

ESCUELA SUPERIOR POLITECNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“AUTOMATIZACIÓN DE UN PROCESO MASIVO DE DATOS DE
CONSUMO DE INTERNET NO COBRADO, A TRAVÉS DE APACHE
HADOOP EN UNA EMPRESA DE TELECOMUNICACIONES.”

TRABAJO DE TITULACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

Autor:

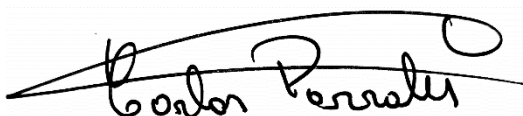
CARLOS JULIO PARRALES CALDERÓN

Guayaquil - Ecuador

2020

AGRADECIMIENTO

Gracias a Dios, a mi esposa y familia que me apoyaron para culminar este gran paso en mi desarrollo profesional, a los maestros del MSIG que con su guía y enseñanzas encaminaron a que cumpla tan preciado objetivo.

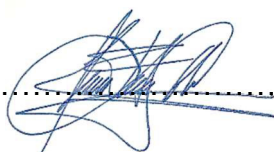


Gordon Perreale

DEDICATORIA

A mi familia y compañeros de estudio del MSIG, Jhonny y Ericka con quienes compartimos risas, esfuerzos y amanecidas para culminar con éxito nuestro objetivo propuesto, que resaltara en nuestra vida personal y profesional.

TRIBUNAL DE SUSTENTACIÓN



Ing. Lenín Eduardo Freire C., MSIG.

COORDINADOR MSIG



Ing. Juan Carlos García P., MSIG.

DIRECTOR DEL PROYECTO DE GRADUACIÓN



Ing. Omar Maldonado D., MSIG.

MIEMBRO DEL TRIBUNAL

RESUMEN

El presente trabajo de titulación tiene como objetivo automatizar un proceso masivo de datos de consumo de internet no cobrado a través de Apache Hadoop, para tal propósito el proceso es llevado a una solución Big Data que comprende un clúster de flujo de datos y un clúster de procesamiento distribuido, el primero que ayudará a obtener y transformar las fuentes del proceso a medida que se vayan generando, y el segundo clúster que permitirá analizar y almacenar la gran cantidad de datos con la que trabaja el proceso.

Durante el trabajo se crearon 4 flujos de datos en la herramienta NIFI de Hortonworks DataFlow, que obtienen cientos de millones de registros diarios y los depositan en el data warehouse Apache Hive, también se creó un proceso HiveQL ejecutado en el clúster Hortonworks Data Platform que diariamente cruza las fuentes previamente cargadas para detectar el consumo de internet que no es cobrado correctamente.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
RESUMEN	V
ÍNDICE GENERAL.....	VI
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS	XII
INTRODUCCIÓN.....	XVIII
CAPITULO 1	1
GENERALIDADES.....	1
1.1 Antecedentes.....	1
1.2 Descripción del problema.....	2
1.3 Solución Propuesta.....	3
1.4 Objetivo General.....	6
1.5 Objetivos Específicos	7
CAPITULO 2.....	8
MARCO TEÓRICO.....	8

2.1 Big Data.....	8
2.2 Procesamiento Distribuido.....	14
2.3 Almacenamiento NoSQL	15
2.4 MapReduce	17
2.5 Flujos de Datos.....	19
CAPITULO 3.....	22
DEFINICIÓN DE LA SITUACION ACTUAL Y DEFINICION DE REQUERIMINETOS.....	22
3.1 Definición de la situación actual	22
3.2 Levantamiento de información de los procesos actuales	25
3.3 Alcance del proyecto	35
CAPITULO 4.....	37
ANÁLISIS, DISEÑO Y DESARROLLO.....	37
4.1 Análisis de requerimientos.....	37
4.2 Diseño de la solución.....	41
4.3 Arquitectura	49
4.4 Modelo de flujos de carga.....	56
4.5 Modelo de datos con Hive	61
4.6 Desarrollo de flujos de carga para archivos.....	73
4.7 Desarrollo de proceso de extracción para bases de datos	85
4.8 Desarrollo de proceso para identificar tráfico no cobrado.....	87

CAPITULO 5.....	88
IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS	88
5.1 Implementación	88
5.2 Resultados de la implementación.....	91
5.3 Plan de pruebas	93
5.4 Pruebas internas	95
5.5 Pruebas de usuario.....	102
5.6 Análisis de los resultados de pruebas	108
CONCLUSIONES Y RECOMENDACIONES.....	111
CONCLUSIONES.....	111
RECOMENDACIONES	112
BIBLIOGRAFIA.....	113
GLOSARIO	115

ABREVIATURAS Y SIMBOLOGÍA

ASN1:	Abstract Syntax Notation One
BASH:	Bourne-again shell
CDR:	Call detail record
EGCDR:	Enhanced G-CDR
EWD:	Enterprise data Warehouse
FTP:	File Transfer Protocol
HDF:	Hortonworks Data Flow
HDP:	Hortonworks Plataforma de Datos
PGWCDR:	Packet Data Network Gateway Charging Data Record

ÍNDICE DE FIGURAS

Figura 1.1. Arquitectura de la solución propuesta.....	4
Figura 2.1. Capas de la arquitectura hadoop.....	10
Figura 2.2. MapReduce overview	19
Figura 4.1. Modelo cascada para desarrollo de proceso masivo de datos ...	38
Figura 4.2. Nodos del clúster Big Data	50
Figura 4.3. Diagrama general del proceso de detección de tráfico no cobrado	55
Figura 4.4. Flujo CDR cobro prepago	56
Figura 4.5. Flujo de extracción del CDR cobro prepago	57
Figura 4.6. Flujo de carga del CDR cobro prepago.....	57
Figura 4.7. Flujo de extracción del CDR consumo PGWCDR	58
Figura 4.8. Flujo de carga del CDR consumo PGWCDR.....	59
Figura 4.9. Flujo de extracción del CDR consumo EGCDR.....	60
Figura 4.10. Flujo de carga del CDR consumo EGCDR	60

Figura 4.11. Flujo NIFI de la carga de CDRs de cobro prepago	74
Figura 4.12. Flujo NIFI extracción de CDRs de cobro post pago	76
Figura 4.13. Flujo NIFI carga de CDRs de cobro post pago	77
Figura 4.14. Flujo NIFI extracción de CDRs de consumo, tipo pgwcdm	79
Figura 4.15. Flujo NIFI carga de CDRs de consumo, tipo pgwcdm	80
Figura 4.16. Flujo NIFI extracción de CDRs de consumo, tipo egcdm	82
Figura 4.17. Flujo NIFI carga de CDRs de consumo, tipo egcdm	84
Figura 5.1. Tablas de CDRs en Hive	91
Figura 5.2. Tabla de tráfico no cobrado	91
Figura 5.3. Ruta de programa java decoder ASN1	92
Figura 5.4. Ruta script HiveQL de proceso de tráfico no cobrado	92
Figura 5.5. Plantillas de flujos de CDR cargados en NIFI Data Flow	92
Figura 5.6. Bitácora de carga de CDRs de cobro	93

ÍNDICE DE TABLAS

Tabla 1. Roles de nodos físicos del Clúster	12
Tabla 2. Servicios HDP	13
Tabla 3. Bases de datos NoSql agrupadas por categoría.....	15
Tabla 4. Características principales Servidor 1	27
Tabla 5. Características principales Servidor 2.....	27
Tabla 6. Layout CDR de cobro prepago	28
Tabla 7. Layout de CDR de consumo	29
Tabla 8. Actores del proceso actual.....	34
Tabla 9. Roles del proceso actual.....	35
Tabla 10. Campos de tabla resultado del proceso actual	41
Tabla 11. Datos del del proceso de ingesta de CDR de cobro prepago	43
Tabla 12. Datos del del proceso de ingesta de CDR de cobro prepago	44
Tabla 13. Datos del del proceso de ingesta de CDR de consumo pgwcdm ...	45
Tabla 14. Datos del del proceso de ingesta de CDR de consumo egcdr.....	46
Tabla 15. Datos de fuente y destino de la tabla de clientes post pago	47

Tabla 16. Datos de fuente y destino de la tabla de clientes prepago.....	48
Tabla 17. Versión Clúster HDF	50
Tabla 18. Características de los nodos del clúster HDF	51
Tabla 19. Componentes instalados en el clúster HDF	51
Tabla 20. Versión del clúster HDP	52
Tabla 21. Características de los name nodos del clúster HDP	52
Tabla 22. Características de los data nodos del clúster HDP	53
Tabla 23. Componentes instalados en el clúster HDP	54
Tabla 24. Propiedades tabla de CDR de cobro post pago.....	61
Tabla 25. Campos de la tabla de CDR de cobro post pago.....	61
Tabla 26. Propiedades tabla de CDR de cobro prepago	63
Tabla 27. Campos de la tabla de CDR de cobro prepago	63
Tabla 28. Propiedades tabla de CDR de consumo del tipo pgwcdr	65
Tabla 29. Campos de la tabla de CDR de consumo del tipo pgwcdr	65
Tabla 30. Propiedades tabla de CDR gdr_pgwcdr_listofservicedata	67
Tabla 31. Campos de la tabla de CDR gdr_pgwcdr_listofservicedata	68

Tabla 32. Propiedades tabla de CDR de consumo del tipo egcdr.....	69
Tabla 33. Campos de la tabla de CDR de consumo del tipo egcdr.....	69
Tabla 34. Propiedades de la tabla de CDR gdr_egcdr_listofservicedata	71
Tabla 35. Campos de la tabla de CDR gdr_egcdr_listofservicedata.....	71
Tabla 36. Propiedades de la tabla resultado del proceso de tráfico no cobrado	72
Tabla 37. Campos de la tabla resultado del proceso de tráfico no cobrado .	73
Tabla 38. Detalle del flujo NIFI de la carga de CDRs de cobro prepago.....	74
Tabla 39. Detalle de script cargaDetallesOneGprs	75
Tabla 40. Detalle de flujo NIFI extracción de CDRs de cobro post pago	76
Tabla 41. Detalle de script descomprime_archivo	76
Tabla 42. Detalle del flujo NIFI carga de CDRs de cobro post pago.....	77
Tabla 43. Detalle de script cargaDetallesGprs.....	78
Tabla 44. Detalle de flujo NIFI extracción de CDRs de consumo, tipo pgwcd	79
Tabla 45. Detalle de script ejecutaDecoder	80

Tabla 46. Detalle de programa decoder-charging.....	80
Tabla 47. Detalle flujo NIFI carga de CDRs de consumo, tipo pgwcdr	81
Tabla 48. Detalle de script cargaDetallesPgwcdr.....	81
Tabla 49. Detalle flujo NIFI extracción de CDRs de consumo, tipo egcdr.....	82
Tabla 50. Detalle de script ejecutaDecoder	83
Tabla 51. Detalle de programa decoder-charging.....	84
Tabla 52. Detalle flujo NIFI carga de CDRs de consumo, tipo egcdr	84
Tabla 53. Detalle de script cargaDetallesEgcdr	85
Tabla 54. Recursos para creación de estructuras de datos.....	89
Tabla 55. Tareas para puesta en producción de estructuras de datos	89
Tabla 56. Recursos para instalación de scripts.....	89
Tabla 57. Tareas para puesta en producción de scripts HiveQL y Bash	90
Tabla 58. Recursos para instalación de flujos Data Flow	90
Tabla 59. Tareas para puesta en producción de flujos Data Flow	90
Tabla 60. Recursos encargados de llevar a cabo las pruebas	94
Tabla 61. Caso de prueba PI001 – carga de CDRs de cobro post pago	95

Tabla 62. Caso de prueba PI002 – carga de CDRs de cobro prepago	96
Tabla 63. Caso de prueba PI003 – carga de CDRs de consumo pgwcdr	97
Tabla 64. Caso de prueba PI004 – carga de CDRs de consumo egcdr	98
Tabla 65. Caso de prueba PI005 – decodificador asn1	99
Tabla 66. Caso de prueba PI006 – carga de clientes prepagos	100
Tabla 67. Caso de prueba PI007 – carga de clientes post pago	101
Tabla 68. Caso de prueba PU001 – generación de reporte.....	102
Tabla 69. Caso de prueba PR001 – rendimiento de carga CDR cobro prepago	103
Tabla 70. Caso de prueba PR002 – rendimiento de carga CDR cobro post pago	104
Tabla 71. Caso de prueba PR003 – rendimiento de carga CDR consumo pgwcdr	105
Tabla 72. Caso de prueba PR004 – rendimiento de carga CDR consumo egcdr	106
Tabla 73. Caso de prueba PR005 – rendimiento del decodificar asn1	107
Tabla 74. Caso de prueba PR006 – rendimiento del proceso de tráfico	107

Tabla 75. Métricas de las cargas de archivos CDRs	108
Tabla 76. Métricas de las cargas tablas de clientes.....	108
Tabla 77. Métricas de ejecución del proceso de tráfico no cobrado	109
Tabla 78. Comparativa del nuevo proceso en Hadoop y el actual proceso	110

INTRODUCCIÓN

En el mundo de las telecomunicaciones se recolectan una gran cantidad y variedad de datos, que deben ser medidos y monitoreados para evitar que existan ingresos no percibidos, que pueden ocurrir por fallas en las plataformas o por errores humanos, esto implica complejidad al momento de almacenarlos, transformarlos y analizarlos.

La compañía de telecomunicaciones tiene una alta tasa de transacciones diarias que deben ser analizadas en un tiempo oportuno, ya que un pequeño error puede derivar en grandes pérdidas, para ello se propone un proceso masivo en una plataforma Big Data que permita escalabilidad en almacenamiento y procesamiento en medida que el negocio lo requiera.

El proceso propuesto consiste en evaluar las transacciones de consumo versus transacciones de facturación, para ello se empleará un clúster de flujo de datos que realice la ingesta de estas transacciones, estas fuentes están contenidas en archivos CDRs que en unos casos deberán ser transformados o decodificados para luego ser cargadas en un data warehouse, también se empleará otro clúster de procesamiento distribuido que permitirá almacenar los datos fuentes y posteriormente procesar la data de consumo y facturación contenida en los repositorios del clúster.

CAPITULO 1

GENERALIDADES

1.1 Antecedentes

La empresa de telecomunicaciones de capital privado brinda servicios a la comunidad desde los años 90, tiene una importante cuota de mercado, segmenta a sus clientes en prepagos y post pagos, siendo los clientes prepagos su mayoría.

La compañía tiene gran cantidad de transacciones al día que deben ser validadas para asegurar ingresos, dado que un pequeño fallo puede representar una pérdida significativa para el negocio.

1.2 Descripción del problema

En la actualidad la empresa de telecomunicaciones registra 300 millones de transacciones diarias de tráfico de datos, esto determina el consumo de datos que deberá ser facturado, sin embargo, el departamento de aseguramiento de ingreso detecta un mes después que no se ha cobrado el 0.04% de estas transacciones, y en ocasiones por algún incidente interno este porcentaje es más representativo.

A pesar de que solo se obtiene tráfico no cobrado del segmento de clientes prepago, dejando fuera a los clientes del segmento pospago, el proceso tarda en ejecutarse hasta 6 días para obtener el reporte del mes. Durante este tiempo se emplean la totalidad de los recursos disponibles para el procesamiento.

Los archivos de tráfico de datos tienen que ser decodificados ya que son binarios con formato ASN1, para esto no se cuenta con decodificadores genéricos, ocasionando que cada vez que se cambia el esquema de algún archivo, se tenga que actualizar el programa decodificador, causando que el mantenimiento del proceso sea complejo. Además, la capacidad de almacenamiento es limitada, permitiendo guardar solo 1 mes de datos.

La facturación es un proceso importante para la compañía, por lo que es necesario obtener el reporte diario de tráfico no cobrado, con el fin de tomar acciones correctivas que eviten pérdida de ingresos.

1.3 Solución Propuesta

El proceso será desarrollado con las herramientas de Hortonworks Hadoop Distribution, dado que ofrece escalabilidad de procesamiento, almacenamiento y análisis de volúmenes grandes de datos. Para desarrollar la solución se considera las siguientes fases: levantamiento y análisis del proceso, diseño de la solución empleando herramientas hadoop, pruebas e implementación de proceso.

La solución propuesta comprenderá un clúster HDP, con 4 Nodos de Datos para servicios de almacenamiento y procesamiento y 4 Nodos Master para gestión de procesamiento y almacenamiento, además de un clúster HDF con 3 Nodos de Flujos de Datos para ingesta de datos.

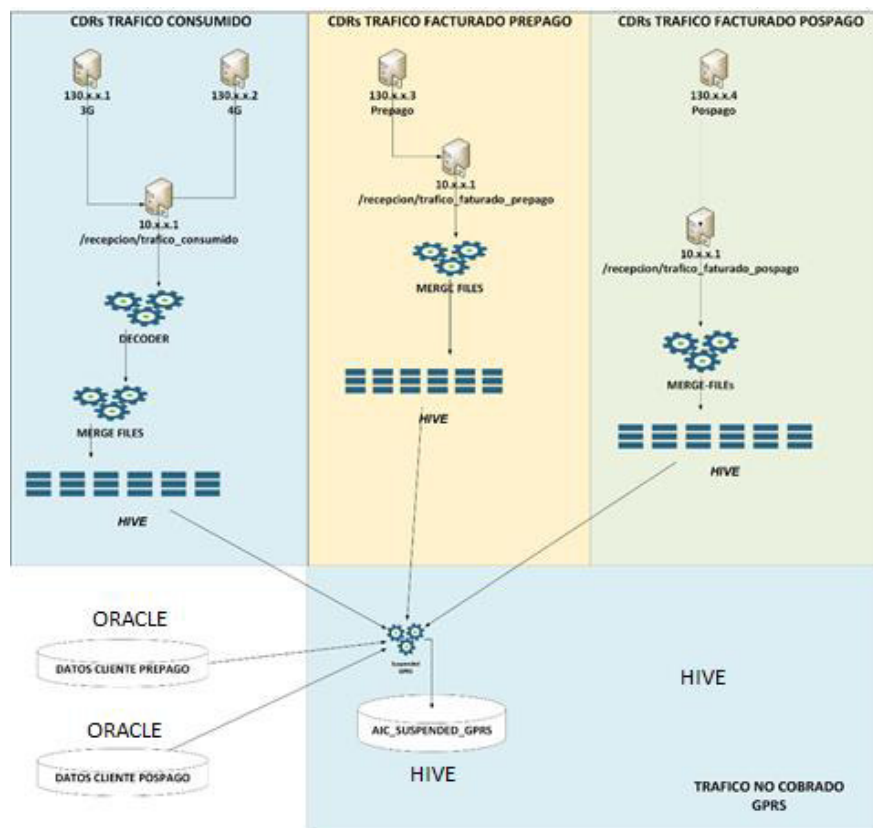


Figura 1.1. Arquitectura de la solución propuesta

Desde el clúster HDF se conectará hacia los servidores de las plataformas de facturación, para extraer y procesar de forma distribuida los archivos CDRs, las plataformas fuentes son: de tráfico consumido, tráfico facturado prepago y de tráfico facturado pospago, se diseñarán 4 flujos para obtener y procesar los archivos.

- Flujo para archivos CDRs de tráfico facturado de clientes prepagos.

- Flujo para archivos CDRs de tráfico facturado de clientes pospagos.
- Flujo para archivos CDRs de tráfico consumido 3G de clientes prepago y post pago.
- Flujo para archivos CDRs de tráfico consumido 4G de clientes prepago y post pago.

Los archivos de tráfico consumido serán decodificados, luego se les aplica un merge para formar un archivo de mayor peso y posteriormente son almacenados en dos estructuras de Hive, una estructura para tráfico 4G y otra estructura para tráfico 3G.

Los archivos de tráfico facturado prepago se les aplicará un merge para formar un archivo de mayor peso y luego se almacenarán en una estructura Hive.

Los archivos de tráfico facturado pospago se les aplicará un merge para formar un archivo de mayor peso y luego se almacenarán en una estructura Hive.

El proceso también requiere información referente al cliente que se encuentra en los sistemas legados de la compañía, para extraer esta información se empleará la aplicación sqoop para cargar estos datos diariamente a estructuras en Hive, se considera cargar 2 estructuras, datos de clientes prepago y datos de clientes pospago.

Las fuentes de tráfico consumido y tráfico facturado serán cruzadas en Hive para obtener el tráfico no cobrado, a estas líneas resultantes se les añadirán los datos de los clientes, y se almacenará el resultado diario en una estructura en Hive.

La solución completará el análisis de todo el segmento de clientes prepagos y pospagos de la compañía, añadiendo las fuentes de facturación que no están siendo consideradas en la actualidad, además los datos de tráfico se cargarán y decodificarán en línea, reduciendo el tiempo de entrega de la información, minimizando el riesgo por ingresos no percibidos.

La herramienta propuesta permite fácil crecimiento en procesamiento y almacenamiento, en caso de aumento de transacciones para el proceso, también posee tolerancia a fallos, por sí ocurriera algún desperfecto en uno de los nodos.

1.4 Objetivo General

Automatizar el proceso para obtener un reporte diario de tráfico no cobrado haciendo uso de la herramienta Hortonworks Hadoop Distribution para una empresa de Telecomunicaciones.

1.5 Objetivos Específicos

1. Levantar y analizar el proceso para obtención de tráfico no cobrado.
2. Diseñar una arquitectura de procesamiento de datos masivos bajo el ecosistema de Hadoop y Hive.
3. Desarrollar el proceso para obtener tráfico no cobrado con la herramienta Hortonworks Hadoop Distribution.
4. Evaluar resultados obtenidos de la implementación.

CAPITULO 2

MARCO TEÓRICO

2.1 Big Data

Big Data puede definirse como el crecimiento exponencial de grandes volúmenes de datos, la necesidad de capturarlos, almacenarlos y analizarlos para conseguir mayores beneficios y oportunidades[1].

No existe definición precisa para big data, en un inicio el volumen de información aumento tanto que ya no era posible procesarla y almacenarla en ordenadores, lo que dio origen a creación de nuevas tecnologías de procesamiento como Map-Reduce, de Google o Hadoop originaria de Yahoo, con estas se pueden manejar grandes cantidades de datos estructurados y no estructurados[2].

El concepto de Big Data aplica para toda información que no puede ser procesada o analizada utilizando procesos o herramientas tradicionales, no se refiere a alguna cantidad en específico, ya que usualmente se emplea cuando se trabaja con petabytes y exabytes de datos.

El gran volumen y variedad de datos esta presente por ejemplo en dispositivos móviles, audio, video, sistemas GPS, sensores digitales que pueden medir y comunicar posicionamiento, movimiento, vibración, temperatura, humedad, y cambios químicos, donde se necesita una velocidad de respuesta rápida para obtener información correcta en el momento oportuno[3].

Las “3V” representan características claves de los sistemas Big Data. La primera característica es volumen, representa la gran cantidad de datos que se va a manipular o analizar. La segunda característica es la velocidad, se refiere al procesamiento de datos en tiempo real o cerca del tiempo real. La tercera característica es la variedad, que se refiere a que el tipo de datos que se almacenan y procesan de son diversos pueden consistir en coordenadas de ubicación, video, datos de navegadores, simulaciones entre otros. Ciertos autores añaden 2V adicionales como características de los sistemas Big Data. La cuarta característica es el valor, que representa el valor comercial que pueden tener los datos almacenados. La quinta característica es la veracidad, que se refiere a la consistencia de los datos.

Una de las tecnologías que intenta superar los desafíos que plantean las características Big Data es Apache Hadoop, que es un software de código abierto cuyo objetivo principal es manejar grandes cantidades de datos en un tiempo razonable, dividiendo los datos en una infraestructura de múltiples nodos para poder procesarlos, además, de almacenar el contenido en nodos disperso para que se pueda encontrar y acceder fácilmente desde el punto más próximo[4].

Capas de arquitectura Hadoop

La arquitectura de Hadoop puede separarse en capas funcionales que ayudan a optimizar el desarrollo y la gestión de los datos, esta se puede dividir en cuatro capas distintivas[5].

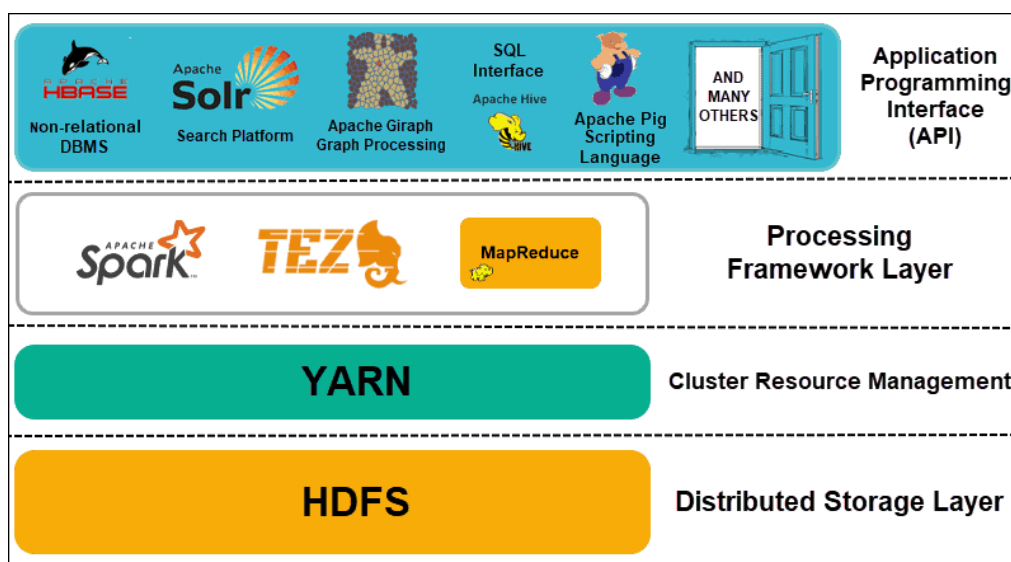


Figura 2.1. Capas de la arquitectura hadoop

Capa de almacenamiento distribuido

La capa de almacenamiento distribuido donde cada nodo de un clúster de Hadoop tiene su propio espacio en disco, memoria, ancho de banda y procesamiento, los datos entrantes se dividen en bloques que luego se almacenan dentro del file system de almacenamiento distribuido, HDFS almacena tres copias de cada conjunto de datos en todo el clúster, los nodos maestros NameNode mantiene los metadatos de los bloques y sus réplicas.

Gestión de recursos del clúster

En esta capa se coordinan las aplicaciones y recursos de manera efectiva, YARN realiza la gestión de recursos de facto para Hadoop, puede asignar recursos a diferentes marcos de trabajo como Apache Pig, Hive, Giraph, Zookeeper, así como el propio MapReduce.

Capa de marco de procesamiento

En la capa de procesamiento se analizan y procesan los conjuntos de datos que ingresan al clúster, estos datos pueden ser estructurados y no estructurados, aquí se mapean, mezclan, ordenan, combinan y reducen en bloques de datos manejables más pequeños, estas operaciones se distribuyen en varios nodos lo más cerca posible de los servidores donde se encuentran los datos.

Interfaz de programación de aplicaciones

Esta capa consta de una variedad de proyectos que se centran en plataformas de búsqueda, transmisión, interfaces fáciles de usar, lenguajes de programación, mensajería, conmutación por error y seguridad, son una parte intrincada de un ecosistema Hadoop completo.

Ecosistema Hadoop Hortonworks

Hortonworks Data Platform (HDP) consta de varios componentes que abordan una amplia gama de funciones, la mayoría de estos componentes se implementan como servicios maestros y de trabajo que se ejecutan en el clúster de forma distribuida[6].

Tabla 1. Roles de nodos físicos del Clúster

Roles nodos físicos	Definición de nodos
Administration Node	Proporciona capacidades para administración de clústeres, este nodo es opcional.
Active NameNode	Ejecuta todos los servicios que administran el almacenamiento de datos HDFS y la administración de recursos YARN. Hay cuatro servicios principales: <ul style="list-style-type: none"> • YARN Resource Manager: Administración de recursos del clúster, incluidos los trabajos de MapReduce. • NameNode: Almacenamiento de datos HDFS. • Journal Manager: Alta disponibilidad. • ZooKeeper: Apoya la coordinación.
Standby NameNode	Se utiliza en modo HA, ejecuta un standby namenode, un segundo Journal Manager y un opcional standby resource manager. En este nodo también se ejecuta Spark History Server y un segundo servicio ZooKeeper.

Roles nodos físicos	Definición de nodos
HA Node	Provee un tercer journal node para HA, el Active NameNodes y Standby NameNodes provee el primer y segundo journal nodes. Aquí también se ejecuta un tercer servicio ZooKeeper.
Edge Node	Es un nodo perimetral que proporciona una interfaz entre los datos y la capacidad de procesamiento que está disponible en el clúster de Hadoop.
Worker Node 1 - N	<p>Ejecuta todos los servicios necesarios para almacenar bloques de datos en los discos duros locales y ejecutar tareas de procesamiento con esos datos.</p> <p>Los servicios primarios que se ejecutan en los Worker Nodes son:</p> <ul style="list-style-type: none"> • Daemon DataNode (para admitir el almacenamiento de datos HDFS). • Demonio NodeManager (para admitir la ejecución del trabajo YARN). • Otros servicios como Hbase y Spark también se ejecutan en Worker Nodes.

La siguiente table lista los servicios principales de Hortonworks Data Platform.

Tabla 2. Servicios HDP

Servicio	Función	Servicios en nodos Master	Servicios en nodos Worker
HDFS	Hadoop distributed filesystem	Active NameNode, Standby NameNode	YARN NodeManager
YARN	Cluster resource management	YARN Resource Manager	HBase Region Server
HBASE	Column-oriented NoSQL Database	HBase Master	
SAPARK	In-memory data processing engine	Spark Master, Spark History Server	Spark Worker
RANGER	Security administration	Ranger Gateway	N/A
AMBARI	Hadoop cluster management	Ambari Server	Ambari Agent

2.2 Procesamiento Distribuido

La computación distribuida es un modelo que resuelve problemas de cómputo masivo utilizando un gran número de computadoras independientes organizadas sobre una infraestructura distribuida actuando como un solo equipo, su objetivo es que estos equipos independientes cooperen para resolver un problema que no puede ser resuelto individualmente[7].

Los sistemas distribuidos tienen la capacidad de procesamiento a gran escala y Hadoop es una herramienta con la que es posible lograrlo. Hadoop tiene 2 componentes principales: MapReduce como motor de ejecución y HDFS como sistema de archivos distribuido. HDFS provee tolerancia a fallos en el almacenamiento a través de la replicación, al dividir los archivos en bloques de datos de igual tamaño y replicarlos en varios nodos HDFS, de modo que, si alguno de los nodos falla, los datos aún pueden recuperarse de otros nodos replicados. MapReduce proporciona tolerancia a fallas a nivel de trabajo, al reasignarlas a otros nodos y también maneja las fallas de los nodos al reprogramar todas las tareas a otros nodos para su nueva ejecución[8].

2.3 Almacenamiento NoSQL

El termino NoSql se utilizó en 1998 para una base de datos relacional que no tenía interfaz sql, en la década del 2000 se volvieron importantes debido a la expansión de internet que requirió necesidad de gestionar enormes cantidades de datos generados por los servicios web.

Las bases de datos NoSql no tienen modelo relacional, se centran en la escalabilidad distribuida y horizontal, son de fácil replicación, se acceden a través de un API, y no tienen modelo de consistencia ACID. Existen diferentes tipos de bases de datos NoSql dependiendo de su aplicación, las que almacenan key-value, las bases de datos columnares, bases de datos documentales y bases de datos graficas[9].

Tabla 3. Bases de datos NoSql agrupadas por categoría

Category	NoSQL databases
Key-Value	Hazelcast Redis Membase/Couchbase Riak Voldemort Infinispan
Wide-Column	Hbase Hypertable Cassandra
Document-Oriented	CouchDB MongoDB Terrastore RavenDB
Graph-Oriented	Neo4J InfiniteGraph InfoGrid HypergraphDB AllegroGraph BigData

Key-Value

Estas bases de datos son tablas hash distribuidas que proporcionan al menos operaciones de get y put. Una base de datos clave-valor asigna elementos de datos a un espacio clave que se utiliza para asignar y localizar pares clave / valor de manera eficiente. Estas bases de datos están diseñadas para escalar a terabytes o incluso petabytes, así como a millones de operaciones simultáneas mediante la adición horizontal de computadoras.

Wide-Column

Las bases de datos Wide Column almacenan datos por columnas, y no imponen un esquema rígido a los datos del usuario. Los datos de estas bases se almacenan por columnas por lo que se pueden usar algoritmos de compresión para disminuir espacio, así también es posible realizar particiones para colocar en la misma ubicación las columnas a las que se accede con mayor frecuencia. La mayoría de las bases de datos en esta categoría están inspiradas en BigTable, una base de datos creada por Google para almacenar datos en el orden de los petabytes. De entre bases destaca Hbase que se basa en HDFS para el almacenamiento y la replicación, permite consultar la base de datos utilizando Hadoop MapReduce a través de lenguajes como Pig y Hive.

Document-Oriented

Las bases de datos orientadas a documentos pueden verse como bases de datos de Key-Value donde el valor a almacenar tiene una estructura conocida determinada en el momento del diseño de la base de datos. Algunos formatos de documentos populares son XML, JSON y BSON.

Graph-Oriented

Estas bases de datos abordan el problema de almacenar datos y sus relaciones como un gráfico, lo que permite consultar y navegar de manera eficiente. Los gráficos se utilizan para representar muchas grandes entidades del mundo real, como mapas y redes sociales.

Las bases de datos NoSQL ahora son parte del diseño de software y ocupan un nicho de mercado importante[10].

2.4 MapReduce

MapReduce es el corazón de Hadoop. Es un paradigma de programación que permite una escalabilidad masiva en cientos o miles de servidores en un clúster de Hadoop. MapReduce se refiere a dos términos separados y distintos que realiza el programa Hadoop. El primero es el trabajo de mapa, que toma datos de entrada y los procesa para producir pares clave / valor, que se barajan y clasifican y se envían como entrada para la tarea de reducción. El trabajo de reducción toma

esos pares clave / valor y luego los combina o agrega para producir los resultados finales. MapReduce implica que el trabajo de reducción siempre se realiza después del trabajo de mapa[11].

La entrada del conjunto de datos a MapReduce primero se transforma en pares de clave y valor, ya que los mapeadores y reductores solo pueden funcionar en este formato.

Mapeador: $(k_1, v_1) / [(k_2, v_2)]$

Reductor: $(k_2, \{v_2j\}) / [(k_3, v_3)]$

donde, k_1 y k_2 son clave de entrada y clave de salida respectivamente, y v_1 y v_2 son valores de entrada y salida respectivamente. k_3 y v_3 son claves y valor finales respectivamente. $\{v_2j\}$ es la lista de valores de salida.

Los mapeadores se ejecutan en paralelo en diferentes divisiones de datos de un conjunto de datos de entrada y generan pares intermedios de claves y valores. Los reductores obtienen estos valores de los mapeadores y calculan el valor final para cada clave. La Figura 1 proporciona el estilo de trabajo de MapReduce con mapeadores y reductores[12].

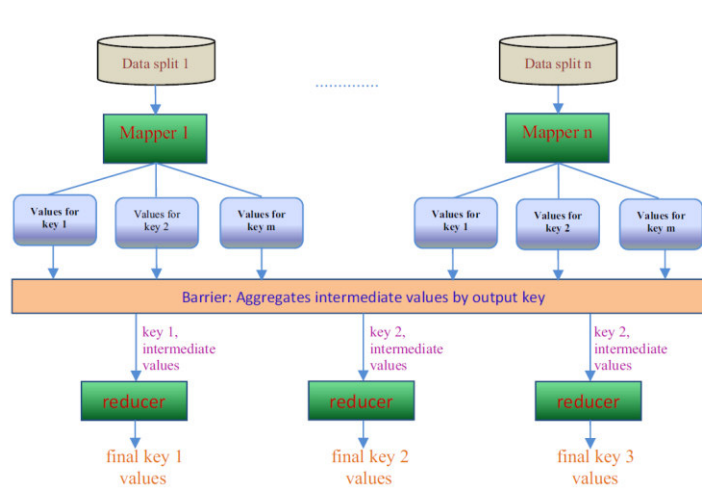


Figura 2.2. MapReduce overview

El modelo MapReduce ofrece las siguientes ventajas:

- Proporciona paralelización y distribución automáticas.
- Es tolerante a fallas. Las tareas individuales pueden ser reintentadas.
- Una abstracción limpia para desarrolladores proporcionada.
- Los programas MapReduce generalmente están escritos en Java, que a su vez es uno de los populares y más utilizados por los desarrolladores.
- Hadoop viene con herramientas de monitoreo y estado estándar.

2.5 Flujos de Datos

Tradicionalmente, las empresas han almacenado grandes volúmenes de datos sin filtrar en centros de datos, esto limita a las organizaciones

que no pueden capitalizar en ideas oportunas. Recientemente, las principales empresas tecnológicas se han centrado en crear soluciones analíticas que puedan obtener valor de los flujos de datos, también conocidos como procesamiento en tiempo real. El procesamiento en tiempo real se produce con una latencia cercana a cero, lo que brinda a las empresas acceso instantáneo a situaciones urgentes que solo se pueden detectar y actuar en el momento[13].

Facebook posee un ecosistema de procesamiento de datos en tiempo real que maneja cientos de Gigabytes por segundo en cientos de canales de datos, es empleado para potenciar muchos de sus casos de uso, incluyendo informes en tiempo real de la voz agregada y anónima de los usuarios de Facebook, análisis para aplicaciones móviles e información para administradores de páginas de Facebook[14].

Para este tipo de trabajo se puede utilizar herramientas como Hortonworks Data Flow (HDF) que analiza el procesamiento de datos en tiempo real. HDF es un conjunto de herramientas que le da al usuario un control de los datos, desde su generación en los dispositivos de borde, o recopilar en tiempo real múltiples tipos de datos de una multitud de fuentes. Hay tres herramientas principales dentro de HDF; Apache NiFi, Apache Kafka y Apache Storm.

Apache NiFi es una tecnología de flujos de datos que utiliza procesamiento basado en flujos, proporciona una GUI y contiene más

de 200 procesadores. Cada procesador realiza una acción sobre los datos, se puede arrastrar y soltar procesadores en el lienzo, ajustar las configuraciones en cada procesador antes de conectarlo al siguiente procesador, creando un flujo de datos en tiempo real. Además, estos procesadores pueden realizar una multitud de acciones, como convertir formatos de datos, agregar atributos a los datos y enrutar los datos en función de los atributos. También hay una colección de procesadores disponibles para ingesta de una multitud de fuentes, desde sitios web, sistemas de archivos locales, bases de datos y fuentes externas, como dispositivos de borde. NiFi también agrega seguridad al transporte de datos con soporte incorporado para SSL, SSH, HTTPS, contenido encriptado y autenticación y autorización basada en roles.

Apache Kafka se utiliza como servicio de mensajería, ya que proporciona un alto rendimiento, entrega confiable y escalabilidad horizontal. Kafka es una plataforma de mensajería de baja latencia para fuentes de datos de transmisión en tiempo real.

Apache Storm es un sistema de cómputo distribuido que realiza el procesamiento en tiempo real de grandes cantidades de datos.

CAPITULO 3

DEFINICIÓN DE LA SITUACION ACTUAL Y DEFINICION DE REQUERIMINETOS

3.1 Definición de la situación actual

La compañía de telecomunicaciones opera en Ecuador desde el año 1993 y desde el 2000 es parte de una multinacional de telecomunicaciones, brinda servicios de telecomunicaciones móviles, cuenta con 8.3 millones de suscriptores, tiene 5.500 puntos de ventas y más de 80 centros de atención al cliente con 3000 empleos directos, tiene la misión de lograr que la población de cada uno de los países en donde operan tenga acceso a productos y servicios de calidad con la más avanzada tecnología en telecomunicaciones.

Entre los servicios que ofrece la compañía se tienen los servicios móviles y los servicios fijos de hogar.

Servicios móviles, entre los que se ofrecen Internet, Minutos, Mensajes y Roaming.

Internet, que es el acceso a la red de Internet desde un dispositivo móvil autónomo (smartphone, tablet, etc.) que permita satisfacer las funciones básicas de la red - comercio, acceso a contenidos y comunicaciones- en cualquier momento y lugar.

Minutos: que es un medio de comunicación que consiste en la transmisión de voz entre teléfonos móviles.

Mensajes: que es un servicio de mensajes cortos.

Roaming: Es un servicio que permite tener cobertura cuando estamos en el extranjero, empleando la línea del país de origen.

Servicios Fijos, entre los que se ofrecen TV fijo, Internet de hogar y llamadas fijas.

TV Fijo: Es un servicio de difusión de TV.

Internet de hogar: que es el acceso a la red de Internet desde un dispositivo instalado en el hogar.

Llamadas fijas: que ofrece transmisión de voz desde un equipo instalado en el hogar.

La compañía segmenta su mercado en clientes prepagos que pagan sus servicios antes de usarlos y clientes post pago que pagan sus servicios después de usarlos con una cuota mensual.

La estructura organizacional de la compañía se compone por el directorio con un presidente ejecutivo y por las siguientes direcciones:

- Dirección de Sistemas
- Dirección Comercial
- Dirección de Servicio al Cliente
- Dirección de Operaciones
- Dirección Técnica
- Dirección Financiera Administrativa
- Dirección de Recursos Humanos
- Dirección de Aseguramiento de Ingresos y Control

La compañía cuenta con las siguientes plataformas tecnológicas de apoyo para consumo y facturación de servicios móviles.

Plataforma de cobro prepago

Es un sistema que obtiene y almacena CDRs de los cobros de eventos de voz, navegación y mensajería de los clientes prepagos. Estos CDRs son almacenados en archivos en formato texto.

Plataforma de cobro post pago

Es un sistema que obtiene y almacena CDRs de los cobros de eventos de voz, navegación y mensajería de los clientes post pago. Estos CDRs son almacenados en archivos en formato texto.

Plataforma de consumo

Es un sistema que obtiene y almacena CDRs de los detalles de eventos de navegación de los clientes prepago y post pago. Estos CDRs son almacenados en archivos binarios en formato asn1.

Sistema comercial para clientes prepagos

Es un sistema que mantiene registro de las distintas transacciones comerciales realizadas por los clientes prepagos.

Sistema comercial para clientes post pago

Es un sistema que mantiene registro de las distintas transacciones comerciales realizadas por los clientes post pago.

3.2 Levantamiento de información de los procesos actuales

La descripción del proceso se basa en la entrevista realizada al ingeniero de desarrollo que es encargado de obtener el reporte de tráfico no cobrado.

El proceso actual para detección de tráfico no cobrado evalúa dos fuentes de CDRs, la primera contiene el cobro del tráfico y la segunda el detalle del evento de navegación (consumo), las dos fuentes son generadas por plataformas distintas, el CDR de consumo se genera mientras ocurre un evento de navegación de internet y en él se detallan los KB de bajadas y subida que deberán cobrarse, mientras el CDR de cobro se genera cuando termina el evento de navegación de internet, detallando los KB y el monto de cobro de dicho evento.

Una vez evaluadas las fuentes y detectadas las inconsistencias se genera un reporte con el detalle de los casos, adicionando información del cliente y estado de la línea reportada.

El proceso en la actualidad se ejecuta una vez al mes debido a limitaciones de recursos con los que se cuenta, también se evalúan datos de un solo mes, esto porque no se cuenta con suficiente espacio en la base de datos para almacenar más tiempo los datos de los CDRs, además solo se está evaluando el segmento de clientes prepago, el segmento post pago no se lo está considerando.

Recursos tecnológicos empleados

- Servidor 1 (GYE)

Tabla 4. Características principales Servidor 1

RAM	130 GB
CPU	60 cores
ALMACENAMIENTO	700 GB

- Servidor 2 (UIO)

Tabla 5. Características principales Servidor 2

RAM	130 GB
CPU	60 cores
ALMACENAMIENTO	800 GB

- Base de Datos Oracle Enterprise Edición

Layout de archivos CDR

CDR de cobro prepago

El CDR de cobro prepago se encuentra en formato texto y consta de 61 campos, siendo los campos 60 y 61 opcionales, los archivos podrán tener registros con 59, 60 o 61 columnas, el detalle de los campos se muestra a continuación.

Tabla 6. Layout CDR de cobro prepago

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
sequencenumber	smallint	grouptype	tinyint
subscriberid	bigint	groupid	tinyint
cdrdate	bigint	rulediscountamount	tinyint
cdrsseconds	int	rulediscountunits	tinyint
sourcedate	bigint	rulediscountname	varchar(24)
sourcetime	int	discountamount0	int
servicetype	smallint	discountschemeid0	tinyint
transactiontype	smallint	discountamount1	tinyint
correlationid	varchar(80)	discountsechemeid1	tinyint
transactionmajornumber	int	discountamount2	tinyint
transactionminornumber	tinyint	discountschemeid2	tinyint
sourcetype	tinyint	accounttype	tinyint
sourceinfo1	bigint	accountstatus	tinyint
sourceinfo2	smallint	balancedeftype	tinyint
ratename	varchar(25)	balancedefid	int
tariffplanid	smallint	balanceamount	bigint
systemtax	tinyint	balanceunits	tinyint
locationtax	tinyint	balancedelta	bigint
servicetax	tinyint	reservationcount	smallint
billingeventid	smallint	networktype	int
timebandgroupid	smallint	gprs_imsi	bigint
categoryid	smallint	apn	varchar(118)
operatorid	bigint	serviceid	smallint
operatortransactiontype	smallint	userequipment	bigint
acountnumber	varchar(20)	charging_id	bigint
suscriberclass	tinyint	rating_group_id	bigint
profileid	smallint	celda	varchar(114)
optionsarray	int	imei	varchar(114)
servicefeedate	bigint	extrabalance	varchar(114)
suspendeddate	bigint	extrabalance2	varchar(114)
frozendate	bigint		

CDR de consumo

El CDR de consumo se encuentra en formato binario ASN1, existen 2 tipos de estos CDRs, cada uno con su layout, en la actualidad solo se obtienen 84 campos de estos CDRs, a continuación, se muestra el layout de salida luego de la decodificación del archivo ASN1.

Tabla 7. Layout de CDR de consumo

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
RECORDTYPE	string	S-Last Sequence Number	string
Potencialduplicate	string	S-Duration	double
SystemType	int	G-Charging Characteristics	string
Record Sequence Number	bigint	G-Charging Type	string
Served IMSI	string	Dynamic Address Flag	string
RecordOpening Time	timestamp	G-Cause for Record Closing	string
Served IMEI	string	G-Complete	string
NAPI for MSISDN	string	G-Uplink	double
Served MSISDN	string	G-Downlink	double
SGSN address	string	G-Quality of Service Requested	string
GGSN Address	string	G-Quality of Service Negotiated	string
Charging ID	bigint	G-Record opening time	timestamp
Access PointName	string	G-Timestamp	timestamp
Served PDP TypeOrg	string	G-First Sequence Number	string
Served PDP TypeNumber	string	G-Last Sequence Number	string
Served PDP Address	string	G-Node ID	string
MS NW Capability	string	G-Duration	double
Routing Area	int	1-Routing Area	string
Location Area Code	int	1-Location Area	string
Cell Identity	string	1-Cell Identity	string

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
SGSN Change	string	2-Routing Area	string
Diagnostics 1	string	2-Location Area code	string
Diagnostics 2	string	2-Cell Identity	string
Diagnostics 3	string	3-Routing Area	string
Diagnostics 4	string	3-Location Area code	string
Diagnostics 5	string	3-Cell Identity	string
Network-initiated PDP Context	string	4-Routing Area	string
Charging Characteristics	string	4-Location Area code	string
Node ID	int	4-Cell Identity	string
Service Degradation	string	5-Routing Area	string
RNC Address	string	5-Location Area code	string
S-Charging Type	string	5-Cell Identity	string
PDP HLR Index	string	M-Sequence Number	string
S-Cause for Record Closing	string	M-Duration	double
S-Complete	string	M-Cause for Record Closing	string
S-Uplink	double	Service Centre	string
S-Downlink	double	Recording Entity	string
S-Quality of Service Requested	string	Calling Party Number	string
S-Quality of Service Negotiated	string	Destination Number	string
S-Record Opening Time	timestamp	Message Reference	string
S-Timestamp	timestamp	SMS Result	string
S-First Sequence Number	string	PLMNIdentifier	string

El cdr ASN1 se decodifica según estándar ETSI/3GPP, TS 32.251-860, en el siguiente documento adjunto se detalla la definición del archivo.



CDRF_R8_Org.asn

Flujo del proceso

El reporte se lo obtiene los primeros días del mes una vez que las fuentes estén disponibles, normalmente se ejecuta al 5to día del mes y consta de los siguientes pasos.

Transferencia de archivos CDRs de la plataforma de cobro prepago a servidores de trabajo GYE y UIO

Los CDRs de la plataforma de cobro prepago se extraen vía SFTP, esta transferencia de archivos de un mes toma 8 horas, esto lo realiza los primeros días del mes, tomando los datos del mes anterior.

Una vez terminada la extracción se realiza una verificación manual de los archivos recibidos versus los archivos que existen en la fuente, se validan cantidad de líneas de los archivos.

Los archivos están comprimidos y en su nombre se describe la fecha y hora de generación.

20201107_073336_gprs_96_3_archive.cdr_cOk.Z

Transferencia de archivos CDRs de la plataforma de consumo a servidores de trabajo GYE y UIO

Los CDRs de la plataforma de consumo se extraen vía SFTP, un servidor extrae los CDRs de GYE y otro servidor extrae los CDRs de UIO.

Transferir los archivos de un mes toma 1 día, durante ese tiempo se descargan los datos GYE y UIO en paralelo, esto lo realizan los primeros días del mes, tomando los datos del mes anterior.

Una vez terminada la extracción se realiza una verificación manual de los archivos recibidos versus los archivos que existen en la fuente, se validan cantidad de archivos.

Los archivos binarios tienen la extensión dat y contienen la fecha y hora de generación.

PGWGYE_INT2020112000311244256375.dat

Decodificación de archivos CDRs de la plataforma de consumo de formato ASN1 a texto

Los archivos de consumo que se descargaron en los servidores GYE y UIO son decodificados empleando un programa java que extrae solamente 84 columnas del CDR y genera un nuevo archivo texto, este proceso toma 1 día cuando se toman todos los recursos de los dos servidores, y hasta 4 días cuando otros procesos se ejecutan en los equipos.

La decodificación se la realiza según estándar ETSI/3GPP, TS 32.251-860, en la actualidad la decodificación de cada archivo toma alrededor de 4 segundos.

El resultado del decodificador no está considerando todos los campos del CDR.

Carga de archivos de plataforma de cobro a tabla en base de datos Oracle

Los archivos de CDRs de cobro prepago son descomprimidos y posteriormente movidos a un directorio para ser vistos como tabla externa a una base de datos Oracle.

Carga de archivos de plataforma de consumo a tabla en base de datos Oracle

Los archivos de CDRs de consumo son movidos a un directorio para ser vistos como tabla externa a una base de datos Oracle.

Replica de tabla de clientes prepagos a base de datos Oracle

Mediante dblink entre bases de datos Oracle se realiza una réplica de la tabla de clientes prepagos a la base de datos Oracle que contiene las tablas externas de CDRs.

Ejecución de proceso para detectar tráfico de datos no cobrado

En el proceso para detectar el tráfico no cobrado se agrupan los registros de la tabla externa de CDRs de consumo por número celular, código de cobro, id de registro y se suma el monto cobrado, luego se compara con la tabla externa de CDRs de cobro con la misma agrupación, las diferencia positivas y negativas son registradas en una

tabla final y al siguiente mes se agregan al análisis, ya que estas diferencias pueden desaparecer días después que lleguen los CDRs de esas transacciones. Las fechas de las transacciones no se consideran al obtener las diferencias ya que puede existir un desfase en la fecha del CDR de consumo y la fecha del CDR de cobro.

En los datos de los CDRs de cobro se deben filtrar solo registro cuyo campo ratinggroup sea 52, 30, 31, 54, 58 o 101, además para la fuente post pago adicional se filtran registros cuyo campo transactiontype sea igual a 308.

El proceso para la generación del reporte de tráfico no cobrado está a cargo de un ingeniero de desarrollo que se encarga de validar fuentes, ejecutar el proceso e informar las líneas con inconsistencia.

ACTORES DEL PROCESO

En el proceso intervienen los siguientes actores.

Tabla 8. Actores del proceso actual

Nombre	Cargo	Departamento	ROL
Ing. Desarrollo 01	Ingeniero Senior	Aseguramiento de Ingresos y Control	Ingeniero de Desarrollo

ROLES DEL PROCESO

Existen los siguientes roles para el proceso.

Tabla 9. Roles del proceso actual

Actor	Descripción	Responsabilidades
Ing. Desarrollo 01	Es la persona encargada de entregar reporte de tráfico no cobrado.	Validar datos de entrada
		Ejecutar proceso
		Entregar reporte

3.3 Alcance del proyecto

Este proyecto busca detectar ingresos no percibidos por errores en la facturación de datos de todo el segmento de clientes de la compañía, esto comprende:

- Entregar un reporte diario de líneas que tengan tráfico no cobrado, considerando todo el segmento de clientes, tanto prepagos como post pagos.
- Reducir el tiempo de entrega del reporte de tráfico no cobrado a un día.
- Decodificación de todos los campos del CDR de consumo.
- Almacenamiento de 6 meses de los CDRs de cobro y de consumo.

- Implementar el proceso en una herramienta escalable y con procesamiento distribuido en el ecosistema Hadoop.

CAPITULO 4

ANÁLISIS, DISEÑO Y DESARROLLO

4.1 Análisis de requerimientos

La empresa de telecomunicaciones requiere un proceso en la plataforma hadoop que genere un reporte diario de clientes con tráfico de datos no cobrado, el reporte se obtiene al comparar registros de CDRs de cobro y CDRs de consumo, más datos adicionales de los clientes obtenidos del sistema comercial de la compañía.

Para el desarrollo del proceso se empleará el modelo cascada, que comprenderá las siguientes etapas.



Figura 4.1. Modelo cascada para desarrollo de proceso masivo de datos

Fuentes del proceso

Fuente de Trafico cobrado prepago: contienen los archivos CDRs de cobro del segmento de clientes prepago, estos archivos serán depositados en los equipos Hadoop por el proveedor de la plataforma de cobro prepago.

Estos archivos están en formato texto y pueden cargarse sin transformación a la base de datos.

Fuente de Trafico cobrado post pago: contienen los archivos CDRs de cobro del segmento de clientes post pago, estos archivos serán depositados en los equipos Hadoop por el proveedor de la plataforma de cobro post pago.

Estos archivos están en formato texto, y contienen 2 columnas opcionales al final de cada registro, es decir en los archivos existirán registros con 61, 60 o 59 columnas dependiendo de si existen las columnas opcionales.

Fuente de Trafico consumido (PGWCDR): contienen los archivos CDRs de consumo de los tipos PGWCDR, estos archivos se almacenan por 10 días en los servidores del proveedor, y tocara tomarlos desde dichos equipos.

Estos archivos están en formato ASN1 y requerirán una decodificación antes de ser almacenados en la base de datos.

Fuente de Trafico consumo (EGCDR): contienen los archivos CDRs de consumo de los tipos EGCDR, estos archivos se encuentran en servidores del departamento de sistemas de la empresa, y tocara tomarlos desde dichos equipos.

Estos archivos están en formato ASN1 y requerirán una decodificación antes de ser almacenados en la base de datos.

Datos de los clientes del segmento prepago: estos datos se encuentran en una base de datos Oracle.

Datos de los clientes del segmento post pago: estos datos se encuentran en una base de datos Oracle.

El proceso tomará las distintas fuentes y las moverá a la base Hive en el ecosistema Hadoop, para las fuentes CDRs se empleará la herramienta NIFI, para las fuentes de bases de datos relacionales se empleará la herramienta sqoop.

Los CDRs de cobro no requerirán mayor transformación para ser subidos a Hive, mientras los CDRs de consumo deberán pasar por un proceso de decodificación antes de ser subidos a Hive.

Los CDRs de cobros se cargarán a Hive en línea, una vez sean depositados en las rutas de recepción, NIFI tomará los archivos y los subirá a Hive, posteriormente este archivo procesado se moverá a una ruta de respaldo donde permanecerá hasta por 3 días.

Los CDRs de consumo se cargarán a Hive en línea, una vez sean depositados en las rutas de recepción, NIFI tomará los archivos, los enviará a decodificar y los subirá a Hive, posteriormente este archivo procesado se moverá a una ruta de respaldo donde permanecerá hasta por 3 días.

El proceso de decodificación arrojará todas las columnas del CDR de consumo para que sea almacenado en HIVE en su totalidad.

La carga de los datos del cliente de las bases de datos relacionales se ejecutará antes de lanzar el proceso de verificación de cobro.

El reporte con tráfico no cobrado se lo entregará todos los días antes de las 8 am, como archivo de texto csv separados por comas y tendrá el siguiente layout.

Tabla 10. Campos de tabla resultado del proceso actual

CAMPOS	TIPO DE DATO
subscriberid	string
charging_id	string
cdr_timestamp_min	string
cdr_timestamp_max	string
duration	double
duration_tn3_cbs	double
diferencia	double
id_subproducto	string
costo	double
fecha	bigint

Los datos de CDRs se almacenarán por 6 meses en Hive, por lo que se requerirá un proceso de depuración para eliminar datos mayores a la ventana definida.

4.2 Diseño de la solución

El proceso para generación del reporte de tráfico no cobrado se creará en la plataforma hadoop distribución Hortonworks, comprenderá las siguientes tareas.

- Ingesta de CDRs de cobro prepago
- Ingesta de CDRs de cobro post pago

- Decodificador de CDRs ASN1
- Ingesta de CDRs de consumo PGWCDR
- Ingesta de CDRs de consumo EGCDR
- Carga de tabla de clientes prepago
- Carga de tabla de clientes post pago
- Proceso para detección de tráfico no cobrado

Ingesta de CDRs de cobro prepago

La ingesta de CDRs se realizará en la herramienta NIFI del clúster HDF, la herramienta permitirá de forma distribuida leer archivos CDRs y depositarlos en la base Hive.

Nifi tomara los archivos CDRs desde una ruta de recepción en el file system de uno de los nodos HDF y los moverá a una ruta de procesados en el mismo file system, para la tarea se requerirán 244 GB en file system para tener los 3 días de respaldo de los archivos CDRs, para file system hadoop (HDFS) se estiman 15 GB diarios, 2.6 TB para mantener 6 meses, esto por la réplica que se tenga configurada en el clúster.

Luego que NIFI lea los archivos CDRs los enviara a hdfs y posteriormente a una estructura en Hive.

Tabla 11. Datos del del proceso de ingesta de CDR de cobro prepago

SERVIDOR ORIGEN	NODO HDF 02
RUTA RECEPCION	/repcion/cdr_cobro/prepago
RUTA PROCESADOS	/repcion/cdr_cobro/prepago/procesados
RUTA HDFS	/raw/cdr/huawei_gdr
ESQUEMA HIVE	COLECTOR
TABLA HIVE	HUAWEI_GDR

Ingesta de CDRs de cobro post pago

La ingesta de CDRs se realizará en la herramienta NIFI del clúster HDF, la herramienta permitirá de forma distribuida leer archivos CDRs y depositarlos en la base Hive.

Nifi tomara los archivos CDRs desde una ruta de recepción en el file system de uno de los nodos HDF, los descomprimirá y los moverá a una ruta de procesados en el mismo file system, para la tarea se requerirán 400 GB en file system para tener los 3 días de respaldo de los archivos CDRs, para file system hadoop (HDFS) se estiman 20 GB diarios, 3.5 TB para mantener 6 meses, esto por la réplica que se tenga configurada en el clúster.

Luego que NIFI lea los archivos CDRs los enviara a hdfs y posteriormente a una estructura en Hive.

Tabla 12. Datos del del proceso de ingesta de CDR de cobro prepago

SERVIDOR ORIGEN	NODO HDF 03
RUTA RECEPCION	/repcion/cdr_cobro/postpago
RUTA PROCESADOS	/repcion/cdr_cobro/postpago/procesados
RUTA HDFS	/raw/cdr/gdr_gprs_crudo
ESQUEMA HIVE	COLECTOR
TABLA HIVE	GDR_GPRS_CRUDO

Decodificador de CDRs ASN1

Se creará una aplicación java que decodifique el CDRs ASN1, el programa recibirá el nombre del archivo ASN1 y generara un archivo decodificado en una ruta especificada en otro parámetro del programa, la salida del archivo decodificado será un formato JSON, que contendrá todos los campos del CDR.

También se creará una estructura en Hive con JsonSerDe, que leerá el archivo json generado por el decodificador.

Ingesta de CDRs de consumo PGWCDR

La ingesta de CDRs se realizará en la herramienta NIFI del clúster HDF, la herramienta permitirá de forma distribuida leer archivos CDRs y depositarlos en la base Hive.

Nifi tomara los archivos CDRs desde una ruta de recepción en el file system del nodo EDGE del HDP, los decodificara empleando el

decodificar java y los moverá a una ruta de procesados en el mismo file system, para la tarea se requerirán 560 GB en file system para tener los 3 días de respaldo de los archivos CDRs, para file system hadoop (HDFS) se estiman 38 GB diarios, 6.6 TB para mantener 6 meses, esto por la réplica que se tenga configurada en el clúster.

Luego que NIFI lea los archivos CDRs decodificados los enviara a hdfs y posteriormente a una estructura en Hive.

Tabla 13. Datos del del proceso de ingesta de CDR de consumo pgwcdr

SERVIDOR ORIGEN	NODO EDGE HDP
RUTA RECEPCION	/repcion/cdr_consumo/pgwcdr/gye /repcion/cdr_consumo/pgwcdr/uio
RUTA PROCESADOS	/repcion/cdr_consumo/pgwcdr/gye/procesados /repcion/cdr_consumo/pgwcdr/uio/procesados
RUTA HDFS	/raw/cdr/gdr_pgwcdr
ESQUEMA HIVE	COLECTOR
TABLA HIVE	gdr_charging_pgwcdr

Ingesta de CDRs de consumo EGCDR

La ingesta de CDRs se realizará en la herramienta NIFI del clúster HDF, la herramienta permitirá de forma distribuida leer archivos CDRs y depositarlos en la base Hive.

Nifi tomará los archivos CDRs desde una ruta de recepción en el file system del nodo EDGE del HDP, los decodificará empleando el decodificar java y los moverá a una ruta de procesados en el mismo file system, para la tarea se requerirán 10 GB en file system para tener los 3 días de respaldo de los archivos CDRs, para file system hadoop (HDFS) se estiman 370 MB diarios, 63 GB para mantener 6 meses, esto por la réplica que se tenga configurada en el clúster.

Luego que NIFI lea los archivos CDRs decodificados los enviara a hdfs y posteriormente a una estructura en Hive.

Tabla 14. Datos del del proceso de ingesta de CDR de consumo egcdr

SERVIDOR	EQUIPO SISTEMAS
ORIGEN	
RUTA RECEPCION	/repcion/cdr_consumo/EGCDR_GYE01blackberry.net /repcion/cdr_consumo/EGCDR_GYE01CORPORATEAPN /repcion/cdr_consumo/EGCDR_UIO01blackberry.net /repcion/cdr_consumo/EGCDR_UIO01CORPORATEAPN
RUTA HDFS	/raw/cdr/gdr_egcdr
ESQUEMA HIVE	COLECTOR
TABLA HIVE	gdr_charging_egcdr

Carga de tabla de clientes post pago

Para la carga de clientes se empleará la herramienta SQOOP, el programa se conectará y realizará la copia de la tabla hacia Hive 1 vez al día antes de que empiece el proceso de tráfico no cobrado.

La ejecución del programa sqoop se la realizará por medio de un Shell script, que también limpiará la tabla final Hive para que sea refrescada en cada carga.

Tabla 15. Datos de fuente y destino de la tabla de clientes post pago

BASE ORIGEN	AXIS
MOTOR BASE ORIGEN	ORACLE
TABLA ORIGEN	CL_SERVICIOS_CONTRATADOS
ESQUEMA HIVE	COLECTOR
TABLA HIVE	CL_SERVICIOS_CONTRATADOS

Carga de tabla de clientes prepago

Para la carga de clientes se empleará la herramienta SQOOP, el programa se conectará y realizará las copias de las tablas hacia Hive 1 vez al día antes de que empiece el proceso de tráfico no cobrado.

La ejecución del programa sqoop se la realizará por medio de un Shell script, que también limpiará la tabla final Hive para que sea refrescada en cada carga.

Tabla 16. Datos de fuente y destino de la tabla de clientes prepago

BASE ORIGEN	EMPREP
MOTOR BASE ORIGEN	ORACLE
TABLAS ORIGEN	inf_subscriber inf_subscriber_his
ESQUEMA HIVE	COLECTOR
TABLAS HIVE	inf_subscriber inf_subscriber_his

Proceso para detección de tráfico no cobrado

El proceso para detectar tráfico no cobrado se lo realizara en un script HIVEQL, este script será llamado desde un shell script y se ejecutara después de que se confirmen las cargas de las tablas de clientes de prepago y post pago.

El script unirá las tablas de cobro prepago y cobro post pago en una tabla temporal cobro_tmp, también unirá las tablas de consumo PGWCDR y consumo EGCDR en otra tabla temporal consumo_tmp, luego se compararán las tablas cobro_tmp y consumo_tmp para obtener las diferencias, luego se incluirán datos adicionales de los clientes cruzando con las tablas de clientes, finalmente se insertarán las líneas con inconsistencia en una tabla final.

Los campos de CDRs que el proceso emplea para la validación son:

Tablas de CDRs de cobro

- subscriber_id
- charging_id
- duration
- cdr_timestamp

Tablas de CDRs de consumo

- servedmsisdn
- chargingid
- datavolumefbcuplink
- datavolumefbcdowndlink
- recordopeningtime

4.3 Arquitectura

La implementación del proceso se la realizara en un ecosistema hadoop en la distribución Hortonworks, esta solución la componen un clúster HDF que se encargara de la ingesta de archivos CDRs y otro clúster HDP que se encargara de almacenar y procesar los datos.

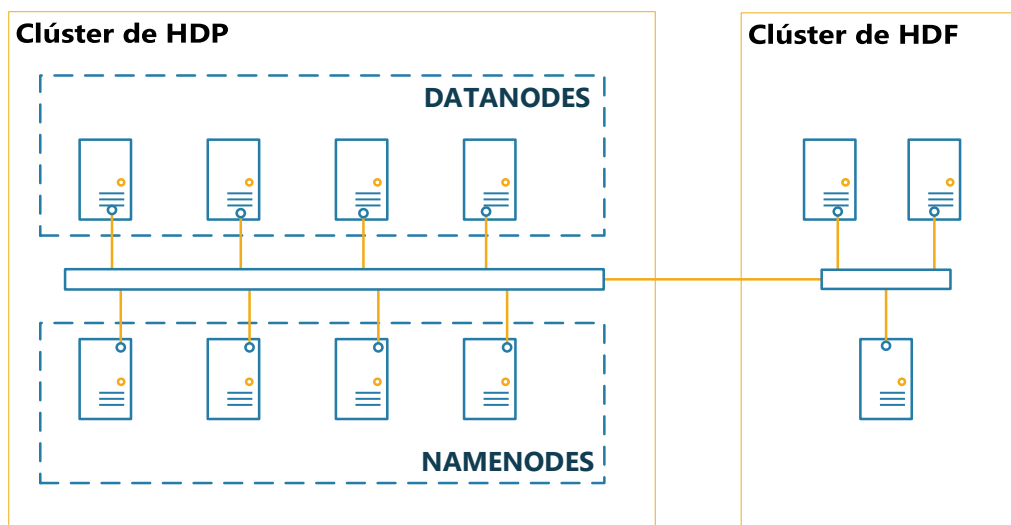


Figura 4.2. Nodos del clúster Big Data

HORTONWORKS DATA FLOW (HDF)

El clúster HDF estará compuesto por 3 nodos, en los cuales se habilitará el servicio NIFI donde se crearán los flujos de cargas para los archivos CDRs, a continuación, se especifican las características de la herramienta.

Versión

Tabla 17. Versión Clúster HDF

Plataforma	HORTONWORKS HDF
Versión	HDP 3.2.0

Equipos

DATA FLOW NODE: 3 (Tres) Máquinas Virtuales con los siguientes recursos.

Tabla 18. Características de los nodos del clúster HDF

DATAFLOW NODE		
Componente	Detalle	Cantidad
Procesador	Virtual Procesor (vCORE)	10
Memoria	Virtual Memory (vMEM GB)	106
Almacenamiento (OS + SERVICIOS)	Virtual Storage (vStorage GB) @ SAS	2000
Almacenamiento (CACHE)	Virtual Storage (vStorage GB) @ SSD	400
Conectividad LAN Administración	NIC Ports @ 1Gbps ETH	2
Conectividad LAN Datos	NIC Ports @ 1Gbps ETH	2
Sistema Operativo	Sistema Operativo Red Hat Enterprise Server 7.2+	1

Componentes

Tabla 19. Componentes instalados en el clúster HDF

Componente	Descripción	Uso
Apache Ambari	Herramienta de Administración	Componente de administración del cluster Hadoop Hortonworks Data Flow
NIFI	Core HADOOP	Apache NiFi, ingesta de datos
ZooKeeper	Un servidor de código abierto que coordina de manera confiable los procesos distribuidos	Servicio centralizado que proporciona coordinación distribuida altamente confiable

HORTONWORKS DATA FLOW (HDF)

En el clúster HDP se habilitarán los servicios HDFS, HIVE, MAP REDUCE, YARN y SQOOP que permitirán el procesamiento y el almacenamiento de los datos.

Versión

Tabla 20. Versión del clúster HDP

Plataforma	HORTONWORKS HDP
Versión	HDP 3.0.1

Equipos

NAME NODE: 4 (Cuatro) Máquinas Virtuales con los siguientes recursos.

Tabla 21. Características de los name nodos del clúster HDP

MANAGEMENT NODE		
Componente	Detalle	Cantidad
Procesador	Virtual Procesor (vCORE)	10
Memoria	Virtual Memory (vMEM GB)	96
Almacenamiento (OS + SERVICIOS)	Virtual Storage (vStorage GB) @ SAS	1300
Almacenamiento (CACHE)	Virtual Storage (vStorage GB) @ SSD	200
Conectividad LAN Administración	NIC Ports @ 1Gbps ETH	2
Conectividad LAN Datos	NIC Ports @ 10Gbps ETH	2
Sistema Operativo	Sistema Operativo Red Hat Enterprise Server	1

DATA NODE: 4 (CU) Máquinas Virtuales con los siguientes recursos.

Tabla 22. Características de los data nodos del clúster HDP

DATA NODE		
Componente	Detalle	Cantidad
Procesador	Virtual Procesor (vCORE)	40
Memoria	Virtual Memory (vMEM GB)	160
Almacenamiento Interno (OS)	Virtual Storage (vStorage GB) @ SAS	1200
Almacenamiento Interno (DATA)	Virtual Storage (vStorage TB) @ NLSAS==>	12
Conectividad LAN Administración	NIC Ports @ 1Gbps ETH	2
Conectividad LAN Datos	NIC Ports @ 1Gbps ETH	2
Sistema Operativo	Sistema Operativo Red Hat Enterprise Server 7.2+	1

Componentes

Tabla 23. Componentes instalados en el clúster HDP

Componente	Descripción	Uso
Apache Ambari	Herramienta de Administración	Componente de administración del cluster Hadoop
Apache HDFS	Core HADOOP	Core Hadoop FS
Apache Hive	Motor SQL HADOOP	EDW
Apache Yarn	Administración de recursos	Gestiona recursos y programa tareas en procesamiento distribuido
Apache Map Reduce	Procesa grandes volúmenes de datos	Software para escribir aplicaciones que se procesan en paralelo y con tolerancia a fallos
Apache Ranger	Seguridad completa para Enterprise Hadoop	Proporciona una plataforma centralizada para definir, administrar y gestionar políticas de seguridad de manera consistente en todos los componentes de Hadoop
Apache ZooKeeper	Un servidor de código abierto que coordina de manera confiable los procesos distribuidos	ZooKeeper proporciona un servicio de configuración distribuida, un servicio de sincronización y un registro de nombres para sistemas distribuidos

PROCESO TRAFICO NO COBRADO

El proceso de tráfico de datos no cobrado se ejecutará en la plataforma Hadoop Hortonworks, empleará el clúster HDF para extraer los archivos CDRs, transformarlos en el caso de los archivos codificados y depositarlos en Hive, en el clúster HDP se ejecutará el proceso de tráfico no cobrado y se almacenaran los datos de CDRs y datos resultantes del proceso en el EWD Hive.

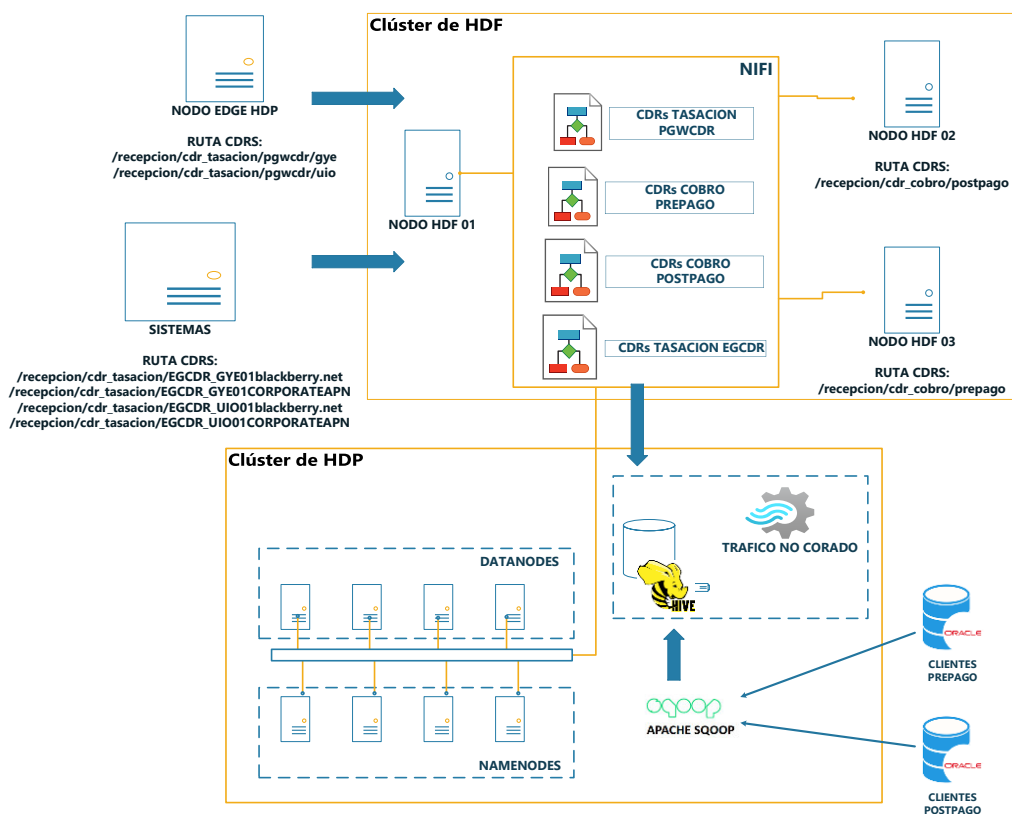


Figura 4.3. Diagrama general del proceso de detección de tráfico no cobrado

4.4 Modelo de flujos de carga

Se está considerando crear 4 flujos de carga uno por cada tipo de CDRs que requerirá el proceso.

Flujo CDR cobro prepago

Se detalla el modelo para el flujo del CDR de cobro prepago.

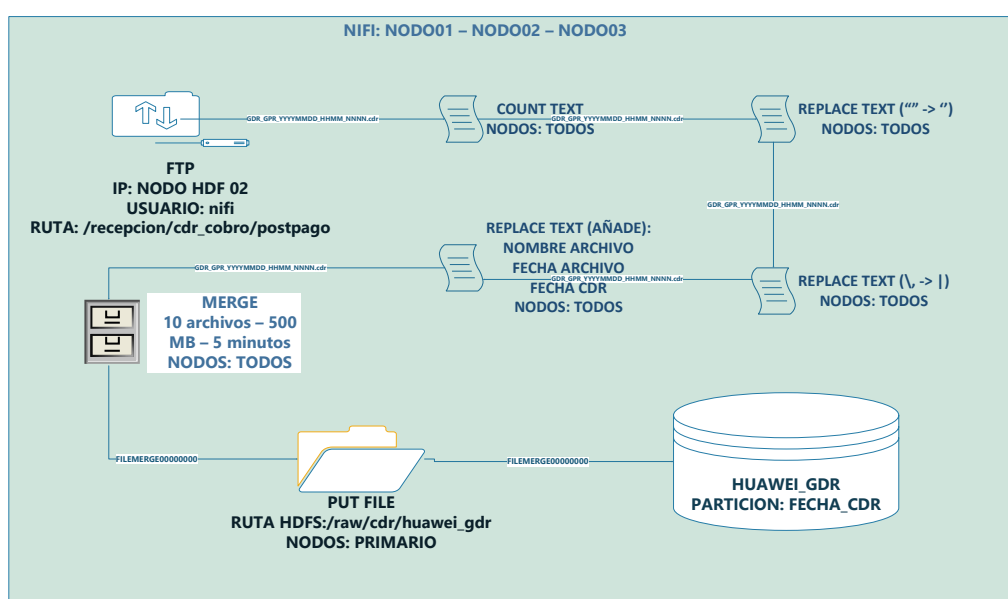


Figura 4.4. Flujo CDR cobro prepago

Flujo CDR cobro prepago

Se detalla el modelo para el flujo del CDR de cobro post pago, existirán 2 flujos el primero descomprime los archivos y el segundo cargara los archivos a Hive.

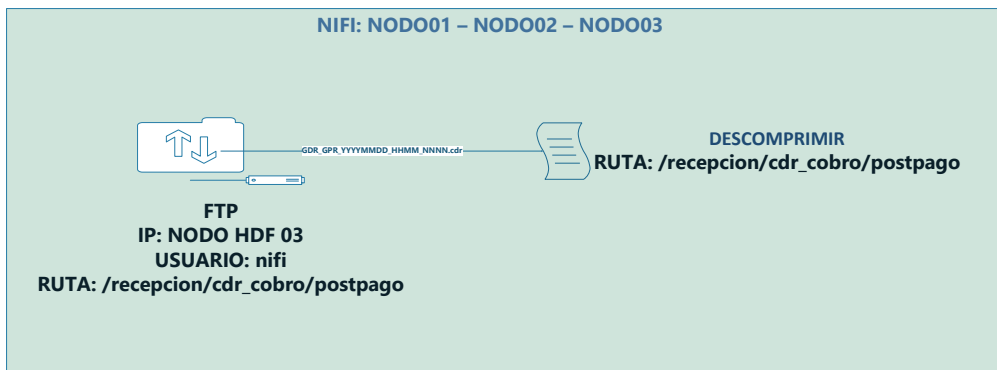


Figura 4.5. Flujo de extracción del CDR cobro prepago

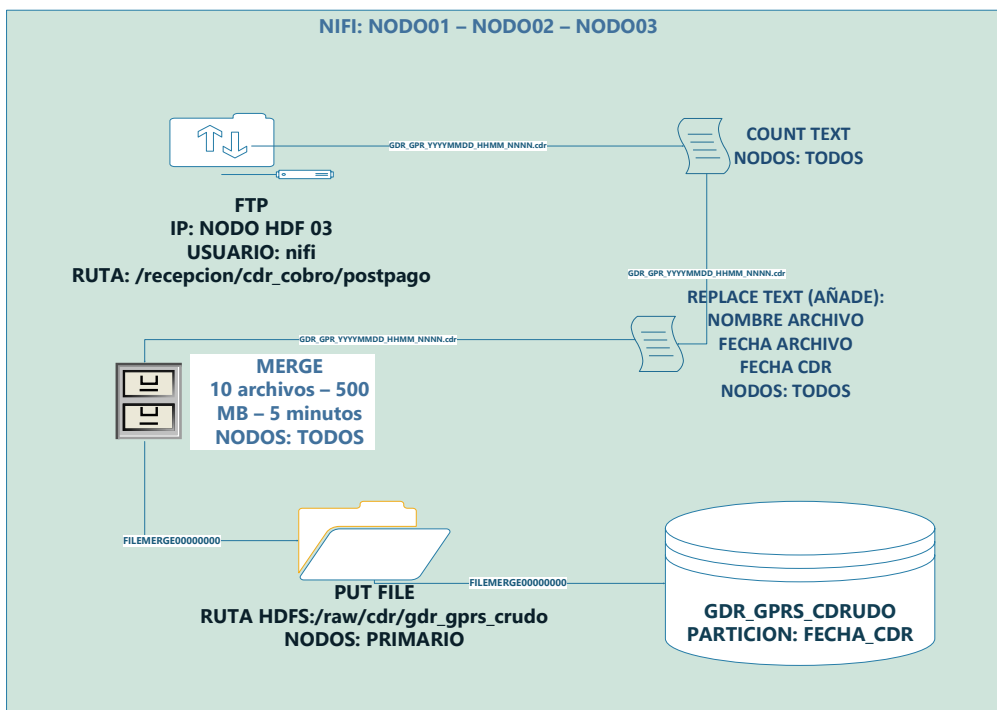


Figura 4.6. Flujo de carga del CDR cobro prepago

Flujo CDR consumo PGWCDR

Se detalla el modelo para el flujo del CDR de consumo PGWCDR, existirán 2 flujos el primero decodifica los archivos y el segundo cargara los archivos a Hive.

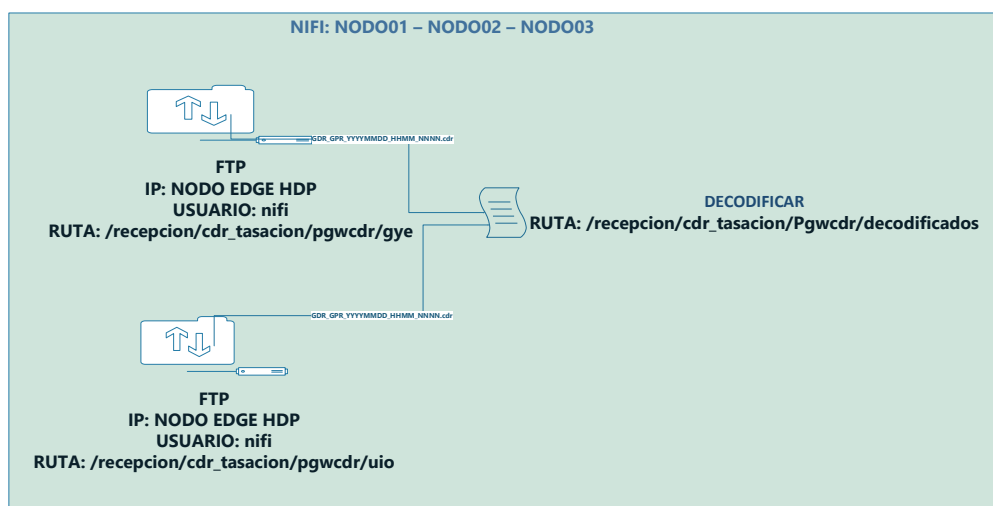


Figura 4.7. Flujo de extracción del CDR consumo PGWCDR

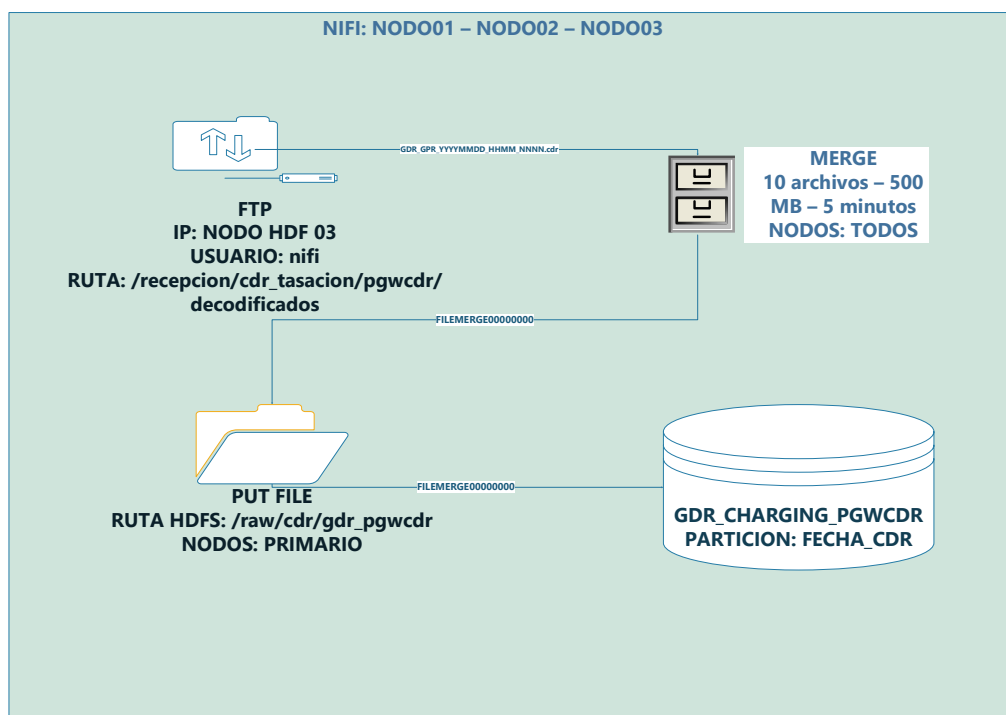


Figura 4.8. Flujo de carga del CDR consumo PGWCDR

Flujo CDR consumo EGCDR

Se detalla el modelo para el flujo del CDR de consumo PGWCDR, existirán 2 flujos el primero decodifica los archivos y el segundo cargara los archivos a Hive.

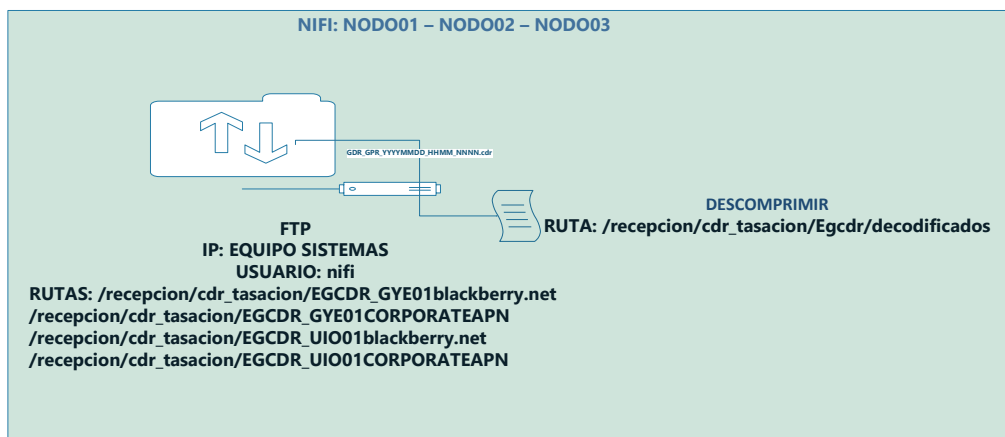


Figura 4.9. Flujo de extracción del CDR consumo EGCDR

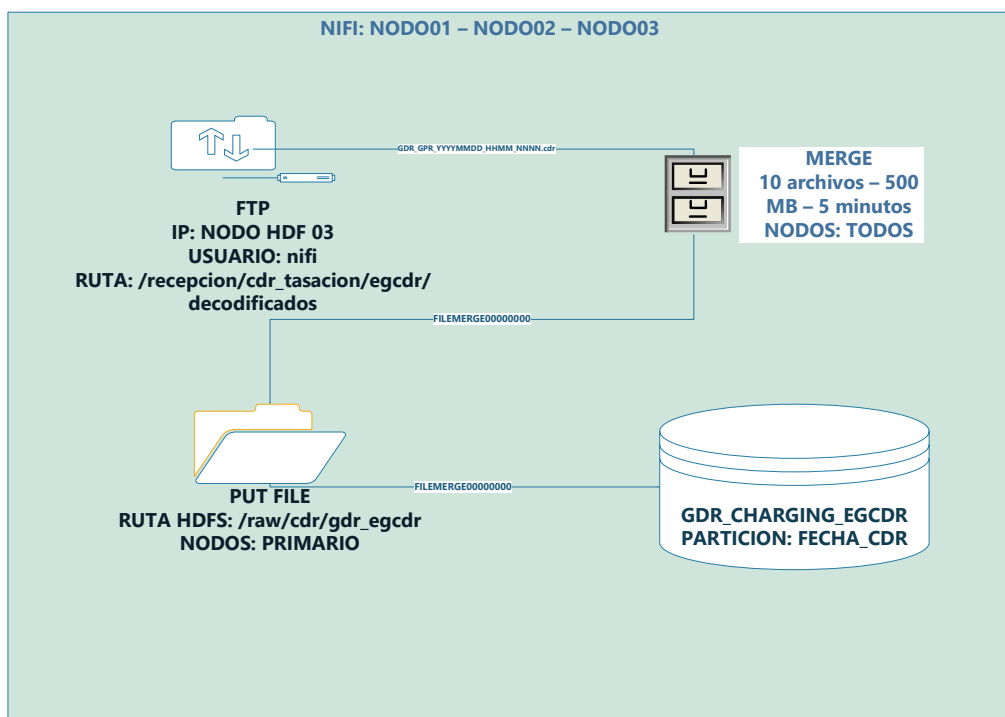


Figura 4.10. Flujo de carga del CDR consumo EGCDR

4.5 Modelo de datos con Hive

Los datos se almacenarán en el EDW de Hadoop Hive, se crearán estructuras para CDRs, datos de clientes y estructura del proceso de verificación de tráfico de datos no cobrado, todas son estructuras independientes, sin relación entre ellas.

Tabla 24. Propiedades tabla de CDR de cobro post pago

Tabla de CDR de cobro post pago	
Tabla	GDR_GPRS_CRUDO
Esquema	COLECTOR
Partición	DIARIA
Tamaño	20 GB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 25. Campos de la tabla de CDR de cobro post pago

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
sequencenumber	smallint	groupid	tinyint
subscriberid	bigint	rulediscountamount	tinyint
cdrdate	bigint	rulediscountunits	tinyint
cdrsseconds	int	rulediscountname	varchar(24)
sourcedate	bigint	discountamount0	int
sourcetime	int	discountschemeid0	tinyint
servicetype	smallint	discountamount1	tinyint
transactiontype	smallint	discountsechemeid1	tinyint
correlationid	varchar(80)	discountamount2	tinyint

CAMPOS	TIPO DE DATO
transactionmajornumber	int
transactionminornumber	tinyint
sourcetype	tinyint
sourceinfo1	bigint
sourceinfo2	smallint
ratename	varchar(25)
tariffplanid	smallint
systemtax	tinyint
locationtax	tinyint
servicetax	tinyint
billingeventid	smallint
timebandgroupid	smallint
categoryid	smallint
operatorid	bigint
operatortransactiontype	smallint
accountnumber	varchar(20)
suscriberclass	tinyint
profileid	smallint
optionsarray	int
servicefeedate	bigint
suspendeddate	bigint
frozendate	bigint
grouptype	tinyint

CAMPOS	TIPO DE DATO
discountschemeid2	tinyint
accounttype	tinyint
accountstatus	tinyint
balancedeftype	tinyint
balancedefid	int
balanceamount	bigint
balanceunits	tinyint
balancedelta	bigint
reservationcount	smallint
networktype	int
gprs_imsi	bigint
apn	varchar(118)
serviceid	smallint
userequipment	bigint
charging_id	bigint
rating_group_id	bigint
celda	varchar(114)
imei	varchar(114)
extrabalance	varchar(114)
extrabalance2	varchar(114)
nombre_archivo	string
fecha_archivo	string
fecha_cdr	string

Tabla 26. Propiedades tabla de CDR de cobro prepago

Tabla de CDR de cobro prepago	
Tabla	HUAWEI_GDR
Esquema	COLECTOR
Partición	DIARIA
Tamaño	15 GB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 27. Campos de la tabla de CDR de cobro prepago

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
subscriber_id	bigint	total_balance_delta	bigint
profile_id	int	total_balance_curr_id	varchar(100)
account_status	smallint	total_fu_balance_type	varchar(100)
account_type	smallint	total_fu_balance_amount	bigint
calling_imsi	bigint	total_fu_balance_delta	bigint
calling_imei	bigint	total_fu_balance_curr_id	varchar(100)
called_imsi	bigint	total_bonus_balance_type	varchar(100)
cdr_date	bigint	total_bonus_balance_amount	bigint
cdr_sseconds	int	total_bonus_balance_delta	bigint
cdr_timestamp	bigint	total_bonus_balance_curr_id	varchar(100)
time_submission	bigint	total_bonus_fu_balance_type	bigint
time_source	bigint	total_bonus_fu_balance_amount	bigint
latitud	varchar(20)	total_bonus_fu_balance_delta	bigint
longitud	varchar(20)	total_bonus_fu_balance_curr_id	bigint
cdr_name	varchar(100)	transfer_amt	bigint
sequence_number	bigint	adjust_amt	bigint
scr_cdr_no	bigint	recharge_amt	bigint

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
lob	smallint	sub_key	bigint
service_category	bigint	procesamiento_fecha	bigint
usage_service_type	varchar(100)	procesamiento_id	bigint
bill_cycle_id	bigint	called_home_network_code	bigint
start_time_of_bill_cycle	bigint	last_effect_offering	bigint
transaction_type	smallint	bearer_capability	varchar(50)
external_transaction_type	varchar(50)	payment_type	varchar(10)
operator_id	bigint	ext_trans_type	varchar(50)
duration	bigint	usage_measure_id	bigint
b_party_number	varchar(100)	debit_amount	bigint
destination_name	varchar(50)	reason_code	varchar(32)
main_offering	bigint	channel_id	varchar(20)
tpi	varchar(256)	access_method	bigint
red	smallint	card_sequence	varchar(50)
tipo_red	bigint	payment_method	varchar(100)
red_destino	smallint	transid	varchar(100)
apn	varchar(50)	thirdid	varchar(100)
url	varchar(50)	offerprice	varchar(100)
rating_group	smallint	card_cos_id	int
service_flow	varchar(10)	card_amount	bigint
roam_state	smallint	debit_from_advance_prepaid	bigint
onnet_flag	smallint	free_unit_amount_of_times	bigint
call_type	smallint	chargepartyindicator	varchar(10)
hotline_indicator	bigint	charging_id	varchar(50)
location_id	varchar(50)	session_id	varchar(256)
cur_expire_time	bigint	obj_type	varchar(10)
cur_expire_time_date_fu	bigint	obj_id	bigint
service_id	bigint	calling_home_network_code	bigint
content_id	bigint	calling_roam_country_code	bigint
service_type	bigint	calling_roam_network_code	bigint
content_type	bigint	called_home_country_code	bigint
service_capability	bigint	dept_id	bigint
provision_type	bigint	offeringtrace	string
category_type	bigint	additionalinfo	string

CAMPOS	TIPO DE DATO	CAMPOS	TIPO DE DATO
fn_flag	smallint	nombre_archivo	string
ctd_detalle	bigint	fecha_archivo	string
total_balance_type	varchar(100)	fecha_cdr	string
total_balance_amount	bigint		

Tabla 28. Propiedades tabla de CDR de consumo del tipo pgwcdr

Tabla de CDR de consumo PGWCDR	
Tabla	GDR_CHARGING_PGWCDR
Esquema	COLECTOR
Partición	DIARIA
Tamaño	38 GB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 29. Campos de la tabla de CDR de consumo del tipo pgwcdr

CAMPOS	TIPO DE DATO
recordtype	string
servedimsi	string
pgwaddress	array<struct<ipbinaryaddress:array<struct<ipbinv4address:string>>>>
chargingid	bigint
servingnodeaddress	array<struct<ipbinaryaddress:array<struct<ipbinv4address:string>>>>
accesspointnameni	string
pdppdtype	string

CAMPOS	TIPO DE DATO
servedpdppdnaddress	array<struct<ipaddress:array<struct<ipbinv4address:string>>>>
dynamicaddressflag	string
listoftrafficvolumes	struct<changeofcharcondition:array<struct<datavolumeegprsdownlink:string, datavolumeegprsuplink:string, changecondition:string,changetime:string, userlocationinformation:struct<plmn:int, lac:int,sac:int>,epc qosinformation:array<struct<qci:int,arp:int>>>>
recordopeningtime	bigint
duration	string
causeforrecclosing	string
recordsequencenumber	bigint
nodeid	string
recordextensions	array<struct<extensiontype:int,length:int, servicelist:array<struct<servicecode:int, uplinkvolume:int,downlinkvolume:int, usageduration:int,url:string>>>>
localsequencenumber	bigint
apnselectionmode	string
servedmsisdn	string
chargingcharacteristics	string
chchselectionmode	string
servingnodeplmnidentifier	string
servedimei	string
servedimeisv	string
ratype	string
mstimezone	string
userlocationinformation_plmn	string
userlocationinformation_lac	string
userlocationinformation_sac	string
userlocationinformation_ci	string
userlocationinformation_tai	string
userlocationinformation_ecgi	string
listofservicedata	array<struct<ratinggroup:int, chargingrulebasename:string, localsequencenumber:int, timeoffirstusage:string,timeoflastusage:string, timeusage:string,serviceconditionchange:string,

CAMPOS	TIPO DE DATO
	qosinformationneg:array<struct<qci:int,arp:int>>, sgsnaddress:array<struct<ipbinv4address:string>>, datavolumeefbcuplink:string, datavolumeefbcdnlink:string, timeofreport:string, userlocationinformation:struct <plmn:int,lac:int,sac:int>>>
servingnodetype	array<string>
nombre_archivo	string
fecha_archivo	string
fecha_cdr	bigint

Tabla 30. Propiedades tabla de CDR gdr_pgwcdr_listofservicedata

Tabla de CDR de consumo PGWCDR detalle de consumo	
Tabla	GDR_PGWCDR_LISTOFSERVICEDATA
Esquema	COLECTOR
Partición	DIARIA
Tamaño	9 GB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 31. Campos de la tabla de CDR gdr_pgwcdr_listofservicedata

CAMPOS	TIPO DE DATO
nombre_archivo	string
localsequencenumber	bigint
servedmsisdn	string
chargingid	bigint
recordopeningtime	string
ratinggroup	int
chargingrulebasename	string
localsequencenumber_serv	int
timeoffirstusage	string
timeoflastusage	string
timeusage	string
serviceconditionchange	string
qosinformationneg	array<struct<qci:int,arp:int>>
sgsnaddress	array<struct<ipbinv4address:string>>
datavolumeefbcuplink	string
datavolumeefbcdowndownlink	string
timeofreport	string
plmn	int
lac	int
sac	int
fecha_archivo	string
fecha_cdr	bigint

Tabla 32. Propiedades tabla de CDR de consumo del tipo egcdr

Tabla de CDR de consumo EGCDR	
Tabla	GDR_CHARGING_EGCDR
Esquema	COLECTOR
Partición	DIARIA
Tamaño	38 GB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 33. Campos de la tabla de CDR de consumo del tipo egcdr

CAMPOS	TIPO DE DATO
recordtype	string
networkinitiation	string
servedimsi	string
ggsnaddress	array<struct<ipbinaryaddress:array<struct<ipbinv4address:string>>>>
chargingid	bigint
sgsnaddress	array<struct<ipbinaryaddress:array<struct<ipbinv4address:string>>>>
accesspointnameni	string
pdptype	string
servedpdpaddress	array<struct<ipaddress:array<struct<ipbinv4address:string>>>>
dynamicaddressflag	string
listoftrafficvolumes	array<struct<qosnegotiated:string,datavolumegprsuplink:string,datavolumegprsdnlink:string,changecondition:string,changetime:string,userlocationinformation:struct<plmn:int,lac:int,sac:int,ci:int>>>
recordopeningtime	string
duration	string
causeforreclosing	string
recordsequencenumber	bigint

CAMPOS	TIPO DE DATO
nodeid	string
recordextensions	array<struct<extensiontype:int,length:int,servicelist:array<struct<servicecode:int,uplinkvolume:bigint,downlinkvolume:bigint,usageduration:bigint,url:string>>>>
localsequencenumber	bigint
apnselectionmode	string
servedmsisdn	string
chargingcharacteristics	string
chchselectionmode	string
sgsnplmnidentifier	string
servedimei	string
servedimeisv	string
rattype	string
mstimezone	string
userlocationinformation_plmn	string
userlocationinformation_lac	string
userlocationinformation_sac	string
userlocationinformation_ci	string
listofservedata	array<struct<ratinggroup:int,chargingrulebasename:string,localsequencenumber:bigint,timeoffirstusage:string,timeoflastusage:string,timeusage:string,serviceconditionchange:string,qosinformationneg:string,sgsnaddress:array<struct<ipbinv4address:string>>,sgsnplmnidentifier:struct<id:string,mcc:string,mnc:string>,datavolumefbcuplink:string,datavolumefbcdnlink:string,timeofreport:string,rattype:string,userlocationinformation:struct<plmn:int,lac:int,sac:int,ci:int>>>
nombre_archivo	string
fecha_archivo	string
fecha_cdr	bigint

Tabla 34. Propiedades de la tabla de CDR gdr_egcdr_listofservicedata

Tabla de CDR de consumo EGCDR detalle de consumo	
Tabla	GDR_EGCDR_LISTOFSERVICEDATA
Esquema	COLECTOR
Partición	DIARIA
Tamaño	170 MB diarios
Formato	ORC
Transacciones	Solo insert

Tabla 35. Campos de la tabla de CDR gdr_egcdr_listofservicedata

CAMPOS	TIPO DE DATO
nombre_archivo	string
localsequencenumber	bigint
servedmsisdn	string
chargingid	bigint
recordopeningtime	string
ratinggroup	int
chargingrulebasename	string
localsequencenumber_serv	int
timeoffirstusage	string
timeoflastusage	string
timeusage	string
serviceconditionchange	string
qosinformationneg	string
sgsnaddress	array<struct<ipbinv4address:string>>
id	string
mcc	string
mnc	string
datavolumeefbcuplink	string
datavolumeefbcdowndlink	string
timeofreport	string

CAMPOS	TIPO DE DATO
ratype	string
plmn	int
lac	int
sac	int
ci	int
fecha_archivo	string
fecha_cdr	bigint

Tabla 36. Propiedades de la tabla resultado del proceso de tráfico no cobrado

Tabla de resultado del proceso de tráfico no cobrado	
Tabla	SUS_SUSPENDED_GPRS_SUB
Esquema	COLECTOR
Partición	DIARIA
Tamaño	240 MB diarios
Formato	ORC
Transacciones	truncate, insert

Tabla 37. Campos de la tabla resultado del proceso de tráfico no cobrado

CAMPOS	TIPO DE DATO
subscriberid	string
charging_id	string
cdr_timestamp_min	string
cdr_timestamp_max	string
duration	double
duration_tn3_cbs	double
diferencia	double
id_subproducto	string
costo	double
fecha	bigint

4.6 Desarrollo de flujos de carga para archivos

Los flujos de carga se desarrollan en la herramienta Apache NiFi Version 1.11.4 en su interfaz web.

Flujo CDRs de cobro prepago

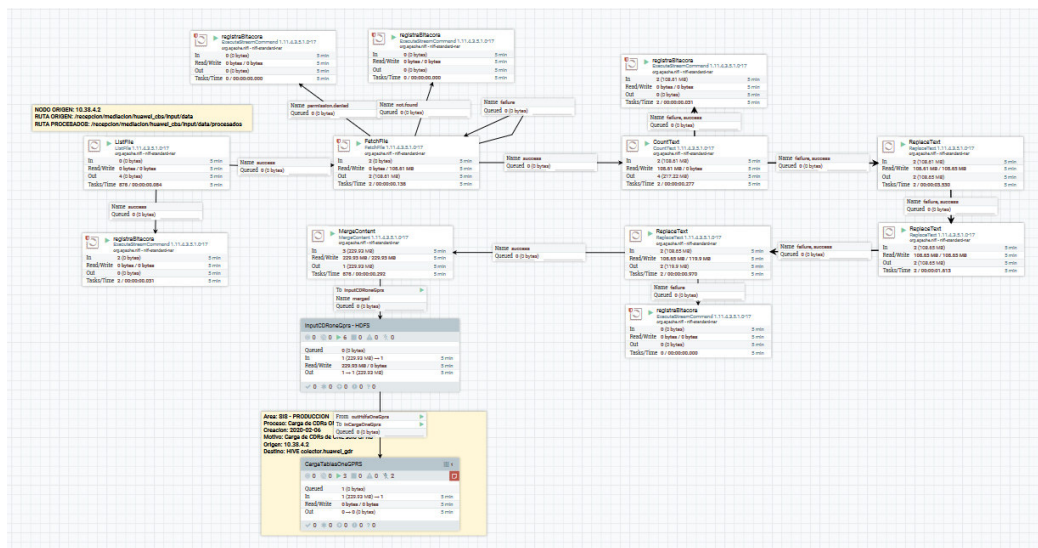


Figura 4.11. Flujo NIFI de la carga de CDRs de cobro prepago

Tabla 38. Detalle del flujo NIFI de la carga de CDRs de cobro prepago

Process group name: Carga_CDRs_One_GPRS	
ListFile	<p>Recupera una lista de archivos del sistema de archivos local, crea un FlowFile que representa el archivo.</p> <p>Properties: Input Directory: /recepcion/mediacion/huawei_cbs/input/data File Filter: *.cdr</p>
FetchFile	<p>Lee el contenido de un archivo del disco y lo transmite al FlowFile.</p> <p>Properties: File to Fetch: \${absolute.path}/\${filename} Completion Strategy: Move File Move Destination Directory: /recepcion/mediacion/huawei_cbs/input/data/procesados</p>
CountText	<p>Cuenta varias métricas sobre el texto entrante que se generaran como atributos del FlowFile</p> <p>Properties: Count Lines: True</p>
ReplaceText	<p>Actualiza el contenido de un FlowFile evaluando una expresión regular (regex) y reemplazando la sección del contenido que coincide con la expresión regular con algún valor alternativo.</p>

Process group name: Carga_CDRs_One_GPRS	
	Properties: Search Value: (?<gp>""") Replacement Value: ""
ReplaceText	Actualiza el contenido de un FlowFile evaluando una expresión regular (regex) y reemplazando la sección del contenido que coincide con la expresión regular con algún valor alternativo. Properties: Search Value: (?<gp>\\,) Replacement Value:
ReplaceText	Actualiza el contenido de un FlowFile evaluando una expresión regular (regex) y reemplazando la sección del contenido que coincide con la expresión regular con algún valor alternativo. Properties: Search Value: \$ Replacement Value: ,\${filename},\${filename:substring(8,16)},{now():toNumber():format('yyyyMMdd')}
MergeContent	Fusiona un grupo de FlowFiles en función de una estrategia definida por el usuario y los empaqueta en un solo FlowFile. Properties: Maximum Number of Entries: 20 Maximum Group Size: 500 MB Max Bin Age: 5 minutes
PutHDFS	Escribe datos de FlowFile en el sistema de archivos distribuido de Hadoop (HDFS). Properties: Directory: /raw/cdr/huawei_gdr
ExecuteStreamCommand	Ejecuta un comando externo sobre el contenido de un FlowFile. Properties: Command Path: /procesos/aic/cdr_cbs/bin/cargaDetallesOneGprs.sh

Tabla 39. Detalle de script cargaDetallesOneGprs

Script HIVEQL: cargaDetallesOneGprs.sh	
Sentencia:	<pre>insert into colector.huawei_gdr partition(fecha_cdr) select * from colector.huawei_gdr_external</pre>

Flujo CDRs de cobro post pago

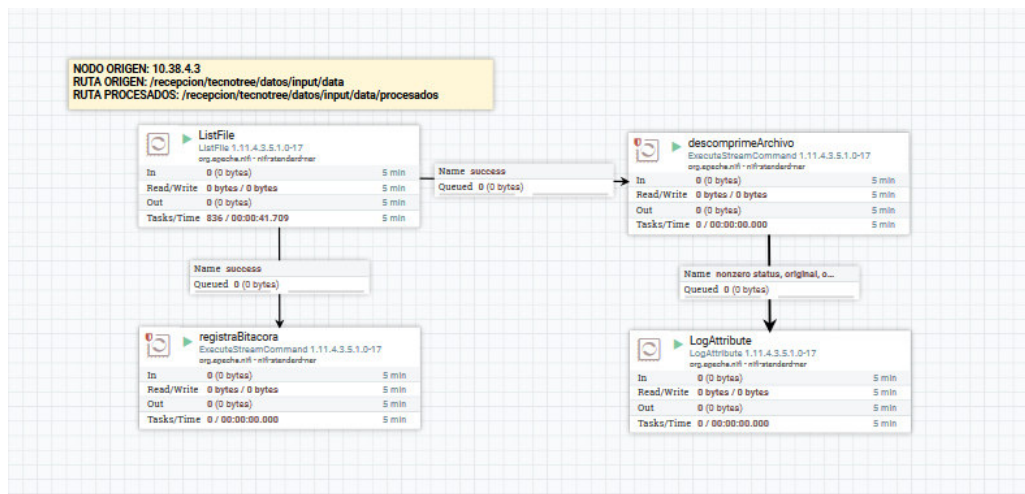


Figura 4.12. Flujo NIFI extracción de CDRs de cobro post pago

Tabla 40. Detalle de flujo NIFI extracción de CDRs de cobro post pago

Process group name: Carga_CDRs_One_GPRS	
ListFile	Recupera una lista de archivos del sistema de archivos local, crea un FlowFile que representa el archivo. Properties: Input Directory: /recepcion/tecnotree/datos/input/data File Filter: .*archive\.cdr_cOk.gz
ExecuteStreamCommand	Ejecuta un comando externo sobre el contenido de un FlowFile. Properties: Command Path: /procesos/aic/cdr_tecnotree/bin/descomprime_archivo.sh

Tabla 41. Detalle de script descomprime_archivo

Script BASH: descomprime_archivo.sh	
Sentencia:	gunzip \$path/"\$filename awk -F"," 'NF==59 { print \$0", , " } NF==60 { print \$0", " } NF==61 { print \$0 }' \$path/"\$archivounzip > \$path/"\$archivounzip".cdr"

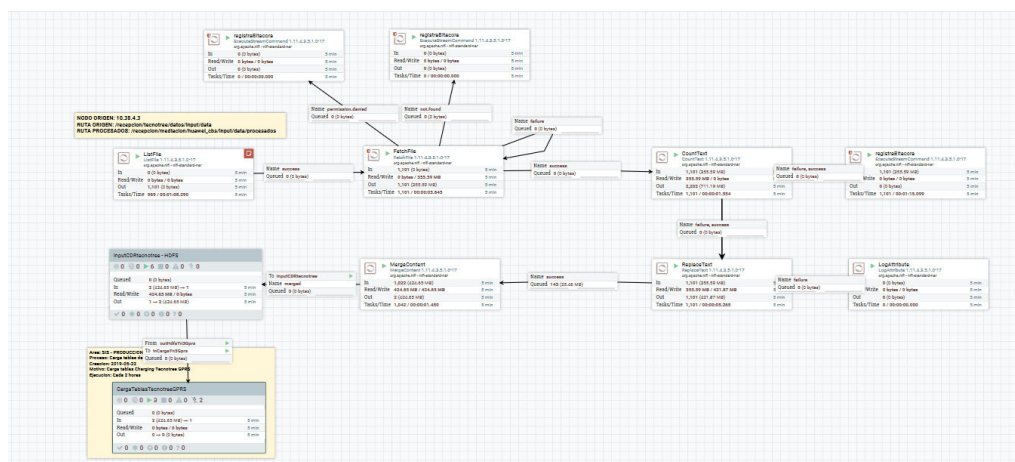


Figura 4.13. Flujo NIFI carga de CDRs de cobro post pago

Tabla 42. Detalle del flujo NIFI carga de CDRs de cobro post pago

Process group name: Carga_CDRs_GPRS_Tecnotree	
ListFile	<p>Recupera una lista de archivos del sistema de archivos local, crea un FlowFile que representa el archivo.</p> <p>Properties: Input Directory: /recepcion/tecnotree/datos/input/data File Filter: .*archive\.cdr_cOk.cdr</p>
FetchFile	<p>Lee el contenido de un archivo del disco y lo transmite al FlowFile.</p> <p>Properties: File to Fetch: \${absolute.path}/\${filename} Completion Strategy: Move File Move Destination Directory: /recepcion/tecnotree/datos/input/data/procesados</p>
CountText	<p>Cuenta varias métricas sobre el texto entrante que se generaran como atributos del FlowFile</p> <p>Properties: Count Lines: True</p>
ReplaceText	<p>Actualiza el contenido de un FlowFile evaluando una expresión regular (regex) y reemplazando la sección del contenido que coincide con la expresión regular con algún valor alternativo.</p> <p>Properties: Search Value: \$ Replacement Value: ,\${filename},\${now():toNumber():format('yyyyMMddHHmmss')},\${now():toNumber():format('yyyyMMdd')}</p>

Process group name: Carga_CDRs_GPRS_Tecnotree	
MergeContent	<p>Fusiona un grupo de FlowFiles en función de una estrategia definida por el usuario y los empaqueta en un solo FlowFile.</p> <p>Properties: Maximum Number of Entries: 5000 Maximum Group Size: 500 MB Max Bin Age: 5 minutes</p>
PutHDFS	<p>Escribe datos de FlowFile en el sistema de archivos distribuido de Hadoop (HDFS).</p> <p>Properties: Directory: /raw/cdr/gdr_gprs_crudo</p>
ExecuteStreamCommand	<p>Ejecuta un comando externo sobre el contenido de un FlowFile.</p> <p>Properties: Command Path: /procesos/aic/cdr_tecnotree/bin/cargaDetallesGprs.sh</p>

Tabla 43. Detalle de script cargaDetallesGprs

Script HIVEQL: cargaDetallesGprs.sh	
Sentencia:	<pre>insert into colector.gdr_gprs_crudo partition(fecha_cdr) select * from colector.gdr_gprs_crudo_external</pre>

Flujo CDRs de consumo PGWCDR

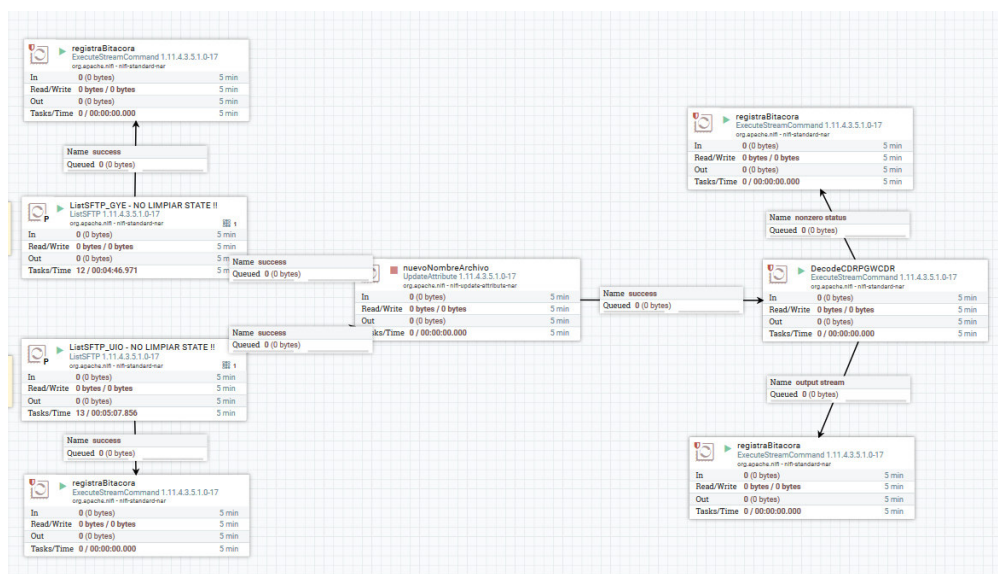


Figura 4.14. Flujo NIFI extracción de CDRs de consumo, tipo pgwcdr

Tabla 44. Detalle de flujo NIFI extracción de CDRs de consumo, tipo pgwcdr

Process group name: Extraccion_PGWCDR	
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties: HostName: 10.31.32.29 Username: usr_ftp_hdf IRemote Path: /recepcion_cg/charging/pgwcdr/gye/input/data File Filter Regex: ^PGWGYE.*\..dat</p>
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties: HostName: 10.31.32.29 Username: usr_ftp_hdf IRemote Path: /recepcion_cg/charging/pgwcdr/uiu/input/data File Filter Regex: ^PGWUIO.*\..dat</p>
UpdateAttribute	<p>Este procesador actualiza los atributos de un FlowFile utilizando propiedades o reglas que agrega el usuario.</p> <p>Properties:</p>

Process group name: Extraccion_PGWCDR	
	nombreBad: \${filename:substringBefore('.')}.err nombreNameProc: \${filename:substringBefore('.')}.enproc nuevoNombre: \${filename:substringBefore('.')}.cdr pathDecoder: /recepcion/charging/pgwcdr/decodificados
ExecuteStreamCommand	Ejecuta un comando externo sobre el contenido de un FlowFile. Properties: Command Path: /procesos/aic/charging/bin/ejecutaDecoder.sh

Tabla 45. Detalle de script ejecutaDecoder

Script BASH: ejecutaDecoder.sh	
Sentencia:	ssh usr_ftp_hdf@10.31.32.29 /usr/java/jdk1.8.0_112/bin/java -jar /procesos/aic/charging/bin/decoder-charging-edge-v2.jar \$archivoOrigen \$archivoEnProc \$archivoDestino \$archivoError

Tabla 46. Detalle de programa decoder-charging

Programa Java: decoder-charging-edge-v2.jar	
Parametros:	archivoOrigen: nombre del archivo origen archivoEnProc: nombre de temporal que se crea mientras se está decodificando el archivo. archivoDestino: nombre del archivo decodificado archivoError: nombre de archive con líneas que no pudieron ser decodificadas.

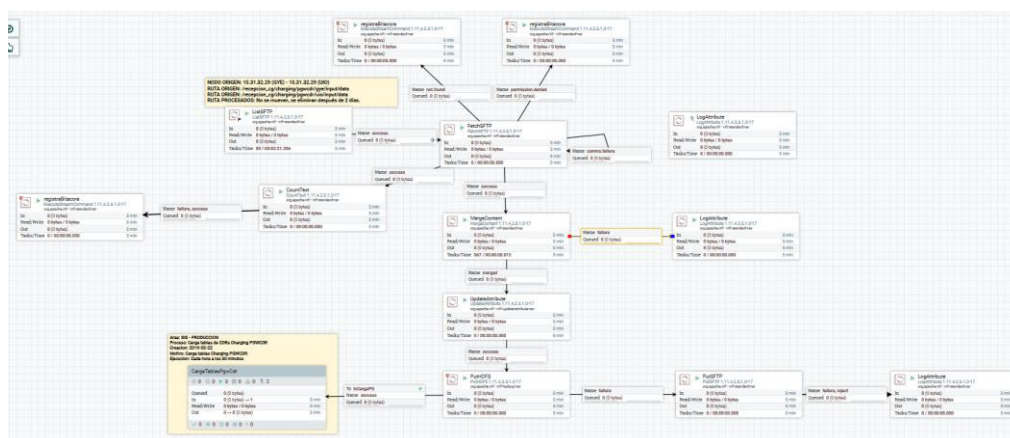


Figura 4.15. Flujo NIFI carga de CDRs de consumo, tipo pgwcdr

Tabla 47. Detalle flujo NIFI carga de CDRs de consumo, tipo pgwcdr

Process group name: Carga_CDR_PGWCDR	
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties: HostName: 10.31.32.29 Username: usr_ftp_hdf IRemote Path: /recepcion/charging/pgwcdr/decodificados File Filter Regex: ^.*\..cdr</p>
FetchSFTP	<p>Obtiene el contenido de un archivo de un servidor SFTP remoto y sobrescribe el contenido de un FlowFile.</p> <p>Properties: File to Fetch: \${path}/\${filename} Completion Strategy: Move File Move Destination Directory: /recepcion/charging/pgwcdr/procesados</p>
MergeContent	<p>Fusiona un grupo de FlowFiles en función de una estrategia definida por el usuario y los empaqueta en un solo FlowFile.</p> <p>Properties: Maximum Number of Entries: 120 Maximum Group Size: 500 MB Max Bin Age: 5 minutes</p>
PutHDFS	<p>Escribe datos de FlowFile en el sistema de archivos distribuido de Hadoop (HDFS).</p> <p>Properties: Directory: /raw/cdr/gdr_pgwcdr</p>
ExecuteStreamCommand	<p>Ejecuta un comando externo sobre el contenido de un FlowFile.</p> <p>Properties: Command Path: /procesos/aic/charging/bin/cargaDetallesPgwcdr.sh</p>

Tabla 48. Detalle de script cargaDetallesPgwcdr

Script HIVEQL: cargaDetallesPgwcdr.sh	
Sentencia:	<pre>insert into colector.gdr_charging_pgwcdr partition(fecha_cdr) select * from colector.gdr_pgwcdr_external; insert into colector.gdr_pgwcdr_listofservicedata partition(fecha_cdr) select * from colector.gdr_pgwcdr_listofservicedata_external;</pre>

Flujo CDRs de consumo EGCDR

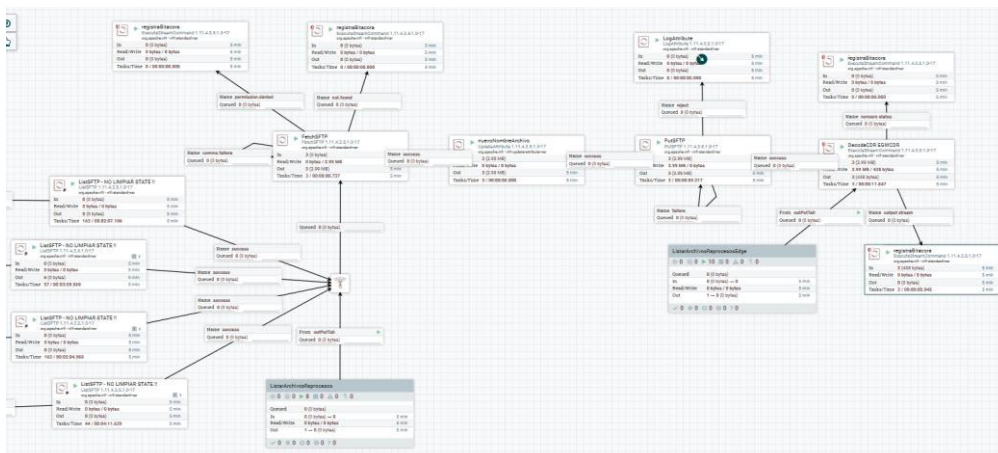


Figura 4.16. Flujo NIFI extracción de CDRs de consumo, tipo egcdr

Tabla 49. Detalle flujo NIFI extracción de CDRs de consumo, tipo egcdr

Process group name: Extraccion_EGCDR	
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties: HostName: 130.2.18.36 Username: usr_ftp_hdf IRemote Path: /proc_cdr/gsioper/CG/GYE01/eGCDR/MASIVO/EGCDR_GYE01 blackberry.net/PROCESADOS File Filter Regex: ^EG.*\dat.proc</p>
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties: HostName: 130.2.18.36 Username: usr_ftp_hdf IRemote Path: /procesos/gsioper/CG/GYE01/eGCDR/CORPORATIVO/EGCDR_GYE01CORPORATEAPN/PROCESADOS File Filter Regex: ^EG.*\dat.proc</p>
ListSFTP	<p>Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile.</p> <p>Properties:</p>

Process group name: Extraccion_EGCDR	
	HostName: 130.2.18.36 Username: usr_ftp_hdf IRemote Path: /proc_cdr/gsioper/CG/UIO01/eGCDR/MASIVO/EGCDR_UIO01bl ackberry.net/PROCESADOS File Filter Regex: ^EG.*\dat.proc
ListSFTP	Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile. Properties: HostName: 130.2.18.36 Username: usr_ftp_hdf Remote Path: /procesos/gsioper/CG/UIO01/eGCDR/CORPORATIVO/EGCDR_UIO01CORPORATEAPN/PROCESADOS File Filter Regex: ^EG.*\dat.proc
FetchSFTP	Obtiene el contenido de un archivo de un servidor SFTP remoto y sobrescribe el contenido de un FlowFile. Properties: File to Fetch: \${path}/\${filename} Completion Strategy: None
UpdateAttribute	Este procesador actualiza los atributos de un FlowFile utilizando propiedades o reglas que agrega el usuario. Properties: nombreNameProc: \${filename:substringBefore('.')}.enproc nuevoNombre: \${filename:substringBefore('.')}.cdr pathDecoder: /repcion/charging/pgwcdcr/decodificados
PutSFTP	Envía FlowFiles a un servidor SFTP. Properties: HostName: 130.2.18.36 Username: usr_ftp_hdf Remote Path: /repcion_cg/charging/egcdr/input/data
ExecuteStreamCommand	Ejecuta un comando externo sobre el contenido de un FlowFile. Properties: Command Path: /procesos/aic/charging/bin/ejecutaDecoder.sh

Tabla 50. Detalle de script ejecutaDecoder

Script BASH: ejecutaDecoder.sh	
Sentencia:	ssh usr_ftp_hdf@10.31.32.29 /usr/java/jdk1.8.0_112/bin/java -jar /procesos/aic/charging/bin/decoder-charging-edge-v2.jar \$archivoOrigen \$archivoEnProc \$archivoDestino \$archivoError

Tabla 51. Detalle de programa decoder-charging

Programa Java: decoder-charging-edge-v2.jar	
Parametros:	archivoOrigen: nombre del archivo origen archivoEnProc: nombre de temporal que se crea mientras se está decodificando el archivo. archivoDestino: nombre del archivo decodificado archivoError: nombre de archive con líneas que no pudieron ser decodificadas.

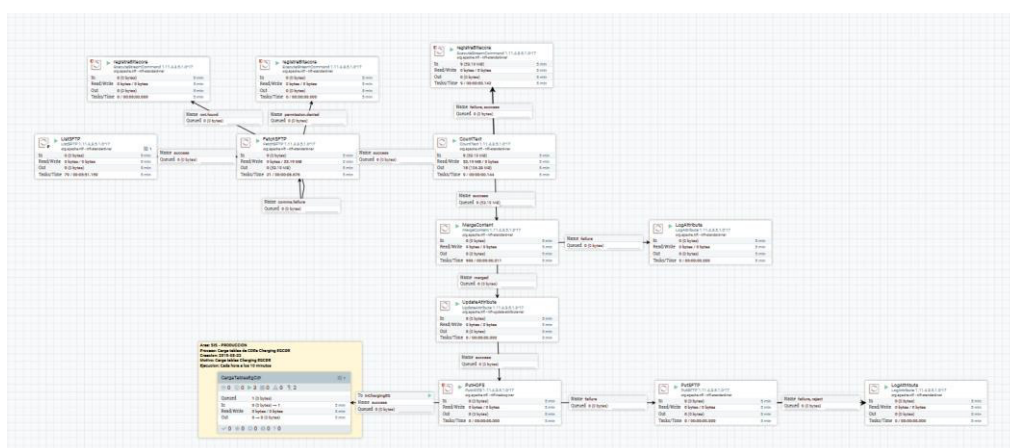


Figura 4.17. Flujo NIFI carga de CDRs de consumo, tipo egcdr

Tabla 52. Detalle flujo NIFI carga de CDRs de consumo, tipo egcdr

Process group name: Carga_CDR_EGWCDR	
ListSFTP	Realiza una lista de los archivos que residen en un servidor SFTP. Para cada archivo que se encuentre en el servidor remoto, se creará un nuevo FlowFile. Properties: HostName: 10.31.32.29 Username: usr_ftp_hdf IRemote Path: /repcion_cg/charging/egcdr/input/data File Filter Regex: ^.*\cdr
FetchSFTP	Obtiene el contenido de un archivo de un servidor SFTP remoto y sobrescribe el contenido de un FlowFile. Properties: File to Fetch: \${path}/\${filename} Completion Strategy: Move File

Process group name: Carga_CDR_EGWCDR	
	Move Destination Directory: /repcion_cg/charging/egcdr/input/data/procesados
MergeContent	Fusiona un grupo de FlowFiles en función de una estrategia definida por el usuario y los empaqueta en un solo FlowFile. Properties: Maximum Number of Entries: 100 Maximum Group Size: 500 MB Max Bin Age: 5 minutes
PutHDFS	Escribe datos de FlowFile en el sistema de archivos distribuido de Hadoop (HDFS). Properties: Directory: /raw/cdr/gdr_egcdr
ExecuteStreamCommand	Ejecuta un comando externo sobre el contenido de un FlowFile. Properties: Command Path: /procesos/aic/charging/bin/cargaDetallesEgcdr.sh

Tabla 53. Detalle de script cargaDetallesEgcdr

Script HIVEQL: cargaDetallesEgcdr.sh	
Sentencia:	insert into colector.gdr_charging_egcdr partition(fecha_cdr) select * from colector.gdr_egcdr_external; insert into colector.gdr_egcdr_listofservicedata partition(fecha_cdr) select * from colector.gdr_egcdr_listofservicedata_external;

4.7 Desarrollo de proceso de extracción para bases de datos

Las extracciones desde bases de datos relacionales hacia hadoop se las desarrolla en Apache Sqoop versión 1.4.7.3.0.1.0-187.

Desarrollo de proceso de extracción para bases de datos

```
sqoop import \  
-libjars ${LIB_JARS} \  
-Dmapreduce.task.timeout=3000000 \  

```

```

-Dorg.apache.sqoop.splitter.allow_text_splitter=true \
--driver oracle.jdbc.driver.OracleDriver \
--class-name CL_SERVICIOS_CONTRATADOS_TMP \
--connect $JDBCURL \
--username $TDUSER \
--password $TDPASS \
--query " $script \$CONDITIONS " \
--split-by "ID_SERVICIO" \
--boundary-query "SELECT MIN(ID_SERVICIO), MAX(ID_SERVICIO) FROM
CL_SERVICIOS_CONTRATADOS " \
--num-mappers 12 \
--target-dir /raw/aic/axis/CL_SERVICIOS_CONTRATADOS \
--escaped-by \\ --optionally-enclosed-by \"\" \
--fields-terminated-by '|'

```

Script sqoop para carga de clientes pospago

```

sqoop import \
-libjars ${LIB_JARS} \
-Dmapreduce.task.timeout=3000000 \
-Dorg.apache.sqoop.splitter.allow_text_splitter=true \
--driver oracle.jdbc.driver.OracleDriver \
--class-name INF_SUBSCRIBER_HIS_TMP \
--connect $JDBCURL \
--username $TDUSER \
--password $TDPASS \
--query " $script \$CONDITIONS " \
--split-by "ID_SERVICIO" \
--boundary-query "SELECT MIN(subs_id), MAX(subs_id) FROM
INF_SUBSCRIBER_HIS " \
--num-mappers 12 \
--target-dir /raw/aic/emprep/ INF_SUBSCRIBER_HIS \
--escaped-by \\ --optionally-enclosed-by \"\" \
--fields-terminated-by '|'

```

```

sqoop import \
-libjars ${LIB_JARS} \
-Dmapreduce.task.timeout=3000000 \
--driver oracle.jdbc.driver.OracleDriver \
--class-name INF_SUBSCRIBER_TMP \
--connect $JDBCURL \
--username $TDUSER \
--password $TDPASS \
--query " $script \$CONDITIONS " \
--split-by "ID_SERVICIO" \
--boundary-query "SELECT MIN(subs_id), MAX(subs_id) FROM INF_SUBSCRIBER
" \
--num-mappers 12 \
--target-dir /raw/aic/emprep/ INF_SUBSCRIBER \
--escaped-by \\ --optionally-enclosed-by \"\" \
--fields-terminated-by '|'

```


4.8 Desarrollo de proceso para identificar tráfico no cobrado

El proceso para identificar tráfico no cobrado se lo desarrolla en lenguaje HIVEQL desde un script bash.

Cruce de tablas de tráfico de cobro y consumo:

```
create table aic_vfs.sus_suspended_gprs_tmp as
select a.subscriberid, a.charging_id, a.cdr_timestamp_min, a.cdr_timestamp_max,
a.duration, nvl(b.duration,0) duration_tn3_cbs
from aic_vfs.sus_cdr_charging_gateway_hist_a a
left outer join aic_vfs.sus_cdr_tecnotree_cbs_hist_a b
on (a.subscriberid = b.subscriberid
and a.charging_id = b.charging_id);
```

Tabla final de tráfico no cobrado con datos adicionales de clientes:

```
insert into aic_vfs.sus_suspended_gprs_sub partition(fecha)
select a.*,
case when nvl(c.service_number,e.service_number) is not null then 'PPA' else
b.ID_SUBPRODUCTO end ID_SUBPRODUCTO,
case when b.ID_SUBPRODUCTO = 'PPA' then (nvl(cast(diferencia as
double),0)/1024/1024) * $costoPPA
when b.ID_SUBPRODUCTO = 'AUT' then (nvl(cast(diferencia as
double),0)/1024/1024) * $costoAUT
when b.ID_SUBPRODUCTO = 'TAR' then (nvl(cast(diferencia as
double),0)/1024/1024) * $costoTAR
else (nvl(cast(diferencia as double),0)/1024/1024) * 0.01 end costo,
$fecha fecha
from aic_vfs.sus_suspended_gprs a
left outer join colector_desa.CL_SERVICIOS_CONTRATADOS_fin b
on (a.subscriberid = b.ID_SERVICIO and cdr_timestamp_min between
b.FECHA_INICIO and b.FECHA_FIN )
left outer join colector_desa.inf_subscriber_fin c
on (a.subscriberid = c.service_number)
left outer join colector_desa.inf_subscriber_his_fin e
on (a.subscriberid = e.service_number and cdr_timestamp_min between e.eff_date
and e.exp_date );
```

CAPITULO 5

IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

5.1 Implementación

Para la implementación se considera que el clúster Big Data de Data Flow y Hadoop ya están operando, por lo que solo comprenderá la puesta en producción de los flujos Data Flow, scripts BASH, HIVEQL, programas JAVA, la creación de estructuras de datos y el encendido de los procesos.

Creación de estructuras de datos

Para crear las estructuras se requerirán los siguientes recursos y tareas.

Tabla 54. Recursos para creación de estructuras de datos

Id	Recurso	Responsable	Descripción
D1	Desarrollador Big Data	Gerente TI	Desarrollador responsable de la puesta en producción
S1	DDL de tablas de CDRs y proceso	D1	Script con sentencias para crear tablas
D2	Ingeniero Producción	Gerente TI	Ingeniero con accesos a equipos y servicios Big Data

Tabla 55. Tareas para puesta en producción de estructuras de datos

Tarea	Recursos	Duración
Abrir sesión ssh en servidor HDF01	D2	5 min
Generar token de conexión a clúster Hadoop con kerberos	D2	5 min
Subir scripts DDL a nodo HDF01	D1 y S1	5 min
Ejecutar script DDL en cliente beeline en servidor HDF01	D1 y S1	5 min

Scripts HIVEQL, BASH y decodificador JAVA

Para instalar los scripts se requerirán los siguientes recursos y tareas.

Tabla 56. Recursos para instalación de scripts

Id	Recurso	Responsable	Descripción
D1	Desarrollador Big Data	Gerente TI	Desarrollador responsable de la puesta en producción
S1	Scripts HiveQL, Bash y decodificador JAVA	D1	Scripts con programas hive y shell script
D2	Ingeniero Producción	Gerente TI	Ingeniero con accesos a equipos y servicios Big Data

Tabla 57. Tareas para puesta en producción de scripts HiveQL y Bash

Tarea	Recursos	Duración
Abrir sesión ssh en servidor HDF01	D2	5 min
Abrir sesión ssh en servidor HDF02	D2	5 min
Abrir sesión ssh en servidor HDF03	D3	5 min
Subir scripts HiveQL, bash y programa JAVA a servidor HDF01	D1 y S1	5 min
Subir scripts HiveQL, bash y programa JAVA a servidor HDF02	D1 y S1	5 min
Subir scripts HiveQL, bash y programa JAVA a servidor HDF03	D1 y S1	5 min

Flujos Data Flow

Para instalar los flujos NIFI requerirán los siguientes recursos y tareas.

Tabla 58. Recursos para instalación de flujos Data Flow

Id	Recurso	Responsable	Descripción
D1	Desarrollador Big Data	Gerente TI	Desarrollador responsable de la puesta en producción
S1	Template de flujos en formato xml	D1	Respaldo de los flujos a crear en herramienta NIFI Data Flow
D2	Ingeniero Producción	Gerente TI	Ingeniero con accesos a equipos y servicios Big Data

Tabla 59. Tareas para puesta en producción de flujos Data Flow

Tarea	Recursos	Duración
Iniciar sesión en herramienta web NIFI	D2	5 min
Subir templates de flujos de CDRs y flujo de proceso de tráfico no cobrado.	D1	15 min
Encender y programar ejecución de flujos CDRs. <ul style="list-style-type: none"> Todos los flujos en stream 	D1	15 min
Encender y programar ejecución de flujo de tráfico no cobrado. <ul style="list-style-type: none"> Ejecución diaria a las 6 am 	D1	15 min

5.2 Resultados de la implementación

Los resultados de la implementación se describen a continuación.

Creación de estructuras de datos

Las estructuras de datos fueron creadas exitosamente en el EWD Hive de Hadoop.

```
0: jdbc:hive2://pgdacl-hdpmg01.conecel.com:21> show tables
+-----+
|          tab_name          |
+-----+
| gdr_charging_egcdr        |
| gdr_charging_pgwcdr       |
| gdr_egcdr_listofservicedata |
| gdr_gprs_crudo            |
| gdr_pgwcdr_listofservicedata |
| huawei_gdr                 |
+-----+
6 rows selected (0.176 seconds)
```

Figura 5.1. Tablas de CDRs en Hive

```
0: jdbc:hive2://pgdacl-hdpmg01.conecel.com:21> show tables like 'sus_suspended_gprs_sub';
+-----+
|          tab_name          |
+-----+
| sus_suspended_gprs_sub    |
+-----+
1 row selected (0.093 seconds)
```

Figura 5.2. Tabla de tráfico no cobrado

Scripts HIVEQL, BASH y decodificador JAVA

Los scripts se subieron a los 3 nodos NIFI.

```
[22:47:46] usr_ftp_hdf-ggcadecodr:/procesos/aic/charging/bin$ ls -ltr decoder-charging-edge.jar
-rwxrwxr-x 1 usr_ftp_hdf usr_ftp_hdf 4315314 Oct 10 2019 decoder-charging-edge.jar
[22:48:02] usr_ftp_hdf-ggcadecodr:/procesos/aic/charging/bin$
```

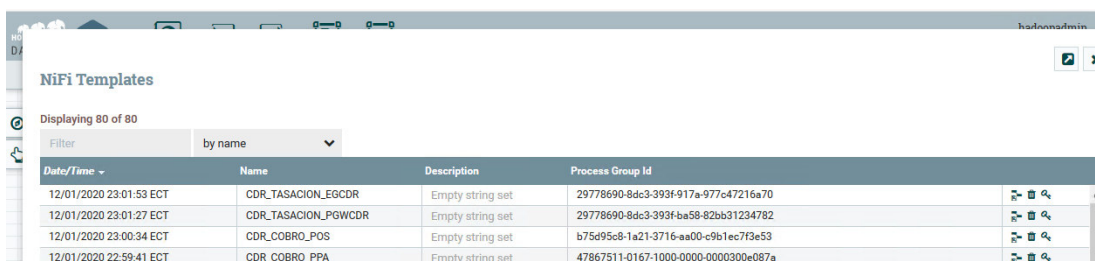
Figura 5.2. Ruta de programa java decoder ASN1

```
[22:49:45] biguser-pgdacl-hdf02:/procesos/aic/suspended/bin$
[22:49:45] biguser-pgdacl-hdf02:/procesos/aic/suspended/bin$ ls -ltr suspendedGPRS.sh
-rwxrwxr-x+ 1 gsioper gsioper 23458 Jul 16 19:32 suspendedGPRS.sh
[22:49:54] biguser-pgdacl-hdf02:/procesos/aic/suspended/bin$
```

Figura 5.3. Ruta script HiveQL de proceso de tráfico no cobrado

Flujos Data Flow

Los flujos fueron cargados correctamente en NIFI Data Flow.



Date/Time	Name	Description	Process Group Id
12/01/2020 23:01:53 ECT	CDR_TASACION_EGCDR	Empty string set	29778690-8dc3-393f-917a-977c47216a70
12/01/2020 23:01:27 ECT	CDR_TASACION_PGWCODR	Empty string set	29778690-8dc3-393f-ba58-82bb31234782
12/01/2020 23:00:34 ECT	CDR_COBRO_POS	Empty string set	675895c8-1a21-3716-aa00-c9b1ec7f3e53
12/01/2020 22:59:41 ECT	CDR_COBRO_PPA	Empty string set	47867511-0167-1000-0000-0000300e087a

Figura 5.4. Plantillas de flujos de CDR cargados en NIFI Data Flow

Ejecución de procesos

Los flujos de carga de archivos fueron encendidos y registran datos en línea.

```

202012012241 (INFO) 20201201,GDR GPP_20201201_2240_5103.cdr,,110307996,LIST,2020-12-01T22:40:43-0500|20201201
202012012241 (INFO) 20201201,GDR GPP_20201201_2240_5103.cdr,GPRS,230516,110307996,FETCH,2020-12-01T22:40:43-0500|20201201
202012012245 (INFO) 20201201,GDR GPR_20201201_2244_5104.cdr,,160809707,LIST,2020-12-01T22:44:54-0500|20201201
202012012245 (INFO) 20201201,GDR GPR_20201201_2244_5104.cdr,GPRS,341215,160809707,FETCH,2020-12-01T22:44:54-0500|20201201
202012012246 (INFO) 20201201,GDR GPP_20201201_2244_5105.cdr,,94819392,LIST,2020-12-01T22:45:25-0500|20201201
202012012246 (INFO) 20201201,GDR GPP_20201201_2244_5105.cdr,GPRS,198286,94819392,FETCH,2020-12-01T22:45:25-0500|20201201
202012012251 (INFO) 20201201,GDR GPR_20201201_2249_5106.cdr,,175207395,LIST,2020-12-01T22:50:00-0500|20201201
202012012251 (INFO) 20201201,GDR GPP_20201201_2250_5107.cdr,,112848582,LIST,2020-12-01T22:50:37-0500|20201201
202012012251 (INFO) 20201201,GDR GPR_20201201_2249_5106.cdr,GPRS,371723,175207395,FETCH,2020-12-01T22:50:00-0500|20201201
202012012251 (INFO) 20201201,GDR GPP_20201201_2250_5107.cdr,GPRS,235907,112848582,FETCH,2020-12-01T22:50:37-0500|20201201
202012012255 (INFO) 20201201,GDR GPR_20201201_2254_5108.cdr,,68523212,LIST,2020-12-01T22:54:34-0500|20201201
202012012255 (INFO) 20201201,GDR GPR_20201201_2254_5108.cdr,GPRS,145393,68523212,FETCH,2020-12-01T22:54:34-0500|20201201
202012012255 (INFO) 20201201,GDR GPP_20201201_2254_5109.cdr,,9321647,LIST,2020-12-01T22:54:36-0500|20201201
202012012255 (INFO) 20201201,GDR GPP_20201201_2254_5109.cdr,GPRS,19489,9321647,FETCH,2020-12-01T22:54:36-0500|20201201
202012012301 (INFO) 20201201,GDR GPR_20201201_2259_5110.cdr,,243099163,LIST,2020-12-01T23:00:24-0500|20201201
202012012301 (INFO) 20201201,GDR GPR_20201201_2259_5110.cdr,GPRS,515839,243099163,FETCH,2020-12-01T23:00:24-0500|20201201
202012012302 (INFO) 20201201,GDR GPP_20201201_2300_5111.cdr,,184305494,LIST,2020-12-01T23:01:33-0500|20201201
202012012302 (INFO) 20201201,GDR GPP_20201201_2300_5111.cdr,GPRS,385420,184305494,FETCH,2020-12-01T23:01:33-0500|20201201
202012012305 (INFO) 20201201,GDR GPR_20201201_2304_5112.cdr,,93367675,LIST,2020-12-01T23:04:38-0500|20201201
202012012305 (INFO) 20201201,GDR GPR_20201201_2304_5112.cdr,GPRS,198092,93367675,FETCH,2020-12-01T23:04:38-0500|20201201
202012012305 (INFO) 20201201,GDR GPP_20201201_2304_5113.cdr,,62539308,LIST,2020-12-01T23:04:55-0500|20201201
202012012305 (INFO) 20201201,GDR GPP_20201201_2304_5113.cdr,GPRS,130739,62539308,FETCH,2020-12-01T23:04:55-0500|20201201
202012012310 (INFO) 20201201,GDR GPR_20201201_2309_5114.cdr,,206810113,LIST,2020-12-01T23:10:15-0500|20201201
202012012310 (INFO) 20201201,GDR GPR_20201201_2309_5114.cdr,GPRS,438745,206810113,FETCH,2020-12-01T23:10:15-0500|20201201
202012012311 (INFO) 20201201,GDR GPP_20201201_2310_5115.cdr,,113884414,LIST,2020-12-01T23:10:58-0500|20201201
202012012311 (INFO) 20201201,GDR GPP_20201201_2310_5115.cdr,GPRS,238100,113884414,FETCH,2020-12-01T23:10:58-0500|20201201
[23:13:12] biguser-pgdacl-hdf02:/procesos/aic/cdr_cbs/logs$

```

Figura 5.5. Bitácora de carga de CDRs de cobro

5.3 Plan de pruebas

El plan de pruebas define condiciones y escenarios de pruebas que evidencien que el producto está completo y que puede operar con todas las funcionalidades definidas por los usuarios.

Se realizarán pruebas internas que comprenderán pruebas de componentes desarrollados y pruebas de rendimiento, también se realizarán pruebas funcionales comprendidas en las pruebas de usuarios.

Entorno de pruebas

Las pruebas se realizarán en el ambiente productivo, ya que ahí se cuenta con todos los accesos a las fuentes del proceso.

- Clúster Hadoop Hortonworks Data Platform
 - 4 nodos de Datos, sistema operativo red hat
 - 4 nodos de Trabajo, sistema operativo red hat
- Cluster Hadoop Data Flow
 - 3 nodos, sistema operativo red hat
- Servidor Edge Hadoop
 - 1 nodo, sistema operativo red hat
- Equipo cliente
 - 1 equipo, sistema operativo Windows 10

Criterios de aceptación o rechazo

Criterio de aceptación, se da por aceptada una prueba cuando no presente errores en su ejecución.

Criterio de suspensión, se da cuando existan errores no controlados.

Recursos para pruebas

Tabla 60. Recursos encargados de llevar a cabo las pruebas

Id	Recurso	Rol	Responsabilidades
D1	Ingeniero QA	Líder de Pruebas	Encargado de la planeación de las pruebas
D2	Ingeniero QA 2	Tester	Encargado de la ejecución de las pruebas
D3	Desarrollador Big Data	Ingeniero de Desarrollo	Encargado de atender incidencias que se presenten durante las pruebas

5.4 Pruebas internas

Para verificar funcionalidad de los componentes se detallan los siguientes casos de pruebas.

Tabla 61. Caso de prueba PI001 – carga de CDRs de cobro post pago

Información General	
Identificador de caso de uso:	PI001
Descripción Prueba:	Carga de CDR cobro post pago
Responsable:	D2
Prerrequisitos	
Deben existir archivos CDRs de cobro post pago en la ruta de recepción.	
Descripción de Casos de Prueba	
Caso: carga de archivo CDR a tabla en base HIVE.	
Entradas: Archivos CDR: 20201107_065119_gprs_122_2_arhive.cdr_cOk 20201107_064204_gprs_85_2_arhive.cdr_cOk	Salidas: Tabla Hive: gdr_gprs_crudo
Componente (s) Probados: Flujo NIFI CDRs cobro prepago	
Instrucciones de Prueba <ol style="list-style-type: none"> Colocar archivo en ruta de recepción de los CDRs de cobro prepago. Encender flujo NIFI. 	
Criterios de Aceptación <ol style="list-style-type: none"> Si los datos del CDR se encuentran en la tabla Hive gdr_gprs_crudo. Si los archivos de entrada con movidos a la ruta de archivos procesados. 	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 62. Caso de prueba PI002 – carga de CDRs de cobro prepago

Información General	
Identificador de caso de uso:	PI002
Descripción Prueba:	Carga de CDR cobro prepago
Responsable:	D2
Prerrequisitos	
Deben existir archivos CDRs de cobro prepago en la ruta de recepción.	
Descripción de Casos de Prueba	
Caso: carga de archivo CDR a tabla en base HIVE.	
Entradas:	Salidas:
Archivos CDR: GDR_GPR_20201127_0959_2731.cdr GDR_GPP_20201127_1000_2732.cdr	Tabla Hive: huawei_gdr
Componente (s) Probados:	
Flujo NIFI CDRs cobro prepago	
Instrucciones de Prueba	
<ol style="list-style-type: none"> Colocar archivo en ruta de recepción de los CDRs de cobro prepago. Encender flujo NIFI. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> Si los datos del CDR se encuentran en la tabla Hive Huawei_gdr. Si los archivos de entrada con movidos a la ruta de archivos procesados. 	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 63. Caso de prueba PI003 – carga de CDRs de consumo pgwcdr

Información General	
Identificador de caso de uso:	PI003
Descripción Prueba:	Carga de CDR consumo pgwcdr
Responsable:	D2
Prerrequisitos	
Deben existir archivos CDRs de consumo pgwcdr en la ruta de recepción.	
Descripción de Casos de Prueba	
Caso: carga de archivo CDR a tabla en base HIVE.	
Entradas:	Salidas:
Archivos CDR: PGWGYE_INT2020120415130345 014190.dat PGWGYE_INT2020120415130445 014191.dat	Tabla Hive: gdr_charging_pgwcdr gdr_pgwcdr_listofservicedata
Componente (s) Probados:	
Flujo NIFI CDRs consumo pgwcdr	
Instrucciones de Prueba	
<ol style="list-style-type: none"> Colocar archivo en ruta de recepción de los CDRs de consumo pgwcdr. Encender flujo NIFI. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> Si los datos del CDR se encuentran en la tabla Hive gdr_charging_pgwcdr y gdr_pgwcdr_listofservicedata. 	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 64. Caso de prueba PI004 – carga de CDRs de consumo egcdr

Información General	
Identificador de caso de uso:	PI004
Descripción Prueba:	Carga de CDR consumo egcdr
Responsable:	D2
Prerrequisitos	
Deben existir archivos CDRs de consumo egcdr en la ruta de recepción.	
Descripción de Casos de Prueba	
Caso: carga de archivo CDR a tabla en base HIVE.	
Entradas: Archivos CDR: EGCDR_UIO01CORPORATEAPN 01679530.dat.proc EGCDR_GYE01CORPORATEAP N01205930.dat.proc	Salidas: Tabla Hive: gdr_charging_egcdr gdr_egcdr_listofservicedata
Componente (s) Probados: Flujo NIFI CDRs consumo egcdr	
Instrucciones de Prueba <ol style="list-style-type: none"> Colocar archivo en ruta de recepción de los CDRs de consumo pgwcdcr. Encender flujo NIFI. 	
Criterios de Aceptación <ol style="list-style-type: none"> Si los datos del CDR se encuentran en la tabla Hive gdr_charging_egcdr y gdr_egcdr_listofservicedata. Si los archivos de entrada con movidos a la ruta de archivos procesados. 	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 65. Caso de prueba PI005 – decodificador asn1

Información General	
Identificador de caso de uso:	PI005
Descripción Prueba:	Decodificación de archivo binario asn1
Responsable:	D2
Prerrequisitos	
Deben existir archivos CDRs de consumo egcdr en la ruta de recepción.	
Descripción de Casos de Prueba	
Caso: carga de archivo CDR a tabla en base HIVE.	
Entradas:	Salidas:
Archivos CDR: EGCDR_UIO01CORPORATEAPN 01679530.dat.proc EGCDR_GYE01CORPORATEAP N01205930.dat.proc	Archivos deodificados: EGCDR_UIO01CORPORATEAPN 01679530.dat.cdr EGCDR_GYE01CORPORATEAP N01205930.dat.cdr
Componente (s) Probados: Decodificador de CDRs asn1	
Instrucciones de Prueba	
<ol style="list-style-type: none"> Colocar archivo en ruta de recepción de los CDRs de consumo pgwcd. r. Encender flujo NIFI de decodificación. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> Si el archivo CDR se decodifica y se mueve a la ruta de decodificados. 	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 66. Caso de prueba PI006 – carga de clientes prepagos

Información General	
Identificador de caso de uso:	PI006
Descripción Prueba:	Carga datos clientes prepago
Responsable:	D2
Prerrequisitos	
Deben existir conexión a base de datos Oracle de clientes prepagos.	
Descripción de Casos de Prueba	
Caso: carga de tabla de clientes en base HIVE.	
Entradas: Tabla: INF_SUBSCRIBER INF_SUBSCRIBER_HIS	Salidas: Tablas Hive: INF_SUBSCRIBER INF_SUBSCRIBER_HIS
Componente (s) Probados: Sqoop para carga de clientes prepagos	
Instrucciones de Prueba 1. Ejecutar script sqoop.	
Criterios de Aceptación 1. Si la cantidad de registros de las tablas fuentes son iguales a la cantidad de registros en Hive. 2. Si la cantidad de clientes activos es igual en origen y destino.	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Tabla 67. Caso de prueba PI007 – carga de clientes post pago

Información General	
Identificador de caso de uso:	PI007
Descripción Prueba:	Carga datos clientes post pago
Responsable:	D2
Prerrequisitos	
Deben existir conexión a base de datos Oracle de clientes post pago.	
Descripción de Casos de Prueba	
Caso: carga de tabla de clientes en base HIVE.	
Entradas:	Salidas:
Tabla: SERVICIOS	Tablas Hive: SERVICIOS
Componente (s) Probados: Sqoop para carga de clientes post pago	
Instrucciones de Prueba 1. Ejecutar script sqoop.	
Criterios de Aceptación 1. Si la cantidad de registros de las tablas fuentes son iguales a la cantidad de registros en Hive. 2. Si la cantidad de clientes activos es igual en origen y destino.	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

5.5 Pruebas de usuario

Para evaluar la funcionalidad del proceso implementado se realizarán los siguientes casos de pruebas.

Tabla 68. Caso de prueba PU001 – generación de reporte

Información General	
Identificador de caso de uso:	PU001
Descripción Prueba:	Generación de reporte de tráfico no cobrado
Responsable:	D2
Prerrequisitos	
Se debieron ejecutar los casos de pruebas PI006 y PI007.	
Descripción de Casos de Prueba	
Caso: Generación de reporte.	
Entradas:	Salidas:
Parámetros: Fecha proceso Correo destinatario	Archivo csv
Componente (s) Probados: Proceso detección de tráfico no cobrado	
Instrucciones de Prueba 1. Iniciar flujo NIFI de tráfico no cobrado.	
Criterios de Aceptación 1. Si llega al mail destinatario con archivo csv. 2. Si al evaluar casos reportados en los archivos se confirman las diferencias. 3. Si no se presentan errores al ejecutar proceso.	
Fecha de creación: 01/12/2020	
Revisiones y observaciones:	

Para evaluar rendimiento del proceso implementado se realizarán los siguientes casos de pruebas.

Tabla 69. Caso de prueba PR001 – rendimiento de carga CDR cobro prepago

ID Test:	PR001	Tipo: Carga																								
Nombre de la prueba																										
Carga de CDR cobro prepago																										
FLUJO:																										
Parámetros entrada:	Archivos CDRs																									
Parámetros de la prueba																										
<table border="1"> <thead> <tr> <th style="text-align: center;">ARCHIVO CDR</th> <th style="text-align: center;">PESO MB</th> </tr> </thead> <tbody> <tr> <td>GDR_GPR_20201127_0959_2731.cdr</td> <td>201M</td> </tr> <tr> <td>GDR_GPP_20201127_1000_2732.cdr</td> <td>139M</td> </tr> <tr> <td>GDR_GPR_20201127_1004_2733.cdr</td> <td>73M</td> </tr> <tr> <td>GDR_GPP_20201127_1004_2734.cdr</td> <td>58M</td> </tr> <tr> <td>GDR_GPR_20201127_1009_2735.cdr</td> <td>327M</td> </tr> <tr> <td>GDR_GPP_20201127_1011_2736.cdr</td> <td>206M</td> </tr> <tr> <td>GDR_GPR_20201127_1014_2737.cdr</td> <td>178M</td> </tr> <tr> <td>GDR_GPP_20201127_1015_2738.cdr</td> <td>128M</td> </tr> <tr> <td>GDR_GPR_20201127_1019_2739.cdr</td> <td>215M</td> </tr> <tr> <td>GDR_GPR_20201127_0959_2731.cdr</td> <td>201M</td> </tr> <tr> <td>GDR_GPP_20201127_1000_2732.cdr</td> <td>139M</td> </tr> </tbody> </table>			ARCHIVO CDR	PESO MB	GDR_GPR_20201127_0959_2731.cdr	201M	GDR_GPP_20201127_1000_2732.cdr	139M	GDR_GPR_20201127_1004_2733.cdr	73M	GDR_GPP_20201127_1004_2734.cdr	58M	GDR_GPR_20201127_1009_2735.cdr	327M	GDR_GPP_20201127_1011_2736.cdr	206M	GDR_GPR_20201127_1014_2737.cdr	178M	GDR_GPP_20201127_1015_2738.cdr	128M	GDR_GPR_20201127_1019_2739.cdr	215M	GDR_GPR_20201127_0959_2731.cdr	201M	GDR_GPP_20201127_1000_2732.cdr	139M
ARCHIVO CDR	PESO MB																									
GDR_GPR_20201127_0959_2731.cdr	201M																									
GDR_GPP_20201127_1000_2732.cdr	139M																									
GDR_GPR_20201127_1004_2733.cdr	73M																									
GDR_GPP_20201127_1004_2734.cdr	58M																									
GDR_GPR_20201127_1009_2735.cdr	327M																									
GDR_GPP_20201127_1011_2736.cdr	206M																									
GDR_GPR_20201127_1014_2737.cdr	178M																									
GDR_GPP_20201127_1015_2738.cdr	128M																									
GDR_GPR_20201127_1019_2739.cdr	215M																									
GDR_GPR_20201127_0959_2731.cdr	201M																									
GDR_GPP_20201127_1000_2732.cdr	139M																									
Datos del equipo de prueba																										
<ul style="list-style-type: none"> • Nodo 2 HDF, 10 cores • 100 GB RAM • REDHAT 7 																										

Tabla 70. Caso de prueba PR002 – rendimiento de carga CDR cobro post pago

ID Test:	PR002	Tipo: Carga																								
Nombre de la prueba																										
Carga de CDR cobro post pago																										
FLUJO:																										
Parámetros entrada:	Archivos CDRs																									
Parámetros de la prueba																										
<table border="1"> <thead> <tr> <th style="text-align: center;">ARCHIVO CDR</th> <th style="text-align: center;">PESO KB</th> </tr> </thead> <tbody> <tr> <td>20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr</td> <td>36K</td> </tr> <tr> <td>20201201_100036_pstp_gprs_76_1_archive.cdr_cOk.cdr</td> <td>33K</td> </tr> <tr> <td>20201201_100036_pstp_gprs_75_1_archive.cdr_cOk.cdr</td> <td>48K</td> </tr> <tr> <td>20201201_100036_pstp_gprs_74_1_archive.cdr_cOk.cdr</td> <td>45K</td> </tr> <tr> <td>20201201_100037_pstp_gprs_79_1_archive.cdr_cOk.cdr</td> <td>40K</td> </tr> <tr> <td>20201201_100036_pstp_gprs_72_1_archive.cdr_cOk.cdr</td> <td>41K</td> </tr> <tr> <td>20201201_100037_pstp_gprs_71_1_archive.cdr_cOk.cdr</td> <td>39K</td> </tr> <tr> <td>20201201_100037_pstp_gprs_73_1_archive.cdr_cOk.cdr</td> <td>38K</td> </tr> <tr> <td>20201201_100124_pstp_gprs_113_3_archive.cdr_cOk.cdr</td> <td>56K</td> </tr> <tr> <td>20201201_100124_pstp_gprs_112_3_archive.cdr_cOk.cdr</td> <td>54K</td> </tr> <tr> <td>20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr</td> <td>36K</td> </tr> </tbody> </table>			ARCHIVO CDR	PESO KB	20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr	36K	20201201_100036_pstp_gprs_76_1_archive.cdr_cOk.cdr	33K	20201201_100036_pstp_gprs_75_1_archive.cdr_cOk.cdr	48K	20201201_100036_pstp_gprs_74_1_archive.cdr_cOk.cdr	45K	20201201_100037_pstp_gprs_79_1_archive.cdr_cOk.cdr	40K	20201201_100036_pstp_gprs_72_1_archive.cdr_cOk.cdr	41K	20201201_100037_pstp_gprs_71_1_archive.cdr_cOk.cdr	39K	20201201_100037_pstp_gprs_73_1_archive.cdr_cOk.cdr	38K	20201201_100124_pstp_gprs_113_3_archive.cdr_cOk.cdr	56K	20201201_100124_pstp_gprs_112_3_archive.cdr_cOk.cdr	54K	20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr	36K
ARCHIVO CDR	PESO KB																									
20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr	36K																									
20201201_100036_pstp_gprs_76_1_archive.cdr_cOk.cdr	33K																									
20201201_100036_pstp_gprs_75_1_archive.cdr_cOk.cdr	48K																									
20201201_100036_pstp_gprs_74_1_archive.cdr_cOk.cdr	45K																									
20201201_100037_pstp_gprs_79_1_archive.cdr_cOk.cdr	40K																									
20201201_100036_pstp_gprs_72_1_archive.cdr_cOk.cdr	41K																									
20201201_100037_pstp_gprs_71_1_archive.cdr_cOk.cdr	39K																									
20201201_100037_pstp_gprs_73_1_archive.cdr_cOk.cdr	38K																									
20201201_100124_pstp_gprs_113_3_archive.cdr_cOk.cdr	56K																									
20201201_100124_pstp_gprs_112_3_archive.cdr_cOk.cdr	54K																									
20201201_100036_pstp_gprs_77_1_archive.cdr_cOk.cdr	36K																									
Datos del equipo de prueba																										
<ul style="list-style-type: none"> • Nodo 3 HDF, 10 cores • 100 GB RAM • REDHAT 7 																										

Tabla 71. Caso de prueba PR003 – rendimiento de carga CDR consumo pgwcd

ID Test:	PR003	Tipo: Carga																								
Nombre de la prueba																										
Carga de CDR consumo pgwcd																										
FLUJO:																										
Parámetros entrada:	Archivos CDRs																									
Parámetros de la prueba																										
<table border="1"> <thead> <tr> <th>ARCHIVO CDR</th> <th>PESO MB</th> </tr> </thead> <tbody> <tr> <td>PGWGYE_INT2020120416131745017789.dat</td> <td>1.1M</td> </tr> <tr> <td>PGWGYE_INT2020120416131945017790.dat</td> <td>1.2M</td> </tr> <tr> <td>PGWGYE_INT2020120416132045017791.dat</td> <td>1.1M</td> </tr> <tr> <td>PGWGYE_INT2020120416132045017792.dat</td> <td>1.2M</td> </tr> <tr> <td>PGWGYE_INT22020120416364800056081.dat</td> <td>14K</td> </tr> <tr> <td>PGWGYE_CORP2020120416313200082518.dat</td> <td>1015K</td> </tr> <tr> <td>PGWGYE_CORP2020120416372400082519.dat</td> <td>1013K</td> </tr> <tr> <td>PGWGYE_CORP2020120416492500082521.dat</td> <td>1014K</td> </tr> <tr> <td>PGWGYE_CORP2020120416552600082522.dat</td> <td>1013K</td> </tr> <tr> <td>PGWGYE_CORP2020120416433600082520.dat</td> <td>1015K</td> </tr> <tr> <td>PGWGYE_INT2020120416131745017789.dat</td> <td>1.1M</td> </tr> </tbody> </table>			ARCHIVO CDR	PESO MB	PGWGYE_INT2020120416131745017789.dat	1.1M	PGWGYE_INT2020120416131945017790.dat	1.2M	PGWGYE_INT2020120416132045017791.dat	1.1M	PGWGYE_INT2020120416132045017792.dat	1.2M	PGWGYE_INT22020120416364800056081.dat	14K	PGWGYE_CORP2020120416313200082518.dat	1015K	PGWGYE_CORP2020120416372400082519.dat	1013K	PGWGYE_CORP2020120416492500082521.dat	1014K	PGWGYE_CORP2020120416552600082522.dat	1013K	PGWGYE_CORP2020120416433600082520.dat	1015K	PGWGYE_INT2020120416131745017789.dat	1.1M
ARCHIVO CDR	PESO MB																									
PGWGYE_INT2020120416131745017789.dat	1.1M																									
PGWGYE_INT2020120416131945017790.dat	1.2M																									
PGWGYE_INT2020120416132045017791.dat	1.1M																									
PGWGYE_INT2020120416132045017792.dat	1.2M																									
PGWGYE_INT22020120416364800056081.dat	14K																									
PGWGYE_CORP2020120416313200082518.dat	1015K																									
PGWGYE_CORP2020120416372400082519.dat	1013K																									
PGWGYE_CORP2020120416492500082521.dat	1014K																									
PGWGYE_CORP2020120416552600082522.dat	1013K																									
PGWGYE_CORP2020120416433600082520.dat	1015K																									
PGWGYE_INT2020120416131745017789.dat	1.1M																									
Datos del equipo de prueba																										
<ul style="list-style-type: none"> • Nodo 1 HDF, 10 cores • 100 GB RAM • REDHAT 7 																										

Tabla 72. Caso de prueba PR004 – rendimiento de carga CDR consumo egcdr

ID Test:	PR004	Tipo: Carga																								
Nombre de la prueba																										
Carga de CDR consumo egcdr																										
FLUJO:																										
Parámetros entrada:	Archivos CDRs																									
Parámetros de la prueba																										
<table border="1"> <thead> <tr> <th style="text-align: center;">ARCHIVO CDR</th> <th style="text-align: center;">PESO MB</th> </tr> </thead> <tbody> <tr> <td>EGCDR_UIO01CORPORATEAPN01683574.dat.proc</td> <td>1.1M</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683573.dat.proc</td> <td>1021K</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683575.dat.proc</td> <td>1022K</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683576.dat.proc</td> <td>1.1M</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683577.dat.proc</td> <td>1018K</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683578.dat.proc</td> <td>1.1M</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683579.dat.proc</td> <td>1.1M</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683580.dat.proc</td> <td>1.1M</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683581.dat.proc</td> <td>1022K</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683582.dat.proc</td> <td>1021K</td> </tr> <tr> <td>EGCDR_UIO01CORPORATEAPN01683574.dat.proc</td> <td>1.1M</td> </tr> </tbody> </table>			ARCHIVO CDR	PESO MB	EGCDR_UIO01CORPORATEAPN01683574.dat.proc	1.1M	EGCDR_UIO01CORPORATEAPN01683573.dat.proc	1021K	EGCDR_UIO01CORPORATEAPN01683575.dat.proc	1022K	EGCDR_UIO01CORPORATEAPN01683576.dat.proc	1.1M	EGCDR_UIO01CORPORATEAPN01683577.dat.proc	1018K	EGCDR_UIO01CORPORATEAPN01683578.dat.proc	1.1M	EGCDR_UIO01CORPORATEAPN01683579.dat.proc	1.1M	EGCDR_UIO01CORPORATEAPN01683580.dat.proc	1.1M	EGCDR_UIO01CORPORATEAPN01683581.dat.proc	1022K	EGCDR_UIO01CORPORATEAPN01683582.dat.proc	1021K	EGCDR_UIO01CORPORATEAPN01683574.dat.proc	1.1M
ARCHIVO CDR	PESO MB																									
EGCDR_UIO01CORPORATEAPN01683574.dat.proc	1.1M																									
EGCDR_UIO01CORPORATEAPN01683573.dat.proc	1021K																									
EGCDR_UIO01CORPORATEAPN01683575.dat.proc	1022K																									
EGCDR_UIO01CORPORATEAPN01683576.dat.proc	1.1M																									
EGCDR_UIO01CORPORATEAPN01683577.dat.proc	1018K																									
EGCDR_UIO01CORPORATEAPN01683578.dat.proc	1.1M																									
EGCDR_UIO01CORPORATEAPN01683579.dat.proc	1.1M																									
EGCDR_UIO01CORPORATEAPN01683580.dat.proc	1.1M																									
EGCDR_UIO01CORPORATEAPN01683581.dat.proc	1022K																									
EGCDR_UIO01CORPORATEAPN01683582.dat.proc	1021K																									
EGCDR_UIO01CORPORATEAPN01683574.dat.proc	1.1M																									
Datos del equipo de prueba																										
<ul style="list-style-type: none"> • Nodo 1 HDF, 10 cores • 100 GB RAM • REDHAT 7 																										

Tabla 73. Caso de prueba PR005 – rendimiento del decodificar asn1

ID Test:	PR005	Tipo: Carga																								
Nombre de la prueba																										
Decodificador asn1																										
FLUJO:																										
Parámetros entrada:	Archivos CDRs																									
Parámetros de la prueba																										
<table border="1"> <thead> <tr> <th>ARCHIVO CDR</th> <th>PESO MB</th> </tr> </thead> <tbody> <tr> <td>PGWGYE_INT2020120416131745017789.dat</td> <td>1.1M</td> </tr> <tr> <td>PGWGYE_INT2020120416131945017790.dat</td> <td>1.2M</td> </tr> <tr> <td>PGWGYE_INT2020120416132045017791.dat</td> <td>1.1M</td> </tr> <tr> <td>PGWGYE_INT2020120416132045017792.dat</td> <td>1.2M</td> </tr> <tr> <td>PGWGYE_INT22020120416364800056081.dat</td> <td>14K</td> </tr> <tr> <td>PGWGYE_CORP2020120416313200082518.dat</td> <td>1015K</td> </tr> <tr> <td>PGWGYE_CORP2020120416372400082519.dat</td> <td>1013K</td> </tr> <tr> <td>PGWGYE_CORP2020120416492500082521.dat</td> <td>1014K</td> </tr> <tr> <td>PGWGYE_CORP2020120416552600082522.dat</td> <td>1013K</td> </tr> <tr> <td>PGWGYE_CORP2020120416433600082520.dat</td> <td>1015K</td> </tr> <tr> <td>PGWGYE_INT2020120416131745017789.dat</td> <td>1.1M</td> </tr> </tbody> </table>			ARCHIVO CDR	PESO MB	PGWGYE_INT2020120416131745017789.dat	1.1M	PGWGYE_INT2020120416131945017790.dat	1.2M	PGWGYE_INT2020120416132045017791.dat	1.1M	PGWGYE_INT2020120416132045017792.dat	1.2M	PGWGYE_INT22020120416364800056081.dat	14K	PGWGYE_CORP2020120416313200082518.dat	1015K	PGWGYE_CORP2020120416372400082519.dat	1013K	PGWGYE_CORP2020120416492500082521.dat	1014K	PGWGYE_CORP2020120416552600082522.dat	1013K	PGWGYE_CORP2020120416433600082520.dat	1015K	PGWGYE_INT2020120416131745017789.dat	1.1M
ARCHIVO CDR	PESO MB																									
PGWGYE_INT2020120416131745017789.dat	1.1M																									
PGWGYE_INT2020120416131945017790.dat	1.2M																									
PGWGYE_INT2020120416132045017791.dat	1.1M																									
PGWGYE_INT2020120416132045017792.dat	1.2M																									
PGWGYE_INT22020120416364800056081.dat	14K																									
PGWGYE_CORP2020120416313200082518.dat	1015K																									
PGWGYE_CORP2020120416372400082519.dat	1013K																									
PGWGYE_CORP2020120416492500082521.dat	1014K																									
PGWGYE_CORP2020120416552600082522.dat	1013K																									
PGWGYE_CORP2020120416433600082520.dat	1015K																									
PGWGYE_INT2020120416131745017789.dat	1.1M																									
Datos del equipo de prueba																										
<ul style="list-style-type: none"> • Nodo 1 HDF, 10 cores • 100 GB RAM • REDHAT 7 																										

Tabla 74. Caso de prueba PR006 – rendimiento del proceso de tráfico

ID Test:	PR006	Tipo: Carga
Nombre de la prueba		
Detección de tráfico no cobrado		
FLUJO:		
Parámetros entrada:	Fecha de carga	
Parámetros de la prueba		
<ul style="list-style-type: none"> • 20201201 		
Datos del equipo de prueba		
<ul style="list-style-type: none"> • Cluster HDP • 4 nodos de datos • 4 nodos de trabajo 		

5.6 Análisis de los resultados de pruebas

En promedio durante un día el clúster HDF está cargando 700 millones de registros a Hive, se procesan 308 mil archivos con 312 GB en y se decodifican 100 mil archivos, el tiempo máximo que tarda el CDR para ser cargado en HIVE y estar disponible para su análisis es de 4 minutos.

Tabla 75. Métricas de las cargas de archivos CDRs

TIPO CDR	PROMEDIO ARCHIVOS DIA	GIGAS PROMEDIO DIA	REGISTRO DIA	TIEMPO PROMEDIO TRASFORMACION (DECODIFICACION)	TIEMPO PROMEDIO CARGA ARCHIVO A HIVE
COBROS PREPAGO	6000	100 GB	134201669	-	3 min
COBROS POST PAGO	200000	70 GB	250568895	-	3 min
CONSUMO PGWCDR	100000	140 GB	348325345	3 seg	4 min
CONSUMO EGCDR	2600	2 GB	5374862	2 seg	4 min

La carga de fuentes de clientes hacia hadoop toma en promedio 20 minutos, se extraen 150 GB y 100 millones de registros.

Tabla 76. Métricas de las cargas tablas de clientes

FUENTE	GIGAS PROMEDIO DIA	REGISTROS DIARIOS	TIEMPO PROMEDIO CARGA A HIVE
CLIENTES PREPAGO	40 GB	30114981	15 min
CLIENTES PREPAGO HISTORIA	20 GB	15806581	10 min
CLIENTES POST PAGO	90 GB	95360951	20 min

El proceso masivo para detectar tráfico no cobrado diariamente está tomando 50 minutos, se están reportando en promedio 5000 casos al día.

Tabla 77. Métricas de ejecución del proceso de tráfico no cobrado

TIEMPO TOTAL	55 minutos
RECURSOS DEL CLÚSTER - MEMORIA RAM	80 GB
RECURSOS DEL CLÚSTER - VCORES	30 VCORES
TIEMPO INICIO	20/12/01 06:30:04
TIEMPO FIN	20/12/01 07:25:34

Según análisis previo, y resultado de pruebas satisfactorias, se puede concluir que las pruebas reúnen parámetros aceptables para el proceso masivo de datos no cobrados desarrollado en la plataforma Hadoop, se logra tener el reporte de tráfico no cobrado a cualquier hora del día, con tiempo de duración no mayor a 1 hora y media.

Tabla 78. Comparativa del nuevo proceso en Hadoop y el actual proceso

	NUEVO PROCESO CLÚSTER HADOOP	ACTUAL PROCEOS 2 SERVIDORES
TIEMPO DECODIFICACION Y CARGA CDRs	Carga en línea.	1 a 4 días (1 día con 2 equipos full)
TIEMPO CARGA TABLAS CLIENTES	25 minutos	25 minutos
RECURSOS - MEMORIA RAM	80 GB (Del clúster HDP)	50 a 200 GB (200 GB con 2 equipos full)
RECURSOS - VCORES	30 VCORES (Del clúster HDP)	25 a 100 Cores (100 cores con 2 equipos full)
TIEMPO PROCESO TRAFICO NO COBRADO	55 minutos	2 horas
TIEMPO TOTAL PROCESO	1 hora 20 minutos	De 1 a 4 días

Si se compara el nuevo proceso en Hadoop y el actual proceso se notan mejoras principalmente en los tiempos de cargas y decodificación de CDRs, los tiempos de cargas de tablas de bases de datos son iguales y también se tienen mejoras en el tiempo de ejecución del proceso que detecta las inconsistencias del tráfico de datos.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

En el ámbito empresarial donde se generan gran cantidad y variedad de datos es necesario implementar procesos en herramientas Big Data que ayuden a transformar estos datos en información relevante para la organización.

1. El propósito de contar con un reporte diario que permita detectar inconsistencia en los cobros se consigue diseñando un proceso que en tiempo real toma datos de CDRs, los transforman y los dejan disponibles para su consumo, esto ahorra tiempo y trabajo, además de que se construye sobre una plataforma escalable y de procesamiento distribuido.
2. El proceso desarrollado permite procesar 700 millones de registros de CDRs de manera distribuida, ocupando un 20% de carga en el clúster HDF y un 30% de carga de trabajo del clúster HDP durante 1 hora 20 minutos de su ejecución lo que hace viable al proceso sin causar mayor impacto a la plataforma.

RECOMENDACIONES

1. Hay que considerar que al momento de la implementación del proceso el clúster Big Data tiene la carga del 30%, en la medida que se creen nuevos procesos los tiempos de ejecución del proceso de tráfico no cobrado pueden aumentar.
2. Considerar aumentar los recursos al clúster si se requiere que el reporte diario tome menos tiempo en ejecutarse.
3. Si en algún momento el proceso desarrollado disputa recursos del clúster con otros procesos se puede definir una cola de trabajo exclusiva en YARN, de esta forma los recursos definidos en su cola de trabajo solo estarán disponibles para el proceso de tráfico no cobrado.
4. Cuando se tengan nuevos layouts de los CDRs de cobro o de consumo, se deberá considerar actualizar el decodificador y estructuras creadas en hive.
5. Se debe evaluar periódicamente el crecimiento de los datos en Hive y HDFS, si estos datos sobrepasan el 80% del total, se puede afectar el rendimiento de los procesos.

BIBLIOGRAFIA

- [1] Luis Joyanes Aguilar, *Big Data, Análisis de grandes volúmenes de datos en organizaciones*. 2013.
- [2] K. Mayer-Schönberger, Viktor & Cukier, *Big Data. La revolución de los datos masivos*. 2013.
- [3] R. Barranco Fragoso, “¿Qué es Big Data?,” *IBM Dev.*, 2012.
- [4] A. A. Academia de Studii Economice (Romania), Tole, and A. Adrian, *Database systems journal*. 2013.
- [5] Vladimir Kaplarevic, “Apache Hadoop Architecture Explained (In-Depth Overview).” <https://phoenixnap.com/kb/apache-hadoop-architecture-explained> (accessed Dec. 11, 2020).
- [6] Dell EMC Solutions, “Hortonworks Hadoop 3.0 Architecture Guide,” *Ready Archit. Hortonworks Hadoop 3.0*, pp. 1–56, Dec. 2019.
- [7] A. E. Martín, S. B. Chávez, N. R. Rodríguez, and M. A. Murazzo, “NoSql en sistemas distribuidos sobre Cluster Hadoop,” *Objeto Conf.*, 2016, doi: 10.1080/13880290590913822.
- [8] C. Singh and R. Singh, “Enhancing Performance and Fault Tolerance of Hadoop cluster,” *Int. Res. J. Eng. Technol.*, 2017.
- [9] A. Meier, M. Kaufmann, A. Meier, and M. Kaufmann, “NoSQL

Databases,” in *SQL & NoSQL Databases*, 2019.

- [10] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, “Persisting big-data: The NoSQL landscape,” *Inf. Syst.*, 2017, doi: 10.1016/j.is.2016.07.009.
- [11] L. . Greeshma and G. Pradeepini, “Big Data Analytics with Apache Hadoop MapReduce Framework,” *Indian J. Sci. Technol.*, 2016, doi: 10.17485/ijst/2016/v9i26/93418.
- [12] T. H. Sardar and Z. Ansari, “Partition based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions,” *Futur. Comput. Informatics J.*, 2018, doi: 10.1016/j.fcij.2018.06.002.
- [13] R. Young, S. Fallon, and P. Jacob, “An Architecture for Intelligent Data Processing on IoT Edge Devices,” 2018, doi: 10.1109/UKSim.2017.19.
- [14] H. Isah and F. Zulkernine, “A Scalable and Robust Framework for Data Stream Ingestion,” 2019, doi: 10.1109/BigData.2018.8622360.

GLOSARIO

Clúster. El término clúster se aplica a los sistemas distribuidos de granjas de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen un único servidor.

Nodo. Un nodo de clúster es un equipo que en conjunto con otros equipos forman un clúster.

ASN1. Es una norma para representar datos independientemente de la máquina que se esté usando y sus formas de representación internas.

NIFI. Apache Nifi es un proyecto de software de Apache Software Foundation diseñado para automatizar el flujo de datos entre sistemas de software.

HADOOP. Apache Hadoop es un entorno de trabajo para software, bajo licencia libre, para programar aplicaciones distribuidas que manejen grandes volúmenes de datos. Permite a las aplicaciones trabajar con miles de nodos en red y petabytes de datos.

HIVE. Apache Hive es una infraestructura de almacenamiento de datos construida sobre Hadoop para proporcionar agrupación, consulta, y análisis de datos.

SQOOP. Sqoop es una aplicación con interfaz de línea de comando para transferir datos entre bases de datos relacionales y Hadoop.