Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Guadalajara

Escuela de Graduados en Ingeniería y Arquitectura (EGIA)

Maestría en Ciencias Computacionales

Extracción en tiempo real de grandes volúmenes de información y análisis de Big Data basado en el sistema de posicionamiento Glonass

AUTOR: **Julio Cesar Roa Gil**

ASESOR: Liliana Ibeth Barbosa Santillán

Guadalajara (Jal), 26 de Noviembre de 2014

Dedicatoria

Quiero dedicar esta tesis a mi madre, gracias a quien pude estudiar la maestría y me apoyó incondicionalmente en todos los momentos,
a mis abuelas y a mis tías que me ayudaron durante todo este tiempo,
a mis mejores amigos que me hicieron la vida imposible,
y en fin, a los amigos y compañeros con quienes me divertí bastante.

Agradecimientos

Quisiera reconocer, antes que nada, el apoyo, fortaleza y paciencia de mi asesor de tesis de maestría, la Dra. Liliana Barbosa. Me beneficié mucho de las charlas que sostuve con ella durante mis estudios. También quisiera agradecer a otros miembros del comité supervisor de tesis por su tiempo y atención.

Quisiera agradecer a mis compañeros de maestría con los cuales pasamos momentos agradables y a los miembros de la Maestría en Ciencias de la Computación del ITESM Campus Guadalajara por su esfuerzo y ayuda en que este programa tenga altos estándares de calidad.

Resumen

El fenómeno de Big Data ha impulsado durante los últimos años una revolución en torno a los datos y su ventaja competitiva en el ámbito empresarial y científico a través de su análisis. Cuando hablamos de Big Data implícitamente nos referimos a grandes volúmenes de información, con una gran variedad de fuentes de información y que se generan a grandes velocidades. Dentro de esta gran variedad de fuentes de información podemos encontrar las redes sociales, las bases de datos tradicionales, los Data Warehousing, los sensores de diferentes dispositivos, los satélites, entre otros.

En este trabajo, la principal fuente de información que se usará es la del conjunto de satélites rusos que componen el sistema Glonass, la cual se empleará para el análisis de Big Data, y su posterior visualización. El sistema Glonass, es uno de los tres principales sistemas de posicionamiento a nivel mundial; otros dos más conocidos son el sistema GPS de procedencia estadounidense y Galileo de procedencia europea. La propuesta de tesis está basada en implementar un método de extracción en tiempo real y almacenamiento de datos, como también, en realizar un análisis e implementación de Big Data a través de diferentes procesos de minería de datos, como el agrupamiento, a partir de los archivos de datos obtenidos de los distintos transmisores del sistema de posicionamiento ruso Glonass.

Los resultados obtenidos han sido la creación de una aplicación ETL para el procesamiento de los datos que funciona tanto en modo secuencial como en paralelo para optimizar el uso de los recursos de hardware, y así mismo el descubrimiento de patrones que se encontraban en el meta-modelo de la base de datos. También el meta-modelo permitió responder a diferentes preguntas sobre los datos almacenados, como la cantidad de fallas y anomalías encontradas.

Contenido

Lista de Tablas

Lista de Figuras

CAPÍTULO 1

Introducción

1.1 Antecedentes

Desde hace un par de años el término de Big Data es utilizado para designar los grandes volúmenes de datos, tanto estructurados como no estructurados, que se están generando en la Sociedad de la Información y el Conocimiento y que, por su tamaño y heterogeneidad, plantean grandes dificultades para ser procesados por el software y los sistemas de gestión de bases de datos tradicionales.

En general, la información que circula por Internet como los textos, documentos, fotografías y vídeos; los grafos sociales (*social networks*); los contenidos sociales aportados por los usuarios (*social data*); los datos de los dispositivos móviles, los datos de los diferentes tipos de satélites, los datos de las redes de sensores y los *RFID* (identificación por radiofrecuencia) [1]; los registros de las actividades de los sitios Web y la indexación de las búsquedas en Internet; la información científica en temas como la astronomía, meteorología, genómica, bioquímica, biológica y otros datos complejos de la investigación científica interdisciplinaria; los registros médicos; la vigilancia militar y policial; los datos generados por las administraciones públicas (*open data*); los datos de las transacciones en los mercados financieros; o los datos de la actividad relacionada con el comercio electrónico, entre otros. Son estos el conjunto o la combinación de todos ellos los que aportan al universo

del Big Data.

Asimismo, el tratamiento de los grandes volúmenes de datos y contenidos plantea nuevos retos tecnológicos para procesarlos de forma eficiente en un tiempo razonable. Esto va a requerir avanzar en las tecnologías para el procesamiento paralelo masivo de bases de datos (*MPP*); en la computación en la nube (*cloud computing*); en los sistemas escalables de almacenamiento; y en otros campos relacionados con los sistemas de archivos y bases de datos distribuidas o en los sistemas de minería de datos (*data mining*). Sin olvidar otras cuestiones de gran calado que pueden afectar la privacidad de las personas como son los criterios éticos y la protección de los datos personales en la explotación y cruce de los datos de diferentes fuentes.

Hay cinco formas generales en que el uso de grandes volúmenes de datos puede crear valor. Primero, los grandes volúmenes de datos dan un valor significativo al hacer que la información sea transparente y útil en una frecuencia mucho mayor. En segundo lugar, como las organizaciones crean y almacenan más datos transaccionales en formato digital, pueden recoger información más precisa y detallada sobre el desempeño de todo, desde los inventarios de productos hasta los días de enfermedad, y por lo tanto exponer la variabilidad y mejorar el rendimiento. Las empresas líderes están utilizando la recolección y análisis de datos para llevar a cabo experimentos controlados para tomar mejores decisiones de gestión.

En tercer lugar, el Big Data permite encontrar una relación más estrecha entre los clientes y los productos o servicios, lo que nos resulta, es una medida mucho más precisa. En cuarto lugar, los análisis sofisticados pueden mejorar sustancialmente la toma de decisiones. Por último, el Big Data se puede usar para mejorar el desarrollo de la próxima generación de productos y servicios. Por ejemplo, los fabricantes están utilizando datos obtenidos de sensores integrados en los productos innovadores para crear ofertas de servicios de post-venta como mantenimiento preventivo (medidas preventivas que se llevan a cabo antes de que ocurra un fallo) [2].

En los últimos años, los inversionistas privados y las empresas de capital de riesgo han invertido cientos de millones de dólares en nuevas empresas que desarrollan nuevas tecnologías para recopilar, almacenar, organizar y analizar volúmenes de datos estructurados y no estructurados a una escala de petabytes. Sin embargo, el Big Data es una potente herramienta para la experimentación, el análisis y la toma de decisiones. Es una oportunidad para experimentar en tiempo real rompiendo con las barreras de los costes y el tiempo requerido en obtener los datos, porque éstos están ahí, de forma masiva, para su explotación. Desde los comportamientos de los consumidores, tal como se ha señalado, hasta los temas reales que preocupan a los ciudadanos en diversos ámbitos, o los comportamientos de todos los agentes que intervienen en los procesos de negocios.

1.2 Problemática

La recolección de los datos no es el principal problema, el que hacer con estos volúmenes de información es el reto de la industria. El reto fundamental de los grandes volúmenes de datos de diferentes fuentes es el encontrar nuevas utilidades que antes no se habían evidenciado. El desafío para las empresas es el desarrollo de métodos que permitan obtener el verdadero valor de esa mina de terabytes de datos.

De manera un poco más específica se presentan todo tipo de problemas en las diferentes etapas o procesos que involucran al Big Data. Por ejemplo, en la etapa de extracción, el proceso de almacenamiento o el proceso de extracción en tiempo real de diferentes fuentes de información, presentan grandes retos al relacionar variables como la velocidad, el volumen y la variedad de los datos a extraer. Algo similar acontece con el pre-procesamiento de los datos, en donde la capacidad de hardware juega un papel de vital importancia, al poder ejecutar una "limpieza" de la información en el menor tiempo posible.

Para el análisis y visualización de la información se hace necesario presentarla de la manera más fácil y sencilla de entender por cualquier persona, y en este punto, es donde

implica un reto al utilizar técnicas y metodologías que resuman y muestren la información de forma clara y precisa.

Cuando se habla de problemas referentes al Big Data es importante mencionar que para que se pueda clasificar o catalogar como un problema de Big Data se deben cumplir con las siguientes cuatro dimensiones o características:

- **Volumen:** Hoy en día los datos son generados por computadoras, redes e interacciones humanas, como el *social data*. Es el tamaño o escala de los datos expresado en cantidades de miles de Gigabytes, o en Terabytes, Pentabytes, Exabytes o en Zetabytes.

- **Velocidad:** Se refiere a la velocidad en que se generan los datos, por ejemplo 10 TB por hora. También describe la velocidad con que se pueden analizar los datos. Para ser un poco más precisos es la información generada en Tiempo Real o *Real Time*.

- **Variedad:** Desde datos estructurados hasta datos no estructurados, como por ejemplo, audio, video, texto, imágenes, sensores, etc. Es la variedad de fuentes de datos que se encuentran ya sea interna como en una organización o externa, como internet o una combinación de ambas.

- **Veracidad:** Esta dimensión es la más importante para obtener un resultado más preciso y confiable, dado que hace referencia a la fuente u origen de los datos, pues esta debe de ser una fuente real de información y no una fuente ficticia.

1.3 Entorno, delimitación y definición del problema

La delimitación del tema propuesto para la presente tesis puede quedar resumida al siguiente objetivo:

Implementar un método óptimo de almacenamiento para el proceso de extracción de información del sistema de posicionamiento global Glonass en tiempo real, el cual permita posteriormente realizar un análisis de Big Data a partir de la información previamente

almacenada.

El propósito específico de la propuesta de tesis es proponer un método de almacenamiento que permita administrar de manera eficiente los recursos de hardware para realizar la extracción de información del sistema de posicionamiento Glonass en tiempo real. Este método contemplará el almacenamiento de los metadatos y de los datos de manera independiente.

Una vez almacenada la información se realizará un análisis de Big Data a través de técnicas de minería de datos que nos permitan extraer el conocimiento implícito que se encuentra en los datos. Para esto se extraerá un volumen aproximado o igual a los 100 Gigabytes (GB).

El análisis o aplicación del estudio de los datos será definido una vez que se cuente con la información almacenada, dado que *a priori*, no se puede definir el segmento de mercado o nicho de investigación. Esto se debe que al momento de la extracción en tiempo real se encuentran multitud de peticiones en los satélites y no se cuenta con una forma explícita de definir qué tipo de información se quiere descargar.

1.4 Objetivos

1.4.1 Objetivo general

- Extraer información (en volumen de Big Data) en tiempo real basado en el sistema de posicionamiento Glonass.

1.4.2 Objetivos específicos

- Implementar un método para el almacenamiento de los datos.
- Identificar la estructura de datos extraídos.

- Implementar un modelo de datos para los metadatos de la información extraída.
- Proponer una arquitectura apta para el almacenamiento de la información en tiempo real.
- Identificar una aplicación de Big Data de los datos extraídos.
- Utilizar técnicas de minería de datos para el análisis e identificación de conocimiento de los datos extraídos.
- Realizar un análisis de Big Data a partir de los datos obtenidos.

1.5 Justificación

Una de las principales tendencias hoy en día es el Big Data. Al ser una tendencia nueva tanto en el campo empresarial como en el investigativo no se cuentan con muchos profesionales que conozcan a profundidad sobre el tema.

También es poco común que en Latinoamérica se trabaje con los sistemas de posicionamiento global, y menos si este sistema no es tan conocido por los profesionales del área de tecnología, como lo es GLONASS, lo que propone un mayor grado de complejidad en la investigación.

Además, es una oportunidad única en la que se relacionan temas como infraestructura, hardware, bases de datos, extracción de datos en tiempo real, almacenamiento de la información, datos no estructurados, arquitectura de software, internet y soluciones de Cloud Computing.

Este proyecto permite adquirir experiencia y desarrollar habilidades necesarias para llegar a convertirse en un "científico de datos", el cual ha sido catalogado como "el trabajo más sexy del siglo XXI"[3].

En general, este proyecto abarca un desafio en el área informática al reunir una gran cantidad de conceptos y tecnologías, y da un avance más en la investigación de aplicaciones y análisis de Big Data haciendo uso de los sistemas de posicionamiento global.

1.6 Hipótesis

En esta sección se presentan las hipótesis hechas en el presente trabajo de tesis.

- **H1:** La extracción y almacenamiento en tiempo real de grandes volúmenes de información es posible mediante una aplicación receptora y una estructura de directorios bien definida.
- **H2:** La estructura de los datos está basada en el formato RINEX.
- **H3:** Al menos tres usos significativos se le puede dar a los datos descargados.
- **H4:** El porcentaje de errores en los archivos descargados no ha de ser mayor al 2 %.

1.7 Metodología

En la presente sección se explicarán de forma detallada las distintas fases por las que ha ido avanzando esta investigación, describiendo la metodología propuesta, basada en una combinación de la investigación cuantitativa y el método para la gestión de proyectos informáticos, *MERISE* [4], adaptada a las necesidades propias del proyecto. Además se usará la metodología *SCRUM* para el seguimiento y control durante todo el proceso de la elaboración de la tesis.

MERISE es un método de concepción, de desarrollo y de realización de proyectos informáticos. La meta de este método es llegar a realizar un sistema de información. El método está basado en la separación de los datos y de los procedimientos a efectuarse en más modelos conceptuales y físicos. La separación de los datos y los procedimientos asegura una vida más larga del modelo.

Las fases de la metodología son:

- Estudio preliminar (fase de planificación).
- Estudio detallado (fase de análisis y diseño de la solución).
- Implementación y puesta en marcha (fase de desarrollo y producción).

En cuanto a *SCRUM* [5], es un marco de trabajo para la gestión y desarrollo de software enfocado en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. *SCRUM* es un modelo de referencia que determina un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

1.7.1 Estrategia de investigación

El siguiente esquema muestra el proceso que se ha seguido y se seguirá:

- Formulación del problema. (Abordado en el capítulo 1).
    - Introducción
    - Antecedentes
    - Problemática
    - Definición de los objetivos
    - Alcance y delimitación
- Fase exploratoria. (Abordado en el capítulo 2, Estado de la Técnica)
    - Elaboración del marco teórico
        - Revisión de la literatura
        - Extracción y recopilación de la información
    - Construcción y redacción del marco teórico
- Diseño de la investigación

- Estudio exploratorio
- Formulación de la hipótesis y detección de variables

- Recopilación de datos mediante la metodología MERISE
  - Estudio preliminar
    - Análisis de situación actual
    - Propuesta de solución global
  - Estudio detallado
    - Definición funcional de la situación
  - Implementación
    - Distribución de datos y tratamiento
    - Codificación y verificación de los programas
  - Realización y puesta en marcha
    - Implementación de medios técnicos
    - Implementación de medios organizativos

- Trabajo de gabinete
  - Presentación de los datos
  - Estructura del informe
  - Referencias y Bibliografía

CAPÍTULO 2

Marco Teórico

2.1 Estado del arte

En este capítulo se describe el estado del arte con un preámbulo de los primeros sistemas de posicionamiento y luego se hace referencia a la aplicación de técnicas de minería de datos basados en datos de los diferentes sistemas de posicionamiento. También se mencionan técnicas de minería de datos aplicadas a optimizar el almacenamiento de información en tiempo real, y varias aplicaciones del uso intensivo de datos extraídos de los satélites de posicionamiento.

Además se nombran varios de los centros de investigación que actualmente están trabajando o han trabajado a partir de los datos de los sistemas de georreferenciación, como también los diversos modelos de negocio que surgen de estos datos.

2.1.1 Los primeros sistemas de posicionamiento

El antecedente inmediato de los sistemas de posicionamiento modernos de hoy en día es el sistema de navegación satelital marina (NNSS), también llamado Sistema de Tránsito o *Transit System*. Este sistema se concibió a finales de 1950 y se desarrolló en la década de 1960 por el servicio militar de EE.UU., sobre todo, para determinar las coordenadas y tiempo de los buques en el mar y para uso militar en tierra. El uso civil de este sistema de satélites fue finalmente liberado a finales de 1964 para su manipulación; el sistema se

utiliza en todo el mundo tanto para la navegación como la topografía.

La primera oferta razonable en el uso de satélites para la navegación nació durante la investigación de la posible aplicación de las tecnologías de radio - astronomía para aeronavegación dirigido por el Prof. V.S. Shebshaevich, en la Academia Militar de Ingeniería de Leningrado Mozhaiskii en 1957.

Las investigaciones adicionales para aumentar la precisión de las definiciones de navegación, el apoyo mundial, la aplicación diaria y la independencia de las condiciones meteorológicas, permitieron el desarrollo del sistema *Tsikada* ruso (o también conocido como Cicada) que transmite las mismas dos frecuencias portadoras como Tránsito, y es similar a la misma con respecto a las precisiones alcanzables. Diez satélites de órbita baja se desplegaron en dos constelaciones complementarias: uno militar compuesto por una constelación de seis satélites, y una red civil compuesta por una constelación de cuatro satélites. Al contrario de Tránsito, el sistema *Tsikada* sigue funcionando.

2.1.2 Los sistemas de posicionamiento: actuales y futuros

El sistema de navegación con tiempo y rango (NAVSTAR), y el Sistema de Posicionamiento Global (GPS) fue desarrollado por el área militar de EE.UU. para superar las deficiencias de los sistemas anteriores.

El Sistema Global de Navegación por Satélite (GLONASS) es el equivalente ruso del GPS y es operado por el ejército ruso. GLONASS difiere de GPS en términos del segmento de control, el segmento de espacio, y la estructura de la señal.

Galileo es la contribución europea al futuro de los GNSS (Global Navigation Satellite System). Un sistema chino llamado *Compass*, que es la evolución del sistema regional de primera generación *Beidou*, se encuentra actualmente en fase de desarrollo.

Como se mencionó, GNSS [6] implica varios sistemas existentes, como GPS, GLONASS

o Galileo. Además, estos sistemas se complementan con los sistemas de aumentación basados en el espacio (*space-based augmentation systems - SBAS*) o sistemas de aumentación basado en tierra (*ground-based augmentation systems - GBAS*). Ejemplos de *SBAS* son el sistema de los EE.UU. de área amplia de aumento (*wide-area augmentation system - WAAS*), la órbita de servicio europeo de superposición de navegación (*European geostationary navigation overlay service - EGNOS*) o el satélite de transporte multifuncional japonés (*multifunctional transport satellite - MTSAT*) basado en el espacio del sistema de aumentación (*space-based augmentation system - MSAS*). Estos sistemas aumentan la órbita media (*medium earth orbit - MEO*) con constelaciones de satélites geoestacionarios.

2.1.3 Glonass

GLONASS es la abreviatura del ruso "*Globalñaya Navigatsionnaya Sputnikovaya Sistema*", traducida a su equivalente del español, esto significa Sistema Global de Navegación por Satélite o en inglés, *Global Navigation Satellite System*.

A mediados de 1970, la ex Unión de Repúblicas Socialistas Soviéticas (URSS) inició el desarrollo de GLONASS en base a las experiencias con el sistema de satélites *Doppler Tsikada*. La *Academician M.F. Reshetnev's State Unitary Enterprise of Applied Mechanics* ha sido la principal contratista responsable de la elaboración y la aplicación general del sistema.

Los subcontratistas son el Instituto Ruso de Investigación Científica de la Industria Espacial (*Russian Scientific-Research Institute of Space Industry*) y el Instituto Ruso de Radionavegación y Hora (*Russian Institute of Radionavigation and Time*). Estos institutos son los responsables de la vigilancia y el control, y también participaron para un desarrollo adecuado de los receptores y relojes.

Según *Coordination Scientific Information Center*, y como se define en el documento de control de interfaz GLONASS, el propósito de GLONASS es proporcionar un número ilimitado de usuarios en el aire, mar, y cualquier otro tipo de usuarios en cualquier tiempo

un posicionamiento tridimensional, de medición de velocidad y de tiempo en cualquier parte el mundo o en el espacio cercano a la Tierra.

Al ser GLONASS operado por las fuerzas militares rusas, casi no hubo información detallada cuando se dio a conocer. Más tarde, este déficit de información cambió. En 1988, en una reunión de la Comisión Especial sobre el Futuro de Sistemas de Navegación Aérea (*Special Committee on Future Air Navigation Systems*) de la Organización de Aviación Civil Internacional (*International Civil Aviation Organization - OACI*), se presentó un documento con detalles técnicos de GLONASS y la URSS ofreció el uso gratuito de este sistema satelital. Más adelante, en marzo de 1995, el Gobierno de la Federación de Rusia lanzó el Decreto número 237, donde el Ministerio de Defensa de la Federación Rusa, la Agencia Espacial Federal Rusa y el Ministerio de Transporte de la Federación de Rusia se comprometieron a proporcionar el despliegue del sistema de navegación global por satélite GLONASS, y el inicio de su operación con su asignación completa en 1995 con el fin de dar servicio a los usuarios civiles, militares nacionales y los usuarios civiles extranjeros de acuerdo a los compromisos existentes.

Las pruebas de vuelo del sistema de alta altitud de navegación por satélite, GLONASS, se iniciaron en octubre de 1982 con el lanzamiento del *Kosmos-1413*. El sistema GLONASS se puso en las pruebas de funcionamiento en 1993. Para 1995 se formó la órbita completa compuesta por 24 satélites. El sistema proporciona navegación global continua de todos los tipos de usuarios con diferentes niveles de requisitos de calidad para apoyo a la navegación. La reducción de la financiación de la industria espacial en 1990 condujo a la degradación de la constelación GLONASS. Años más tarde el Presidente y el Gobierno ruso aprobaron una serie de documentos de política, incluyendo el programa federal "Sistema Global de Navegación" para proporcionar los sistemas de seguridad y del progreso. Apoyó la creación de un campo de navegación global para determinar las coordenadas de los objetos con un alto grado de exactitud y fiabilidad, la introducción de tecnologías de navegación por satélite en la gestión del tráfico de información, la mejora de la seguridad en el sector del transporte por carretera del país, una importante reducción de los costos de operación.

2.1.4 Satélites y minería de datos

Actualmente, la enorme cantidad de información almacenada en las organizaciones de cualquier sector del mercado es una fuente potencial de conocimiento para ser explorado y extraído. Igualmente los satélites de posicionamiento se han vuelto una gran fuente de información para diversas empresas, especialmente, aquellas que trabajan o investigan a partir de datos geo-espaciales.

Es en este punto donde diferentes procesos, técnicas, metodologías y áreas del conocimiento se unen para sacar provecho y dar soluciones a los grandes retos que implica la utilización de datos de posicionamiento, su infraestructura, su extracción, su procesamiento y su aplicación.

La extracción de conocimiento a partir de las fuentes existentes de información es un área de desarrollo clave para desbloquear las relaciones desconocidas entre los diferentes puntos de datos. La minería de datos es una técnica que utiliza métodos de inteligencia artificial para extraer relaciones previamente desconocidas. Se convierte en un factor cuando se deben analizar grandes volúmenes de datos, como en el caso de las agrupaciones de satélites.

Recientemente son más las empresas, organizaciones, centros de investigación y agencias que están recurriendo a la utilización de técnicas de minería de datos. Lo hacen por varias razones, entre ellas la necesidad de responder a las cambiantes necesidades de los clientes, y del mercado. Las mismas herramientas que utilizan las empresas también se pueden aplicar a la tecnología usada en los satélites.

Por ejemplo, los registros en las empresas contienen información histórica acerca de la misma empresa, los clientes y las transacciones, y por lo general se asocian a empresas, como los bancos, las grandes tiendas por departamentos, tarjetas de crédito, seguros, telecomunicaciones, salud y agencias gubernamentales. Las empresas de este tipo acumulan

decenas de miles de registros cada día. Estos registros están en la forma de los datos operativos tales como los puntos de venta, entrada de pedidos, catalogación, entre otros.

De manera similar los satélites también acumulan registros, pero en la forma de puntos de datos de telemetría. Varios cientos de miles de puntos de datos tienen el formato de tramas de telemetría en cada paso de tiempo. Estos datos se almacenan en bases de datos para ser analizados. Los puntos de datos son la base de los datos históricos que se acumulan y pueden ser extraídos para relacionar la información.

Una evidencia de esto, es el análisis de los datos de telemetría[7] por satélite a través de la minería de datos, lo cual provee en gran parte las mismas ventajas que son frecuentes en la comunidad empresarial. Aquí, la identificación y categorización de los parámetros se llevó a cabo dentro del almacén de datos (*Data Warehouse*) donde los datos son preparados (normalizados y reformateados). Una vez procesados estos datos son pasados a un *Data Mark*, donde los expertos y los usuarios pueden encontrar la información de interés ubicada en categorías.

La minería de datos es el área que mayores aportes ha realizado para el uso intensivo de los datos y el descubrimiento de nuevo conocimiento, aunque también ha sido utilizado para la optimización de los recursos de procesamiento y almacenamiento, como se evidencia a través de un sistema de distribución de imágenes por satélite en línea [8], en el cual se extrajo el conocimiento sobre el uso de las imágenes con el objetivo de detectar el uso potencial de una imagen real de la base de datos del satélite. Los resultados obtenidos indicaron que el uso de técnicas de minería de datos puede ayudar en la automatización de la elección de las imágenes que se procesan y se almacenan antes. Como consecuencia, se puede representar una mejora en el servicio al cliente y puede conducir a un mejor uso del espacio de almacenamiento y los recursos de procesamiento.

2.1.5 Aplicaciones con datos satelitales

Los datos telemétricos son la única fuente para identificar y predecir las anomalías en los satélites artificiales. Existen personas especializadas en el análisis de estos datos en tiempo real, pero su gran volumen hace que este análisis sea extremadamente difícil. Por tal motivo, se aplicó la técnica de algoritmos de agrupamiento para ayudar a los operadores y analistas a realizar la tarea de análisis de la telemetría [9]. Se consideraron dos casos reales de las anomalías de satélites en misiones espaciales de Brasil, lo que permitió evaluar y comparar la eficacia de dos algoritmos de agrupamiento, *K-means* y expectativa de maximización (*Expectation Maximization-EM*). Se logró demostrar su eficacia en varios canales de telemetría que tendían a ofrecer valores atípicos y, en estos casos, podrían apoyar a los operadores de satélites, permitiendo la anticipación de anomalías. Sin embargo, para los problemas silenciosos, donde solo había una pequeña variación en un solo canal, los algoritmos no eran tan eficientes.

Las técnicas actuales de los modelos de seguimiento de detección de ciclones y de mediciones *in situ*, no proporcionan una verdadera cobertura global, a diferencia de las observaciones satelitales remotas. Sin embargo, es poco práctico usar una sola órbita satelital para la detección y seguimiento de estos eventos de una manera continua, debido a la cobertura espacial y temporal limitada. Una solución para aliviar tales problemas persistentes es la de utilizar los datos de los sensores desde múltiples satélites en órbita. Este enfoque aborda los retos únicos asociados con el descubrimiento de conocimiento y minería de flujos de datos por satélites heterogéneos. Dicha orientación consiste en tres componentes principales [10]: la extracción de características de cada medición del sensor, un clasificador para el descubrimiento del conjunto de ciclones, y el intercambio de conocimientos entre las distintas mediciones de los sensores remotos basados en un filtro lineal de *Kalman* para el seguimiento de predicción de ciclones. Los resultados experimentales sobre datos históricos de huracanes demuestran el rendimiento superior de este enfoque en comparación con otros trabajos[11, 12].

Otro tipo de aplicaciones se han mostrado también en los satélites de transmisión de televisión [13] (TV *broadcasting*). Con el fin de dar sentido a los datos alarmantes en el sistema de monitoreo del satélite, se introdujo un algoritmo mejorado para descubrir los episodios más frecuentes en la televisión de la red de monitoreo de radiodifusión por satélite. Los episodios frecuentes dentro de una determinada escala de los datos alarmantes se extraen con el fin de resumir el estado de los modelos de alarmas.

La minería de datos espacial es la extracción de conocimiento implícito, las relaciones espaciales u otros patrones que no están almacenados de forma explícita en la base de datos espacial. Bajo este enfoque se coloca la derivación de información de datos espaciales. Las coordenadas geográficas de los "*Hot Spots*" en las regiones de incendios forestales, que se extraen de las imágenes de los satélites, son estudiadas y utilizadas en la detección de los posibles puntos de fuego (*hot spots*) [14]. Dentro de las aplicaciones se encontraron que las falsas alarmas pueden ocurrir en los puntos de acceso derivados. Dada que esta información falsa puede ser identificada mediante la comparación del resplandor detectado en varias bandas; se utilizó la agrupación y la transformación de *Hough* para determinar los patrones regulares en los puntos de acceso derivados y clasificarlos como falsas alarmas. Esta implementación demuestra la aplicación de la minería de datos espacial para reducir falsas alarmas del conjunto de puntos obtenidos a partir de las imágenes.

Por último, esta investigación realiza un análisis de Big Data basado en los datos de los satélites de posicionamiento. Se presenta el desarrollo y el uso de un nuevo entorno de modelado distribuido de riesgo geológico para el análisis y la interpretación de los conjuntos de datos de terremotos de gran escala [15]. Este trabajo se da en un entorno analítico distribuido de tiempo real donde los análisis y las simulaciones están estrechamente acopladas, integrando implementaciones de minería de imágenes de alto rendimiento, las cuales se ejecutan en servidores dedicados. Se logró simular terremotos en una escala micro y macro basados en imágenes (*imageodesy*) y en los datos históricos de estos.

CAPÍTULO 3

Planeación

3.1 Cronograma

El cronograma mostrado a continuación contiene las actividades realizadas en el transcurso del desarrollo del proyecto de tesis.

Se ilustra desde el descubrimiento del tema de investigación hasta su posterior defensa. Cabe aclarar que todas las actividades fueron definidas, revisadas y supervisadas por el asesor de tesis.

CRONOGRAMA DE ACTIVIDADES

FASE DEL PROYECTO	AÑO - MES															
	2013					2014										
	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
1.PROBLEMA DE INVESTIGACIÓN																
Descubrimiento del tema de interés	■															
Revisión de artículos, reportes e informes sobre el tema de interés	■	■														
Planteamiento del problema				■												
Definición de la metodología			■													
Definición de las preguntas de investigación			■													
Revisión bibliográfica			■	■	■	■										
Alcance y limitaciones				■												
Definición de objetivos generales				■												
Definición de objetivos específicos				■												
Motivaciones		■														
Lectura y resumen del libro GNSS – Global Navigation Satellite Systems GPS, GLONASS, Galileo, and more				■	■	■	■	■								
Reuniones de seguimiento	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
2. CAPÍTULO 1 - INTRODUCCIÓN																
Redacción de la introducción, antecedentes y problématica					■											
Redacción del entorno y delimitación del tema					■											
Redacción de objetivos generales y específicos					■											
Redacción de las preguntas de investigación					■	■										
Redacción de la justificación					■											
Redacción de las hipótesis					■	■										
Redacción de la metodología					■											
3. CAPÍTULO 2 - ESTADO DEL ARTE																
Redacción del estado del arte				■	■	■										
4. VIABILIDAD TÉCNICA																
Registro para acceso a datos en Tiempo Real (Real Time) en el IGS								■	■							
Instalación del software cliente BNC Ntrip Client								■								
Prueba de descarga por servicio FTP del proveedor IGS								■								

Figura 3.1: Parte 1 - Cronograma de actividades con diagrama de Gantt

FASE DEL PROYECTO	AÑO - MES															
	2013					2014										
	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
Prueba de obtención de datos a través del software cliente																
Realizar un benchmarketing de proveedores de datos y software para el GNSS Glonass																
Benchmarketing de aplicaciones de software para decodificar los archivos RINEX																
5. CAPÍTULO 3 - PLANEACIÓN																
Realización del cronograma de actividades																
Realización del presupuesto del proyecto																
6. CAPÍTULO 4 - DESCRIPCIÓN DEL SISTEMA																
Obtención de datos en tiempo real. Descarga de un terabyte de información																
Modelado del proceso de descarga																
Definición de la estrutura de directorios																
Definición y propuesta de la arquitectura del sistema																
Descripción de estándares RINEX y RTCM																
Revisión de los capítulos 1 y 2																
Revisión y pruebas de herramientas ETL																
Revisión de software estadístico y de visualización de datos																
Modelado de la arquitectura del sistema																
Modelado del esquema para los metadatos																
Instalación de la base de datos																
Redacción de la arquitectura del sistema																
Redacción de los formatos RINEX y RTCM																
Redacción del protocolo NTRIP																
Redacción del proceso de descarga																
Corrección del capítulo 1 y 2																

Figura 3.2: Parte 2 - Cronograma de actividades con diagrama de Gantt

FASE DEL PROYECTO	AÑO - MES															
	2013					2014										
	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
7. CAPÍTULO 5 - EXPERIMENTOS																
Diseño, construcción y pruebas de aplicación de extracción y carga de información en la base de datos																
Pruebas con los datos almacenados en la base de datos y weka																
Ejecución de la técnica clustering sobre los datos																
Análisis de los resultados del clustering																
Revisión y corrección de pruebas de clustering																
Redacción de las pruebas clustering																
Pruebas con el software estadístico																
Definición de los métodos estadísticos a utilizar																
Revisión de los capítulos 1,2,3 y 4																
Ejecución de los métodos estadísticos																
Corrección de errores de los capítulos 1,2,3 y 4																
8. CAPÍTULO 6 - RESULTADOS																
Definición de gráficas a utilizar																
Realización de gráficas sobre los datos																
Redacción de los resultados de las pruebas de clustering																
Redacción y análisis sobre los métodos estadísticos implementados																
Redacción de otros resultados																
9. CAPÍTULO 7 - CONCLUSIONES																
Redacción de las conclusiones																
10. DEFENSA DE TESIS																
Presentación de la tesis																

Figura 3.3: Parte 3 - Cronograma de actividades con diagrama de Gantt

3.2 Presupuesto

A continuación se presenta el presupuesto elaborado para llevar a cabo el proyecto de tesis. La notación usada para la separación de miles es el carácter punto (.).

PRESUPUESTO					
RUBROS	FUENTES			CANTIDAD	TOTAL
	ESTUDIANTE	ITESM	OTRO		
Recursos Humanos				2	**$100.000**
Investigador	X			1	$50.000
Asesor de tesis		X		1	$50.000
Recursos Hardware				1	**$17.374.500**
Utilización de Workstation personal	X			1	$17.374.500
Recursos Software				10	**$0**
Microsoft Office 2013		X		1	$0
BNC NTRIP CLIENT	X			1	$0
Tex Maker	X			1	$0
PDF Reader	X			1	$0
Notepad ++	X			1	$0
IDE Desarrollo	X			1	$0
Librerias adicionales	X			1	$0
Weka	X			1	$0
Herramienta ETL	X			1	$0
Visualizadores y administradores de bases de dato:	X			1	$0
Recursos bibliográficos				1	**$0**
Acceso a bases de datos especializadas		X		1	$0
Recursos Varios				**3**	**$12.500**
Comunicaciones	X			1	$2.000
Material de impresión	X			1	$500
Viajes y Salidas de campo	X			0	$0
Servicios técnicos (Asesorias, consultorias,etc)	X			0	$0
Publicaciones	X			0	$0
Otros	X			0	$0
AUI (Admistración, Utilidad, Imprevistos)	X			1	$10.000
TOTAL					**$17.487.000**

Figura 3.4: Presupuesto del proyecto de tesis.

CAPÍTULO 4

Descripción del sistema

En el presente capítulo se describirán los formatos de los archivos descargados, como también el proceso de obtención de datos, la arquitectura implementada, algunos benchmarketings de software de post-procesamiento, y la explicación del meta-modelo de los datos.

Se muestra el funcionamiento del proceso de descarga, el diseño y desarrollo de la aplicación que extrae y carga los datos hacia el meta-modelo.

4.1 Obtención de datos

El primer paso y más importante para el desarrollo del proyecto fue la consecución de los datos, ya que en base a ellos se implementa la solución propuesta. Para este proceso se realizó un *benchmarketing* de los principales proveedores de datos a nivel mundial. Una vez hecho esto se seleccionó el que nos podía dar mayores accesos en tiempo real.

A continuación, se muestra una tabla con el resultado del *benchmarketing*.

Organización	**Tiempo real**	**Software o FTP**	***Open data***	***URL***
UNAVCO/UNIDATA	Si	Local Data Manager	Si	www.unidata.ucar.edu
CDDIS	No	Servicio FTP	Si	cddis.nasa.gov
IGS	No	Servicio FTP	Si	www.igs.org
IGS	Si	BNC NTrip Client	Si	www.igs.org
Jet Propulsion Laboratory	Si	GIPSY	No	gipsy-oasis.jpl.nasa.gov
EUREF Permanent Network	No	Servicio FTP	Si	www.epncb.oma.be
EUREF Permanent Network	Si	BNC NTrip Client	Si	www.epncb.oma.be
Continuously Operating Reference Station	No	Servicio FTP	Si	www.ngs.noaa.gov/CORS/
Ordnance Survey	No	Servicio FTP	Si	www.ordnancesurvey.co.uk
British Isles continuous GNSS Facility	No	Servicio FTP	Si	www.bigf.ac.uk
Royal Observatory of Belgium	Si	BNC NTrip Client	Si	gnss.be
Royal Observatory of Belgium	No	Servicio FTP	Si	gnss.be

Tabla 4.1: *Benchmarketing*: proveedores de datos satelitales.

El proveedor de datos seleccionado fue el IGS *International GNSS Service* con su servicio de datos en tiempo real a través del protocolo NTRIP usando el cliente *BNC Ntrip Client* proporcionado por esta misma organización. Se escogió este proveedor porque que cuenta con muchas oraganizaciones aliadas y afiliadas para su servicio de transmisión de datos en tiempo real, además de que proporciona un cliente de software que es de libre uso. De igual manera es la organización que más agrupa usuarios y servicios de datos de los GNSS y de alguna manera es la más conocida en ese medio.

Se realizó el registro a través del sitio web del IGS, solicitando un número grande *streams*, alrededor de 1000, con el objetivo de descargar desde diferentes *broadcasters* afiliados a esta organización la mayor cantidad de datos en el menor tiempo posible. La respuesta vía email por parte de los administradores tardó aproximadamente una semana, en donde se decia que el registro fue exitoso y que en los próximos días enviarían los datos de autentificación. Al pasar varias semanas no se obtuvo respuesta alguna, por lo que se les escribió nuevamente tanto por parte del investigador como del asesor de la tesis.

Al pasar casi un mes y aún sin recibir notificación alguna por parte de la organiza-

ción, se decidió con el asesor, solicitar el registro para el proyecto *Multi-GNSS Experiment (MGEX)*, el cual tiene por objetivo rastrear, cotejar y analizar todas las señales GNSS disponibles. Este proyecto nos proporcionó los accesos necesarios para la descarga de los datos. La única restricción de este proyecto es que se otorga el acceso a máximo 15 *streams*, de los cuales 5 *streams* pertenecen a la red del IGS, 5 a la red del proyecto MGEX y los últimos 5 a la red *EUREF Permanent Network*. El otorgamiento de los accesos tardó un día.

Después de realizar varias pruebas con los accesos otorgados, se definieron de manera aleatoria los 15 *broadcasters* pertenecientes a la red.

En las siguientes tablas se muestran los mapeos realizados de las diferentes estaciones por cada una de las agencias pertenecientes de las cuales se descargaron los datos:

Caster Host: mgex.igs-ip.net **Caster Port:** 2101 **Network:** IGS

Mounpoint (ID)	**Ciudad**	**País**	**Formato**	**Sistema**	**Agencia**	***Bitrate***
AREG7	Arequipa	Peru	RTCM 3.2	GPS + GLO GAL + BDS SBAS	Jet Propulsion Laboratory	2400
CONX7	Concepción	Chile	RTCM 3.2	GPS + GLO GAL + SBAS	No publicado	9600
LPGS7	La Plata	Argentina	RTCM 3.2	GPS + GLO GAL	Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum	3000
RIO27	Rio Grande Do Sul	Brasil	RTCM 3.2	GPS + GLO GAL	No publicado	3000
SCRZ7	Santa Cruz	Bolivia	RTCM 3.2	GPS + GLO GAL	No publicado	2200

Tabla 4.2: *Broadcasters* pertenecientes a la red del proyecto *MGEX*.

Caster Host: www.euref-ip.net **Caster Port:** 80 **Network:** EUREF

Mounpoint (ID)	Ciudad	País	Formato	Sistema	Agencia	*Bitrate*
AJAC0	Ajaccio	Francia	RTCM 3.1	GPS + GLO	Institut National de l'Information Geographique et Forestiere	4000
ALAC0	Alicante	España	RTCM 3.0	GPS + GLO	No publicado	5000
BOGI0	Borowa Gora	Polonia	RTCM 3.0	GPS + GLO	Institute of Geodesy and Cartography	4000
VIS00	Visby	Suecia	RTCM 3.0	GPS + GLO	Lantmateriet, the Swedish mapping, cadastral and land registration authority	6000
ZOUF0	Cercivento	Italia	RTCM 2.3	GPS + GLO	Centro Ricerche Sismologiche	5500

Tabla 4.3: *Broadcasters* pertenecientes a la red *EUREF*.

Caster Host: www.igs-ip.net **Caster Port:** 2101 **Network**: IGS

Mounpoint (ID)	Ciudad	País	Formato	Sistema	Agencia	*Bitrate*
ADH10	Abu Dhabi	Emiratos Arabes Unidos	RTCM 3.0	GPS + GLO	No publicado	9600
ALIC0	Alice Springs	Australia	RTCM 3.1	GPS + GLO	Geoscience Australia	1600
BRAZ0	Brasilia	Brasil	RTCM 3.0	GPS + GLO	Brazilian Institute of Geography and Statistics	5000
CNMR0	Saipan	Estados Unidos	RTCM 3.0	GPS + GLO	NOAA-National Geodetic Survey	2000
FFMJ1	Frankfurt	Alemania	RTCM 3.1	GPS + GLO	Bundesamt fuer Kartographie und Geodaesie Department of Geodaesie	2400

Tabla 4.4: *Broadcasters* pertenecientes a la red *IGS*.

4.2 Formatos

En esta sección se describen en detalle los formatos utilizados por los archivos y se explica el funcionamiento del protocolo de comunicación NTRIP para el intercambio de datos procedentes de los sistemas de posicionamiento global.

4.2.1 Raw Data

Los datos primarios (también conocidos como datos en bruto) es un término para los datos recogidos a partir de una fuente o *source*. Los datos primarios no han sido objeto de

procesamiento o cualquier otra manipulación.

Los datos primarios generalmente son las entradas o *inputs* para un programa informático. Estos datos pueden tener los siguientes atributos: posiblemente contienen errores, no están validados; están en diferentes formatos; no están codificados o están sin formato.

Aunque los datos en bruto tienen el potencial de convertirse en "información", la extracción y la organización, en la mayoría de ocasiones requiere del análisis y del formato correcto para la presentación de estos.

4.2.2 Formato RTCM

La Comisión Técnica de Radio para Servicios Marítimos (*Radio Technical Commission for Maritime Services, RTCM*) es una organización internacional de normalización o estandarización. Es una agencia científica, profesional y educativa no lucrativa. Los miembros pertenecientes al RTCM no son individuos particulares sino agencias gubernamentales y no gubernamentales.

En los Estados Unidos, la Comisión Federal de Comunicaciones (*Federal Communications Commission*) y la Guardia Costera de EE.UU. utilizan estándares RTCM para especificar sistemas de radar, y sistemas diferenciales de GPS, entre otros.

Los mensajes consisten en una secuencia de palabras con 30-bits cada uno. Los últimos seis bits de cada palabra son bits de paridad. Cada mensaje comienza con una cabecera que es de dos o tres palabras de largo. La primera palabra contiene un preámbulo fijo, el identificador de tipo de mensaje, y el identificador de la estación de referencia. La segunda palabra contiene la etiqueta del sistema de tiempo, el número de secuencia, la longitud del mensaje, y un indicador de la salud de las estaciones de referencia. En algunos mensajes hay una tercera palabra que se agrega al encabezado. El mensaje total tiene una longitud máxima de 33 palabras.

Los DGPS (*Differential Global Positioning System*) convecionales requieren de los tipos de mensajes 1, 2 y 9 para proporcionar la precisión del medidor. La operación RTK (*Real Time Kinematics*) se basa en los tipos de mensajes del rango entre 18 a 21 para proporcionar una precisión centimétrica. Varios sistemas utilizan el formato de mensaje RTCM para transmitir información confidencial. Por ejemplo, el tipo de mensaje 59, en particular, se puede utilizar como un canal de comunicación para transmitir mensajes cortos. Los mensajes relacionados con GLONASS están disponibles desde la versión 2.2. En la tabla 4.5 se ilustran los tipos de mensaje para la versión 2.3.

RTCM en su versión 3, se ha definido para aumentar la eficiencia de la transmisión de la información y para aumentar la integridad de la operación de paridad. La versión 3 ha sido especialmente diseñada para las operaciones de RTK, donde un gran volúmen de datos se han de transmitir. El mensaje consta de un preámbulo de 8 bits, la longitud del mensaje identificador de 10 bits, y 6 bits adicionales en la cabecera reservados para un uso futuro. El campo de datos tiene una longitud máxima de 1024 bytes seguida de una comprobación de redundancia cíclica de 24 bits (*cyclic redundancy check, CRC*).

Para la transmisión de datos RTCM a través de Internet se realiza por medio del protocolo de Internet NTRIP que ha sido definida por la Agencia Federal Alemana para la Cartografía y Geodesia. NTRIP se basa en el protocolo de transferencia de hipertexto (HTTP). Mientras tanto, el formato NTRIP ha sido asumido oficialmente por el RTCM.

Nombre del documento	Referencia	Versión	Comentarios
Recommended Standards for Differential GNSS(Global Navigation Satellite Systems) Service	10402.3	2.3	Este estándar se utiliza en todo el mundo para GNSS diferenciales, tanto marítima como terrestre.
Differential GNSS (Global Navigation Satellite Systems) Services	10403.1	3.1	Es una alternativa más eficiente para la referencia 10402.3
Standard for Networked Transport of RTCM via Internet Protocol (Ntrip)	10410.0	1	Protocolo de nivel de aplicación que soporta la transmisión de datos de cualquier GNSS a través de Internet.
Standard for Differential Navstar GPS Reference Stations and Integrity Monitors (RSIM)	10401.2	2	Esta norma se ocupa de los requisitos de rendimiento para el equipo que emite correcciones DGNSS.

Tabla 4.5: Estándares *RTCM*.

Tipo de mensaje	Función
1	Correciones para DGPS
2	Correciones para el diferencial delta de GPS
3	Parámetros para las estaciones de referencia de GPS
9	Conjunto parcial de satélites GPS
10	Correcciones diferenciales para P-Code
11	Correcciones delta para GPS C/A-code L1,L2
15	Mensaje de retraso ionosférico
17	GPS ephemerides
18	Fase de *carrier* sin corregir para RTK
19	Código de pseudorangos sin corregir para RTK
20	Correcciones para la fase de *carrier* para RTK
21	Correciones de los códigos de pseudorangos para la fase de *carrier* para RTK
31	Correcciones diferenciales para GLONASS
32	Parámetros para las estaciones de referencia de GLONASS
59	Mensaje propietario

Tabla 4.6: Tipos de mensaje, *RTCM* versión 2.3.

4.2.3 Formato RINEX

RINEX o *Receiver Independent Exchange* es un formato de texto estandarizado para recopilar las medidas u observaciones proporcionadas por los GNSS. Además permite el procesado *off-line* por varias aplicaciones de software, independientemente de cual sea el fabricante tanto del receptor como de la aplicación informática.

La versión más común en la actualidad es la 2.10, que permite el almacenamiento de medidas de pseudodistancias, fase de portador y Doppler.

En las siguientes tablas se describen los campos que contiene el formato RINEX en su versión 2.11 para el tipo de archivo *Observation File*, el cual fue utilizado en el desarrollo de la presente investigación:

Cabecera del archivo o *header*

Campo	Descripción	Ejemplo
RINEX VERSION / TYPE	• Versión del formato (2.11) • Tipo de archivo ('O'for Observation Data) • Sistema: nulo o 'G': GPS • 'R': GLONASS • 'S': Geostationary signal payload • 'E': Galileo • 'M': Mixed	2.11 Observation data Mixed
PGM / RUN BY / DATE	• Nombre del programa o aplicación que creó el archivo • Nombre de la agencia o persona que creo el archivo • Fecha de creación del archivo	BNC 2.10 Julio 25-abr.-14 18:09
COMMENT	Comentarios adicionales	RTCM 3 www.igs-ip.net/ADH10
MARKER NAME	Nombre de la antena del origén de los datos	ADH10
OBSERVER / AGENCY	Nombre de la agencia u observador	gAGE UPC: Technical University of Catalonia
REC # / TYPE / VERS	Número del receptor, tipo y versión de software	IR2200716006 ASHTECH UZ-12 CQ00
ANT # / TYPE	Número y tipo de antena	482 AOAD/M_T NONE
APPROX POSITION XYZ	Posición aproximada de la antena en las coordenadas X, Y y Z	4789028.4701 176610.0133 4195017.0310
ANTENNA: DELTA H/E/N	• Altura de la superficie inferior de la antena sobre la fuente de origen • Singularidades del centro de la antena con respecto a la fuente de origen, al este y al norte (todas las unidades en metros)	0.9030 0.0000 0.0000
WAVELENGTH FACT L1/2	• Factores por defecto de longitud de onda para L1 y L2 (GPS solamente) 1: ambigüedades de ciclo completo 2: las ambigüedades de ciclo medio (cuadratura) 0 (en L2): Frecuencia simple de instrumento cero o en blanco • El registro del factor de longitud de onda es opcional para GPS y obsoleto para los otros sistemas. Los factores de longitud de onda por defecto es 1. Si existe el registro, este debe preceder a algún registro específico de satélite	1 1
# / TYPES OF OBSERV	• Número de diferentes tipos de observaciones almacenados en el archivo • Tipos de observaciones • Código de observación • Código de frecuencia • Si hay más de 9 tipos de observaciones se continúa en la siguiente línea • Los siguientes tipos de observaciones están definidos	8 C1 P1 L1 S1 C2 P2 L2 S2

Tabla 4.7: *Header Observation file*, primera parte, RINEX versión 2.11.

Campo	Descripción	Ejemplo
# / TYPES OF OBSERV	• C: Pseudorang GPS: C/A, L2C GLONASS: C/A GALILEO: ALL P: Pseudorange GPS y GLONASS L: Carrier phase D: Doppler frequency S: Intensidades de señal sin procesar o valores SNR dada por el receptor de las observaciones de la fase respectivas • Códigos de frecuencias: • 1 GPS: L1 GLONASS: G1 GALILEO: E2-L1-E1 SBAS: L1 • 2 GPS: L2 GLONASS: G2 • 5 GALILEO: E5a SBAS: L5 • 6 GALILEO: E6 • 7 GALILEO: E5b • 8 GALILEO: E5a+b • Unidades: Phase: Ciclos completos Pseudorange: metros Doppler: Hz SNR: Depende del receptor	8 C1 P1 L1 S1 C2 P2 L2 S2
TIME OF FIRST OBS	• Tiempo del primer registro observable 4 dígitos para el año Mes Día Hora Minuto Segundo • Sistema del tiempo GPS: Sistema de tiempo GPS GLO: Sistema de tiempo UTC GAL: Sistema de tiempo de GALILEO Campo obligatorio cuando el archivo es mixto, es decir, contiene registros de los sistemas GPS/GLONASS	2014 4 25 18 9 29.0000000 GPS
END OF HEADER	Fin de la cabecera o *header* del archivo	

Tabla 4.8: *Header Observation file*, segunda parte, RINEX versión 2.11.

Cuerpo o *body* del archivo

Campo	Descripción	Ejemplo
EPOCH/SAT or EVENT FLAG	• Epoch o época 2 dígitos para el año Mes Día Hora Minuto Segundo • Epoch flag o bandera de época 0: OK 1: Falla eléctrica entre la época anterior y la actual 3: Nueva ocupación de sitio 4: Continua información de cabecera 5: Evento externo 6: Registros de deslizamiento de ciclo seguidos opcionalmente del informe de detectado y reparado • Campo obligatorio cuando el archivo es mixto, es decir, contiene registros de los sistemas GPS/GLONASS Si hay más de 12 satélites se continua en la siguiente línea	14 04 25 18 09 29.0000000 0 8R 7R 6R21R11R23R22R13R12
OBSERVATIONS	• Valores de las observaciones hechas para cada uno de los tipos especificados en la cabecera o header del archivo. El orden de los valores es dado por el orden de los tipos de observaciones de la cabecera (campo # / TYPES OF OBSERV) • El penúltimo dígito de cada valor representa la pérdida del indicador de bloqueo o Loss of lock indicator (LLI). Solo aplica para los tipos L1 y L2: 0 o en blanco: OK o no conocido Bit 0 set: Bloqueo perdido entre el anterior y el actual registro de observación: posible deslizamiento de ciclo Bit 1 set: Factor de longitud de onda opuesta a la definida para el satélite por un HECHO DE ONDA (WAVELENGTH FACT) anterior, L1 / 2 o contraria a los valores predeterminados. Válido sólo para la época actual Bit 2 set: Valor bajo Antispoofing • El último dígito de cada valor representa la intensidad de la señal para ese tipo de observable: 1: Intensidad mínima posible 5: Umbral de tasa buena S/N 9: Intensidad de señal máxima 0 o en blanco: no conocido	21889866.636 0.000 117178181.112 45.000 0.000 21889875.056 91138588.078 41.000

Tabla 4.9: *Body Observation file*, RINEX versión 2.11.

4.2.4 Protocolo NTRIP

Ntrip fue desarrollado por la Agencia Alemana Federal de Cartografía y Geodesia (BKG) y el Departamento de Ciencias de la Computación de la Universidad de Dortmund. Ntrip fue lanzado en septiembre de 2004 como "*RTCM Recommended Standards for Networked Transport of RTCM via Internet Protocol (Ntrip), Version 1.0*". La versión actual del protocolo es la Versión 2.0 bajo la enmienda 1 del 28 de junio de 2011.

Networked Transport of RTCM via Internet Protocol NTRIP es un protocolo para la transmisión de GPS diferencial (DGPS) de datos a través de Internet, de acuerdo con las especificaciones publicadas por la RTCM. Ntrip es un protocolo genérico, sin estado, basado en el protocolo de transferencia de hipertexto HTTP/1.1 y el cual se ha mejorado para los flujos de datos (*data streams*) de los GNSS.

NTRIP ha sido diseñado para la difusión de los datos de corrección diferencial u otros tipos de *streaming* de datos GNSS a usuarios fijos o móviles a través de Internet, lo que permite conexiones simultáneas de computadores, portátiles o PDA's. Ntrip es compatible con el acceso inalámbrico (WI-FI) a Internet por medio de redes de IP móvil como GSM, GPRS, EDGE, UMTS.

NTRIP se implementa en tres componentes de software del sistema: NTRIPClients, NTRIPServers y NTRIPCasters. El NTRIPCaster es el programa servidor HTTP real mientras que NtripClient y NTRIPServer están actuando como clientes HTTP.

NTRIP es un protocolo estándar abierto. El protocolo se puede descargar libremente desde BKG y hay una implementación de código abierto disponible desde *software.rtcm-ntrip.org*.

Arquitectura del protocolo NTRIP

El protocolo se compone de los siguientes elementos:

1. *NtripSources*, que generan flujos de datos en una ubicación específica,
2. *NTRIPServers*, que transfieren los flujos de datos de una fuente al *NTRIPCaster*,
3. *NTRIPCaster*, es el componente principal del sistema, y
4. *NTRIPClients*, que finalmente acceden a los flujos de datos de *NtripSources* deseados en el *NTRIPCaster*.

En la siguiente figura se ilustra la arquitectura de componenetes del protocolo:

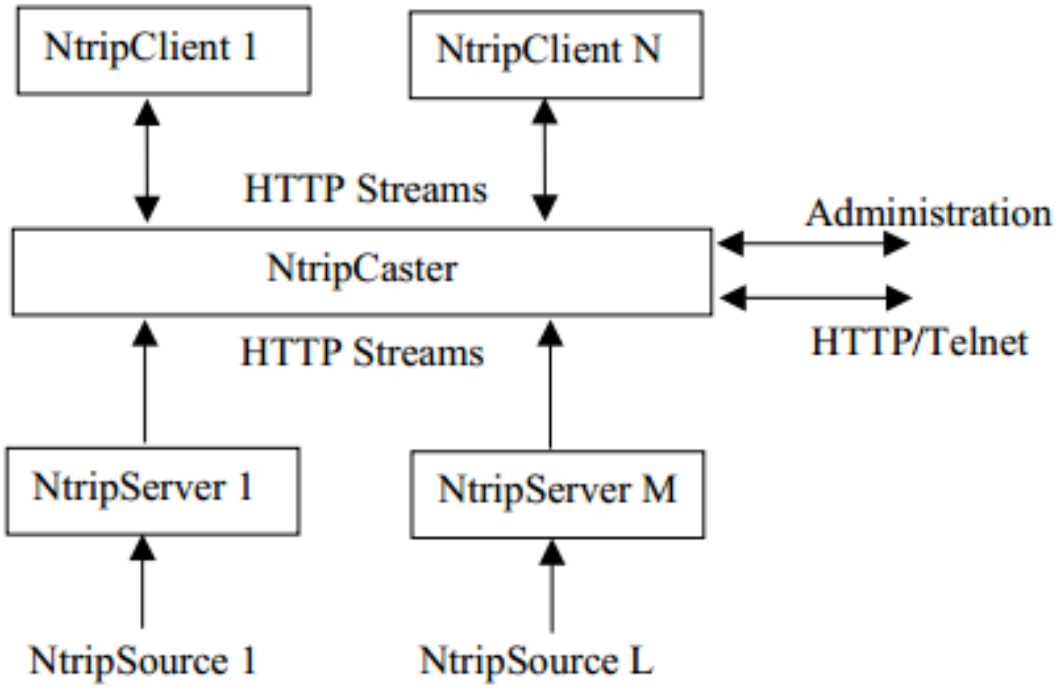

Figura 4.1: Arquitectura protocolo NTRIP.

4.2.5 Nomenclatura archivos RINEX

El software BNC sigue las conveciones del formato RINEX en su versión 2.0. Este software aún no soporta los nombres extendidos que vienen en la versión 3.02.

Los nombres de los archivos son derivados por el software para los 4 primeros caracteres de la clave o identificación del *mounpoint*. Por ejemplo para el *mounpoint* ALAC0 de la

ciudad de Alicante, España, el nombre de su *observation file* sería:

ALAC(ddd)(h).(yy)O,

donde (ddd) es el día del año, la (h) es una letra la cual corresponde a una hora larga en el formato de tiempo UTC; y (yy) son los últimos dos dígitos del año.

Sin embargo, si existe más de un *stream* con la misma identificación de *mounpoint*, la cadena de identificación se divide en dos partes o dos subcadenas y ambas partes conforman el nombre del archivo, como se muestra a continuación:

AL(ddd)(h)_AC.(yy)O.

Los nombres de los archivos para todos los intervalos menor a una hora, tienen la convención para los *observation files* de 15 minutos, como sigue:

ALAC(ddd)(h)(mm).(yy)O,

donde (mm) es el minuto de inicio dentro de la hora.

4.3 Arquitectura del sistema

En la presente sección se describe cada uno de los elementos que componen la arquitectura implementada y la cual se propone como solución.

La arquitectura general del sistema de tranferencia y extracción del conocimiento para los GNSS (STECG) está dividida en 4 grandes capas: componentes externos, comunicación, software y almacenamiento.

- Componentes externos: Son todos aquellos elementos que conforman la parte físi-

ca del sistema, tales como los satélites de la constelación GLONASS, las antenas receptoras, las estaciones de control de datos y los *broadcasters*.

- Comunicación: Es la capa que permite la transmisión de flujos de datos o *data streams* a través de la red.
- Software: Está compuesto por todos los elementos de software utilizados y desarrollados para la extracción, transformación y almacenamiento de los datos.
- Almacenamiento: Esta capa se compone del metamodelo lógico y la base de datos donde se guarda la información relevante de los datos descargados.

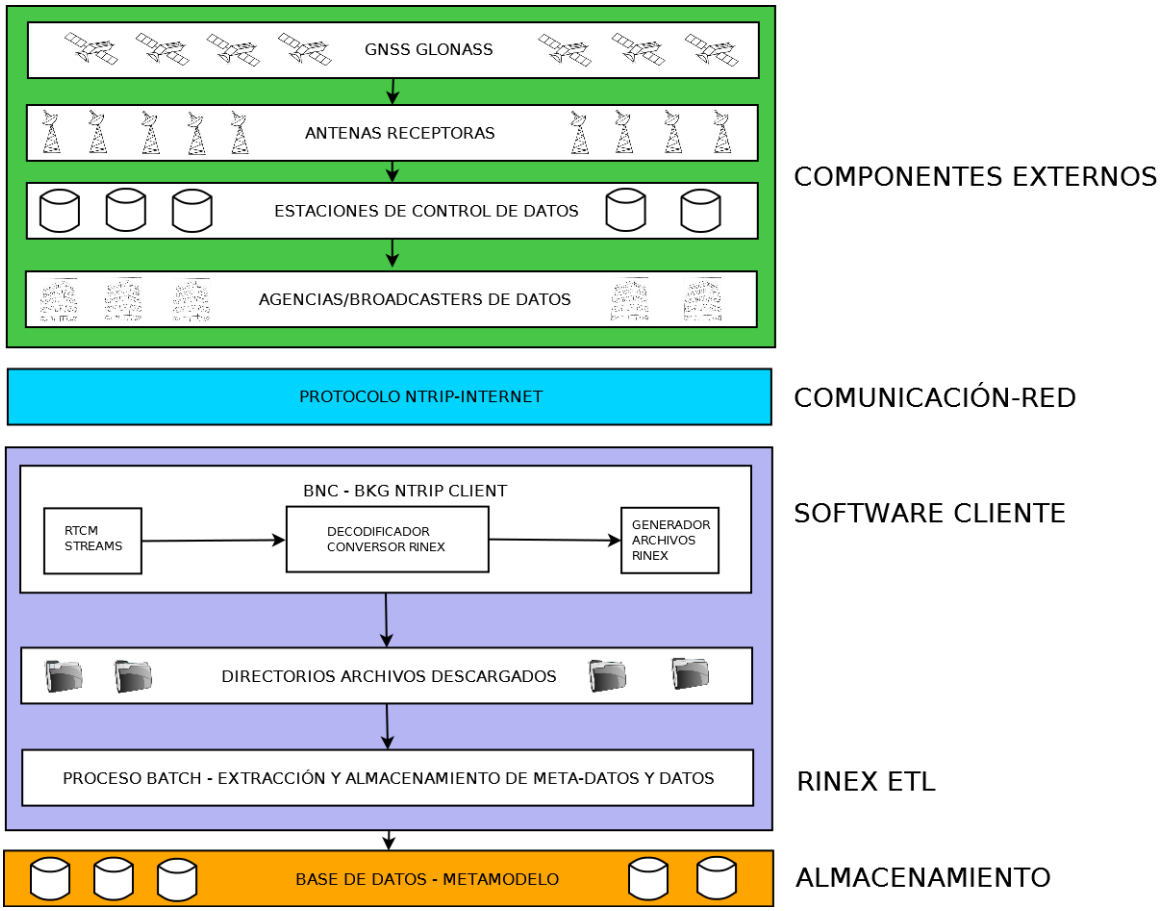

Figura 4.2: Arquitectura del sistema

4.3.1 Componentes externos

A continuación, se relacionan todos los elementos que pertenecen a esta capa de la arquitectura.

1. **GLONASS:** Es un GNSS que tiene una constelación de 31 satélites (24 en activo, 3 satélites de repuesto, 2 en mantenimiento, uno en servicio y uno en pruebas) situados en tres planos orbitales con 8 satélites cada uno. Son la fuente principal donde se producen los datos contenidos en los archivos RINEX.

2. **Antenas receptoras:** Son las antenas que reciben las señales con los datos procedentes de GLONASS. Son el intermediario entre la comunicación de GLONASS y las estaciones de control.

3. **Estaciones de control de datos:** Son las estaciones que se encargan de la operación, control y supervisión de los GNSS. Allí se almacenan los datos trasmitidos por GLONASS. Se realiza intercambio de información con los GNSS en caso de sincronización de eventos y reconfiguración de algún satélite.

4. **Agencias/Broadcasters de datos:** Son aquellas agencias, organizaciones o institutos que recoletan, almacenan, procesan e investigan los datos enviados por los GNSS y a su vez retrasnmiten estos datos por Internet a cualquier interesado en el procesamiento e investigación científica de esta información. Cabe aclarar que la retransmisión de los datos son solo de las frecuencias civiles y que están abiertas al público en general.

4.3.2 Comunicación

En este componente encontramos el protocolo NTRIP, el cual permite la transmisión de los flujos de datos o *data streams* de los datos generados por un GNSS a través de internet y deja el paso hacia un software cliente que recibe la información.

4.3.3 Componentes de software

1. **BNC - BKG Ntrip Client:** Es uno de los elementos más importantes del sistema dado que es un programa que de forma simultánea recupera, decodifica, transforma y procesa los flujos de datos de algún GNSS en tiempo real. También cuenta con algunas funciones de post-procesamiento a partir de los archivos RINEX o SP3 generados por la aplicación.

   Este software a su vez está compuesto de tres elementos principales, los cuales son:

   - ***RTCM Streams*:** Son la principal entrada de datos del sofware, son los flujos de datos descargados desde las agencias u organizaciones que pertenecen a las distintas redes como el IGS o el EUROREF. Estos flujos de datos vienen en el formato RTCM.
   - **Decodificador:** La función de este elemento es decodificar los *data streams* que llegan en el formato RTCM y realizar la transformación al formato RINEX versión 2.11.
   - **Generador de archivos RINEX:** Una vez que el decodificador ha cumplido con la transformación, este componente se encarga de almacenar los archivos RINEX en el directorio especificado.

2. **Directorio de archivos:** Es el esqueleto del proceso de almacenamiento para el software. Se debe tener una estructura bien definida para poder distinguir los diferentes tipos de archivos generados.

   En la siguiente figura se ilustra el directorio de archivos definido para la aplicación:

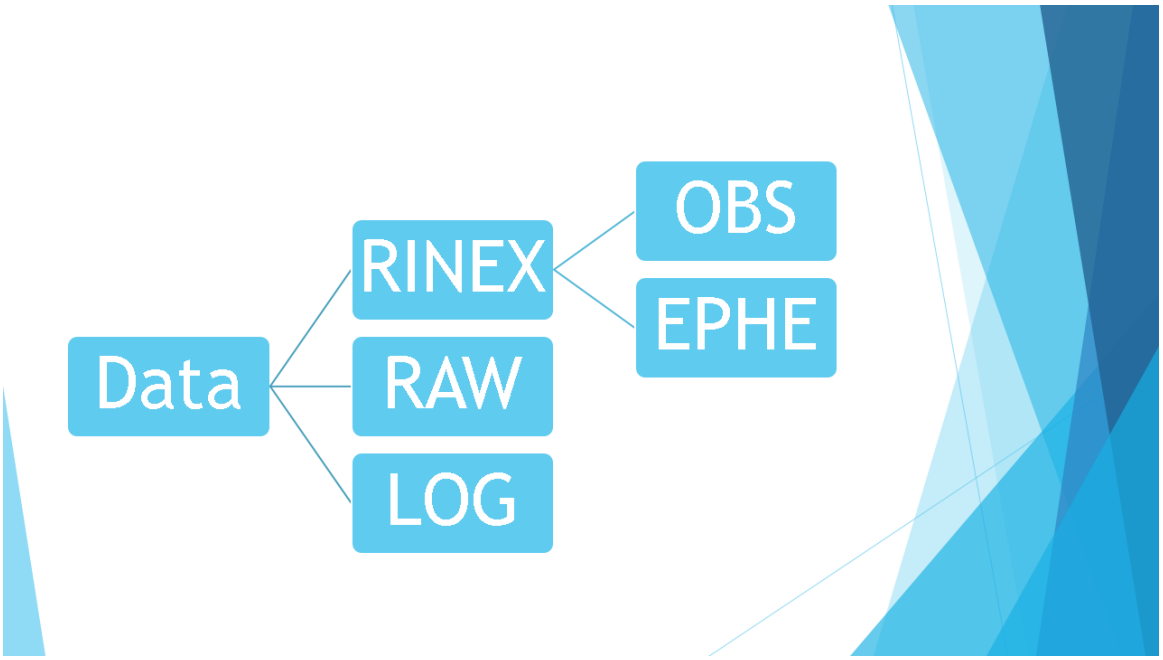

Figura 4.3: Estructura de directorios - BKG Ntrip Client

3. **Herramienta ETL (*Extraction, Transformation, Load*):** Se desarrolló un software específico para cumplir con el propósito de extraer, transformar y realizar el cargue de los datos a la base de datos.

   Esta aplicación tiene el nombre de RINEX ETL y fue programada con el lenguaje Java dada la portabilidad que tiene este lenguaje a los diferentes sistemas operativos que se encuentran actualmente en el mercado. El principal objetivo de la aplicación es la lectura de los *Observation Files*, extracción e inserción de los datos objetivo de esta investigación en la base de datos.

   La aplicación se desarrolló para que se pueda ejecutar de manera secuencial o serial, y de manera paralela, haciendo uso de los cores de los multi-procesadores.

   La arquitectura de la aplicación está basada en dos capas:

   - ***Parser*:** Es la capa principal de la aplicación dado que se encarga de la lectura, extracción y transformación de los datos obtenidos de los *Observation files*. En

esta capa se implementa la interface *Runnable* de Java, la cual permite realizar procesamiento en paralelo a través de *Threads* o hilos.

- ***Data Access Object (DAO)*:** Esta capa suministra la interfaz común entre la aplicación y la base de datos (componente de almacenamiento), es decir, la que permite la comunicación entre el componente de almacenamiento y la aplicación.

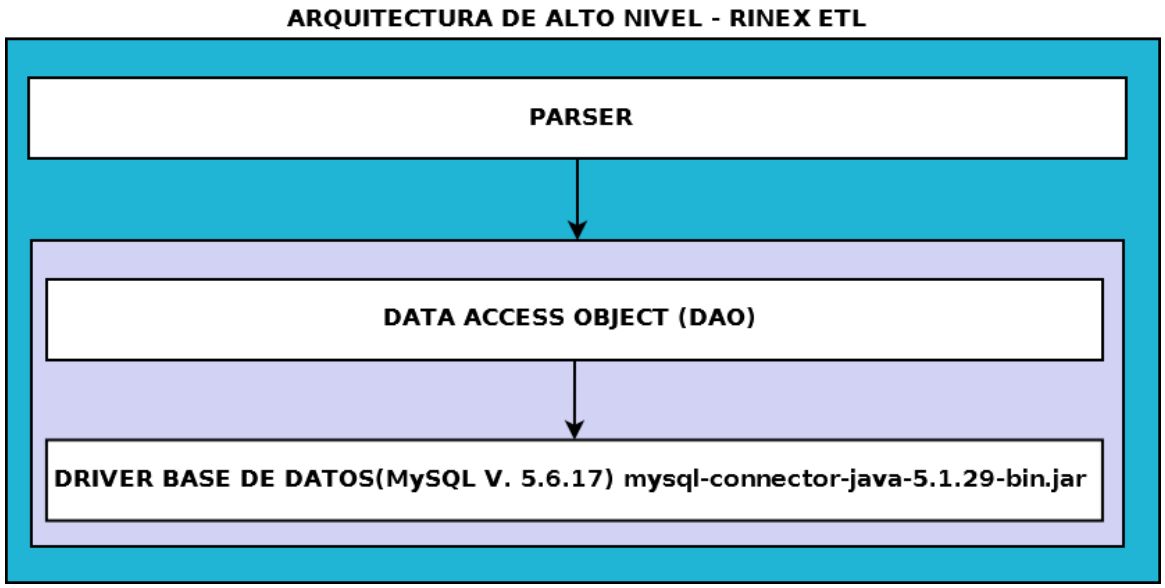

Figura 4.4: Arquitectura de alto nivel, RINEX ETL

Entrando un poco más en detalles, la aplicación está compuesta por sus propios componentes y paquetes, como también hace uso de unas librerías de uso libre(*Freeware*), las cuales hacen parte de los componentes reusables del programa. La aplicación tiene una alta dependencia de estas librerías para la comunicación con la base de datos tanto para la forma secuencial como para la paralela, tal como se puede evidenciar en la siguiente figura.

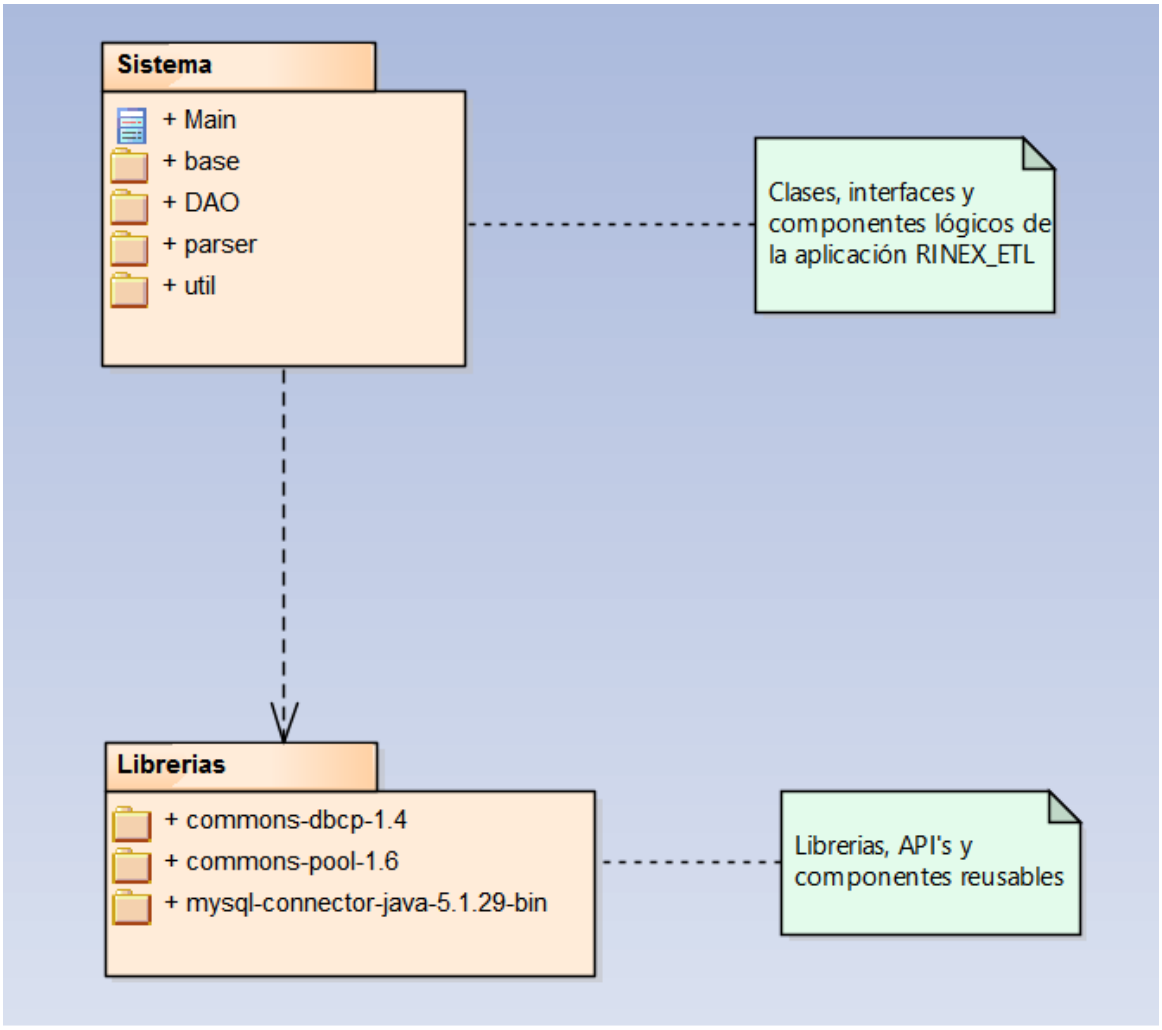

Figura 4.5: Diagrama de componentes, RINEX ETL

Igualmente dentro de las librerias usadas, se tienen unas dependencias, las cuales están relacionadas con aquellos componentes para la creación de un *Pool* de conexiones hacia la base de datos. Este *pool* es utilizado para la forma paralela dada la concurrencia que se maneja en este tipo de ejecución.

En la figura de abajo se muestran las dependencias internas entre las librerías usadas por la aplicación.

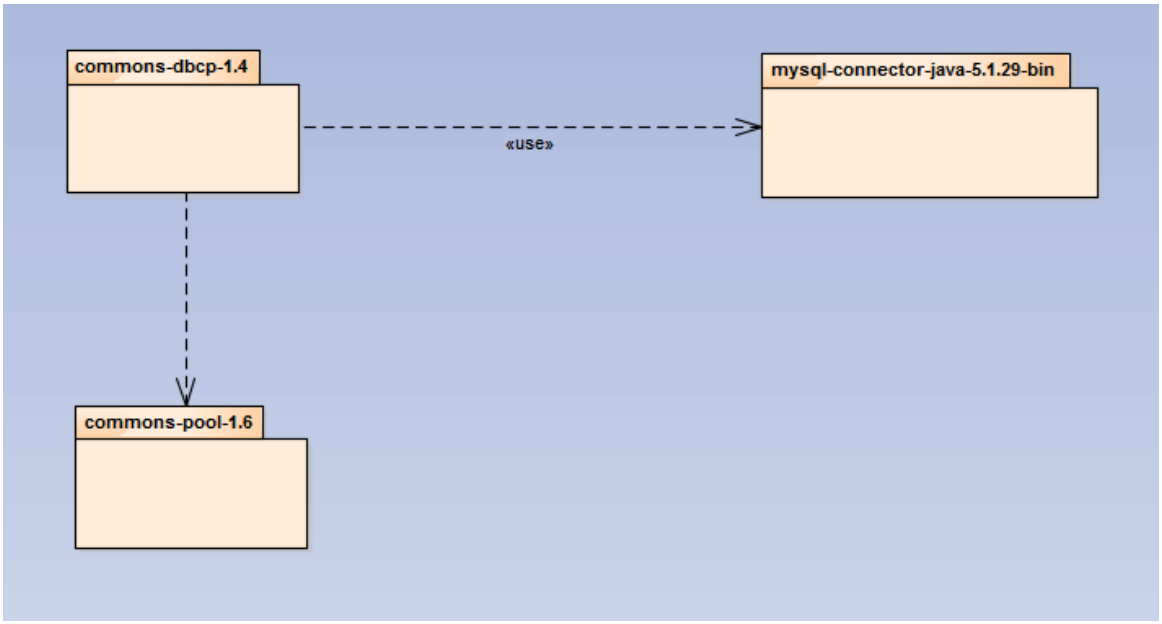

Figura 4.6: Diagrama de librerias, RINEX ETL

Continuando con la descripción del programa, se presenta la distribución interna o de paquetes, en la cual se observan las clases que hacen parte de cada uno de los mismos.

- ***Default*:** Se encuentra la clase que ejecuta la aplicación y mantiene la configuración especificada para la misma.
- **Base:** Se encuentran las clases o estructuras de datos utilizados a lo largo de la aplicación como la metadata y los datos de las observaciones.
- ***Data Access Object (DAO)*:** Se encuentran las clases que implementan la inserción de los datos extraídos y la comunicación con el componente de almacenamiento.
- ***Parser*:** Es la clase que se encarga de la lectura, extracción y transformación de los datos obtenidos de los *Observation files*.
- **Util:** Están las constantes utilizadas en la mayor parte de la aplicación, como las clases que hacen uso del *driver* de la base de datos para la creación de conexiones hacia esta.

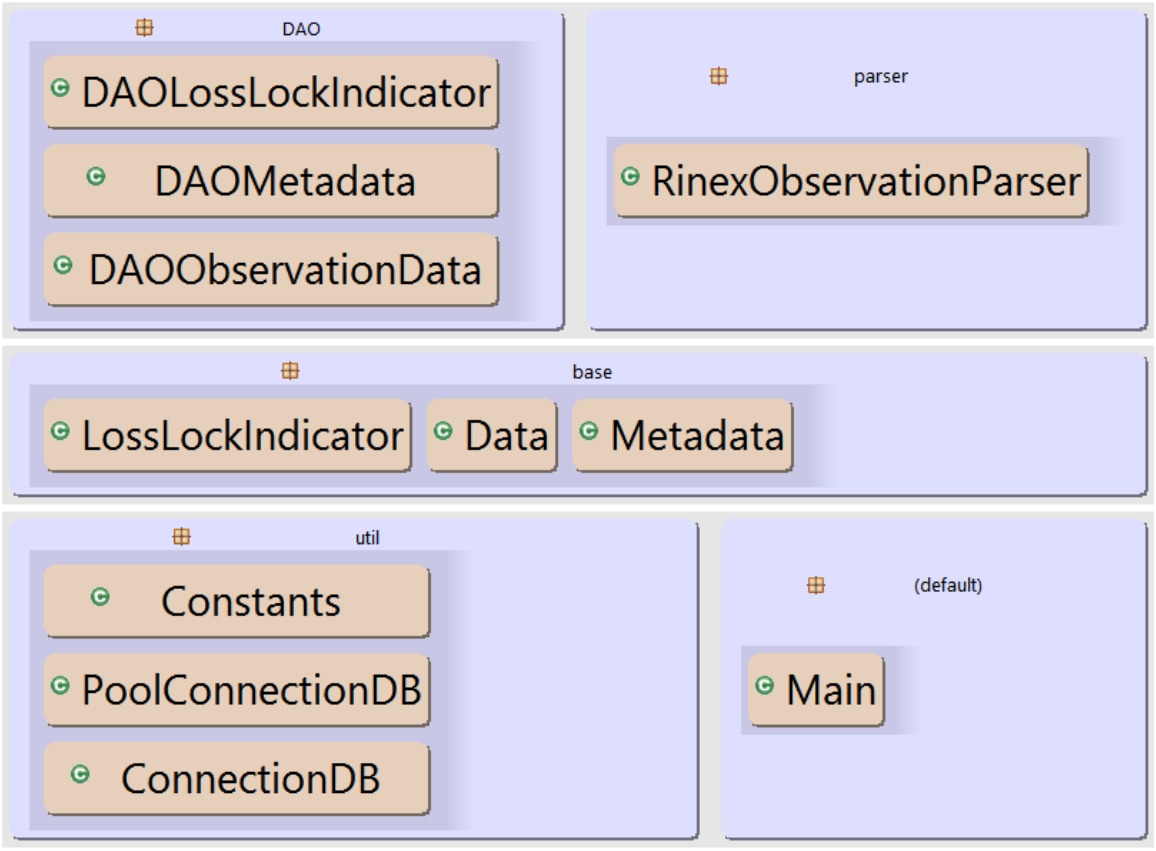

Figura 4.7: Diagrama de paquetes, RINEX ETL

Finalizando con la especificación del software desarrollado y como última parte de detalle se muestra el diagrama de clases con sus respectivas relaciones entre las mismas, sus atributos y sus métodos.

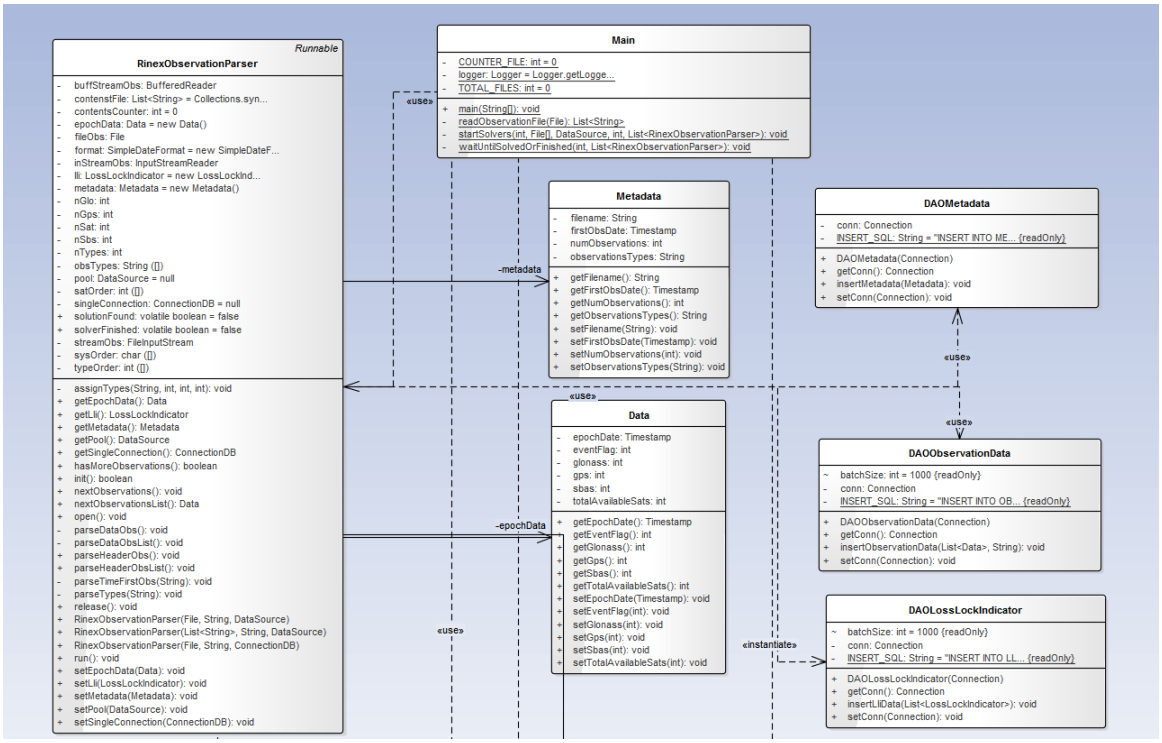

Figura 4.8: Diagrama de clases, parte 1, RINEX ETL

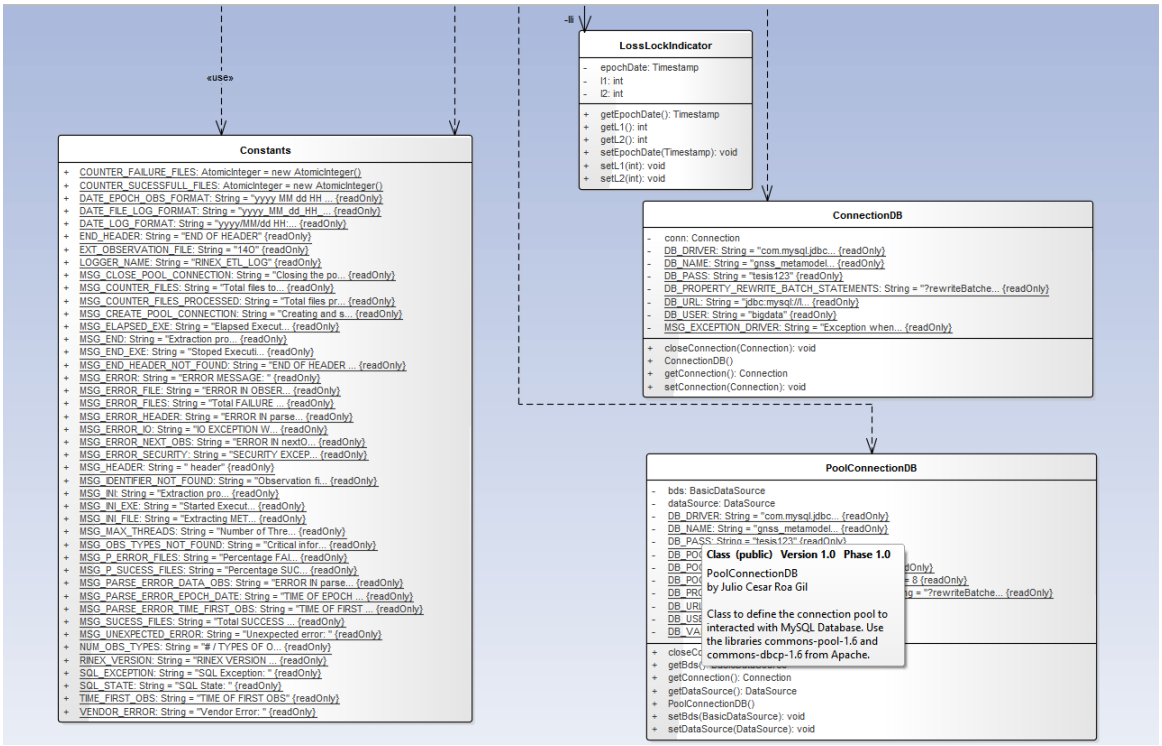

Figura 4.9: Diagrama de clases, parte 2, RINEX ETL

Las dos principales clases utilizadas son: *Main y RinexObservationParser*, para la extracción y transformación. El *Main* es el punto de partida de la aplicación y es en la cual se configura el directorio de archivos a leer, el modo de operar (secuencial o paralelo), es decir, la especificación del número de hilos, y la ruta para el archivo *log* o bitácora del programa. El *RinexObservationParser* se encarga de abrir, leer y cerrar los *Observation files*, y a su vez realizar el paso de los datos extraídos hacia los *Data Access Objects* para su posterior inserción en la base de datos.

Los *DAO's* por su parte realizan la comunicación con la base de datos y la operación de escritura o inserción en la misma. Esta operación fue optimizada a través de las librerias JAVA haciendo los llamados a las funciones de *Bulk Inserts* o *Batch processing* con lo cual se mejora notablemente el rendimiento cuando es necesario introducir grandes cantidades de datos.

Las clases *ConnectionDB* y *PoolConnectionDB* que se encargan de implementar como tal la comunicación directa con la base de datos al hacer uso del *Driver mysql-connector-java-5.1.29-bin.jar*. La primera clase implementa una sola conexión y se utiliza para la ejecución secuencial, y la segunda clase implementa un *Pool* de conexiones, o mantiene varias conexiones abiertas para que los diferentes *Threads* las utilicen. Este *pool* de conexiones es configurable para colocar los límites de conexiones abiertas, en espera, y en ejecución.

4.3.4 Componente de almacenamiento

Aquí encontramos la base de datos con el meta-modelo definido para almacenar y extraer la información de interés definida para esta investigación. La base de datos es relacional por lo que nuestro meta-modelo se diseñó bajo un diagrama de entidad/relación, el cual se ilustra a continuación.

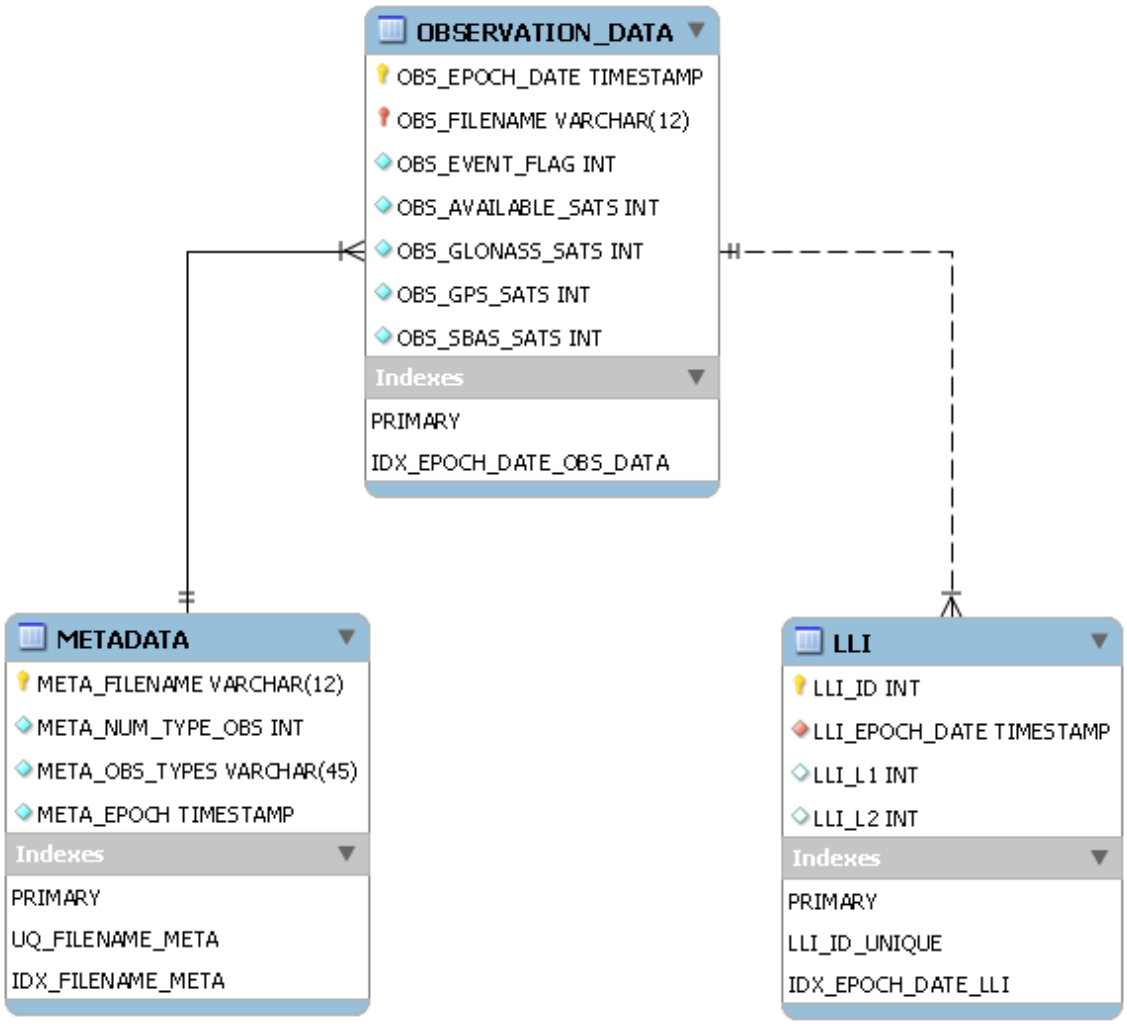

Figura 4.10: Meta-modelo - Diagrama Entidad/Relación

Como se puede observar existen dos tablas, las cuales representan de cierta manera el *header* y el *body* de los *observations files* en formato RINEX. Estas tablas contienen los campos necesarios para poder relacionar la información más importante y extraer así el conocimiento de este conjunto de información. La tercera tabla es mucho más específica, y almacena los datos referentes al *Loss of Lock Indicator (LLI)* de los tipos de observaciones L1 y L2, su razón de ser es la de encontrar los posibles deslizamientos de los ciclos en las *Epoch Dates* con el objetivo de analizar las medidas realizadas por los *GNSS*.

4.4 Proceso de descarga

Esta sección explica cómo funciona el proceso de descarga implementado en esta investigación y se hace por medio de un diagrama de flujo para entender el proceso de una manera visual.

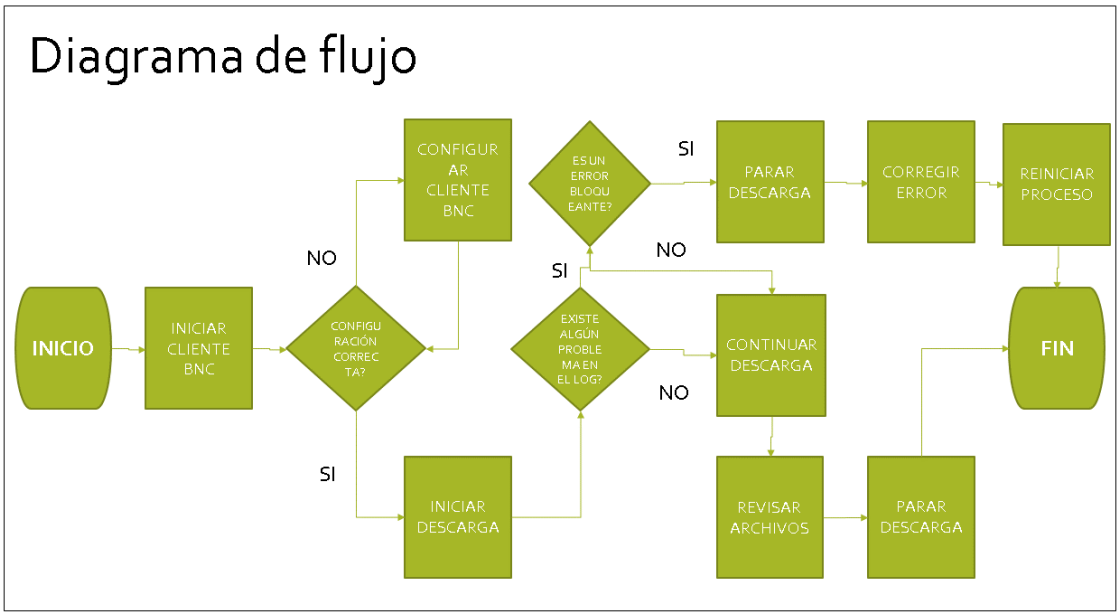

Figura 4.11: Diagrama de flujo - Proceso de descarga

CAPÍTULO 5

Experimentos

En el presente capítulo se describirán los experimentos realizados con los *Observation Files*, la base de datos, y el ambiente en el cual fueron desarrollados.

5.1 Ambiente de pruebas

A continuación se describen las principales características del equipo portátil o *Workstation* en los cuales se realizaron los experimentos.

Item	**Especificación**
Procesador	Procesador Intel Quad Core i7 4700HQ 4GHz
Sistema operativo	Windows 8
Chipset	Intel HM77 Chipset
Memoria Ram	16 GB, DDR3 1800 MHz SDRAM
Panel TFT-LCD	C17.3"16:9 FHD EWV LED Backlight
Tarjeta Gráfica	NVIDIA GeForce GTX 770M 3GB GDDR5 VRAM
Almacenamiento	1TB HDD 5400 RPM + 256GB SDD

Tabla 5.1: Especificaciones *Workstation* Asus G750J

Item	Especificación
CUDA Cores	1536
Base Clock (MHz)	1046
Boost Clock (MHz)	1085
Texture Fill Rate (billion/sec)	134
Velocidad de la memoria	7.0 Gbps
Configuración estándar de la memoria	2048 MB
Interface de memoria	GDDR5
Ancho de banda de memoria (GB/seg)	224.3

Tabla 5.2: Especificaciones tarjeta gráfica, NVIDIA GeForce GTX 770M

Se hizo uso de un disco duro externo para almacenar los archivos descargados. En la siguiente tabla se muestran sus especificaciones técnicas:

Item	Especificación
Capacidad	2TB
Interfaz	USB 2.0 y USB 3.0
Sistema operativo	Windows XP o superior Max OS X 10.4.6 Tiger o superior 10.5 Leopard o 10.6 Snow Leopard (32-bit kernel)

Tabla 5.3: Especificaciones disco duro externo, Seagate GoFlex Desk

Referente a la parte de software, se utilizaron las siguientes herramientas:

Software	Descripción	Versión
Eclipse IDE	Entorno de desarrollo y ejecución de pruebas (JAVA)	Luna
MySQL Workbench	Entorno de desarrollo y ejecución de pruebas (SQL)	6.1
VisualVM	*Profiler* para la *Java Virtual Machine* (JVM)	1.3.8
MySQL Server	Servidor de Base de datos relacionales	5.6.19

Tabla 5.4: Herramientas de software

5.2 Especificaciones generales

Las siguientes especificaciones aplican para todas las pruebas hechas con los datos de los archivos descargados.

Item	Especificación
Cantidad de archivos a procesar	40775
Tamaño de los datos	80.9 Gigabytes
Porcentaje máximo de error	2
Número de ejecuciones	3

Tabla 5.5: Especificaciones generales para las pruebas

Las métricas que se utilizaron en el desarrollo de los experimentos se detallan a continuación:

Métrica	Especificación
Cantidad de archivos procesados con éxito	número natural
Cantidad de archivos procesados con fallas	número natural
Tiempo de ejecución	segundos, minutos y horas

Tabla 5.6: Métricas generales para las pruebas

5.3 Pruebas

1. **Secuencial:** Consiste en ejecutar la aplicación RINEX ETL de manera secuencial o con un único hilo de ejecución.

2. **Paralelo con procesadores:** Consiste en ejecutar la aplicación RINEX ETL de manera paralela a través de hilos, los cuales hacen uso de los *cores* de los procesadores. Esta prueba se realizó con:

   - 8 *Threads*.

- 10 *Threads.*
- 16 *Threads.*
- 32 *Threads.*
- 50 *Threads.*

3. **Clustering:** Consiste en ejecutar la aplicación *Weka* y realizar la conexión con la base de datos del meta-modelo. Después se realizó una consulta a la base de datos con la cual se cargaron los datos en la herramienta de minería de datos y se ejecutó el proceso de *Clustering* con el objetivo de encontrar alguna anomalía en el *Loss of Lock Indicator (LLI)* o algún patrón sobre los datos descargados y de los cuales podamos inferir e intepretar posibles mejoras. Para cumplir este objetivo se utilizó el algoritmo de *K-Means*, con el cual se definieron 4 *clusters.*

4. **Consultas SQL:** A través del lenguaje SQL se determinaron los siguientes objetivos:
   - Identificar la cantidad de fallas en las épocas (*epoch*)
   - Identificar la cantidad de posibles deslizamientos de ciclo para el tipo de observación L1
   - Identificar la cantidad de posibles deslizamientos de ciclo para el tipo de observación L2
   - Determinar la frecuencia de los posibles deslizamientos de ciclo para el tipo de observación L1
   - Determinar la frecuencia de los posibles deslizamientos de ciclo para el tipo de observación L2

CAPÍTULO 6

Resultados

En el presente capítulo se describirán los resultados de los experimentos realizados con los *Observation Files*, la base de datos, y el ambiente en el cual fueron desarrollados.

6.1 Descarga de archivos

Esta sección resume la distribución de los diferentes tipos de archivos descargados a través del software *BNC-Ntrip Client*.

Tipo de archivos	**Tamaño(MB)**
Observation Files	80,900
Ephemerides Files	105
Raw Data	14,500
Log Files	4.25

Tabla 6.1: Distribución de los archivos descargados

Tamaño total de los archivos descargados: 95.5 GB.

La siguiente tabla nos muestra la distribución de los *Observation Files*, entre los archivos correctos o legibles y los corruptos o ilegibles, ya sea por problemas de la red o de comunicación entre el cliente y los *broadcasters* o fallas eléctricas del lado del cliente.

Item	Cantidad	Porcentaje
Total de archivos	40,792	100
Total de archivos correctos	40,775	99.96
Total de archivos corruptos	17	0.04

Tabla 6.2: Distribución de los *Observation Files*

Como resultado obtuvimos que menos del 1 % fueron archivos totalmente ilegibles y por ende no hicieron parte de los experimentos hechos.

La presente gráfica nos muestra la tendencia de descarga a lo largo de 4 meses.

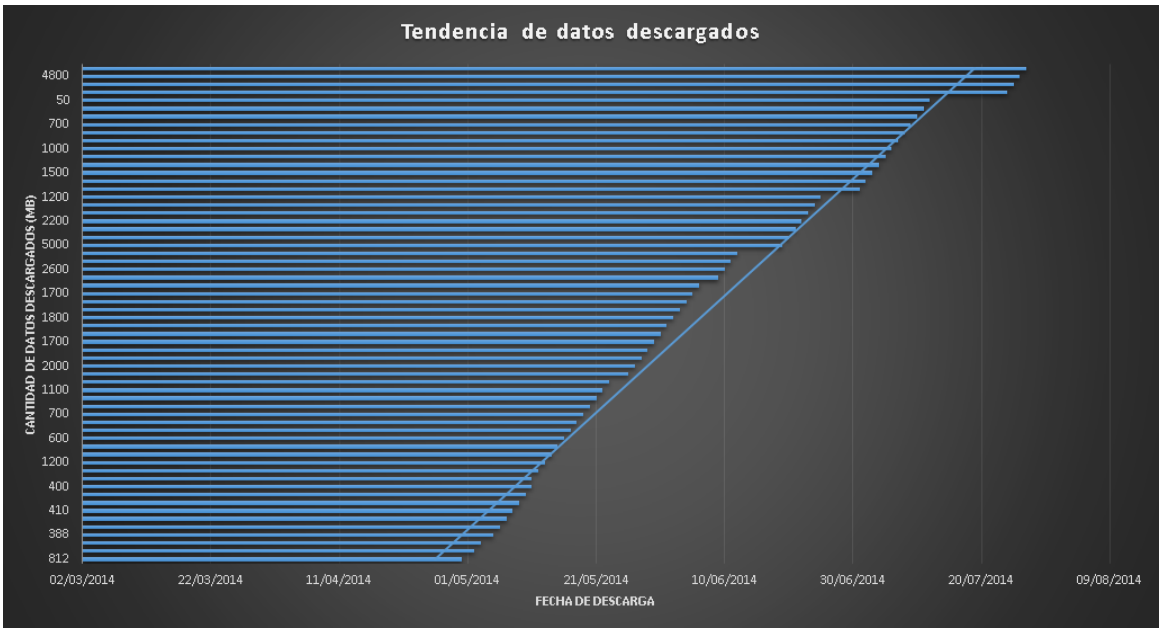

Figura 6.1: Tendencia de archivos descargados

La figura 6.1 nos indica que al cabo del tiempo de descarga de datos definido, se necesitó mayor tiempo de ejecución de la aplicación para obtener una mayor cantidad de datos para ser procesados, con el objetivo de llegar a una muestra considerable de datos en un menor tiempo.

6.2 Resultados de las pruebas

En la presente sección se presentan los resultados obtenidos en las diferentes pruebas. Se describen los resultados del tiempo de ejecución como también se ilustran con gráficas diferentes métricas de rendimiento, como lo son el uso del CPU y de la memoria.

Para determinar y monitorear las diferentes métricas se hizo de la herramienta *VisualVM*, la cual es una herramienta que muestra información detallada de las aplicaciones *Java* que se ejecutan en la *Java Virtual Machine (JVM)*.

6.2.1 Base de datos

La presente tabla muestra la cantidad de registros insertados en la base de datos, en las tablas METADATA, OBSERVATION_DATA, y LLI, pertenecientes al meta-modelo.

Tabla	**Cantidad de registros**
METADATA	40,775
OBSERVATION_ DATA	34'200,319
LLI	12'089,001

Tabla 6.3: Distribución de registros en las tablas del meta-modelo

Dentro de esta cantidad de registros no se encontró ninguna falla entre las *epoch dates*. Esto quiere decir que durante el tiempo de toma de las métricas u observaciones por parte de los *GNSS* no se generó ningún evento que alterara el valor observado.

En la siguiente tabla se muestra la cantidad de registros referentes al *Loss of Lock Indicator* con un posible deslizamiento de ciclo en los tipos de observaciones L1 y L2, almacenados en la tabla LLI.

Tipo de observación	Cantidad de registros
L1	2'517,286
L2	1'431,922

Tabla 6.4: Cantidad de registros con posible deslizamiento de ciclo, L1 y L2

A continuación se muestran las frecuencias y la cantidad de posibles deslizamiento de ciclos referentes al *Loss of Lock Indicator* de los tipos de observaciones L1 y L2 por día, es decir, durante el tiempo de obtención de los datos.

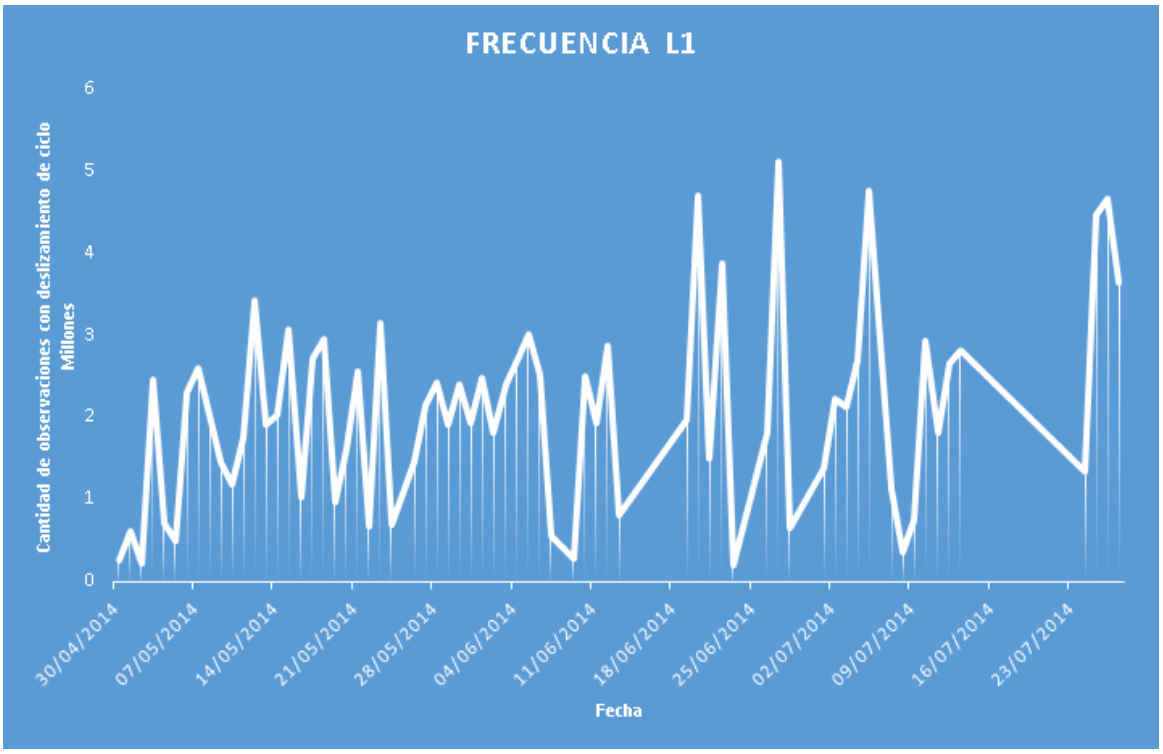

Figura 6.2: Frecuencia L1

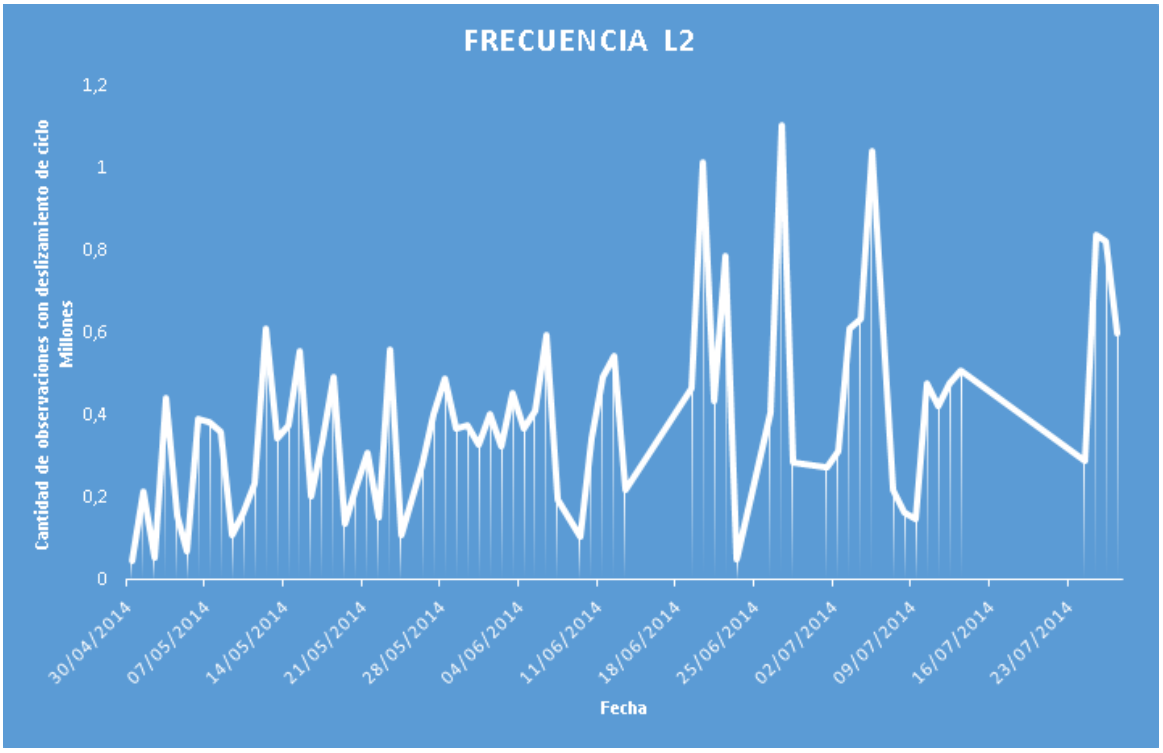

Figura 6.3: Frecuencia L2

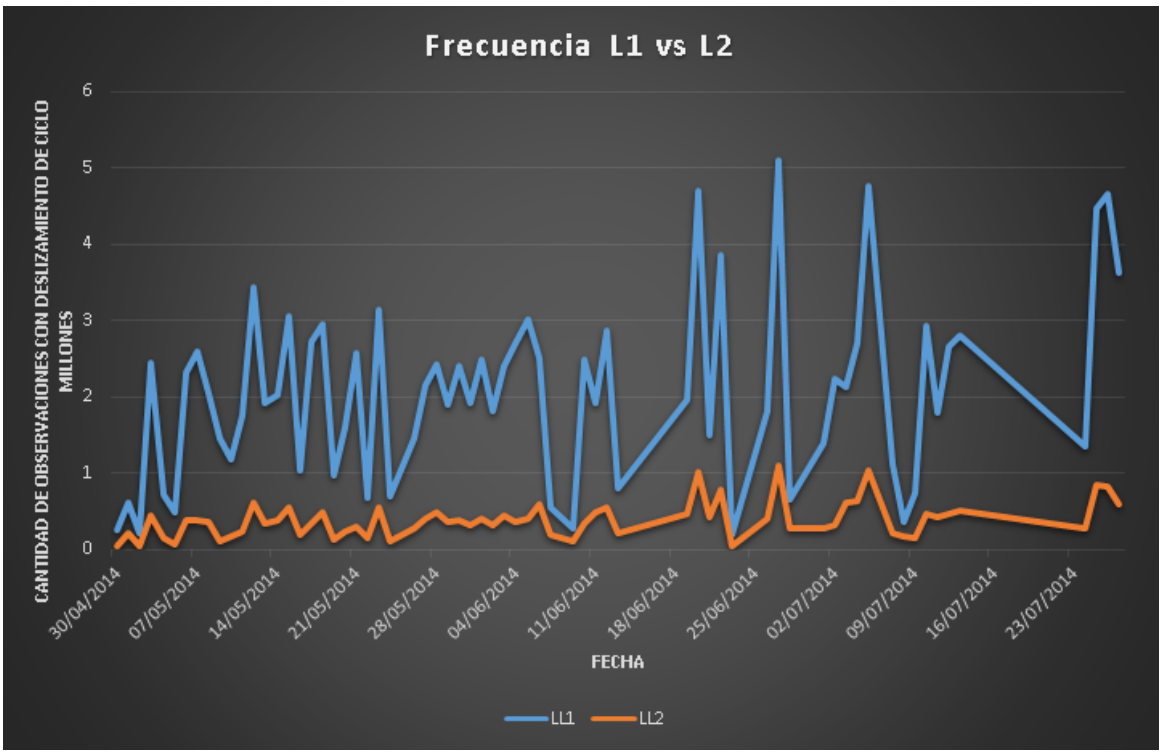

Figura 6.4: Frecuencia L1 y L2

La cantidad máxima de observaciones con posible deslizamiento de ciclo para L1 fue de 5.107.444 y de 1.105.203 para L2. Mientras que los valores mínimos fueron 199.511 y 46.604 para L1 y L2 respectivamente.

Con esta información interpretamos que una de las anomalías más comunes en los tipos de observaciones L1 y L2 es el deslizamiento de ciclo, igualmente esta no representa como tal una falla del valor observado, solo nos indica que se debe realizar un proceso de corrección para hallar el valor correcto observado. Estas correcciones se realizan a través de diferentes métodos, los cuales no hacen parte de esta investigación.

Las figuras 6.5, 6.6 y 6.7 nos resumen por cada uno de los meses, las frecuencias del *Loss of Lock Indicator* para L1 y L2.

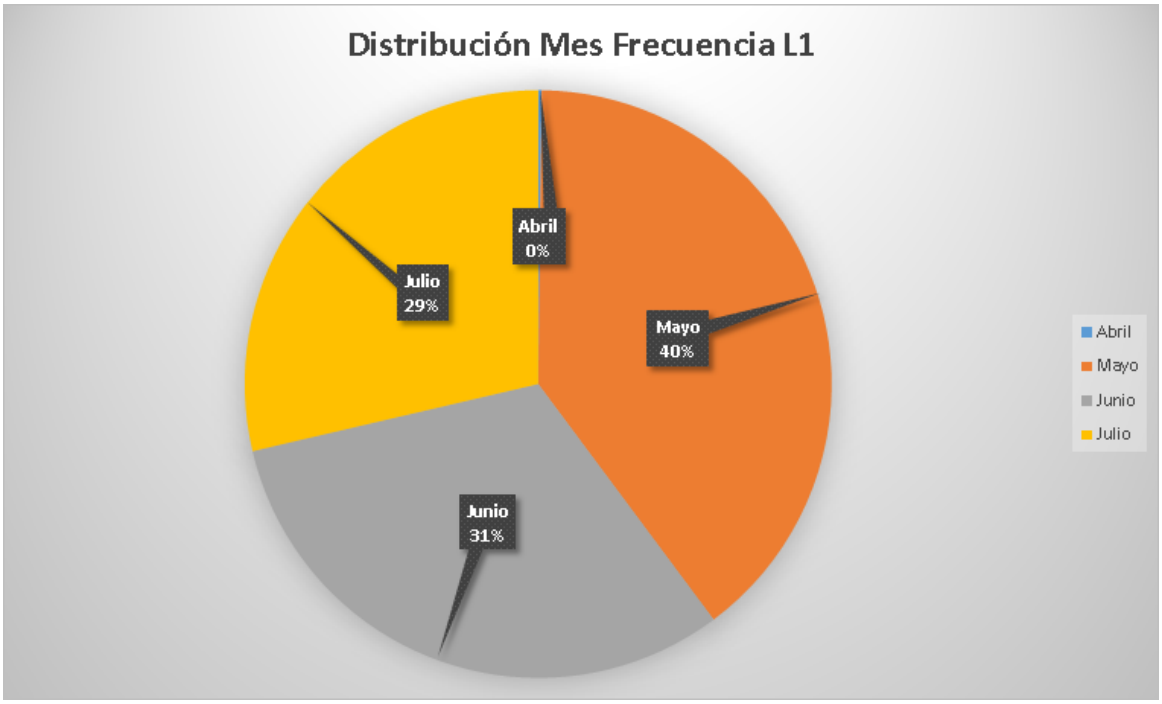

Figura 6.5: Frecuencia mes, L1

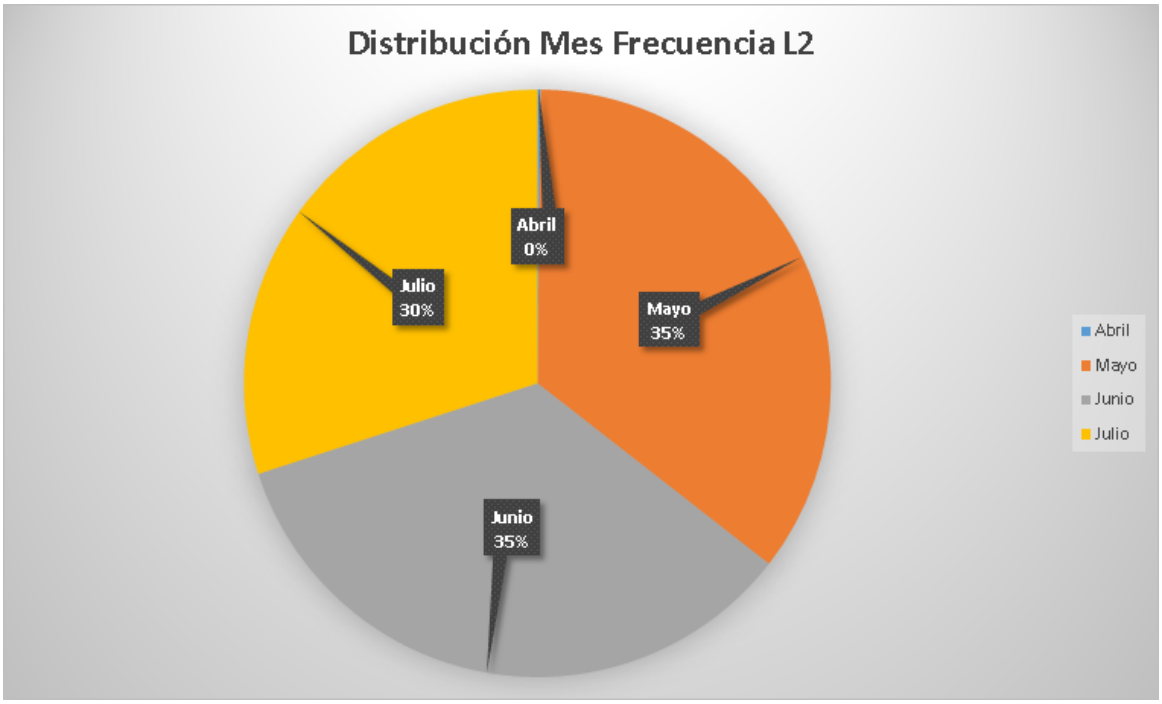

Figura 6.6: Frecuencia mes, L2

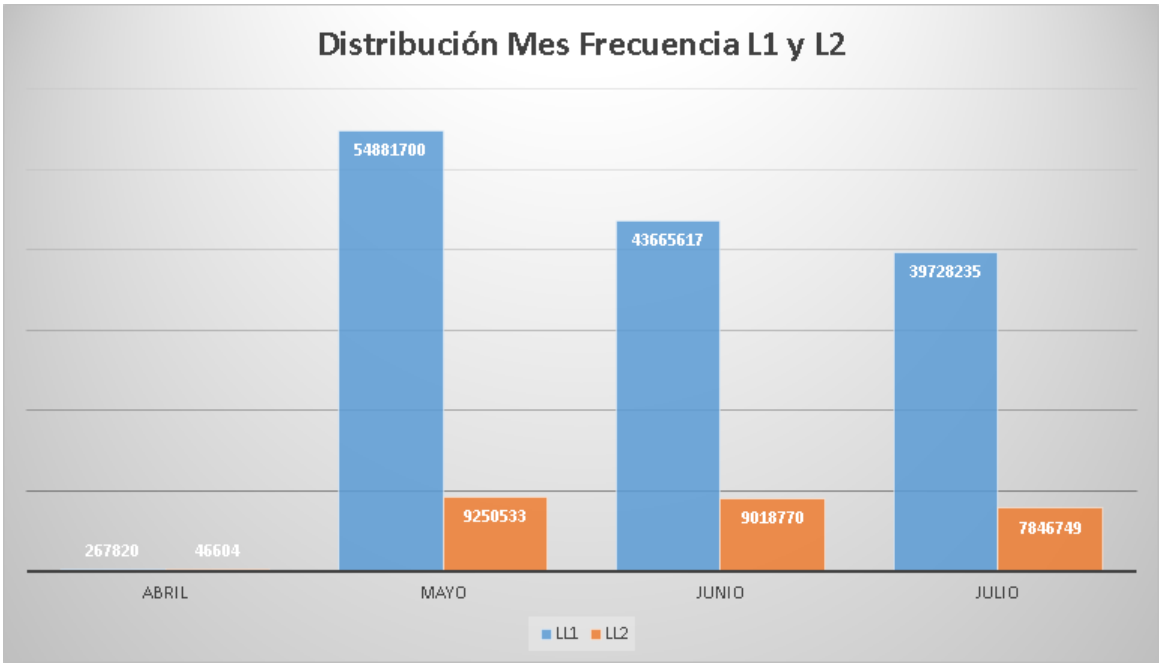

Figura 6.7: Frecuencia mes, L1 y L2

6.2.2 Modo secuencial

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo serial.

Número de ejecución	Tiempo de ejecución (segundos)	Tiempo de ejecución (minutos)	Tiempo de ejecución (horas)
1	5962.77	99.38	1.66
2	6294.33	104.91	1.75
3	5949.65	99.16	1.65
Promedio	**6068.92**	**101.15**	**1.69**

Tabla 6.5: Resultados en modo secuencial

En las figuras 6.8, 6.9, 6.10, 6.11 y 6.12, se evidencia el rendimiento en términos de uso del CPU del *workstation*, la memoria RAM utilizada por la *Java Virtual Machine (JVM)*, el tiempo de ejecución del hilo principal, la cantidad de hilos y *demons* con su rango (en tiempo) de "vida".

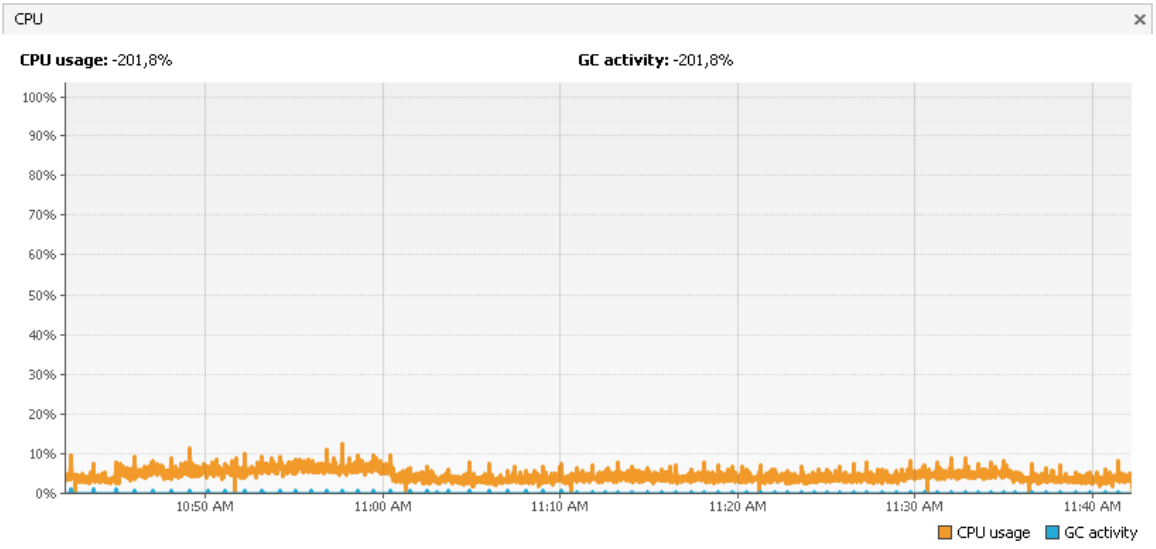

Figura 6.8: *Performance CPU*, con un hilo principal

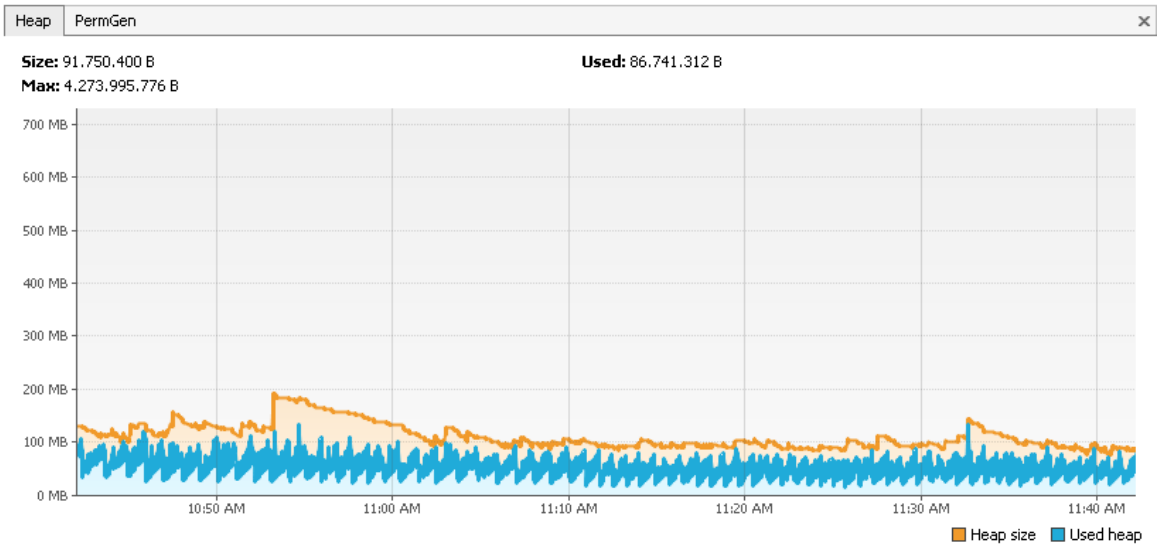

Figura 6.9: *Performance - Heap memory*, con un hilo principal

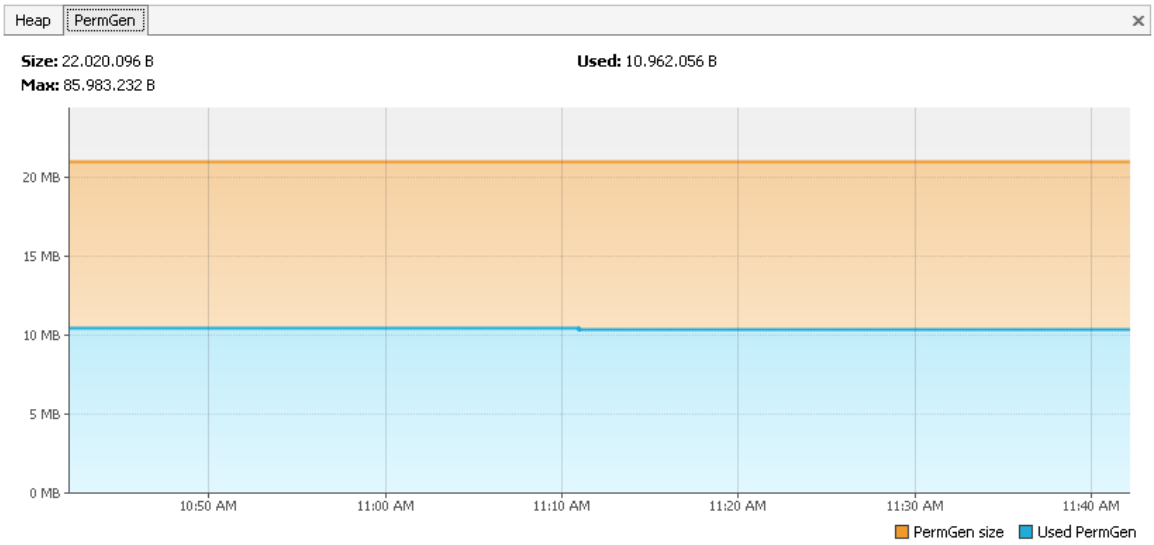

Figura 6.10: *Performance - Permanent Generation heap*, con un hilo principal

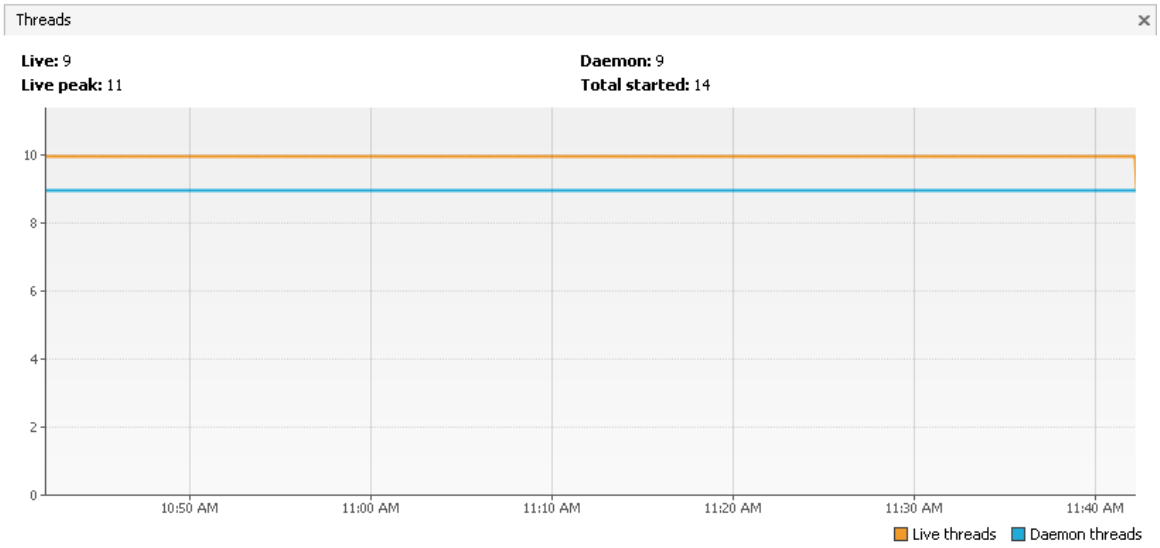

Figura 6.11: *Performance - Live demons and threads*, con un hilo principal

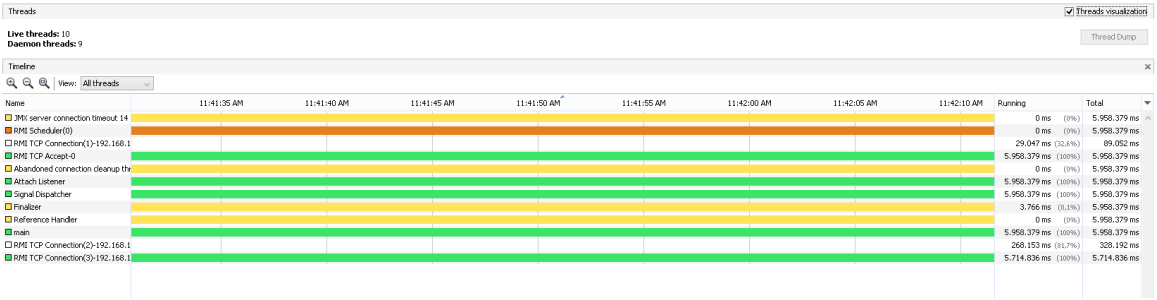

Figura 6.12: *Thread timeline*, con un hilo principal

6.2.3 Modo paralelo, 8 *Threads*

La configuración del *pool* de conexiones usada fue:

Tamaño inicial = 16 conexiones.

Número máximo de conexiones activas = 14.

Número máximo de *prepared statements* activas = 14.

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo paralelo utilizando 8 hilos.

Número de ejecución	Tiempo de ejecución (segundos)	Tiempo de ejecución (minutos)	Tiempo de ejecución (horas)
1	2797.30	46.62	0.78
2	3045.28	50.75	0.85
3	2859.42	47.66	0.79
Promedio	**2900.67**	**48.34**	**0.81**

Tabla 6.6: Resultados en modo paralelo con 8 hilos

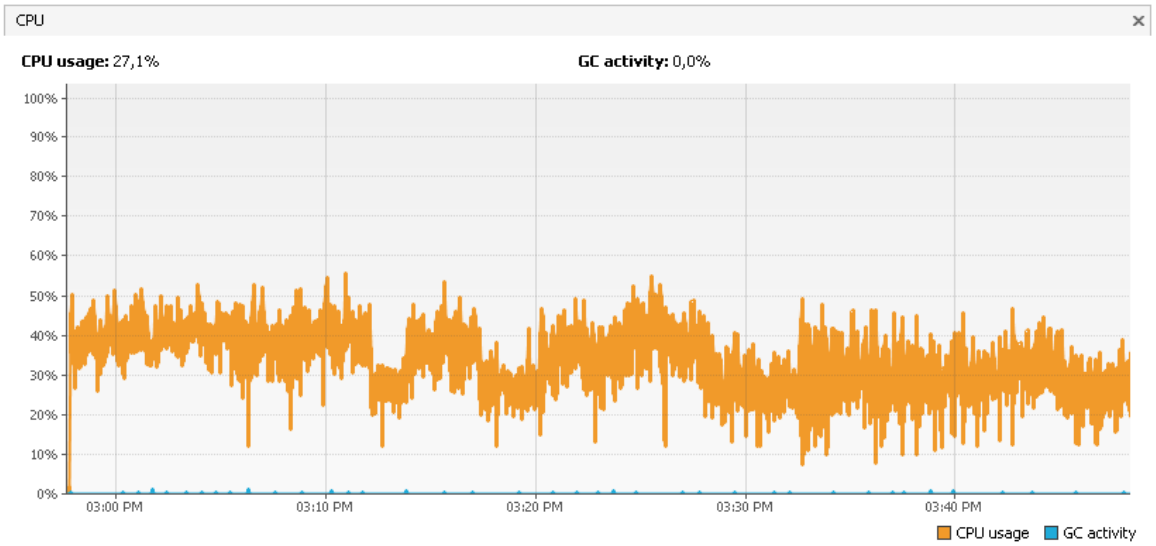

Figura 6.13: *Performance CPU*, con 8 hilos

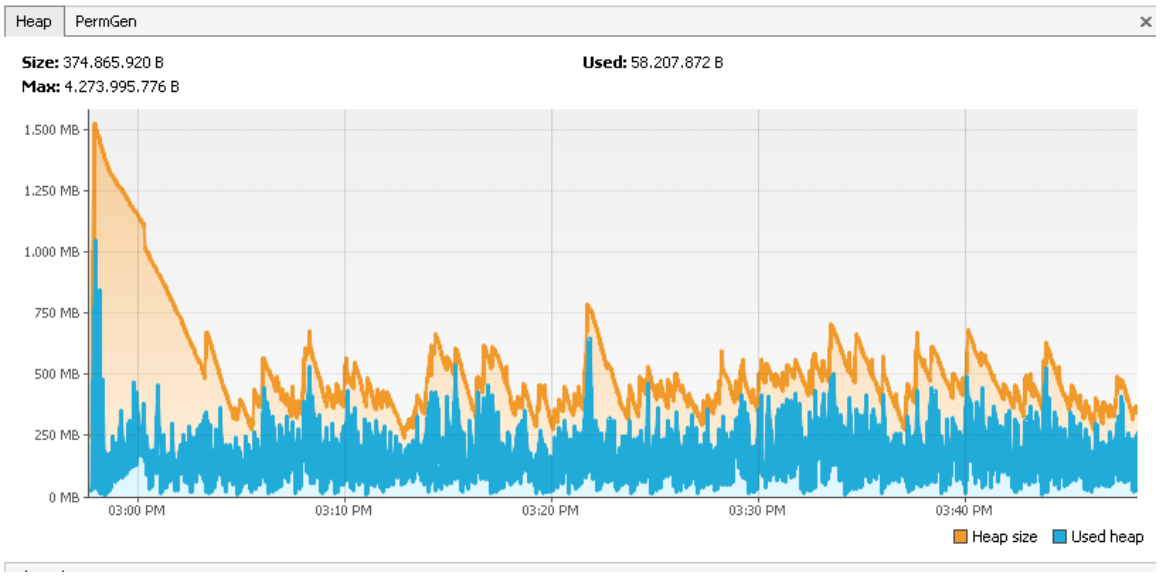

Figura 6.14: *Performance - Heap memory*, con 8 hilos

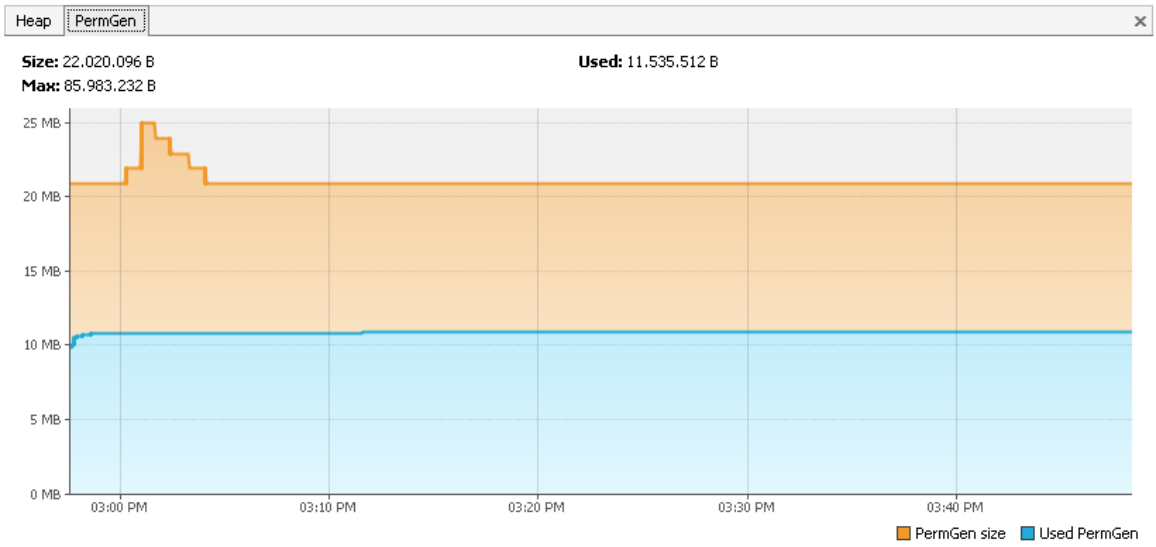

Figura 6.15: *Performance - Permanent Generation heap*, con 8 hilos

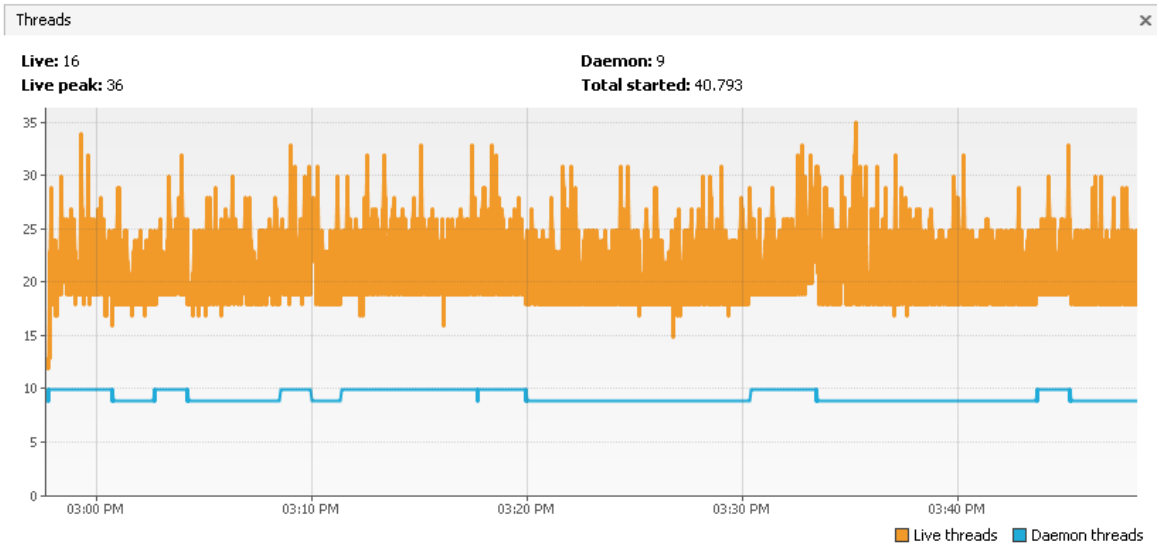

Figura 6.16: *Performance - Live demons and threads*, con 8 hilos

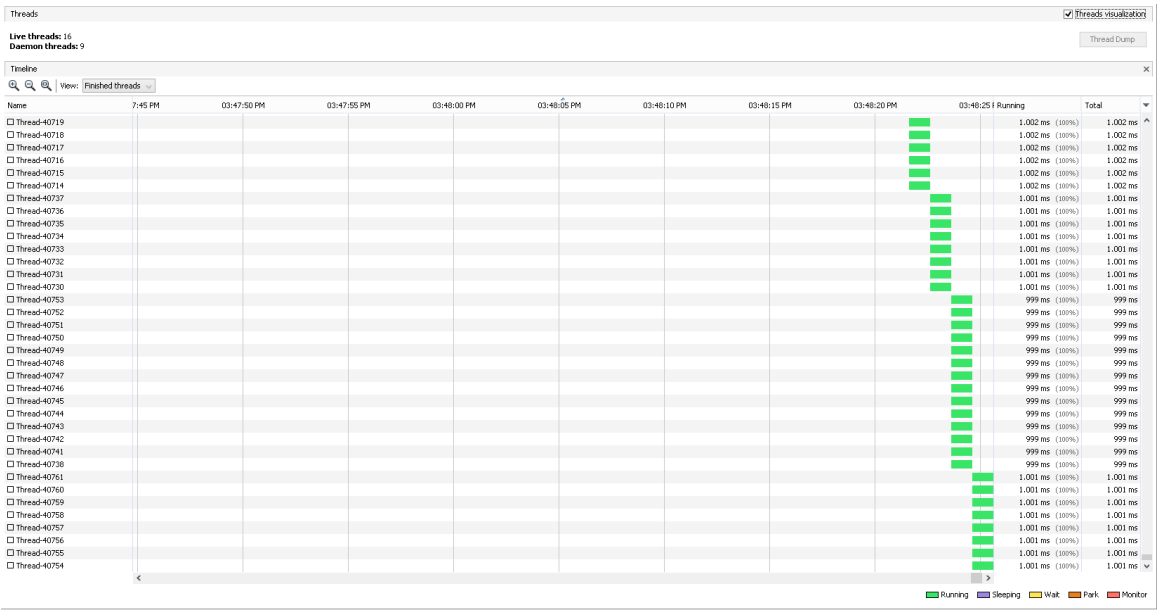

Figura 6.17: *Thread timeline*, con 8 hilos

6.2.4 Modo paralelo, 10 *Threads*

La configuración del *pool* de conexiones usada fue:

Tamaño inicial = 25 conexiones.

Número máximo de conexiones activas = 20.

Número máximo de *prepared statements* activas = 20.

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo paralelo utilizando 10 hilos.

Número de ejecución	**Tiempo de ejecución (segundos)**	**Tiempo de ejecución (minutos)**	**Tiempo de ejecución (horas)**
1	2655.83	44.26	0.74
2	2811.69	46.86	0.78
3	2861.03	47.68	0.79
Promedio	**2776.18**	**46.27**	**0.77**

Tabla 6.7: Resultados en modo paralelo con 10 hilos

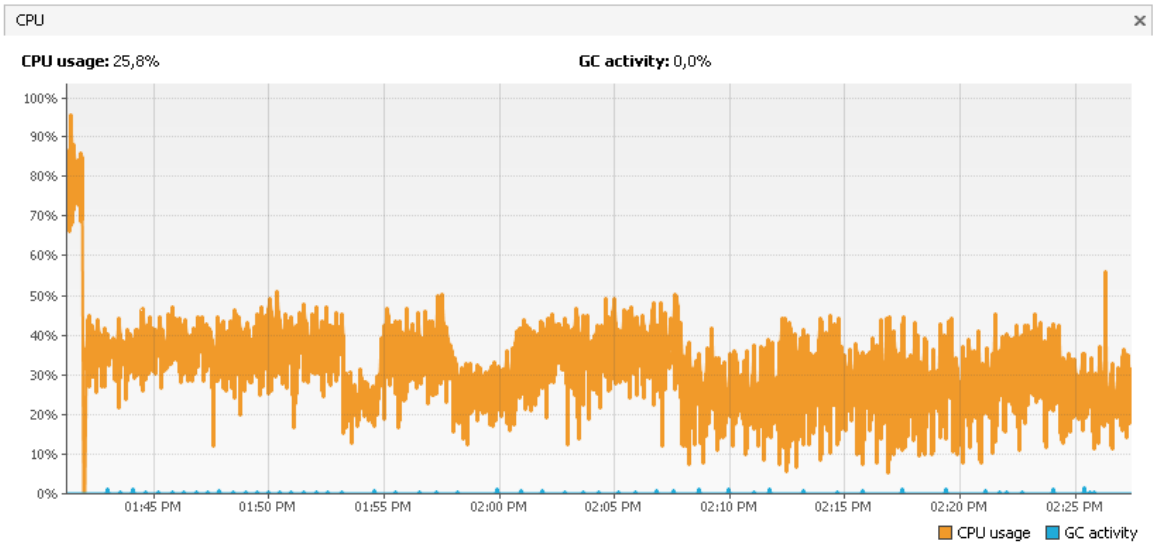

Figura 6.18: *Performance CPU*, con 10 hilos

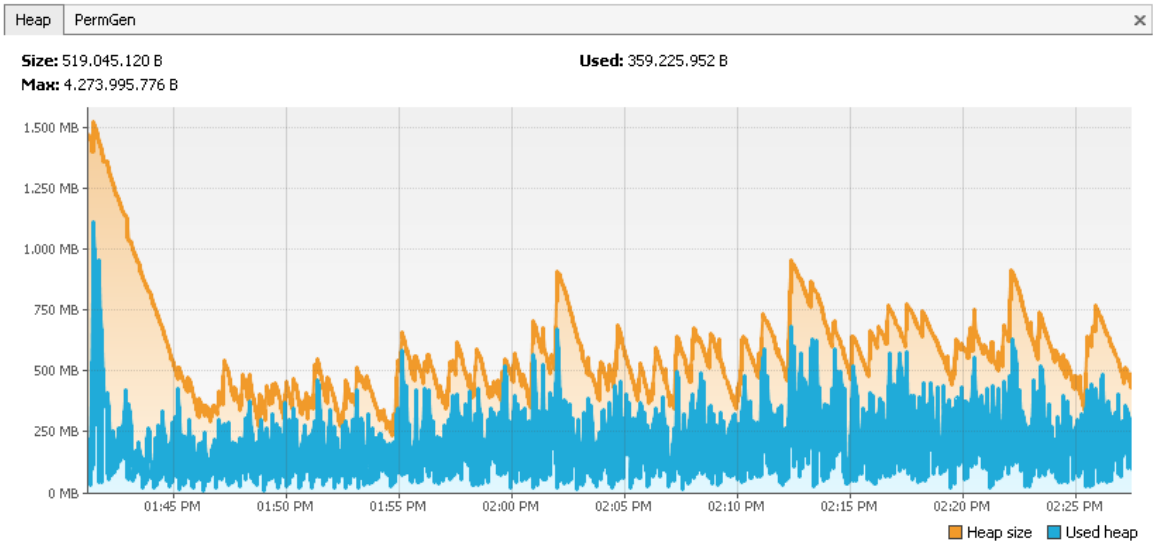

Figura 6.19: *Performance - Heap memory*, con 10 hilos

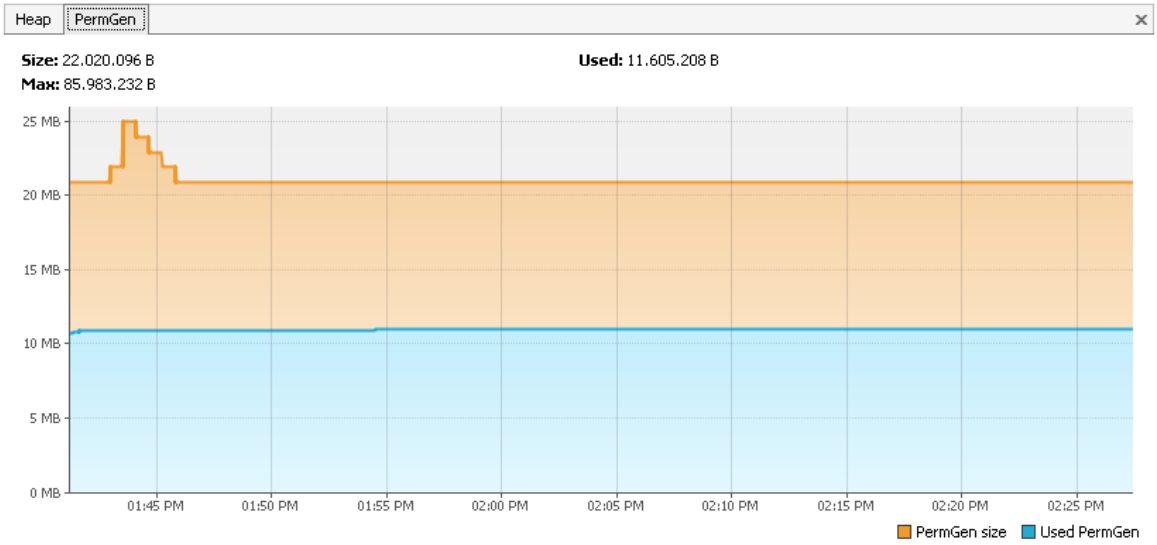

Figura 6.20: *Performance - Permanent Generation heap*, con 10 hilos

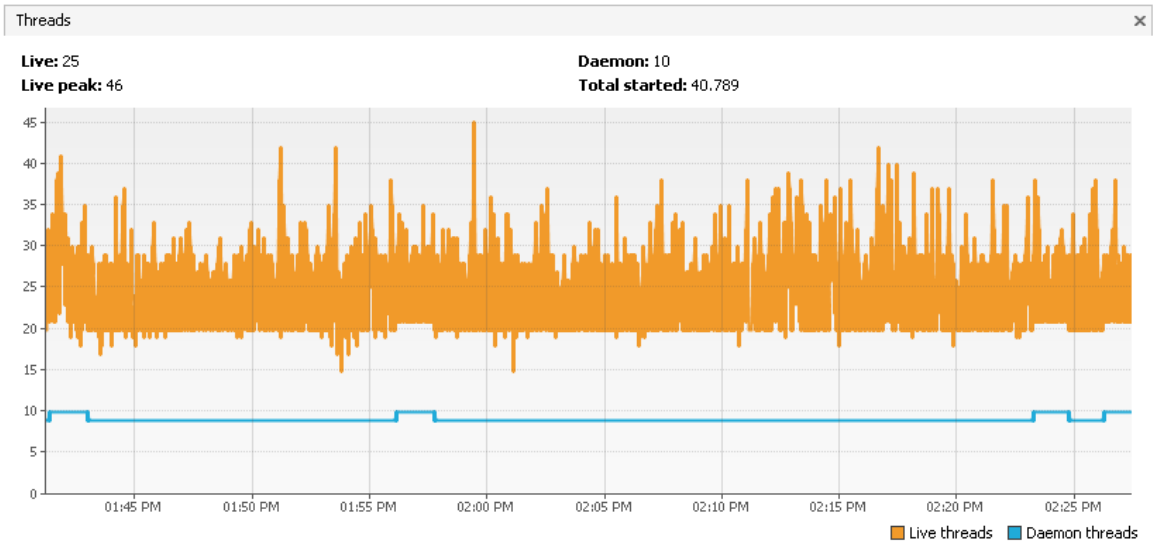

Figura 6.21: *Performance - Live demons and threads*, con 10 hilos

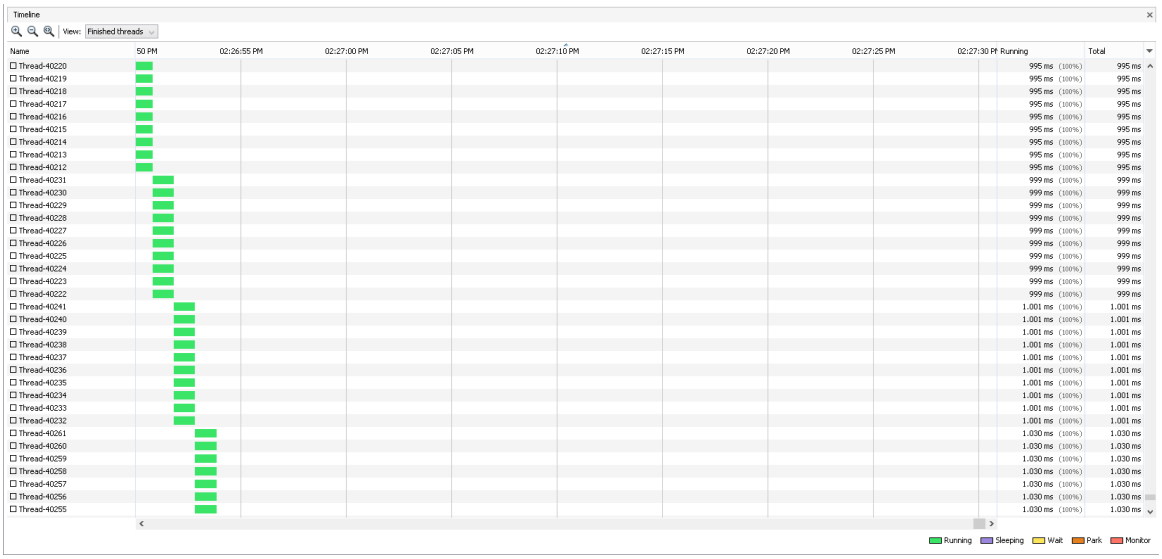

Figura 6.22: *Thread timeline*, con 10 hilos

6.2.5 Modo paralelo, 16 *Threads*

La configuración del *pool* de conexiones usada fue:

Tamaño inicial = 35 conexiones.

Número máximo de conexiones activas = 32.

Número máximo de *prepared statements* activas = 32.

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo paralelo utilizando 16 hilos.

Número de ejecución	Tiempo de ejecución (segundos)	Tiempo de ejecución (minutos)	Tiempo de ejecución (horas)
1	3999.67	66.66	1.11
2	4026.75	67.11	1.12
3	4044.03	67.40	1.12
Promedio	**4023.48**	**67.06**	**1.12**

Tabla 6.8: Resultados en modo paralelo con 16 hilos

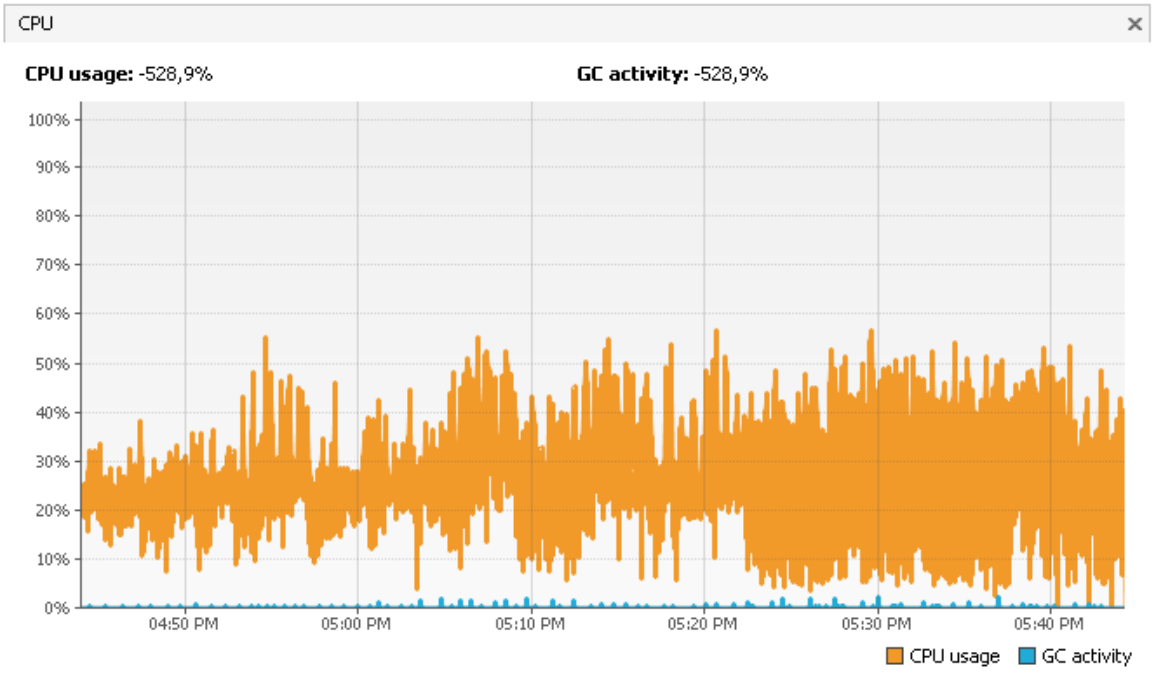

Figura 6.23: *Performance CPU*, con 16 hilos

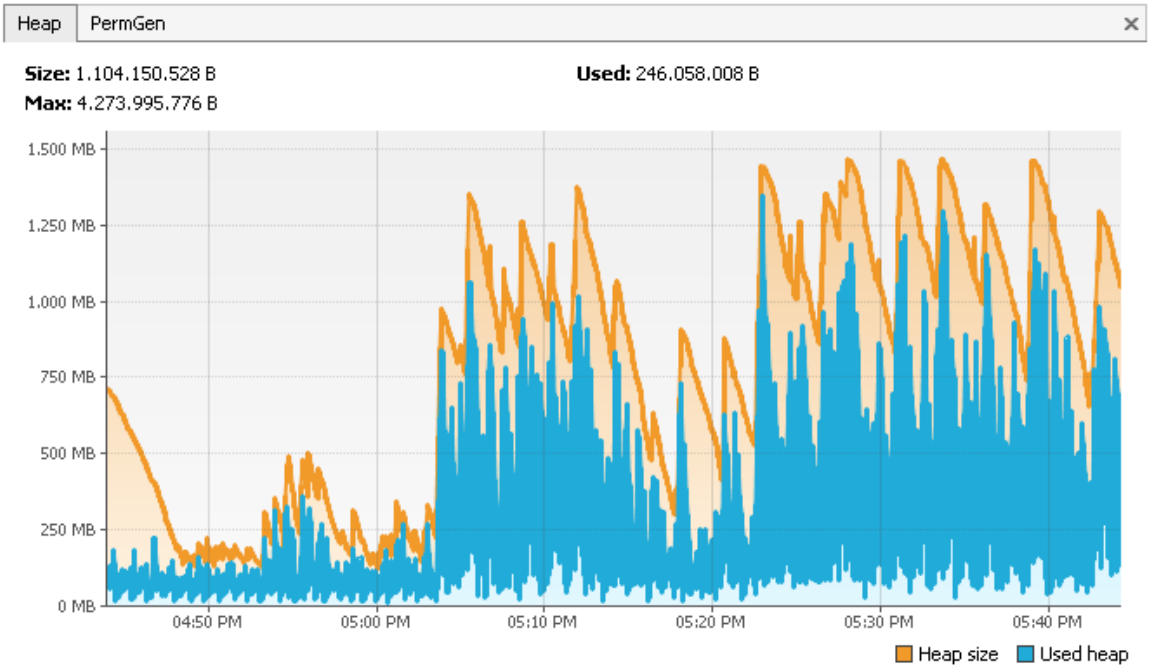

Figura 6.24: *Performance - Heap memory*, con 16 hilos

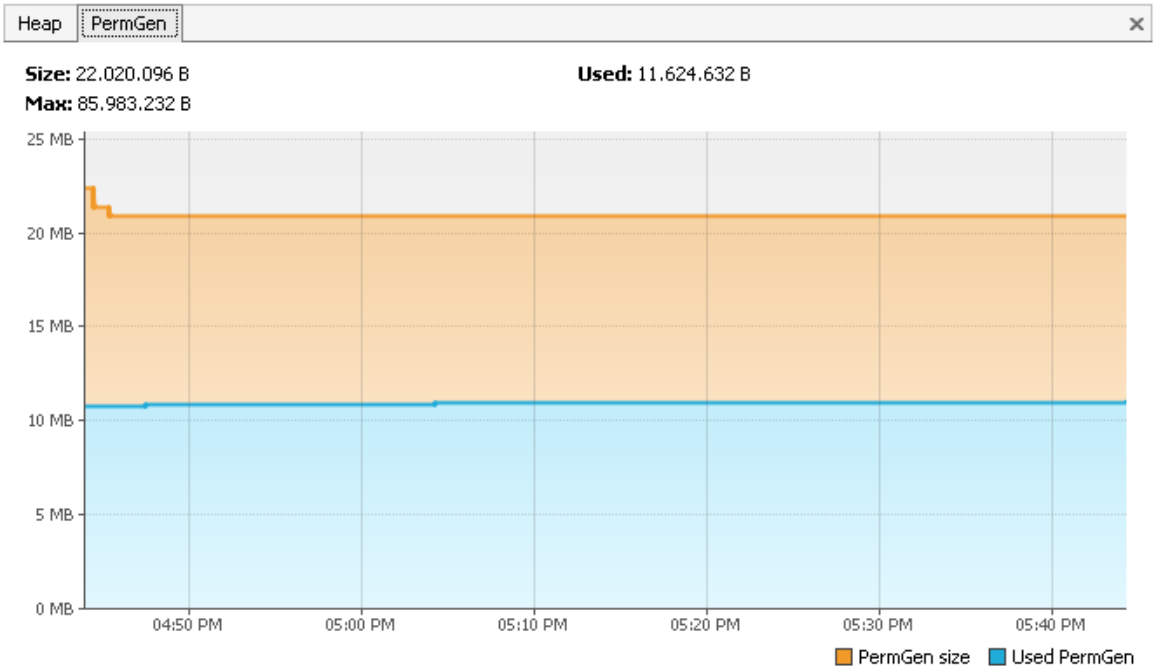

Figura 6.25: *Performance - Permanent Generation heap*, con 16 hilos

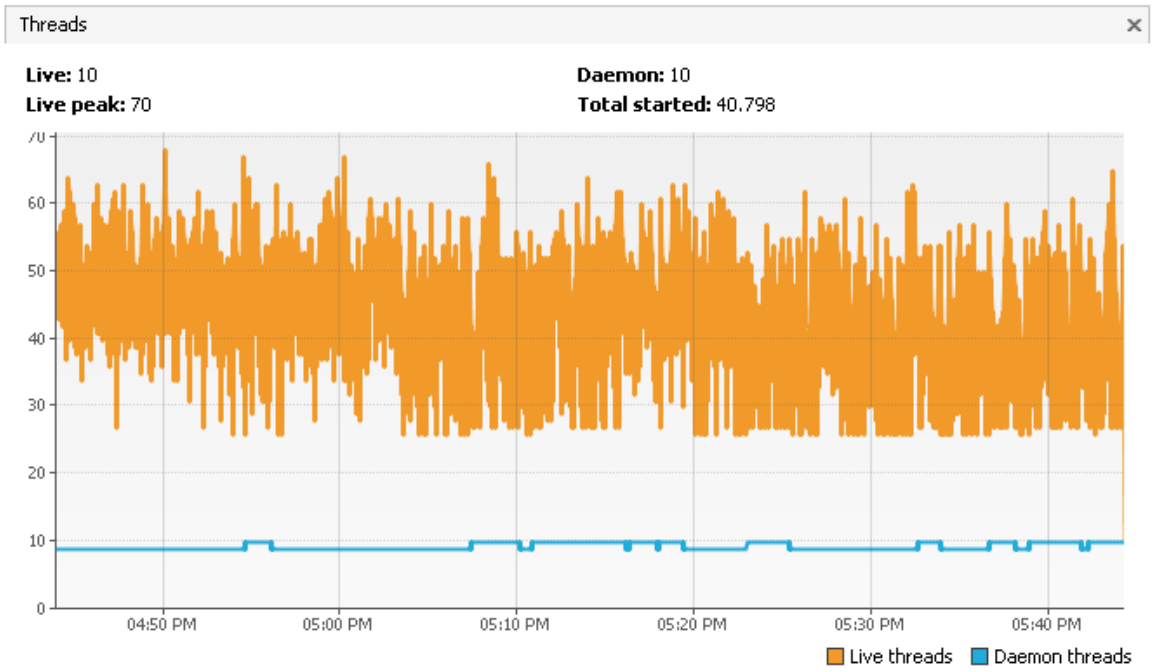

Figura 6.26: *Performance - Live demons and threads*, con 16 hilos

Figura 6.27: *Thread timeline*, con 16 hilos

6.2.6 Modo paralelo, 32 *Threads*

La configuración del *pool* de conexiones usada fue:

Tamaño inicial = 70 conexiones.
Número máximo de conexiones activas = 64.
Número máximo de *prepared statements* activas = 64.

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo paralelo utilizando 32 hilos.

Número de ejecución	**Tiempo de ejecución (segundos)**	**Tiempo de ejecución (minutos)**	**Tiempo de ejecución (horas)**
1	28420.19	473.67	7.89
2	28268.55	471.14	7.85
3	28735.94	478.93	7.98
Promedio	**28474.89**	**474.58**	**7.91**

Tabla 6.9: Resultados en modo paralelo con 32 hilos

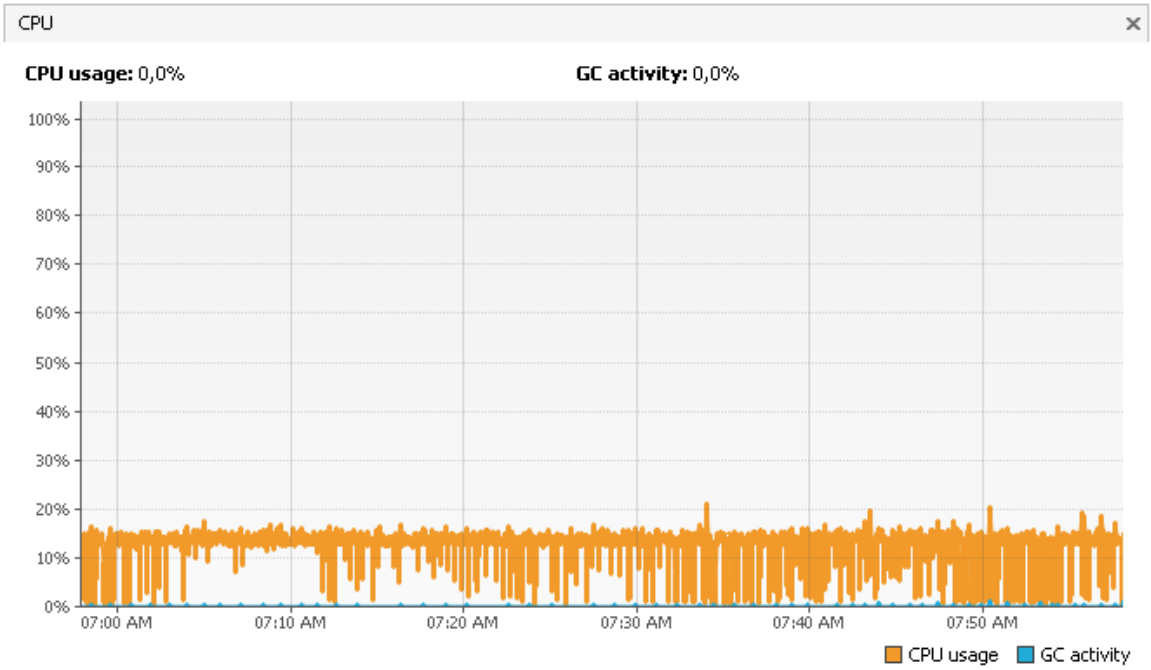

Figura 6.28: *Performance CPU*, con 32 hilos

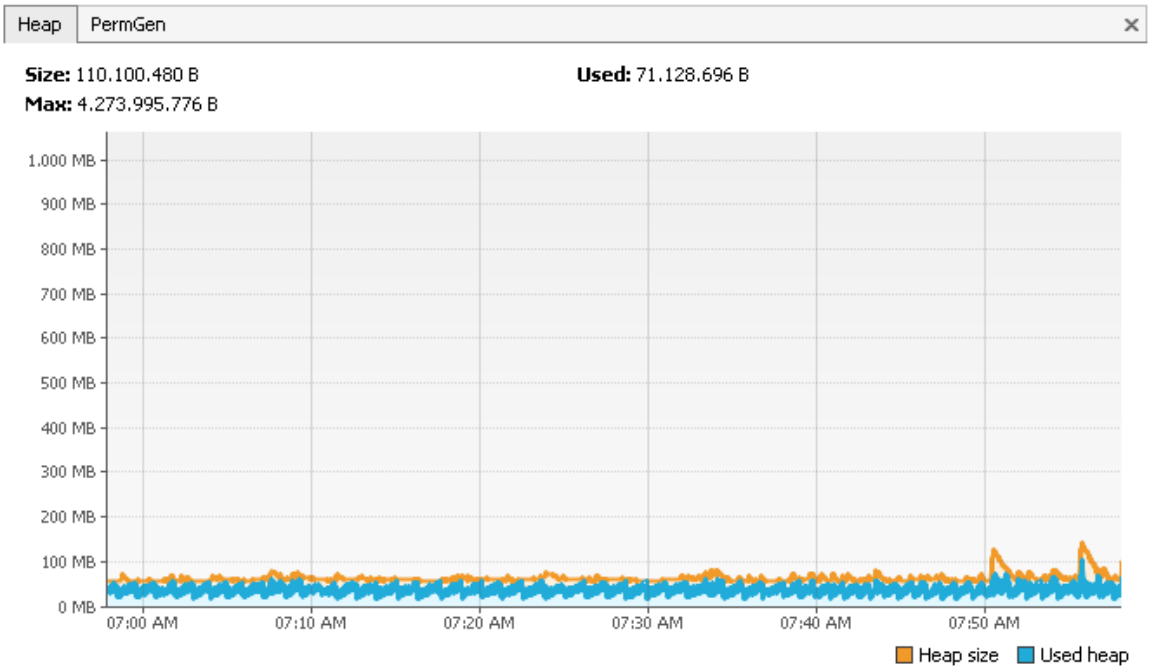

Figura 6.29: *Performance - Heap memory*, con 32 hilos

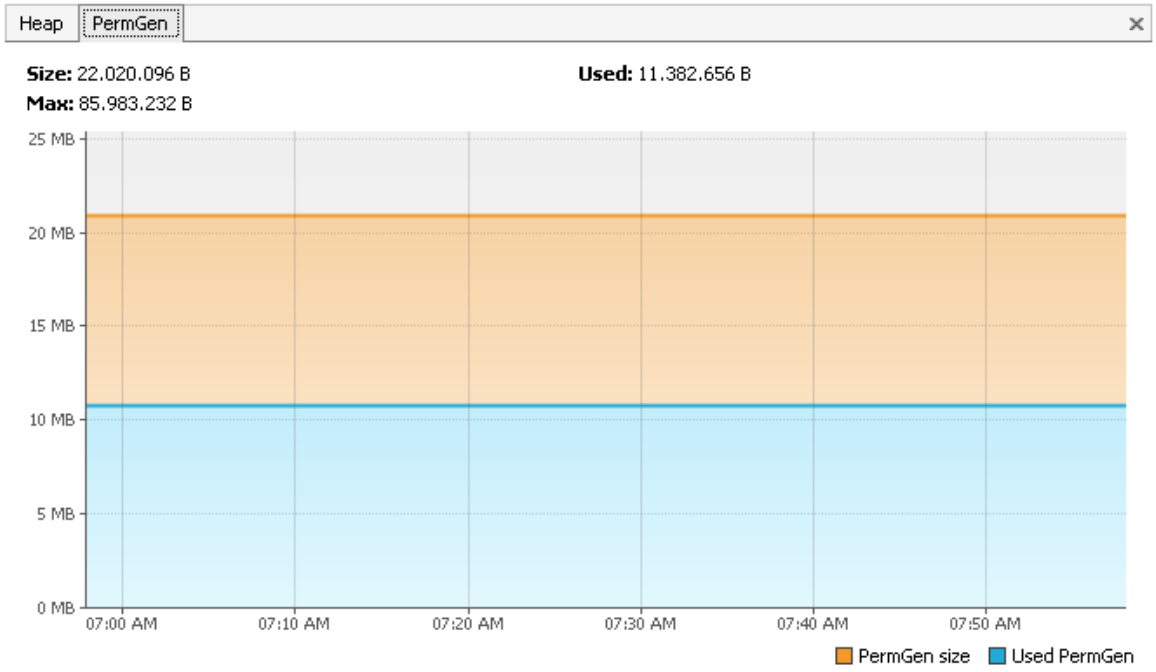

Figura 6.30: *Performance - Permanent Generation heap*, con 32 hilos

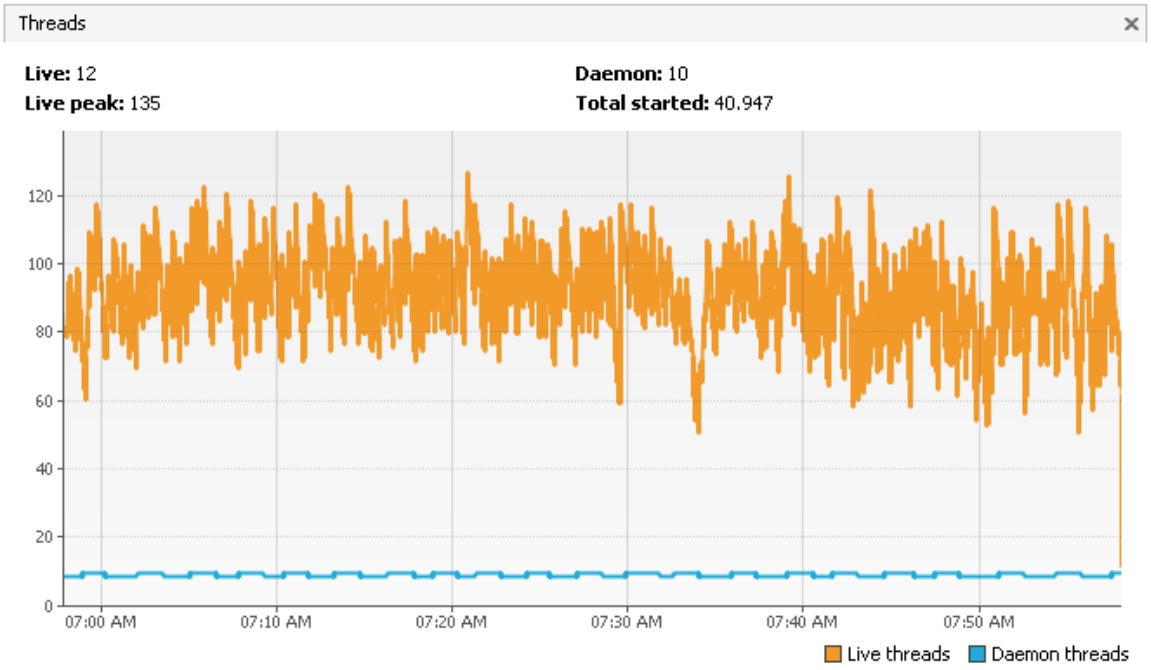

Figura 6.31: *Performance - Live demons and threads*, con 32 hilos

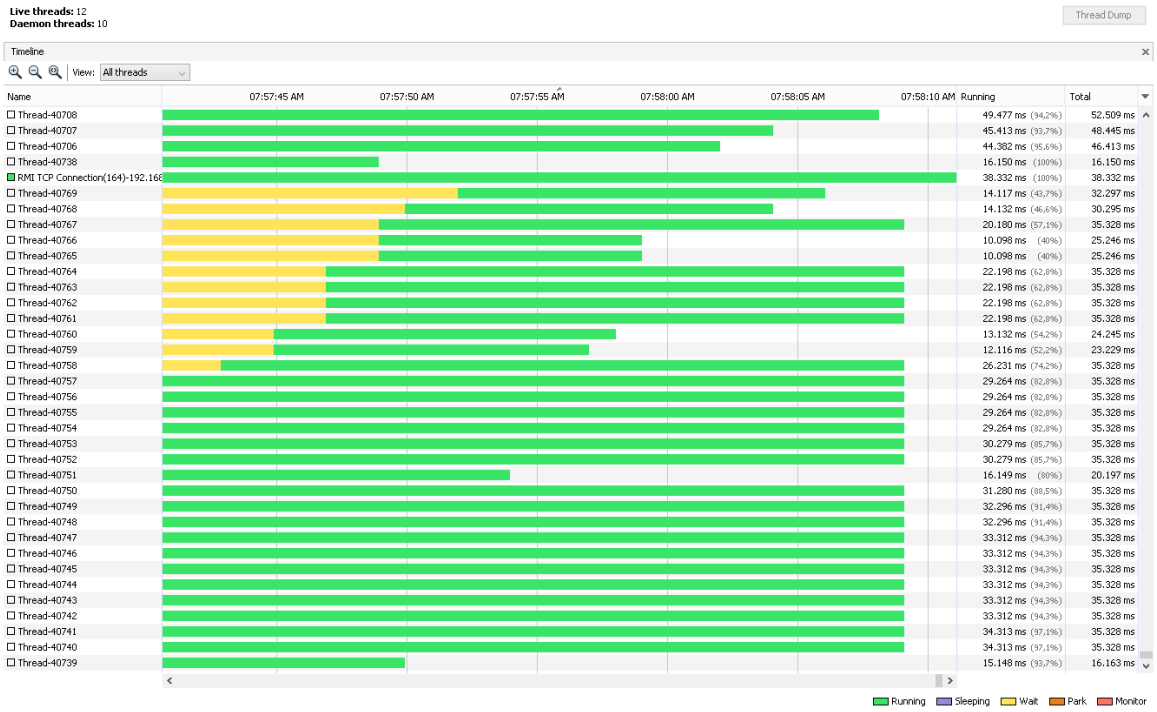

Figura 6.32: *Thread timeline*, con 32 hilos

6.2.7 Modo paralelo, 50 *Threads*

La configuración del *pool* de conexiones usada fue:

Tamaño inicial = 100 conexiones.

Número máximo de conexiones activas = 75.

Número máximo de *prepared statements* activas = 75.

En la siguiente tabla se muestran los resultados obtenidos de las ejecuciones en modo paralelo utilizando 50 hilos.

Número de ejecución	Tiempo de ejecución (segundos)	Tiempo de ejecución (minutos)	Tiempo de ejecución (horas)
1	30543.12	509.05	8.48
2	31362.10	522.70	8.71
3	32208.99	536.82	8.95
Promedio	**31371.40**	**522.86**	**8.71**

Tabla 6.10: Resultados en modo paralelo con 50 hilos

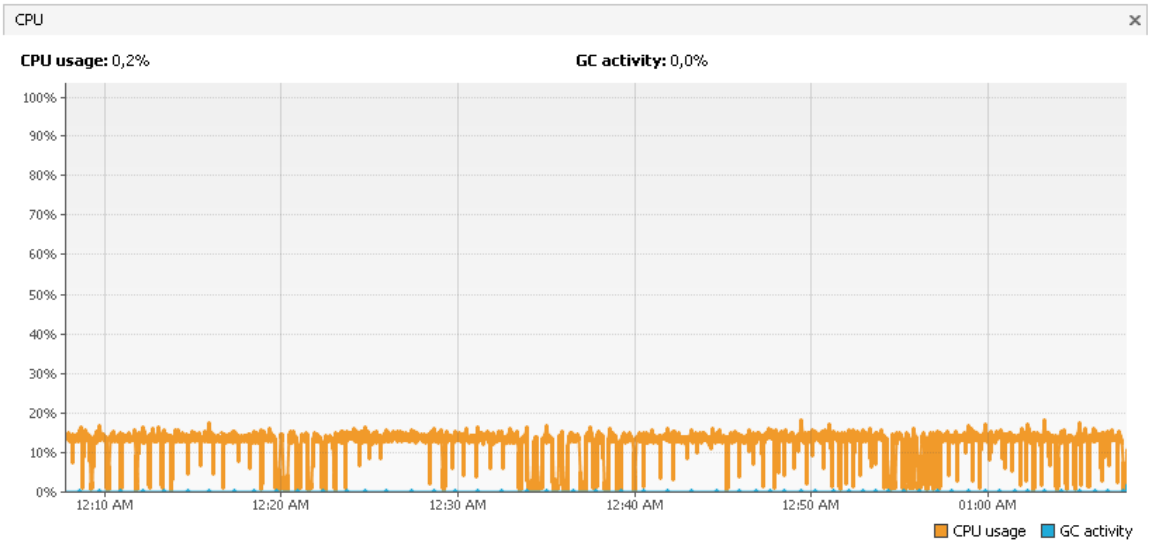

Figura 6.33: *Performance CPU*, con 50 hilos

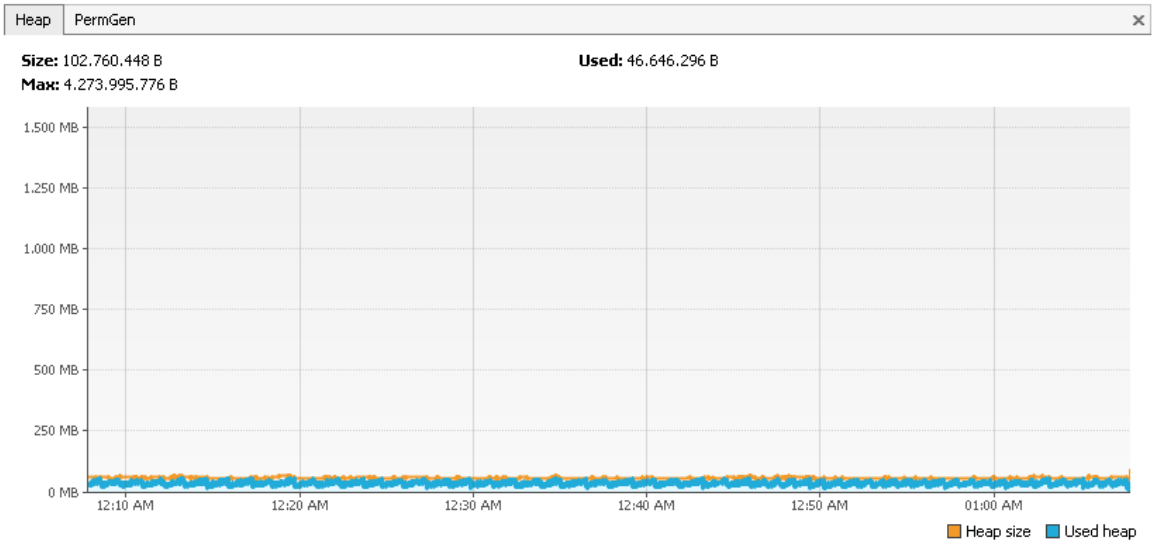

Figura 6.34: *Performance - Heap memory*, con 50 hilos

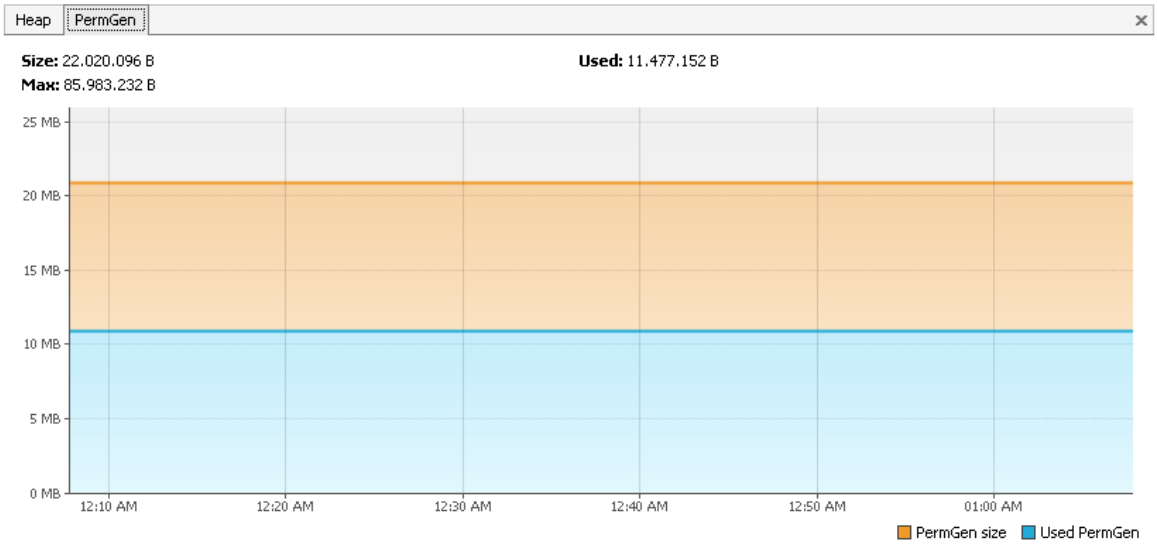

Figura 6.35: *Performance - Permanent Generation heap*, con 50 hilos

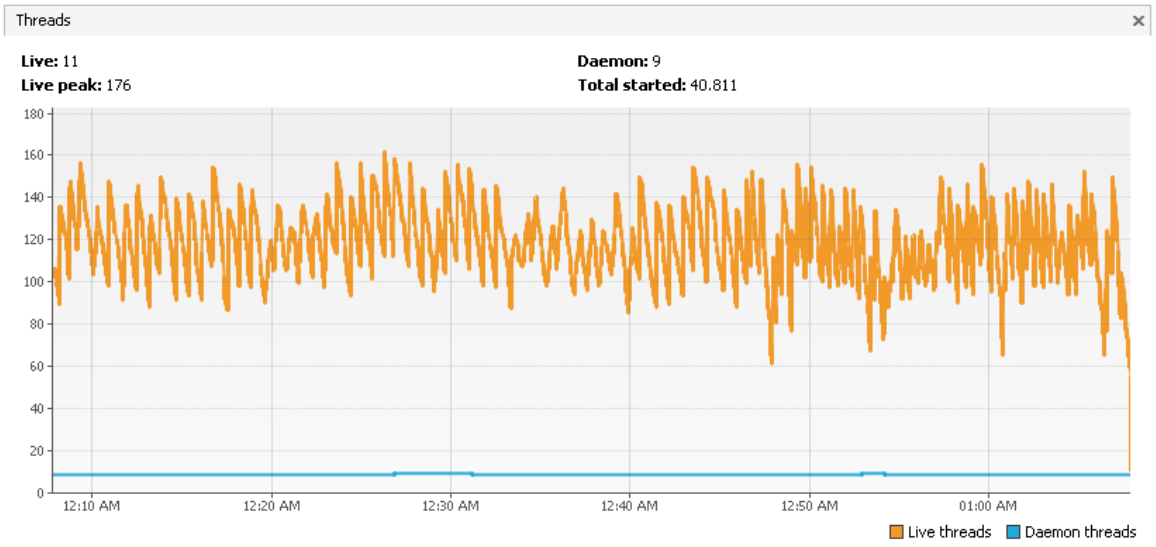

Figura 6.36: *Performance - Live demons and threads*, con 50 hilos

Figura 6.37: *Thread timeline*, con 50 hilos

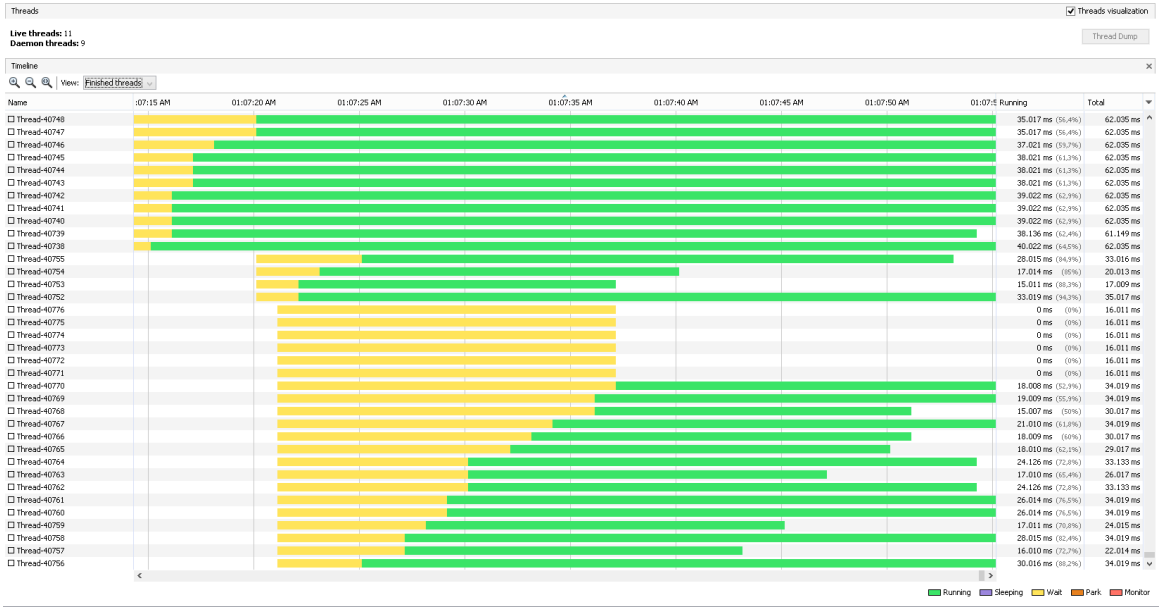

Figura 6.38: *Thread timeline*, segunda parte, con 50 hilos

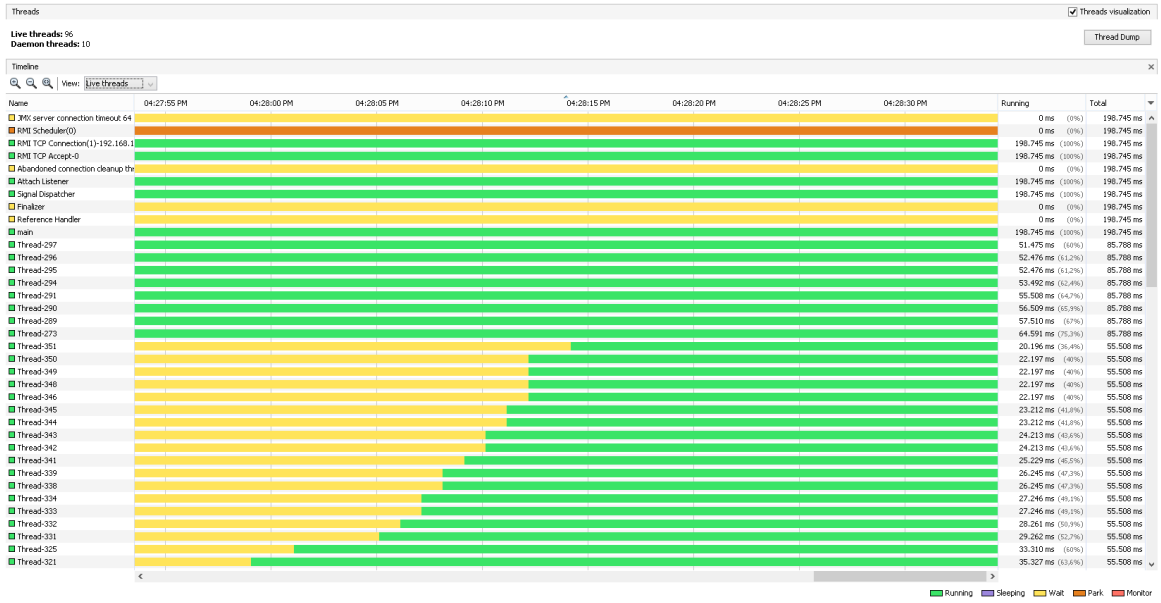

Figura 6.39: *Thread timeline*, tercera parte, con 50 hilos

6.2.8 Resumen de las pruebas

En la presente sección se muestra una tabla resumen con los resultados arrojados de las diferentes pruebas.

Prueba	Cantidad de hilos	Tiempo promedio de ejecución (minutos)	Total archivos procesados con éxito	Total archivos procesados con fallas	Aceleración
Secuencial	1	101.15	40775	0	1X
Paralelo Procesadores	8	48.34	40775	0	2.09X
Paralelo Procesadores	10	46.27	40775	0	2.19X
Paralelo Procesadores	16	67.06	40775	0	1.51X
Paralelo Procesadores	32	474.58	40775	0	0.21X
Paralelo Procesadores	50	522.86	40775	0	0.19X

Tabla 6.11: Resumen: resultados de las pruebas

Como se puede evidenciar en la tabla 6.11, el mejor tiempo de ejecución y el mejor aprovechamiento de los recursos computacionales se logró con la prueba en paralelo con 10 hilos de ejecución.

6.2.9 Clustering

A continuación se detallan los resultados de la prueba de *clustering* para los valores tomados de los *observation files* de aquellas épocas que tuvieran algún deslizamiento de ciclo en los tipos L1 y L2:

Número de instancias: 3,874,181
Número de iteraciones realizadas para construir el modelo: 28
Tiempo que tardó en construirse el modelo: 227.85 segundos.

Atributo	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Epoch date	2014-04-30 22:35:13	2014-04-30 23:32:31	2014-04-30 22:16:12	2014-04-30 22:16:07
Satélites disponibles	19.03	15.67	18.17	15.76
Satélites Glonass	8.14	6.83	8.41	6.18
Satélites GPS	10.89	8.84	9.77	9.58
Satélites SBAS	0	0	0.0005	0
Cantidad de deslizamientos de ciclo en L1 y L2	55.62	59.55	18.94	21.38

Tabla 6.12: Resultados *clustering* centroides

Es necesario aclarar que los valores expuestos hacen referencia a la media o promedio de los valores observados.

Cluster	Cantidad de instancias	Porcentaje
1	821,240	21
2	1,394,169	36
3	758,616	20
4	900,156	23

Tabla 6.13: Resultados *clustering* instancias

En la presente tabla se exponen los resultados de la prueba de *clustering*, en donde los datos se encuentran agrupados por días.

Número de instancias: 134
Número de iteraciones realizadas para construir el modelo: 5
Tiempo que tardó en construirse el modelo: 0.02 segundos.

Atributo	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Epoch date	2014-05-08	2014-05-01	2014-05-21	2014-04-30
Satélites disponibles	9.83	17.73	18	16.35
Satélites Glonass	8.33	8.37	8.55	6.27
Satélites GPS	1.5	9.37	9.44	10.08
Satélites SBAS	0	0	0	0
Cantidad de deslizamientos de ciclo en L1 y L2	801,016.67	496,652.85	2,431,711.04	1,396,281.27

Tabla 6.14: Resultados *clustering* centroides agrupado por días

Cluster	Cantidad de instancias	Porcentaje
1	6	4
2	52	39
3	27	20
4	49	37

Tabla 6.15: Resultados *clustering* instancias agrupado por días

6.3 Análisis de los resultados

De acuerdo a lo presentado en la sección anterior se observó que el rendimiendo de la aplicación no mejoró al configurarle un gran número de hilos, sino que por el contrario, su rendimiento cayó a tal punto que es mejor ejecutar el programa de modo secuencial que de modo paralelo. Este resultado no fue algo tan esperado por la investigación, dado que siempre se procuro por buscar el mejor rendimiento posible.

Sin embargo, se encontró mejoría al correr el programa con 8 o 10 hilos, lo cual aumentó el rendimiento casi el triple de veces. Esto se debe a que a los programas paralelos dependen de encontrar el mejor balance entre el hardware disponible y el compilador. Se evidenció que entre más se sobrepasaba la capacidad de *cores* del *workstation*, mayor era

el tiempo que tomaba en ejecutar la aplicación, al crear colas de procesos para la asignación dínamica de recursos y se creaba mayor *overhead* en la comunicación entre los *threads*.

Como era de esperarse, el *Perm Space* se mantuvo sin mayores cambios a lo largo de las diferentes pruebas, con algunos picos (25 MB) cuando se optimiza el hardware del *workstation*, es decir, cuando se hace un uso efectivo de los multiprocesadores.

Por el contrario, el *Heap Space* tuvo diferentes cambios a lo largo de las pruebas. En el modo serial se comportó casi de manera constante con un uso aproximadamente de unos 128MB. En el modo paralelo con 8 y 10 *Threads* obtuvo un pico al comienzo de la ejecución de un poco más de 1000MB, esto debido a la primera asignación de recursos por parte del Sistema Operativo, mientras se distribuía entre los *cores*. Luego al distribuir de manera eficiente la carga se visualiza un promedio de unos 450MB, con ciertos picos hasta casi los 700MB debido a los diferentes tamaños de los archivos.

Con el uso 16 *Threads* resultó en un gasto considerable de memoria, pues en 3/4 de la ejecución se mantuvo un alto consumo de espacio, el promedio fue de aproximadamente los 1000MB. Esto quiere decir que la cola de procesos obtuvo una rápida asignación de recursos por parte del sistema operativo y se mantuvo con muchos objetos en memoria dado que la recuperación de memoria fue más lenta que la instanciación de objetos de la aplicación.

En cuanto a las pruebas con más de 16 *Threads*, se obtuvieron valores constantes, alrededor de los 90MB en el *Heap Space*, lo cual lo interpretamos de que la cola de procesos era muy grande y por lo cual el sistema operativo distribuyó las cargas de trabajo sin mayor prioridad entre los diferentes *cores*, lo cual hizo que estos valores no cambiaran a lo largo de la ejecución y se presentara un rendimiento menor que en el modo secuencial.

Se puede evidenciar fácilmente a través de los gráficos de "vida" de los *Threads* el tiempo promedio y de la cantidad de hilos activos en cierto período de ejecución, y en el cual

se comprueba una gran cantidad de hilos activos al realizarce un mayor paralelismo en el ambiente de pruebas.

Para detallar un poco la diferencia entre el *Heap Space* y el *Perm Space* es que el primero almacena los datos de los objetos instanciados mientras que el segundo guarda las definiciones de las clases cargadas en la *JVM*. También es importante que el ciclo de vida del *Heap Space* esté ligado con la aplicación y que el *Perm Space* esté ligado con la *JVM*.

Por último, haciendo referencia a las pruebas de *clustering*, no fueron concluyentes los resultados con los datos colocados a la herramienta Weka. No se puede dar ningún resultado relevante porque los grupos contaron con información muy similar y no hubo una clara evidencia de diferenciación entre ellos.

CAPÍTULO 7

Conclusiones

7.1 Conclusiones

Existe un método económico para extraer las observaciones y diferentes lecturas de datos de los sistema de posicionamiento global en tiempo real, y con los cuales se puede trabajar en diferentes áreas de la ciencia.

La velocidad y tamaño de la descarga de los archivos depende de la red de comunicaciones y de la disponibilidad de los diferentes transmisores o *broadcasters*, por lo que hay una dependencia externa para poder extraer los datos.

El desarrollo de aplicaciones que hacen uso de las optimizaciones del lenguaje de programación dan mayor confiabilidad y eficiencia, como el uso y buena administración de hilos de proceso, uso de un *pool* de conexiones hacia la base de datos, uso de inserción en conjunto o *batch*.
Esto es principalmente útil para aquellas aplicaciones que tienen una alta carga y manejo de datos, y así mismo una masiva comunicación con una base de datos, en donde una administración eficiente del *Driver* marca la diferencia en el rendimiento. Haciendo uso de esas prácticas se logró que en menos de 60 minutos se procesaran alrededor de 81 GB de datos.

Se comprobó que debe existir un balance en el nivel de paralelismo que se realiza en

el software para que aproveche al máximo los recursos de hardware que se poseen.

A través de un meta-modelo bien definido, es posible encontrar respuestas a preguntas no formuladas explícitamente, por ejemplo encontrar las frecuencias de las fallas.

Las técnicas de minería de datos son eficientes, en particular, la de agrupación o *clustering* con el algoritmo de K-Means, es útil para encontrar diferentes patrones que no se encuentran de manera explícita, dentro de una gran cantidad de datos. Para el caso de esta investigación, estos patrones no se encontraron fácilmente y por consiguiente los resultados no fueron concluyentes. Esto quiere decir que se debe complementar el algoritmo de K-Means con algún otro.
Se requiere de un mayor esfuerzo para encontrar los patrones y esto depende de la definición y organización de los datos, como la especificación de las características a agrupar.

Los usos que se le pueden dar a los datos descargados puede variar, pero principalmente, nos sirve para la detección de fallas entre las diferentes épocas (*epoch date*), que en este caso particular no mostró ningún error; la detección y cuantificación de fallas de los tipos de observaciones L1 y L2, como se evidencio a través del *Loss of Lock Indicator*; y por último para realizar análisis a través de técnicas de minería de datos con el objetivo de encontrar patrones ocultos.

7.2 Trabajo futuro

Referente a el trabajo futuro hay bastante labor que realizar en esta área de investigación, no solo por el hecho de ser un tendencia relativamente nueva, sino por el hecho que involucra diferentes tipos de tecnologías.

En ese sentido se debería explotar el computo paralelo para la aplicación RINEX ETL haciendo uso del *framework* de CUDA. También realizar la implementación de un sistema de archivos distribuido como HDFS (*Hadoop Distributed File System*) dentro del sistema

de tranferencia y extracción del conocimiento para los GNSS (STECG).

Por otro lado, se podría implementar el algoritmo de *Map Reduce* con el objetivo de mejorar los procesos de extracción y transformación. Además, ampliar el alcance de los datos recolectados y procesados para proporcionar más servicios, como por ejemplo, el cálculo de coordenadas y visualizarlos a través de Google Maps. También mejorar la aplicación adicionando un módulo de post-procesamiento para realizar las correciones de los datos medidos y un componente integrado de minería de datos y visualización.

Evaluar otras técnicas de minería de datos y otros algoritmos que permitan identificar los patrones ocultos de esa enorme cantidad de datos.

Implementar una interfaz de usuario con patrones de usabilidad para una fácil adopción de la aplicación, como así mismo convertir el programa en una aplicación web que este disponible para el uso científico. Se podría implementar hasta el punto de convertirse en una aplicación que haga uso de un *grid* computacional.

Bibliografía

[1] E. BRYNJOLFSSON and A. MCAFEE, "The big data boom is the innovation story of our time," *McKinsey & Company*, Nov. 2011. [Online]. Available: http://www.theatlantic.com/business/archive/2011/11/the-big-data-boom-is-the-innovation-story-of-our-time/248215/

[2] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Hung Byers, "Big data: The next frontier for innovation, competition, and productivity," *The Atlantic*, May 2011. [Online]. Available: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation/

[3] D. J. P. Thomas H. Davenport, "Data scientist: The sexiest job of the 21st century." *Harvard Business Review*, vol. 90, pp. 70–76, 2012.

[4] D. E. Avison, "Merise: A european methodology for developing information systems," *European Journal of Information Systems*, vol. 1, p. 183Ű192, 1991. [Online]. Available: http://dx.doi.org/10.1057/ejis.1991.33

[5] S. W. Ambler, "Scaling scrum," *Dr. Dobb's Journal*, vol. 5, pp. 52–54, May 2008.

[6] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems GPS, GLONASS, Galileo, and more.* Wien; New York: Springer, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-211-73017-1

[7] L. Self, "Use of data mining on satellite data bases for knowledge extraction." in *FLAIRS Conference*, 2000, pp. 149–152. [Online]. Available: http://www.aaai.org/Papers/FLAIRS/2000/FLAIRS00-029.pdf

[8] D. N. R. Azevedo, "Application of data mining techniques to the storage management and online distribution of satellite images." IEEE, Oct. 2007, pp. 955–960. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4389731

[9] D. R. Azevedo, A. M. Ambrosio, and M. Vieira, "Applying data mining for detecting anomalies in satellites." IEEE, May 2012, pp. 212–217. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6214776

[10] S.-S. Ho and A. Talukder, "Automated cyclone discovery and tracking using knowledge sharing in multiple heterogeneous satellite data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 928–936. [Online]. Available: http://dl.acm.org/citation.cfm?id=1402001

[11] V. F. Dvorak and U. States., *Tropical cyclone intensity analysis using satellite data [microform] / Vernon F. Dvorak.* U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Environmental Satellite, Data, and Information Service Washington, D.C, 1984.

[12] M. R. Sinclair, "Objective Identification of Cyclones and Their Circulation Intensity, and Climatology," *Wea. Forecasting*, vol. 12, no. 3, pp. 595–612, Sep. 1997.

[13] Y. Li, Y. Wang, J. Yan, and Y. Qi, "The application of data mining in satellite TV broadcasting monitoring." IEEE, Apr. 2009, pp. 357–359. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5193970

[14] S. C. Tay, W. Hsu, K. H. Lim, and L. C. Yap, "Spatial data mining: Clustering of hot spots and pattern recognition," in *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, vol. 6, 2003, p. 3685âĂŞ3687. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1295237

[15] Y. Guo, J. G. Liu, M. Ghanem, K. Mish, V. Curcin, C. Haselwimmer, D. Sotiriou, K. K. Muraleetharan, and L. Taylor, "Bridging the macro and micro: A computing intensive earthquake study using discovery net."

in *SC*. IEEE Computer Society, 2005, p. 68. [Online]. Available: http://dblp.uni-trier.de/db/conf/sc/sc2005.html#GuoLGMCHSMT05

[16] E. Lenz, "Networked transport of rtcm via internet protocol (ntrip) Ű application and benefit in modern surveying systems," in *Standards, Quality Assurance and Calibration*, 2004.

[17] H. G. G. Weber, D. Dettmering, "Networked transport of rtcm via internet protocol (ntrip)," *Federal Agency for Cartography and Geodesy*, 2003.

[18] *BKG Ntrip Client (BNC)*, Version 2.10 ed., FEDERAL AGENCY FOR CARTOGRAPHY AND GEODESY, FRANKFURT, ALEMANIA, Diciembre 2013.

[19] D. Erickson, J. Daniel, M. Allen, A. R. Ganguly, F. M. Hoffman, S. Pawson, L. Ott, and E. Neilson, "Data mining geophysical content from satellites and global climate models." in *ICDM Workshops*, Y. Saygin, J. X. Yu, H. Kargupta, W. W. 0010, S. Ranka, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, 2009, pp. 214–216. [Online]. Available: http://dblp.uni-trier.de/db/conf/icdm/icdmw2009.html#EricksonDAGHPON09

[20] T. S. Korting, L. M. G. Fonseca, M. I. S. Escada, F. C. da Silva, and Dos, "Geodma - a novel system for spatial data mining," in *Data Mining Workshops, 2008. ICDMW '08. IEEE International Conference on*, 2008, pp. 975–978. [Online]. Available: http://dx.doi.org/10.1109/ICDMW.2008.22

[21] H. Cheng, P.-N. Tan, C. Potter, and S. A. Klooster, "Data mining for visual exploration and detection of ecosystem disturbances." in *GIS*, W. G. Aref, M. F. Mokbel, and M. Schneider, Eds. ACM, 2008, p. 60. [Online]. Available: http://dblp.uni-trier.de/db/conf/gis/gis2008.html#ChengTPK08

[22] Y.-B. Yang, H. L. 0002, and J. Jiang, "Cloud analysis by modeling the integration of heterogeneous satellite data and imaging." *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 36, no. 1, pp. 162–172, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/tsmc/tsmca36.html#YangLJ06

[23] J. M. Dow, R. E. Neilan, and C. Rizos, "The International GNSS Service in a changing landscape of Global Navigation Satellite Systems," *Journal of Geodesy*, vol. 83, pp. 191–198, Mar. 2009.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[25] D. Gibson, J. M. Kleinberg, and P. Raghavan, "Clustering categorical data: An approach based on dynamical systems," *VLDB Journal: Very Large Data Bases*, vol. 8, no. 3–4, pp. 222–236, 2000. [Online]. Available: http://citeseer.ist.psu.edu/cache/papers/cs/157/http:zSzzSzcs.cornell.eduzSzInfozSzPeoplezSzkleinberzSzvldb98.pdf/gibson98clustering.pdf

[26] R. Motta, A. de Andrade Lopes, B. M. Nogueira, S. O. Rezende, A. M. Jorge, and M. C. F. de Oliveira, "Comparing relational and non-relational algorithms for clustering propositional data." in *SAC*, S. Y. Shin and J. C. Maldonado, Eds. ACM, 2013, pp. 150–155. [Online]. Available: http://dblp.uni-trier.de/db/conf/sac/sac2013.html#MottaLNRJO13

[27] E.-J. Son, I.-S. Kang, T.-W. Kim, and K.-J. Li, "A spatial data mining method by clustering analysis," in *Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems*, ser. GIS '98. New York, NY, USA: ACM, 1998, pp. 157–158. [Online]. Available: http://doi.acm.org/10.1145/288692.288720

[28] C. Li, M. Wang, L. Lim, H. Wang, and K. C.-C. Chang, "Supporting ranking and clustering as generalized order-by and group-by," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 127–138. [Online]. Available: http://doi.acm.org/10.1145/1247480.1247496

[29] goGPS project, "gogps project - open source positioning software." [Online]. Available: http://www.gogps-project.org/

[30] Oracle, "Java documentation," consultado el 3 de julio de 2014. [Online]. Available: http://docs.oracle.com/javase/7/docs/api/

[31] T. A. S. Foundation, "Apache commons dbcp configuration." [Online]. Available: http://commons.apache.org/proper/commons-dbcp/configuration.html

[32] ——, "Apache license, version 2.0," January 2004. [Online]. Available: http://www.apache.org/licenses/LICENSE-2.0

[33] Oracle, "Mysql connector/j developer guide," consultado el 31 de julio de 2014. [Online]. Available: http://dev.mysql.com/doc/connector-j/en/

[34] F. S. Foundation, "Gnu lesser general public license, version 3," June 2007. [Online]. Available: https://www.gnu.org/licenses/lgpl.html

[35] E. Microsoft, "Tendencias big data en empresas globales 2013," Feb. 2013, consultado el 25 de octubre de 2013. [Online]. Available: http://blogs.technet.com/b/microsoftlatam/archive/2013/02/15/infograf-205-a-tendencias-big-data-en-empresas-globales-2013.aspx

[36] J. Rick, "Bulk insert into a mysql database," March 2010, consultado el 5 de Agosto de 2014. [Online]. Available: http://jeffrick.com/2010/03/23/bulk-insert-into-a-mysql-database/

[37] J. Sutherland, "Batch writing, and dynamic vs parametrized sql, how well does your database perform?" May 2013, consultado el 5 de agosto de 2014. [Online]. Available: http://java-persistence-performance.blogspot.mx/2013/05/batch-writing-and-dynamic-vs.html

[38] P. Kumar, "Jdbc batch processing example tutorial with insert statements," *JournalDev*, January 2014. [Online]. Available: http://www.journaldev.com/2494/jdbc-batch-processing-example-tutorial-with-insert-statements

[39] M. Abernethy, "Data mining with weka, part 2: Classification and clustering," May 2010, consultado el 5 de septiembre de 2014. [Online]. Available: http://www.ibm.com/developerworks/library/os-weka2/

APÉNDICE A

Consultas SQL

A continuación se muestran alguna de las consultas implementadas a lo largo de esta investigación.

Datos para las pruebas de *clustering*.

```
-- Datos para clustering
SELECT O.OBS_EPOCH_DATE, O.OBS_AVAILABLE_SATS, O.OBS_GLONASS_SATS,
    O.OBS_GPS_SATS, O.OBS_SBAS_SATS, COUNT(L.LLI_L1)
FROM OBSERVATION_DATA O FORCE INDEX FOR JOIN (IDX_EPOCH_DATE_OBS_DATA), LLI L
    FORCE INDEX FOR JOIN (IDX_EPOCH_DATE_LLI)
WHERE O.OBS_EPOCH_DATE = L.LLI_EPOCH_DATE
AND L.LLI_L1 > 0
GROUP BY O.OBS_EPOCH_DATE
UNION
SELECT O.OBS_EPOCH_DATE, O.OBS_AVAILABLE_SATS, O.OBS_GLONASS_SATS,
    O.OBS_GPS_SATS, O.OBS_SBAS_SATS, COUNT(L.LLI_L2)
FROM OBSERVATION_DATA O FORCE INDEX FOR JOIN (IDX_EPOCH_DATE_OBS_DATA), LLI L
    FORCE INDEX FOR JOIN (IDX_EPOCH_DATE_LLI)
WHERE O.OBS_EPOCH_DATE = L.LLI_EPOCH_DATE
AND L.LLI_L2 > 0
GROUP BY O.OBS_EPOCH_DATE;
```

Frecuencia *Loss of lock* en L1 por mes.

```
-- Frecuencia L1 por mes
SELECT MONTH(O.OBS_EPOCH_DATE) AS EPOCH_DATE, COUNT(L.LLI_L1) FROM
    OBSERVATION_DATA O, LLI L
WHERE O.OBS_EPOCH_DATE = L.LLI_EPOCH_DATE
AND L.LLI_L1 > 0
GROUP BY MONTH(O.OBS_EPOCH_DATE);
```

Frecuencia *Loss of lock* en L2 por dia.

```
-- Frecuencia L2 por dia
SELECT DATE(O.OBS_EPOCH_DATE) AS EPOCH_DATE, COUNT(L.LLI_L2) FROM
    OBSERVATION_DATA O, LLI L
WHERE O.OBS_EPOCH_DATE = L.LLI_EPOCH_DATE
AND L.LLI_L2 > 0
GROUP BY DATE(O.OBS_EPOCH_DATE);
```

APÉNDICE B

Código Java, RINEX ETL

En este apéndice se colocan alguna de las partes importantes del código fuente de la aplicación RINEX_ETL.

Clase Main.java.

```
if(maxThreads > 1){
        //Creating and configuring the pool connection
        pool = new PoolConnectionDB();
        dataSource = pool.getDataSource();
        logger.info(Constants.MSG_CREATE_POOL_CONNECTION + "\n");
        System.out.println(Constants.MSG_CREATE_POOL_CONNECTION + "\n");

        logger.info(Constants.MSG_MAX_THREADS + maxThreads + "\n");
        System.out.println(Constants.MSG_MAX_THREADS + maxThreads +"\n");
}else{
        singleConnection = new ConnectionDB();
}
.
.
.
if(maxThreads == 1){
        if(COUNTER_FILE < listOfFiles.length){
```

```
            if (listOfFiles[COUNTER_FILE].isFile()) {
                    System.out.println(Constants.MSG_INI_FILE +
                    listOfFiles[COUNTER_FILE].getName());
                    logger.info(Constants.MSG_INI_FILE +
                    listOfFiles[COUNTER_FILE].getName());

                    filename = listOfFiles[COUNTER_FILE].getName().
                    substring(0,listOfFiles[COUNTER_FILE].getName().lastIndexOf("."));

                    extFilename = listOfFiles[COUNTER_FILE].getName().
                    substring(listOfFiles[COUNTER_FILE].getName().lastIndexOf(".")+1,
                    listOfFiles[COUNTER_FILE].getName().length());

                    if (extFilename.equals(Constants.EXT_OBSERVATION_FILE)){
                            parser = new RinexObservationParser
                            (listOfFiles[COUNTER_FILE],filename,singleConnection);

                            result = parser.init();
                    }
                    else{
                            TOTAL_FILES--;
                    }
            }
    }
    COUNTER_FILE++;
    }else{
            if(COUNTER_FILE < listOfFiles.length){
                    List<RinexObservationParser> solvers = new
                    ArrayList<RinexObservationParser>();

                    startSolvers(maxThreads,listOfFiles,
                    dataSource,COUNTER_FILE,solvers);
```

```
                waitUntilSolvedOrFinished(maxThreads, solvers);
            }
        }
```

Clase RinexObservationParser.java.

```
if(conn != null){
//ArrayList for observations data and loss of lock indicator for each epoch
    date
        List <Data> dataObservations = new ArrayList<Data>();
        List <LossLockIndicator> dataLli = new ArrayList<LossLockIndicator>();

        // Open file streams
        this.open();

        // Parse RINEX observation headers
        this.parseHeaderObs(); /* Header */

        // Parse Rinex observations
        while(hasMoreObservations()){
                this.setEpochData(new Data());
                // Parse One Single Epoch Observation
                this.nextObservations();

                if(this.getEpochData() != null){
                        if(this.getEpochData().getEpochDate() != null){
                                if
                                    (auxObs.compareTo(this.getEpochData().getEpochDate())
                                    == 0){
                                        break;
                                }else{
                                //Adding Data object to the ArrayList for joining
```

```
                                and inserting into DB
                                    dataObservations.add(this.getEpochData());
                                    auxObs =
                                        this.getEpochData().getEpochDate();
                                }
                            }
                        }

                        if (this.getLli() != null){
                            if(this.getLli().getEpochDate() != null){
                                if(auxLli.compareTo(this.getLli().getEpochDate())
                                    != 0){
                                    // Adding LossLockIndicator object to the
                                        ArrayList for joining and  inserting
                                        into DB
                                    dataLli.add(this.getLli());
                                    auxLli = this.getLli().getEpochDate();
                                }
                            }
                        }
                    }
                }
```

Vitae

Julio Cesar Roa Gil nació en Ibagué, Colombia el 25 de julio de 1987. Realizó sus estudios básicos en el Colegio Tolimense de Ibagué, Tolima e hizo sus estudios profesionales en la Universidad de Ibagué, donde obtuvo el título de Ingeniero de Sistemas en noviembre de 2010. Hizo parte de AIESEC, cápituo Tolima, donde fue cordinador de Gestión del Conocimiento o *Knowledge Management* (KM). Trabajó como Coordinador del proyecto "Tolima Digital" para el municipio de Mariquita, Tolima y desarrollador de los mapas de interactivos de la conectividad de la ciudad de Ibagué y del departamento del Tolima. Realizó un intercambio profesional en Belo Horizonte, Brasil, durante una año en la empresa Sydle, considerada una de las mejores empresas para trabajar en Tecnologías de la información (TI) de acuerdo a *Great Place To Work Institute*. Allí mismo desarrollo habilidades en bases de datos Oracle, siendo Administrador de Bases de Datos Junior o *Database Administrator* (DBA) y adquirió las competencias del lenguaje Portugués. Trabajó como desarrollador para un proyecto de integración, especificamente de ETL (*Extract, Transform and Load*) para un importante banco de Colombia, en la ciudad de Bogotá. Actuó como lider técnico, DBA, diseñador y desarrollador para un proyecto de modernización de un sistema transaccional de cajeros automáticos para uno de los mayores grupos bancarios de Colombia. En el ITESM Campus Guadalajara, concluyó sus estudios de Maestría en Ciencias de la Computación en diciembre de 2014.