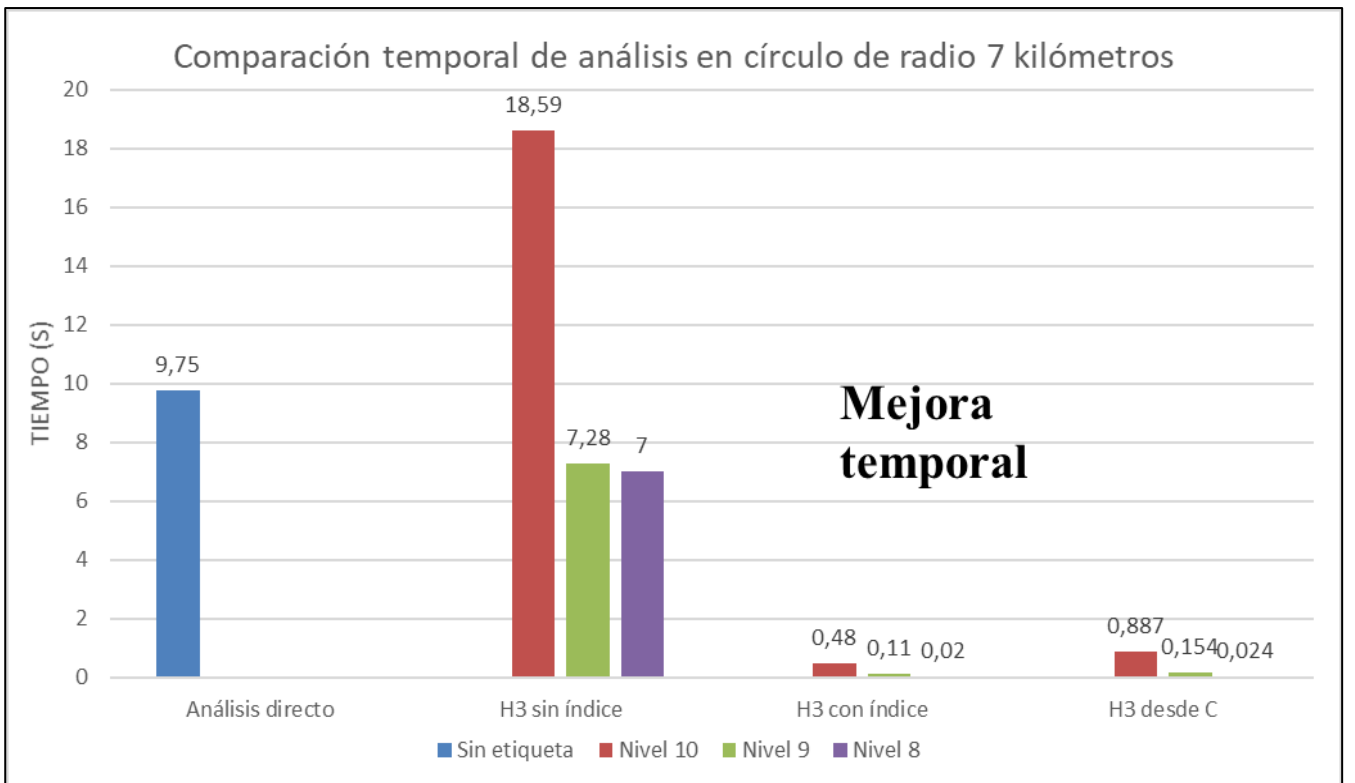
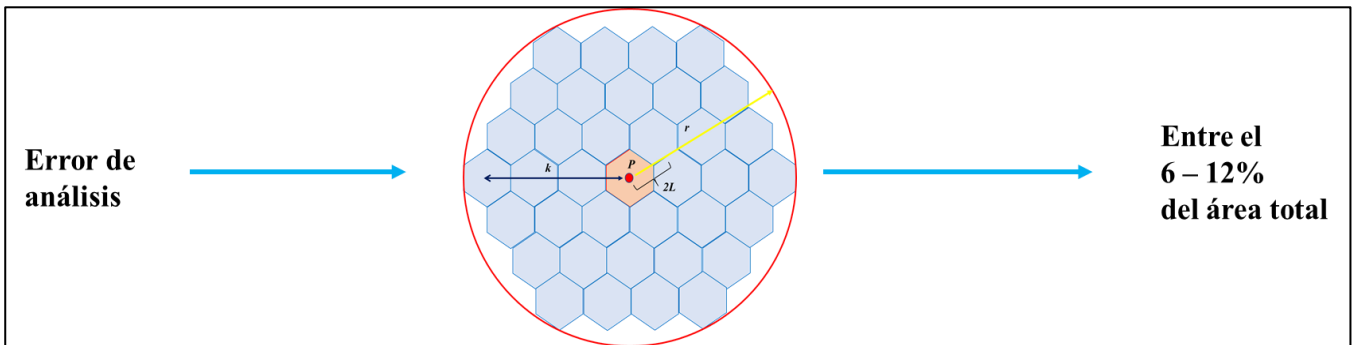
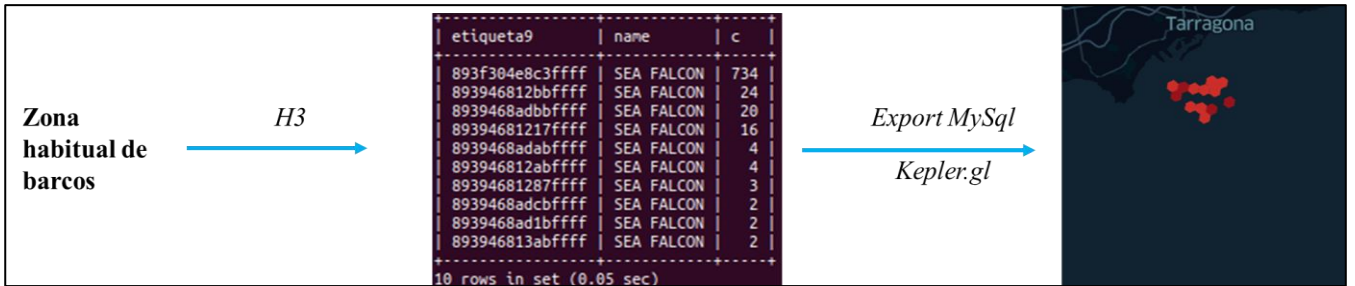
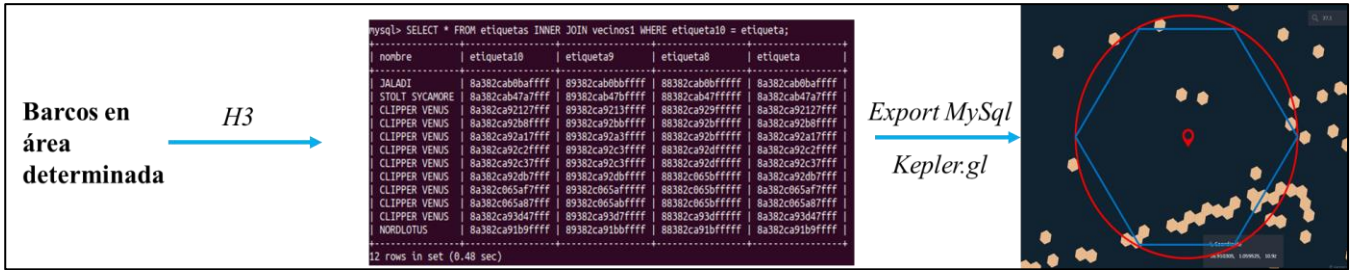


Figura 4-1. Resumen resultados obtenidos.

El caso real del proyecto, en el que se encuadra este trabajo y que ha sido mencionado en capítulos anteriores, requiere de una identificación de situaciones anómalas que se produzcan en la Mar. Es por tanto el indexado de las posiciones una de las maneras de agilizar al máximo este proceso y, como se ha comprobado, una posibilidad de facilitar este trabajo.

Se ha logrado satisfactoriamente tanto la detección de encuentros como la de zonas habituales de buques. El sistema permite localizar las celdas mucho más rápidamente, siendo más que suficiente con niveles de resolución 8 o 9 para sus consultas. Además, se ha podido comprobar cuando los buques se han encontrado con una velocidad variable, diferenciando entre movimientos entre zonas o buques parados. Como bien se sabe, el entorno marítimo es muy complejo y la cinemática de los buques o sus localizaciones durante el periodo de un día no son suficientes para enlazar buques con estos comportamientos, no obstante, es una pieza fundamental.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Revisión de los objetivos

El avance de las tecnologías es continuo y obliga a las instituciones gubernamentales y militares a mantenerse actualizadas y ser capaces de explotarlas en todas sus variantes. Tanto en sus líneas de uso operacional y exclusivo en misiones como en el trabajo del día a día, donde se encuadra el COVAM. La realización de este TFG ha tratado de apoyar el desarrollo tecnológico que las Fuerzas Armadas, y en concreto la Armada, ya están llevando a cabo.

Previamente al desarrollo se realizó un estudio de los sistemas de indexado geoespacial existentes en la actualidad. Tras identificar todas las cualidades, así como las diferentes librerías que utilizaban estos sistemas para su funcionamiento, se seleccionó H3 entre el resto para el desarrollo del trabajo y la realización de las pruebas y experimentos.

Conforme a los objetivos establecidos desde el comienzo de este trabajo se puede determinar que el objetivo general se ha cubierto y cumplido positivamente. Se ha logrado escoger un sistema de indexado geoespacial que facilita el análisis de datos geográficos que recibe el COVAM y además mejora drásticamente el tiempo necesario en las consultas. Por otro lado, se ha comprobado el funcionamiento y la identificación de diferentes escenarios que concluyen de maneras satisfactoria los objetivos secundarios, tales como la detección de zonas habituales o buques dentro de una zona determinada.

La gestión de datos ha jugado un papel fundamental en el desarrollo del trabajo. La capacidad de consulta que aporta MySQL ha apoyado la ejecución de los experimentos, así como la capacidad de utilizar los datos desde programas en C. La base de datos creada es una prueba de la cantidad de gestiones que se pueden realizar, contiene tablas con todos los datos proporcionados por el COVAM, así como la asignación de etiquetas como índices H3 a cada una de las localizaciones de los registros AIS.

Por otro lado, programar en C ha sido necesario para acceder a las funciones contenidas en las librerías de H3, sobre todo a la hora de generar las etiquetas y almacenarlas en la base de datos.

No obstante, los datos que se han utilizado puede que no abarquen el suficiente espacio temporal para determinar las diferentes anomalías en el comportamiento marítimo que propone la propia Armada. Sin embargo, permiten el análisis del sistema de indexado, siendo esta la prioridad.

A modo complementario, a medida que se desarrollaba el trabajo se contempló la posibilidad de representar los índices generados visualmente con alguna aplicación o herramienta de libre acceso. Se comprobó que las funciones de H3 ofrecen la posibilidad de exportar los resultados a archivos KML. Sin embargo, se determinó que la mejor solución era la exportación desde MySQL a archivos CSV de

las consultas o celdas que se quieran representar y utilizar la aplicación de Kepler.gl, cumpliendo un objetivo adicional no contemplado inicialmente.

Concluyendo, desde mi propia opinión como alumno, este trabajo me ha ampliado la visión del avance tecnológico en el que se mueven las Fuerzas Armadas actualmente y en concreto la Armada. Conocer un poco del funcionamiento de una entidad como el COVAM y las tareas en las que se implica día a día, no solo en el mundo militar, sino también en el mundo civil, me ha hecho ver las similitudes entre ambos mundos.

5.2 Líneas futuras

En este trabajo se ha tratado de comprobar como la aplicación de un sistema de indexado espacial es posible a una gran cantidad de datos como son los datos que maneja el COVAM. Además, se ha comprobado como facilita las consultas, la sencillez de aplicación de la librería empleada y la velocidad que aporta a la hora de trabajar con datos espaciales.

Tras la conclusión de este trabajo se proponen diferentes vías como líneas futuras para continuar con proyectos similares:

- Evaluar el error cometido y aplicar un análisis jerárquico a unas zonas determinadas.
- Con las zonas examinadas, incorporar mapas temporales como los creados en Kepler.gl.
- Añadir la incorporación de datos nuevos, bases de datos que se actualicen, con el fin de comprobar el funcionamiento a la hora de ir asignando las celdas formadas por las distintas posiciones geográficas ocupadas.
- Continuar estudiando las diferentes funciones de H3, identificando como se pueden detectar más situaciones en el entorno marítimo.
- Utilizar otras bases de datos, como las bases de datos espaciales estudiadas en el estado del arte o bases de datos de tabla de *hasehes* en memoria, como Redis, y comprobar si aceleran más las consultas.
- Comprobar el funcionamiento con S2 u otro sistema, viendo si existe alguna mejora.
- Utilizar la representación en *dashboards* como Grafana, para juntar este análisis espacial con el resto de datos, representándolos en una interfaz más intuitiva.

6 BIBLIOGRAFÍA

- [1] AJEMA, «Líneas generales de la Armada,» 2017.
- [2] M. p. l. T. E. y. e. R. Demográfico, «www.miteco.gob.es/es,» [En línea]. Available: https://www.miteco.gob.es/es/costas/temas/proteccion-medio-marino/plan-ribera/contaminacion-marina-accidental/trafico_maritimo.aspx#:~:text=Desde%20la%20antig%C3%BCedad%2C%20los%20hombres,de%20mercanc%C3%ADas%20a%20nivel%20internacional.&text=En%20la%20ac. [Último acceso: 2021].
- [3] A. Española, «Tríptico Seguridad Marítima, ENCOMAR,» [En línea]. Available: https://encomar.covam.es/documents/20182/365727/Trifold+Spanish+Navy+MSA+Mission_v2.2.pdf/48157fbf-9271-47c9-84fd-6ce8e04bff39. [Último acceso: 17 Enero 2021].
- [4] Marine Traffic, «Marine Traffic,» [En línea]. Available: <https://www.marinetraffic.com/en/ais/home/centerx:-12.0/centery:25.0/zoom:4>.
- [5] Armada Española, *Plan de Transformación Digital de la Armada*, 2019.
- [6] A. Española, «ENCOMAR,» [En línea]. Available: <https://encomar.covam.es/>. [Último acceso: 17 Enero 2021].
- [7] SAS, «Big Data: Qué es y por qué es importante,» [En línea]. Available: https://www.sas.com/es_es/insights/big-data/what-is-big-data.html. [Último acceso: 19 Enero 2021].
- [8] PowerData, «Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad,» [En línea]. Available: <https://www.powerdata.es/big-data>. [Último acceso: 20 Enero 2021].
- [9] INDRA, «Big Data. Cambio de paradigma,» 23 Septiembre 2014. [En línea]. Available: https://armada.defensa.gob.es/html/especiales/04_semana-naval-madrid/SN_2014/jornadas/J_C_Batanero.pdf. [Último acceso: 20 Enero 2021].
- [10] TodoBI, «Big Data, casos, tecnologías y aplicaciones reales,» [En línea]. Available: <https://todobi.com/big-data-casos-tecnologias-y/>. [Último acceso: Enero 20 2021].

- [11] Marine Digital, «Big Data in Maritime,» [En línea]. Available: https://marine-digital.com/article_bigdata_in_maritime. [Último acceso: 20 Enero 2021].
- [12] M.-E. V. F. O. A. Ioanna Lytra, «A Big Data Architecture for Managing Oceans of Data and Maritime Applications,» 2017. [En línea]. Available: https://www.researchgate.net/publication/322997369_A_big_data_architecture_for_managing_oceans_of_data_and_maritime_applications. [Último acceso: 20 Enero 2021].
- [13] S. Bhattacharjee, «Automatic Identification System (AIS): Integrating and Identifying Marine Communication Channels,» *Marine Navigation*, 25 Diciembre 2020. [En línea]. Available: <https://www.marineinsight.com/marine-navigation/automatic-identification-system-ais-integrating-and-identifying-marine-communication-channels/>. [Último acceso: 6 Febrero 2021].
- [14] COVAM, *Presentación COVAM - ENM*, 2021.
- [15] M. Gallego, «Indra investiga el uso de la Inteligencia Artificial para potenciar la operatividad de la Armada Española,» 31 Enero 2019. [En línea]. Available: <https://bigdatamagazine.es/indra-investiga-el-uso-de-la-inteligencia-artificial-para-potenciar-la-operatividad-de-la-armada-espanola>. [Último acceso: 2021 Enero 20].
- [16] J. A. Micho García, «geotalleres.readthedocs.io/,» 2012. [En línea]. Available: https://geotalleres.readthedocs.io/es/latest/postgis-indexacion-espacial/indexacion_espacial.html#:~:text=La%20indexaci%C3%B3n%20espacial%20es%20una,n%C3%BAmero%20de%20datos%20sean%20eficientes.&text=La%20indexaci%C3%B3n%20organiza%20los%20datos,la%20b%C3. [Último acceso: 18 Enero 2021].
- [17] A. A. Vega, «adrianalonso.es,» 2014. [En línea]. Available: <https://adrianalonso.es/desarrollo-web/indices-geoespaciales-con-mongoddb/#:~:text=MongoDB%20ofrece%20el%20soporte%20para,se%20encuentren%20en%20este%20%C3%A1rea.> [Último acceso: Enero 19 2021].
- [18] Globalmedia, «gmediac.com,» [En línea]. Available: <https://gmediac.com/terminologia-web/significado-de-indexar-en-buscadore-indexacion/>. [Último acceso: Enero 19 2021].
- [19] ArangoDB, «Geo-Spatial Indexes,» [En línea]. Available: <https://www.arangodb.com/docs/stable/indexing-geo.html>. [Último acceso: 21 Enero 2021].
- [20] F. J. L. Pellicer, «Indexación geográfica: la revolucion del topónimo geo-localizador,» [En línea]. Available: <https://www.cervantes.es/imagenes/File/biblioteca/jornadas/Indexacin-geogrifica-version-1.0.pdf>. [Último acceso: 19 Enero 2021].
- [21] R. O. Jesús Rodríguez Lloret, «Introducción a los sistemas de información geográfica,» [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/53645/1/Introducci%C3%B3n%20a%20los%20sistemas%20de%20informaci%C3%B3n%20geogr%C3%A1fica.pdf>. [Último acceso: 19 Enero 2021].
- [22] M. M.-M. I. R.-L. C. B.-E. Neili Machado-García, «Técnicas semánticas en la localización geoespacial de los suelos de Cuba utilizando medidas de similitud espacial,» Diciembre 2016. [En línea]. Available: <http://scielo.sld.cu/pdf/rcta/v25n4/rcta06216.pdf>. [Último acceso: 21 Enero 2021].
- [23] L. McKee, «OGC History (detailed),» 2016. [En línea]. Available: <https://www.ogc.org/ogc/historylong>. [Último acceso: 28 Enero 2021].

- [24] J. H. (OpenGIS), «Simple Feature Access - Part 1: Common Architecture,» 2011. [En línea]. Available: <https://www.ogc.org/standards/sfa>. [Último acceso: 28 Enero 2021].
- [25] Computer Science @SIUe, «Introduction to Spatial Indexing,» 2014. [En línea]. Available: <http://www.cs.siu.edu/~marmcke/docs/cs490/spatialIndexing.html>. [Último acceso: 28 Enero 2021].
- [26] W. Lipscomb, «ResearchGate,» Agosto 2005. [En línea]. Available: https://www.researchgate.net/figure/Generating-geodesic-grids-by-recursive-bisection-and-projection_fig7_307561876. [Último acceso: 28 Enero 2021].
- [27] B. Kunwar, «What are HHCodes and what are they used for?,» 2016. [En línea]. Available: <https://www.careerride.com/oracle-HHCodes.aspx>. [Último acceso: 30 Enero 2021].
- [28] H. Samet, «The Quadtree and Related Hierarchical Data Structure,» Junio 1984. [En línea]. Available: <http://www.cs.umd.edu/~hjs/pubs/SameCSUR84-ocr.pdf>. [Último acceso: 22 Enero 2021].
- [29] Simon Fraser University, «R-Tree,» [En línea]. Available: <https://www2.cs.sfu.ca/CourseCentral/454/jpei/slides/R-Tree.pdf>. [Último acceso: 23 Enero 2021].
- [30] A. Guttman, «R-Trees. A dynamic index structure for spatial searching,» 1984. [En línea]. Available: <http://www-db.deis.unibo.it/courses/SI-LS/papers/Gut84.pdf>. [Último acceso: 2021 Enero 23].
- [31] I. Brodsky, «H3: Uber's Hexagonal Hierarchical Spatial Index,» 27 Junio 2018. [En línea]. Available: <https://eng.uber.com/h3/>. [Último acceso: 21 Enero 2021].
- [32] Uber H3, «Table of Cell Areas for H3 Resolutions,» [En línea]. Available: <https://h3geo.org/docs/core-library/restable>. [Último acceso: 12 Febrero 2021].
- [33] Uber H3, «Overview of the H3 Geospatial Indexing System,» [En línea]. Available: <https://h3geo.org/docs/core-library/overview>. [Último acceso: 15 Febrero 2021].
- [34] S2Geometry, «S2Geometry,» [En línea]. Available: <https://s2geometry.io/>. [Último acceso: 21 Enero 2021].
- [35] E. Veach, «Announcing the S2 Library: Geometry on the Sphere,» 2017. [En línea]. Available: <https://opensource.googleblog.com/2017/12/announcing-s2-library-geometry-on-sphere.html>. [Último acceso: 21 Enero 2021].
- [36] S2Geometry, «S2Geometry: Overview,» [En línea]. Available: <https://s2geometry.io/about/overview>. [Último acceso: 21 Enero 2021].
- [37] O. Procopiuc, «Geometry on the Sphere: Google's S2 Library,» [En línea]. Available: https://docs.google.com/presentation/d/1Hl4KapfAENAO4gv-pSngKwvS_jwNVHRPZTTDzXXn6Q/view#slide=id.i0. [Último acceso: 21 Enero 2021].
- [38] C. S. Perone, «Google's S2, geometry on the sphere, cells and Hilbert curve,» 2015. [En línea]. Available: <https://blog.christianperone.com/2015/08/googles-s2-geometry-on-the-sphere-cells-and-hilbert-curve/>. [Último acceso: 21 Enero 2021].
- [39] G. Niemeyer, «Geohas.org is public!,» 26 Febrero 2008. [En línea]. Available: <https://blog.labix.org/2008/02/26/geohashorg-is-public>. [Último acceso: 31 Enero 2021].

- [40] PubNub, «What is Geohashing?,» 2010. [En línea]. Available: <https://www.pubnub.com/learn/glossary/what-is-geohashing/>. [Último acceso: 31 Enero 2021].
- [41] R. S. Krishna Khadka, «Polygeohasher: an optimized way to create geohashes,» 17 Septiembre 2020. [En línea]. Available: <https://www.geospatialworld.net/blogs/polygeohasher-an-optimized-way-to-create-geohashes/>. [Último acceso: 31 Enero 2021].
- [42] Github, «Notes on Geohashing,» [En línea]. Available: <https://eugene-eeo.github.io/blog/geohashing.html>. [Último acceso: 31 Enero 2021].
- [43] Geohash, «Geohash Converter,» 2008. [En línea]. Available: geohash.co. [Último acceso: 31 Enero 2021].
- [44] Geohash, «Geohash Gofreerange,» [En línea]. Available: <http://geohash.gofreerange.com/>. [Último acceso: 31 Enero 2021].
- [45] J. Kockrow, «Naval Grid Calculator,» [En línea]. Available: navalgrid.com. [Último acceso: 23 Enero 2021].
- [46] G. Helgason, «The War in Maps - The Grid System,» 1 Enero 1995. [En línea]. Available: <https://uboat.net/maps/grid.html>. [Último acceso: 23 Enero 2021].
- [47] Geokov, «UTM - Universal Transverse Mercator,» [En línea]. Available: <http://geokov.com/education/utm.aspx>.
- [48] Arista Sur, «Sistema de Coordenadas Geográficas: UTM,» [En línea]. Available: <https://www.aristasur.com/contenido/sistema-de-coordenadas-geograficas-utm>. [Último acceso: 24 Enero 2021].
- [49] NOAA, «Stories and Tales of the Coast & Geodetic Survey - Technology Tales/Geodetic Surveys in the US The Beginning and the next 100 years,» [En línea]. Available: www.history.noaa.gov. [Último acceso: 24 Enero 2021].
- [50] National Geospatial-Intelligence Agency, «Military Map Reading,» 29 Mayo 2009. [En línea]. Available: <https://earth-info.nga.mil/GandG/coordsys/mmr201.pdf>. [Último acceso: 24 Enero 2021].
- [51] Arc Geek, «Calculadora geodésica,» [En línea]. Available: <https://franzpc.com/apps/conversor-coordenadas-geograficas-utm.html>. [Último acceso: 24 Enero 2021].
- [52] J. A. Cavell, «A Guide to Using the USNG,» 2005. [En línea]. Available: https://archive.amerisurv.com/PDF/TheAmericanSurveyor_USNationalGrid-Cavell_June2005.pdf. [Último acceso: 25 Enero 2021].
- [53] M. Herberts, «Working with Geo Data in Warp 10,» 25 Abril 2019. [En línea]. Available: <https://blog.senx.io/working-with-geo-data-in-warp-10/>. [Último acceso: 30 Enero 2021].
- [54] M. Herberts, «Spatio Temporal Indexing in Warp 10,» 7 Mayo 2019. [En línea]. Available: <https://blog.senx.io/spatio-temporal-indexing-in-warp-10/>. [Último acceso: 31 Enero 2021].
- [55] R. H. Güting, «An Introduction to Spatial Database Systems,» 1994. [En línea]. Available: <http://dna.fernuni-hagen.de/papers/IntroSpatialDBMS.pdf>. [Último acceso: 21 Enero 2021].
- [56] B. B. Martínez, «Introducción a los sistemas de gestión de bases de datos,» de *Informática para la ingeniería*, 2020.

- [57] Oracle, «What is MySQL?,» [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Último acceso: 1 Febrero 2021].
- [58] Oracle, «The Main Features of MySQL,» [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/features.html>. [Último acceso: 01 Febrero 2021].
- [59] IBM Informix, «Getting started with spatial data,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.spatial.doc/ids_spat_008.htm. [Último acceso: 23 Enero 2021].
- [60] IBM Informix, «Syntax for creating an R-tree index,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.spatial.doc/ids_spat_074.htm. [Último acceso: 2 Febrero 2021].
- [61] Amazon Web Services, «¿Qué es Amazon Aurora?,» 2021. [En línea]. Available: https://docs.aws.amazon.com/es_es/AmazonRDS/latest/AuroraUserGuide/CHAP_AuroraOverview.html. [Último acceso: 31 Enero 2021].
- [62] S. Chandrasekaran, «Amazon Aurora under the hood: indexing geospatial data using Z-order curves,» 22 Enero 2018. [En línea]. Available: <https://aws.amazon.com/es/blogs/database/amazon-aurora-under-the-hood-indexing-geospatial-data-using-z-order-curves/>. [Último acceso: 2021 Enero 31].
- [63] F. Naudé, «Oracle Spatial Data Option FAQ,» 17 Octubre 1999. [En línea]. Available: <http://www.blacksheepnetworks.com/security/resources/oracle/faqsdo.htm>. [Último acceso: 30 Enero 2021].
- [64] Oracle, «ORACLE SPATIAL 11g : Administración avanzada de Datos Espaciales para la Empresa,» 2007. [En línea]. Available: <https://www.oracle.com/technetwork/es/documentation/317501-esa.pdf>. [Último acceso: 30 Enero 2021].
- [65] Oracle, «Spatial Analytics with Oracle Database,» 26 Noviembre 2019. [En línea]. Available: <https://www.oracle.com/a/tech/docs/sg-spatial-twp-19c.pdf>. [Último acceso: 30 Enero 2021].
- [66] S. Sathyanarayanan, «Ten reasons why We Should Use Linux,» Open Source For U, 30 Marzo 2020. [En línea]. Available: <https://www.opensourceforu.com/2020/03/reasons-to-use-linux/>. [Último acceso: 6 Febrero 2021].
- [67] GitHub, «H3: A Hexagonal Hierarchical Geospatial Indexing System,» 6 Octubre 2020. [En línea]. Available: <https://github.com/uber/h3>. [Último acceso: 7 Febrero 2021].
- [68] G. Macrì, «Kepler.gl - Uber H3 layer,» Enero 2020. [En línea]. Available: <https://docs.kepler.gl/docs/user-guides/c-types-of-layers/j-h3>. [Último acceso: 3 Marzo 2021].
- [69] «Kepler.gl,» [En línea]. Available: <https://kepler.gl/demo>. [Último acceso: 24 Febrero 2021].

ANEXO I: ETIQUETA10.C

En este anexo se incluye el código completo para generar las etiquetas de nivel 8, 9 y 10. Contiene la descripción en cada línea y muestra el acceso a la base de datos y el uso de las distintas funciones de H3 para el cálculo de etiquetas.

```
#include <stdio.h>
#include <stdlib.h>
#include <mysql/mysql.h> //Librerías mysql, h3
#include <h3/h3api.h>
#include <math.h>
#include <inttypes.h>
#define R 6371.0088 // radio de la Tierra

#define HOST "localhost"
#define USER "victor" //Datos de entrada a la base de datos "barcos"
#define PASS "tfg"
#define DATABASE "tfg"

int main (int argc, char *argv[]){
    MYSQL *con;
    MYSQL_ROW row;
    MYSQL_RES *res;
    float lat;
    float longi;
    char name [25];
    char q [1024];
    con = mysql_init(NULL);
    if (!mysql_real_connect(con, HOST, USER, PASS, DATABASE, 0, NULL, 0)){
//COMPROBACIÓN CONEXIÓN A LA BASE DE DATOS
        fprintf(stderr, "%s\n", mysql_error(con));
        return 1;
    }

    mysql_query(con, "SELECT latitude, longitude, name FROM aismediterraneo");
    res = mysql_store_result(con);

    while((row = mysql_fetch_row(res)) !=0){ // SE RECORREN LOS RESULTADOS DE LA
CONSULTA
        printf("Latitud: %s\nLongitud: %s\n Nombre: %s\n", row[0], row[1],
row[2]);
        lat = row [0] ? atof(row[0]) : 0.0;
        longi = row [1] ? atof(row[1]) : 0.0;
        sprintf(name, "%s", row[2]);

        GeoCoord location;
        location.lat = degsToRads(lat); //CODIGO H3 , GENERACIÓN DE ETIQUETAS
SEGÚN RESOLUCION
        location.lon = degsToRads(longi);
        int resolution = 10;
        H3Index indexed = geoToH3(&location, resolution);
        printf("The index 10 is: %" PRIx64 "\n", indexed);

        int resolution1 = 9;
        H3Index indexed1 = geoToH3(&location, resolution1);
        printf("The index 9 is: %" PRIx64 "\n", indexed1);

        int resolution2 = 8;
        H3Index indexed2 = geoToH3(&location, resolution2);
        printf("The index 8 is: %" PRIx64 "\n", indexed2);
```

```
        sprintf(q, "INSERT INTO etiquetas (nombre, etiqueta10, etiqueta9,
etiqueta8) VALUES('%s', '%lx', '%lx', '%lx') ",name, indexed, indexed1,
indexed2);
        mysql_query(con, q); //SE GUARDAN LAS ETIQUETAS EN OTRA TABLA MYSQL
    }
    mysql_free_result(res);
    mysql_close(con);

    return 0;
}
```


ANEXO II: RADIO_EXP3.C

En este anexo se incluye el código utilizado para calcular los buques en un área determinada. Los códigos de los otros dos programas utilizados (radio_exp2.c y radio_exp1.c) son similares. En este programa se generan puntos aleatorios además de contar con un cronómetro.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <mysql/mysql.h> //Librerías mysql, h3
#include <h3/h3api.h>
#include <math.h>
#include <inttypes.h>
#include <unistd.h>
#include <sys/time.h>

#define HOST "localhost"
#define USER "victor" //Datos de entrada a la base de datos "barcos"
#define PASS "tfg"
#define DATABASE "tfg"

long long cronomsec(int startstop) //Función cronómetro en milisegundos
{
    static long long pre_time;
    struct timeval tv;

    if (startstop)
    {
        gettimeofday(&tv, NULL);
        pre_time=tv.tv_sec*1000+tv.tv_usec/1000;
    }
    else{
        gettimeofday(&tv, NULL);
        return tv.tv_sec*1000+tv.tv_usec/1000 - pre_time;
    }
    return 0;
}

int main(int argc, char* argv[]) {
    MYSQL *con;
    MYSQL_ROW row;
    MYSQL_RES *res;
    con = mysql_init(NULL);
    if (!mysql_real_connect(con, HOST, USER, PASS, DATABASE, 0, NULL, 0)){
//Comprobación de conexión MySQL
        fprintf(stderr, "%s\n", mysql_error(con));
        return 1;
    }
    // Generación de posición aleatorios, ÚNICAMENTE EN EL MAR MEDITERRÁNEO
    srand (time(NULL));
    double lat;
    double longi;
    lat = 30.25 +(double) rand() / (double) RAND_MAX * 15.55; //Latitud entre
30.25° N y 45.8° N
    longi = -5.61 +(double) rand()/ (double) RAND_MAX * 41.83; // Longitud entre
5.61° W y 36.22° E
    printf("Coordenadas punto aleatorio:\n Latitud: %f, longitud: %f \n", lat,
longi);
    int resolution;
    printf("Inserte resolución: ");
    scanf("%d", &resolution);
    GeoCoord location;
```

```

location.lat = degsToRads(lat); //Obtención de índice H3
location.lon = degsToRads(longi);

H3Index indexed = geoToH3(&location, resolution);
printf("Índice: %" PRIx64 "\n", indexed);

GeoCoord center;
h3ToGeo(indexed, &center);
printf("Centro de coordenadas: %lf, %lf\n", radsToDegs(center.lat),
      radsToDegs(center.lon));

int k;
k= rand() % 60; //k, número de saltos, toma valores entre 1 y 60
printf("Nº saltos aleatorio: %d \n", k);
int maxNeighboring = maxKringSize(k);
H3Index* neighboring = calloc(maxNeighboring, sizeof(H3Index));
kRing(indexed, k, neighboring); //Funciones H3, se obtiene las celdas vecinas
(neighboring) según el resultado (indexed)
char vecinos [100000] = ""; //Cadena para almacenar todas las celdas
resultantes
char puntero[16]; //Variable que recoge cada celda obtenida
char q[1000000];
printf("Celdas ocupadas en el área:\n");
for (int i = 0; i < maxNeighboring; i++) {
    if (neighboring[i] != 0) {
        sprintf(puntero, "%lx", neighboring[i]);
        strcat(vecinos, "\\");
        strcat(vecinos, puntero); // Se concatenan todos los valores
        strcat(vecinos, "\\, ");
    }
}
vecinos[strlen(vecinos)-2]='\0'; //Se elimina el ultimo
espacio y coma para que cumpla la sintaxis de MySQL

sprintf(q, "SELECT nombre, etiqueta10 FROM etiquetas WHERE etiqueta10
in (%s)", vecinos); //Se guarda en q la sentencia de MySQL, en este caso para un
nivel de resolución 10
cronomsec(1); //Cronómetro a 0 y empieza a contar
mysql_query(con, q); //Query MySQL
res = mysql_store_result(con);
while((row = mysql_fetch_row(res)) !=0){
    printf("Nombre: %s \n Etiqueta en área: %s \n", row[0], row[1]);
}
//Recorre los resultados
printf("Transcurrido: %lld miliseq \n", cronomsec(0)); //Tiempo en
cronometro

free(neighboring);

mysql_free_result(res);
mysql_close(con);

return 0;
}

```

ANEXO III: CONSULTA MYSQL

Se adjunta la captura de la consulta reseñada en el desarrollo del trabajo.

nombre	etiqueta10	etiqueta9	etiqueta8
KIKYO	8a382c14015ffff	89382c14017ffff	88382c1401fffff
VINJERAC	8a382c10500ffff	89382c10507ffff	88382c1051fffff
NA	8a382ca81587fff	89382ca815bffff	88382ca815fffff
KM& EXCELSUS	8a382ca81997fff	89382ca819bffff	88382ca819fffff
MAERSK PITTSBURGH	8a382ca882c7fff	89382ca882fffff	88382ca883fffff
MAERSK PITTSBURGH	8a382ca8874ffff	89382ca887bffff	88382ca887fffff
ALONDRA	8a382ca88c2ffff	89382ca88c3ffff	88382ca88dfffff
MAERSK PITTSBURGH	8a382ca89217fff	89382ca8923ffff	88382ca893fffff
MAERSK PITTSBURGH	8a382ca89257fff	89382ca8927ffff	88382ca893fffff
MAERSK PITTSBURGH	8a382ca895affff	89382ca895bffff	88382ca895fffff
KM& EXCELSUS	8a382ca8c95ffff	89382ca8c97ffff	88382ca8c9bfffff
KM& EXCELSUS	8a382ca8d90ffff	89382ca8d97ffff	88382ca8d9fffff
ALONDRA	8a382ca8ec17fff	89382ca8ec3ffff	88382ca8edfffff
CLIPPER VENUS	8a382ca9109ffff	89382ca910bffff	88382ca917fffff
NORDLOTUS	8a382ca91b9ffff	89382ca91bbffff	88382ca91bfffff
CLIPPER VENUS	8a382ca92127fff	89382ca9213ffff	88382ca929fffff
CLIPPER VENUS	8a382ca92a17fff	89382ca92a3ffff	88382ca92bfffff
CLIPPER VENUS	8a382ca92b8ffff	89382ca92bbffff	88382ca92bfffff
CLIPPER VENUS	8a382ca92c2ffff	89382ca92c3ffff	88382ca92dfffff
CLIPPER VENUS	8a382ca92c37fff	89382ca92c3ffff	88382ca92dfffff
CLIPPER VENUS	8a382ca92db7fff	89382ca92dbffff	88382ca92bfffff
CLIPPER VENUS	8a382ca938b7fff	89382ca938bffff	88382ca939fffff
CLIPPER VENUS	8a382ca93d47fff	89382ca93d7ffff	88382ca93dfffff
THETIS	8a382ca9831ffff	89382ca9833ffff	88382ca983fffff
CLIPPER VENUS	8a382ca98837fff	89382ca9883ffff	88382ca989fffff
CLIPPER VENUS	8a382ca98b07fff	89382ca98b7ffff	88382ca98bfffff
CAPE GUINEA	8a382ca9a917fff	89382ca9a93ffff	88382ca9a9fffff
ALDEBARAN	8a382ca9b52ffff	89382ca9b53ffff	88382ca9b5fffff
CAPE GUINEA	8a382ca9b937fff	89382ca9b93ffff	88382ca9b9fffff
CLIPPER VENUS	8a382ca9c087fff	89382ca9c0bffff	88382ca9c7fffff
CLIPPER VENUS	8a382ca9c0f7fff	89382ca9c0ffff	88382ca9c7fffff
CAPE GUINEA	8a382ca9e00ffff	89382ca9e03ffff	88382ca9e7fffff
CLIPPER VENUS	8a382ca9e817fff	89382ca9e83ffff	88382ca9e9fffff
CLIPPER VENUS	8a382ca9e857fff	89382ca9e87ffff	88382ca9e9fffff
CLIPPER VENUS	8a382ca9eb47fff	89382ca9eb7ffff	88382ca9ebfffff
CLIPPER VENUS	8a382ca9ed27fff	89382ca9ed3ffff	88382ca917fffff
CLIPPER VENUS	8a382ca9ed2ffff	89382ca916ffff	88382ca9edfffff
SEABRAVERY	8a382caa12b7fff	89382caa12bffff	88382caa13fffff
VINJERAC	8a382caa4b1ffff	89382caa4b3ffff	88382caa4bfffff
VINJERAC	8a382caa4b57fff	89382caa4b7ffff	88382caa4bfffff
JALADI	8a382caa80a7fff	89382caa80bffff	88382caa81fffff
STOLT SYCAMORE	8a382caa921ffff	89382caa92fffff	88382caa93fffff
KM& EXCELSUS	8a382caab057fff	89382caab07ffff	88382caab7fffff
KM& EXCELSUS	8a382caab79ffff	89382caab7bffff	88382caab7fffff
JALADI	8a382cab0baffff	89382cab0bbffff	88382cab0bfffff
STOLT SYCAMORE	8a382cab47a7fff	89382cab47bffff	88382cab47fffff
VINJERAC	8a382cb93147fff	89382cb9317ffff	88382cb931fffff
VINJERAC	8a382cb98d5ffff	89382cb98d7ffff	88382cb98dfffff

48 rows in set (0.00 sec)

Figura A3-1. Análisis con etiquetas truncadas.

ANEXO IV: EXTRACTO CELLS8.KML

Este anexo contiene un extracto del archivo KML generado con H3.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2"
xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
  <name>title</name>
  <description>h3 cells</description>
  <Style id="s_circle_hl">
    <IconStyle>
      <scale>1.3</scale>
      <Icon>
<href>http://maps.google.com/mapfiles/kml/shapes/placemark_circle.png</href>
      </Icon>
      <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>
    </IconStyle>
    <LabelStyle>
      <color>ff0000ff</color>
      <scale>2</scale>
    </LabelStyle>
  </Style>
  <StyleMap id="m_ylw-pushpin">
    <Pair>
      <key>normal</key>
      <styleUrl>#s_circle</styleUrl>
    </Pair>
    <Pair>
      <key>highlight</key>
      <styleUrl>#s_circle_hl</styleUrl>
    </Pair>
  </StyleMap>
  <Style id="s_circle">
    <IconStyle>
      <scale>1.1</scale>
      <Icon>
<href>http://maps.google.com/mapfiles/kml/shapes/placemark_circle.png</href>
      </Icon>
      <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>
    </IconStyle>
    <LabelStyle>
      <color>ff0000ff</color>
      <scale>2</scale>
    </LabelStyle>
  </Style>
<Placemark>
<name>8818400013ffff</name>
  <styleUrl>#lineStyle1</styleUrl>
  <LineString>
    <tessellate>1</tessellate>
    <coordinates>
      -3.538523,46.907413,5.0
      -3.544846,46.905834,5.0
      -3.545760,46.901080,5.0
      -3.540352,46.897904,5.0
      -3.534030,46.899482,5.0
      -3.533115,46.904236,5.0
    </coordinates>
  </LineString>
</Placemark>
</Document>
</kml>
```

```
    -3.538523,46.907413,5.0  
  </coordinates>  
</LineString>  
</Placemark>  
<Placemark>
```