



**UNIVERSIDAD
REY JUAN CARLOS**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso Académico 2009/2010

Proyecto de Fin de Carrera

**ANÁLISIS DE CONTENIDOS WEB MEDIANTE
INFORMES DINÁMICOS**

Autor: Eduardo González Servant

Tutor: Javier Martínez Moguerza (URJC)

RESUMEN

Este proyecto pretende mostrarnos como se ha llevado a cabo la implantación de un sistema de procesamiento y análisis de contenido web.

En concreto, este desarrollo ha sido implementado en el departamento de informática de una determinada empresa, cuya necesidad era poder analizar datos que aportasen información sobre las descargas de los diferentes vídeos que se están sirviendo desde su plataforma web.

Para dicha tarea se presentó un piloto de prueba, con el objetivo de ofrecer la posibilidad de realizar análisis mediante informes dinámicos, usando para ello la herramienta QlikView, cuyas características principales comentaremos posteriormente.

Tras la elección de QlikView como herramienta de Business Intelligence por parte del cliente, se definieron algunos requisitos que debería recoger el informe de Descargas de Vídeos. Dicho informe debería contener gran cantidad de información para poder analizarla con un alto nivel de detalle, disponer de algún tipo de recarga automática diaria, y mantener un histórico de datos lo más amplio posible.

Los resultados mostrados en nuestro informe se están usando actualmente para comprobar la calidad y éxito del servicio, así como para estudiar los costes originados.

Posteriormente se estudiarán las diferentes fases que componen el proyecto de análisis de contenido web:

- Obtención, Procesamiento y Carga de datos.
- Diseño de informes.
- Análisis de Datos.

Además se mostrarán algunas imágenes del aspecto de este tipo de informes dinámicos, y se realizarán algunos ejemplos prácticos de consultas dinámicas sobre dichos informes.

RESUMEN.....	2
1.- INTRODUCCIÓN.....	1
1.1.- ¿QUÉ ES BUSINESS INTELLIGENCE?	1
1.2.- SOBRE QLIKVIEW	1
1.3.- SOBRE TALEND	3
2.- OBJETIVOS.....	5
2.1.- DESCRIPCIÓN DEL PROBLEMA	5
2.2.- ESTUDIO DE ALTERNATIVAS.....	5
2.3.- METODOLOGÍA EMPLEADA	6
3.- DESCRIPCIÓN DEL PROYECTO.....	6
4.- FASES DEL PROYECTO.....	8
4.1.- EXTRACCIÓN TRANSFORMACIÓN Y CARGA DE DATOS	8
4.1.1.- <i>Logs de la CDN</i>	8
4.1.2.- <i>Procesos de Extracción, Transformación y Carga de Datos</i>	11
4.1.2.1.- Proceso de descarga y tratamiento de logs	12
4.1.2.2.- Proceso de gestión y planificación de recarga	15
4.1.3.- <i>Creación de ficheros .qvd</i>	17
4.1.3.1.- Recarga de ficheros .qvd del mes anterior.....	19
4.1.3.2.- Recarga de ficheros .qvd del mes actual	19
4.1.4.- <i>Planificación diaria de ejecución de los procesos</i>	20
4.2.- DISEÑO DE INFORMES DINÁMICOS	21
4.2.1.- <i>Script de recarga y creación del modelo de datos</i>	21
4.2.2.- <i>Creación y Diseño de los Gráficos</i>	23
4.3.- ANÁLISIS DE INFORMES DINÁMICOS.....	27
4.3.1.- <i>Entorno para el análisis de los Informes</i>	27
4.3.2.- <i>Acceso a los informes</i>	28
4.3.3.- <i>Análisis de la Información</i>	30
4.3.3.1.- Interacción con los Informes Dinámicos	30
4.3.3.2.- Ejemplos de Análisis	33
5.- DEFINICIONES.....	38
6.- CONCLUSIONES.....	39
7.- BIBLIOGRAFÍA	40
ANEXO.....	41

1.- INTRODUCCIÓN

1.1.- ¿QUÉ ES BUSINESS INTELLIGENCE?

Podría decirse que Business Intelligence, o Inteligencia de Negocio, es la tarea que consiste en interpretar información para la toma de decisiones. Para ello es importante disponer de algún medio que nos muestre claramente dicha información, y mediante el cual podamos sacar diferentes conclusiones.

El análisis y toma de decisiones, sería el último paso del largo camino que forma parte del Business Intelligence, ya que previamente hemos tenido que recoger todos los datos útiles posibles, procesarlos, y mostrar su información de la mejor manera posible para que finalmente sea analizada.

1.2.- SOBRE QLIKVIEW

QlikView es una herramienta de Business Intelligence, mediante la cual podemos crear informes con los que podemos interactuar.

Algunas de sus características más importantes son:

- Simplicidad de uso y aprendizaje.
- Resultados dinámicos mediante interacción con el usuario.
- Reducción de tiempo en desarrollo y respuesta.
- Integrable con gran cantidad de aplicaciones.
- Permite realizar informes muy visibles y atractivos.

Cuando creamos un informe con QlikView, obtenemos un documento cuya extensión será .qvw, el cual no es más que es un archivo cuyo contenido tiene todo lo necesario para un análisis completo de los datos, es decir, estará compuesto por:

- Script de carga de datos.
- Definición de la estructura interna, o modelo de datos.
- Los datos que se van a analizar.

- El diseño y funcionalidad de los informes dinámicos.

La idea fundamental del funcionamiento interno de dichos documentos, es la Lógica Asociativa de QlikView “*La asociación refleja fielmente la forma en que funciona la mente humana: las personas no piensan de forma linear, sino en una dirección que se construye a partir de asociaciones que se van creando*”.

Para llevar a cabo la lógica asociativa se cargan en memoria los datos que forman parte de un documento QlikView, esto nos permite (de una forma muy rápida) mostrarnos los valores obtenidos tras aplicar las métricas necesarias, en función de la selección realizada por el usuario (Análisis en memoria).

Además QlikView nos permite obtener los datos desde gran cantidad de orígenes diferentes.



Para la inserción de los datos en los informes, QlikView dispone de una sencilla interfaz, mediante la cual podremos crear nuestro Script de recarga. En ella disponemos de varias opciones que nos facilitarán bastante el trabajo a la hora de crear nuestro script, e incluso nos permitirán realizar algunas transformaciones en los datos.

A través de los archivos QlikView, se puede distribuir información muy fácilmente. Los análisis no dependen de la ubicación de los datos originales o de las condiciones de red. Los archivos pueden ser p.ej. archivos adjuntos al correo electrónico enviados a otra persona que por cualquier causa técnica no ha podido entrar en la base de datos.

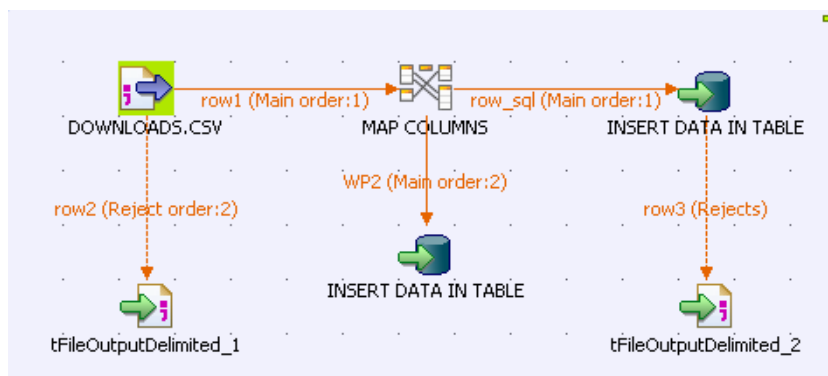
Otra característica importante que hemos usado en cada uno de los informes, es su rapidez a la hora de obtener los datos mediante ficheros .qvd. Dichos ficheros son una especie de XML, preparados para almacenar información comprimida que los informes de QlikView pueden procesar en muy poco tiempo (Entre 10-100 veces más rápido que leer desde cualquier otro origen).

1.3.- SOBRE TALEND

Talend es una herramienta ETL (Extraction, Transformation and Loading) Open Source, su propósito es ofrecer una fácil solución a la tarea de extraer datos desde diferentes orígenes, tratar información de múltiples maneras, y permitirnos almacenar dicha información.

Los desarrollos realizados mediante esta herramienta serán una especie de diagrama de flujo, compuesto por diferentes objetos diseñados para realizar alguna tarea concreta, las cuales podemos ir enlazando entre sí.

Dichos flujos de tareas se ejecutarán secuencialmente, y podremos planificar que secuencia seguirán, crear caminos de ejecución condicionales, e incluso crear bucles.



Contamos con la existencia de numerosos componentes que nos serán de gran utilidad a la hora de construir nuestro flujo de tareas, ya que nos permiten trabajar con numerosos objetos que realizan diferentes operaciones con: Bases de Datos, Ficheros XML, Ficheros .csv, Operaciones por FTP, Envío de mails, etc...

Talend internamente codificará nuestros desarrollos en Java, dicho código estará visible en la pestaña 'Code', mediante la cual podremos ver como nuestros desarrollos están siendo codificados e incluso analizar posibles errores.

A continuación mostraremos una imagen donde podemos ver un ejemplo del contenido en la pestaña anteriormente comentada.

```
Job velocix 1.0
229     tJava_1_onSubJobError(exception, globalMap);
230     }
231
232     public void tFileFetch_1_error(Exception exception,
233     final java.util.Map<String, Object> globalMap)
234     throws TalendException {
235     end_Hash.put("tFileFetch_1", System.currentTimeMillis());
236     tFileFetch_1_onSubJobError(exception, globalMap);
237     }
238
239     public void tFileUnarchive_1_error(Exception exception,
240     final java.util.Map<String, Object> globalMap)
241     throws TalendException {
242     end_Hash.put("tFileUnarchive_1", System.currentTimeMillis());
243     tFileUnarchive_1_onSubJobError(exception, globalMap);
244     }
245
246     public void tMysqlRow_1_error(Exception exception,
247     final java.util.Map<String, Object> globalMap)
248     throws TalendException {
249     end_Hash.put("tMysqlRow_1", System.currentTimeMillis());
250     tMysqlRow_1_onSubJobError(exception, globalMap);
251     }
252
253     public void tPostjob_1_error(Exception exception,
254     final java.util.Map<String, Object> globalMap)
255     throws TalendException {
256     end_Hash.put("tPostjob_1", System.currentTimeMillis());
257     tPostjob_1_onSubJobError(exception, globalMap);
258     }
259     }
260 }
```

Las principales características por las cuales se ha optado por Talend como herramienta para el desarrollo de este proyecto son:

- Su facilidad de uso a la hora de crear los procesos de carga, sin la necesidad de realizar complejos desarrollos en cualquier lenguaje de programación.
- Es una herramienta bastante intuitiva y visual, con lo que reduce el tiempo y la complejidad en el mantenimiento de los procesos.
- Es una herramienta OpenSource, que no requiere ningún tipo de licencia para su uso.

2.- OBJETIVOS

2.1.- DESCRIPCIÓN DEL PROBLEMA

La necesidad principal del cliente era disponer de un sistema de análisis de datos, cuyo objetivo prioritario es poder analizar en detalle diversa información sobre las Descargas de Vídeos desde la web.

Para satisfacer las necesidades del cliente, el proyecto debe cumplir los siguientes requisitos:

1. Usar QlikView, como herramienta de análisis de datos.
2. Poder realizar consultas sobre los datos con el menor tiempo de respuesta posible.
3. Almacenar un Histórico de datos y poder realizar consultas sobre el.
4. Poder cruzar datos desde diferentes orígenes, tales como: Páginas Web, Documentos de texto plano, Bases de Datos, etc...
5. Actualización diaria de los datos, y así poder consultar los datos del día anterior.
6. Disponer de una Interfaz de fácil uso e intuitiva.
7. Realizar un proyecto de fácil mantenimiento.
8. Posibilidad de consulta de los informes por varios usuarios concurrentemente.

2.2.- ESTUDIO DE ALTERNATIVAS

En cuanto a las diferentes alternativas posibles, en la mayor parte de los casos se ha optado por las soluciones que no suponen un gasto adicional en lo que a licencias se refiere. La única excepción es el uso de QlikView como herramienta de análisis de BI, el cual requiere licencias de Desarrollador y de Servidor para poder servir los informes mediante la instalación de QlikView Server.

Para cumplir con los requerimientos demandados por el cliente, hemos utilizado diferentes herramientas.

- En cuanto a Bases de Datos se está usando MySQL.

- Como herramienta ETL y para la gestión de procesos usamos Talend.
- Para el diseño y análisis de Informes de datos usamos QlikView.

2.3.- METODOLOGÍA EMPLEADA

En cuanto a la realización de este proyecto, hemos dividido su desarrollo en tres fases principales:

- Una primera fase donde se definen y desarrollan los procesos de extracción de datos, transformación, gestión de recarga de datos, etc...
- Una segunda fase en la cual creamos el modelo de datos de nuestro informe, codificamos su script de carga, y diseñamos los gráficos y medidores.
- Una última tercera fase, en la cual preparamos el entorno en el cual se alojarán los informes, y analizamos la flexibilidad y posibilidades de interacción con dichos informes.

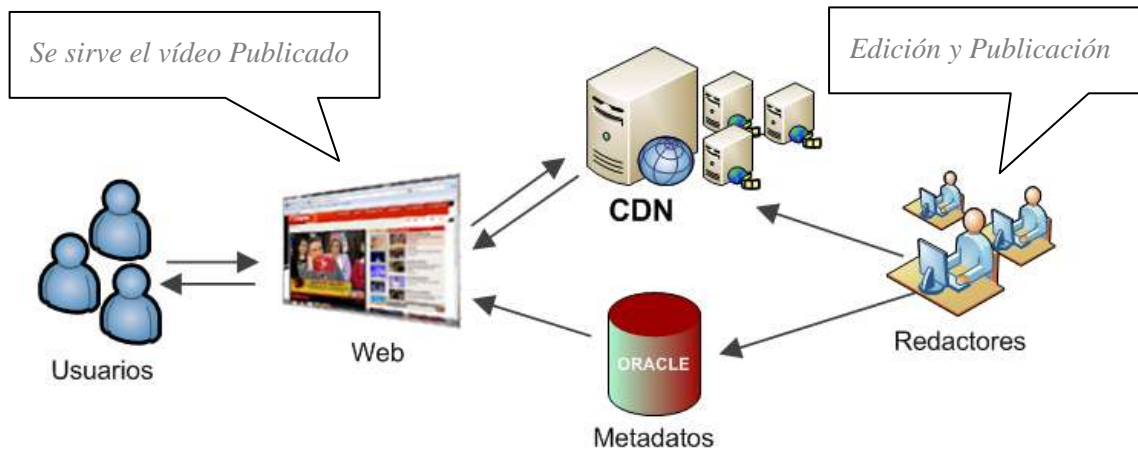
3.- DESCRIPCIÓN DEL PROYECTO

Para desarrollar este proyecto es necesario implementar una arquitectura que permita cruzar y procesar información relacionada con los vídeos servidos desde la web. Dicha información la obtendremos desde diferentes orígenes para su posterior análisis.

Tras analizar en detalle los requerimientos del cliente que se comentaron con anterioridad, se realizaron algunas propuestas respecto a la infraestructura y sistema de procesamiento de datos:

- Uso de ficheros .qvd para mejorar los tiempos de carga de datos en los informes.
- Algunos medidores y gráficos con información sobre: Numero de Descargas, Volumen Descargado, Evolucion de las Descargas, Disponibilidad de los datos cargados, Calidad de servicio ofrecido por los diferentes proveedores, etc...
- Uso de ETL OpenSource (Talend) para facilitar la creación y mantenimiento de los procesos de extracción y carga.
- Etc...

El siguiente esquema nos muestra una idea a un alto nivel sobre cuáles son las diferentes partes que entran en juego en la web a la hora de publicar un vídeo.



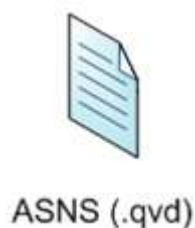
La información necesaria para realizar el procesamiento y posterior análisis de los datos, la obtendremos de diferentes orígenes:



Desde aquí obtenemos los ficheros de log diarios con la información de las descargas de vídeos. Posteriormente analizaremos en detalle la información contenida en dichos ficheros.



Información de los metadatos de los diferentes vídeos servidos (Título, Descripción, Categoría, Tamaño, Bitrate, Duración, etc...). Esta Base de Datos corresponde a una aplicación externa que usan los redactores para subir vídeos y posteriormente publicarlos.



Fichero con un listado de información sobre los diferentes códigos ASNS y su descripción.

Lo usamos para poder cruzar estos datos con las IP de los clientes, y así obtener información sobre el servicio ofrecido por los diferentes proveedores.

Teniendo en cuenta los objetivos y requerimientos definidos, podemos dividir el proyecto en tres fases:

- **Fase 1. Extracción, Transformación y Carga de datos.** En esta fase veremos cómo son los procesos de obtención de datos, así como su planificación diaria, y las diferentes fuentes de información.
- **Fase 2. Diseño de Informes Dinámicos.** Esta es la fase en la cual diseñamos el modelo de datos de nuestro informe, el script de carga de datos, y los medidores que darán vida a nuestro informe dinámico para su posterior análisis.
- **Fase 3. Análisis en Informes Dinámicos.** Mostraremos como acceder a los informes, y demostraremos como sacar el máximo partido de ellos realizando consultas dinámicas.

4.- FASES DEL PROYECTO

4.1.- EXTRACCIÓN TRANSFORMACIÓN Y CARGA DE DATOS

Esta primera fase consiste en: Obtener, Tratar y Almacenar la máxima información posible de los ficheros .log ofrecidos por la Plataforma CDN (desde la cual se sirven los vídeos). En estos ficheros, tenemos información detallada de cada una de las descargas que se han realizado, dicha información puede referirse a descargas de diferentes tipos de contenidos web: Vídeos, Elementos de la web, Objetos Flash, etc...

4.1.1.- Logs de la CDN

Los logs de la CDN son la principal fuente de información para nuestro informe de descargas, dichos logs son ficheros de texto plano que contienen una línea de información por cada descarga de un objeto de la web. El contenido de cada fichero es diario, y sus datos vienen separados por el carácter ‘,’.

Estos ficheros suelen tener un gran tamaño debido al enorme volumen de datos que contienen. Un fichero comprimido de logs con información de las descargas de un solo día, suele ocupar entorno a 150 MB, al descomprimirlo puede ocupar entorno a 900 MB.

Este es el aspecto que tiene este tipo de ficheros:

```
dl, 8, 20070225-085532, 13168, 1900,
BT: 7ba846ccc61b193a16ac98dde955b30b34a685ed, 38070814, 0, 195.112.33.204:49443,
2D5658313230302D4242355A593448535549594B, "Velocix Client 1.2.0.0", 2673, 0, 2673,
20070224-184633, 2, 9699328, rate=128000&tokenid=ABCD1234, 128000
```

Campo	Descripción	Valor de Ejemplo
[Identifier]	Tipo de registro (Siempre es el mismo)	dl
[Download ID]	Identificador único de cada descarga	8
[End Timestamp]	Fecha y Hora del momento en el cual finalizó la descarga. Formato UTC/GMT	20070225-085532
[Velocix Serial Numbre]	Numero identificador del objeto descargado	13168
[Total Download Duration]	Numero de segundos transcurridos desde hasta que se para la descarga	1900
[Protonhash]	Protocolo específico que identifica únicamente el objeto descargado. En el caso de BitTorrent es "BT" seguido del info_hash del objeto	BT:7ba846ccc61b193a16ac98dde955b30b34a685ed
[Total Bytes Reveved]	Numero total de bytes recibidos en la descarga.	38070814
[Total Bytes Uploaded]	Numero total de bytes subidos (Siempre es 0)	0
[Peer Endpoint]	Dirección IP y número de puerto usado por alguna de las máquinas mediante las que un usuario ha realizado la petición de la descarga.	195.112.33.204:49443
[Peer ID]	Código que identificador del Peer.	2D5658313230302D4242355A593448535549594B
[User Agent]	Es información sobre software usado por el Cliente. Suele estar compuesto por "HTTP" seguido de información	"Velocix Client 1.2.0.0" ó HTTP:Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES;

	sobre el navegador y Sistema operativo.	rv:1.9.1.7) Gecko/20091221 Firefox/3.0.3;MEGAUPLOAD 1.0 MEGAUPLOAD 1.0 (.NET CLR 3.5.30729)
[Pieces in File]	Numero de trozos en los que se divide el fichero.	2673
[Initial Pieces]	Numero de trozos del fichero que el cliente tiene al comienzo de la descarga. Este valor siempre será cero.	0
[Final Pieces]	Numero de trozos del fichero en el cliente al final de la descarga.	2673
[Start Timestamp]	Fecha y hora en el cual comenzó la descarga. Formato UTC/GMT	20070224-184633
[Download Sessions]	Numero de diferentes periodos en los cuales se realizó la descarga (Cada vez que paramos la descarga y la continuamos más tarde)	2
[Cache Bytes Downloaded]	Numero de Bytes descargados desde la caché de Velocix	9699328
[Version Specific Bytes]	Código de Autorización del token para la descarga. Será “-“ si no hay token	“-“ ó rate=128000&tokenid=ABCD1234
[Download Rate]	Numero de Bytes por segundo que se sirvieron en la descarga para el token especificado. Si no hay token será “-“	“-“ ó 128000

Estos ficheros son descargados diariamente mediante peticiones por https.

[https://console.velocix.com/logs/yyyymdd/downloads/downloads_\[yyyyymmdd\].csv.gz](https://console.velocix.com/logs/yyyymdd/downloads/downloads_[yyyyymmdd].csv.gz)

Cada fichero contiene información de las descargas realizadas en un día concreto, y su contenido suele estar actualizado al 100% pasados 2 días de su fecha, es decir, el fichero **downloads_20090120.csv** contendrá el 100% de la información de todas las descargas realizadas

ese día si lo descargamos a partir del 22 de Enero del 2009, en el caso de descargarlo antes quizás no esté disponible toda la información de las descargas realizadas durante las últimas horas de ese día.

Para nuestro informe solo tendremos en cuenta las descargas de objetos tipo vídeo, cuyo volumen supone más del 85% del total del volumen descargado por todos los objetos de la web.

4.1.2.- Procesos de Extracción, Transformación y Carga de Datos

La información resultante de estos procesos será almacenada en una Base de Datos MySQL (previamente instalada en el servidor), los datos origen serán los obtenidos desde los ficheros de log.

Debido al gran volumen de datos contenido en cada fichero, se optó por crear una tabla por cada mes a analizar (*actualmente una tabla con datos de un mes puede llegar a ocupar más de 11 GB y contener más de 20 millones de registros*), para almacenar los datos de cada descarga, y otra para las descargas de otro tipo de objetos web, los cuales de momento no son de utilidad para nuestro informe.

```
CREATE TABLE `downloads2010_1` (  
  `Download_ID` bigint(20) NOT NULL,  
  
  `End_Timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  
  `Total_Download_Duration` varchar(50) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `Protohash` varchar(43) CHARACTER SET latin1 DEFAULT NULL,  
  `Total_Bytes_Received` bigint(20) DEFAULT NULL,  
  `Peer_Endpoint` varchar(21) CHARACTER SET latin1 DEFAULT NULL,  
  `Peer_ID` varchar(12) CHARACTER SET latin1 DEFAULT NULL,  
  `User_Agent` text COLLATE latin1_spanish_ci,  
  `Pieces_in_File` int(10) unsigned DEFAULT NULL,  
  `Initial_Pieces` int(10) unsigned DEFAULT NULL,  
  `Final_Pieces` int(10) unsigned DEFAULT NULL,  
  `Start_Timestamp` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `Download_Sessions` int(10) unsigned DEFAULT NULL,  
  `Cache_Bytes_Downloaded` bigint(20) DEFAULT NULL,  
  `FechaDatosDownload` date DEFAULT NULL,  
  `AnnoDatosDownload` int(11) DEFAULT NULL,  
  `MesDatosDownload` smallint(6) DEFAULT NULL,  
  `DiaDatosDownload` smallint(6) DEFAULT NULL,  
  `ASN` int(10) unsigned DEFAULT NULL,  
  PRIMARY KEY (`Download_ID`),  
  KEY `IND_p_DOWNLOADS2010_1_DiaDatosDownload` (`DiaDatosDownload`),  
  KEY `IND_p_DOWNLOADS2010_1_FechaDatosDownload` (`FechaDatosDownload`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

Para la descarga y procesamiento de los datos usamos dos procesos desarrollados en Talend, los cuales detallaremos a continuación.

4.1.2.1.- Proceso de descarga y tratamiento de logs

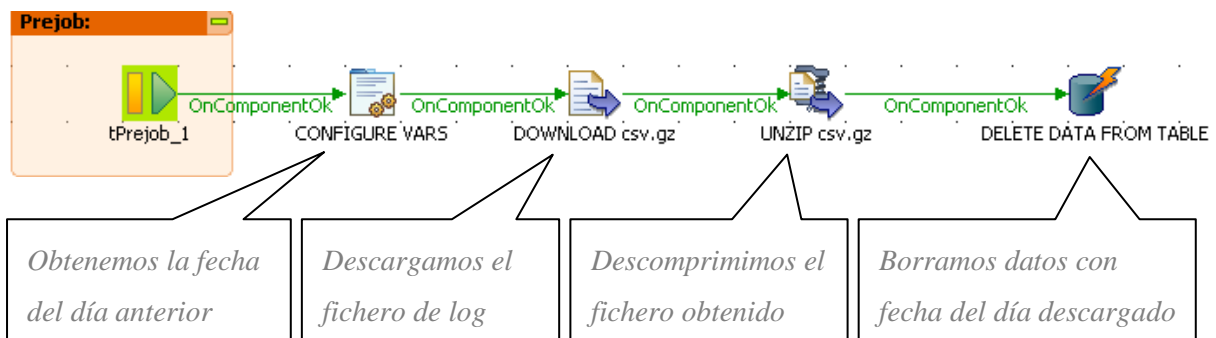
Objetivo:

El objetivo de este primer proceso Talend es el siguiente:

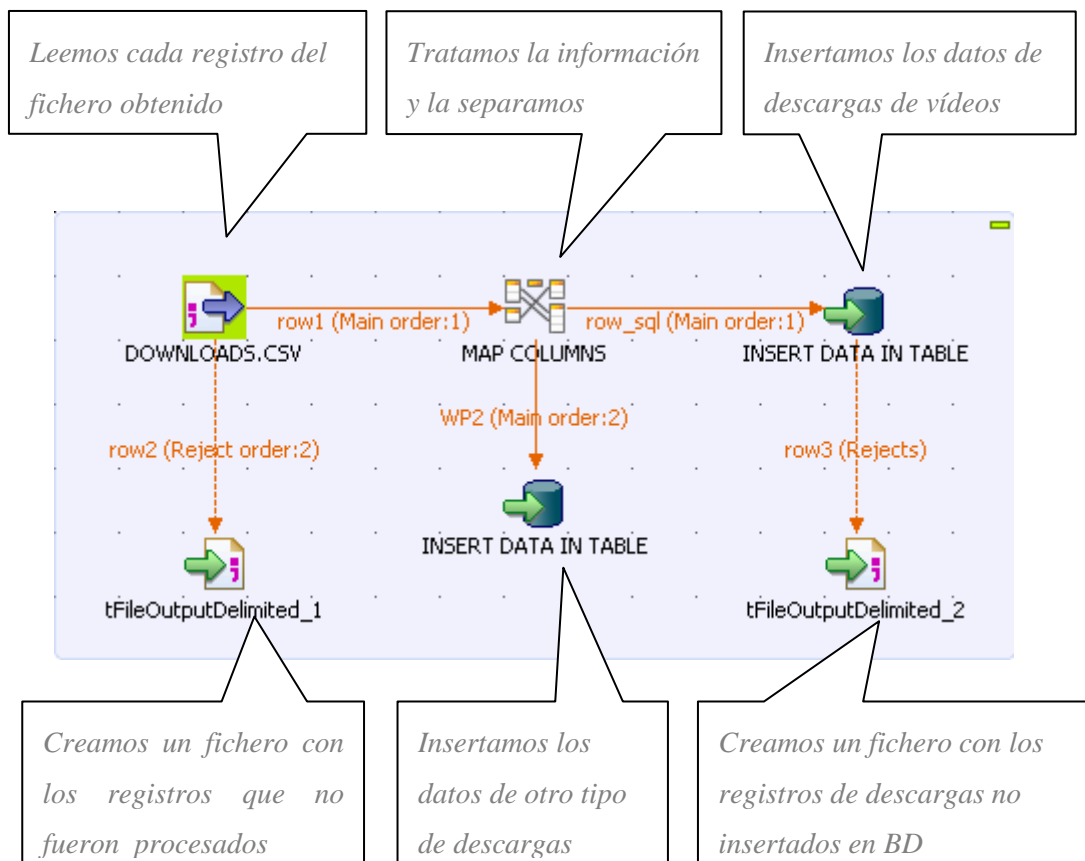
- Descargar el fichero de logs de la CDN.
- Tratar solo los registros válidos referentes a descargas de vídeos.
- Insertar en Base de Datos dichos registros para poder ser analizados, y mantener un histórico.
- Informar del resultado de la ejecución vía email.

Detalle del proceso:

El proceso se compone de tres partes. En la primera parte obtenemos la fecha del día anterior; después descargamos (mediante una petición por https a la CDN) el fichero de log comprimido, y lo descomprimos en un directorio de trabajo; posteriormente eliminamos los registros (con fecha del día anterior) de la tabla correspondiente.



Una vez tenemos preparado el fichero log, obtenemos sus datos para procesarlos y almacenarlos en la tabla correspondiente. En caso de fallar la lectura de datos o la inserción en la tabla de descargas, los datos erróneos se almacenarán en un fichero de errores.



Mediante el componente **MAP COLUMNS** mapeamos y tratamos la información origen.

(Ver imagen en la página siguiente)

Tal y como podemos ver en la imagen posterior, tenemos en la izquierda el esquema que usaremos para leer los registros del fichero de log, y en la parte derecha está el mapeo por campos que usaremos para insertar los datos origen en la tabla de descargas.

La mayoría de los campos se insertarán sin ninguna modificación en la tabla destino, solo cabe destacar la creación de algunos nuevos campos:

- Campos relacionados con la fecha de descarga (formateando el campo origen **Start_Timestamp**).
- Campo 'ASN', que contendrá un código para posteriormente cruzar su información con los nombres de los diferentes proveedores (Telefonica, ONO, Orange, etc...)

Aplicamos un filtro para obtener solo las descargas de videos.
Solo aquellas en las cuales el campo Protohash comience por
'BT:'

Column
Identifier
Download_ID
End_Timestamp
Velocix_Serial_Number
Total_Download_Duration
Protohash
Total_Bytes_Received
Total_Bytes_Uploaded
Peer_Endpoint
Peer_ID
User_Agent
Pieces_in_File
Initial_Pieces
Final_Pieces
Start_Timestamp
Download_Sessions
Cache_Bytes_Downloaded
Version_Specific_Bytes
Download_Rate

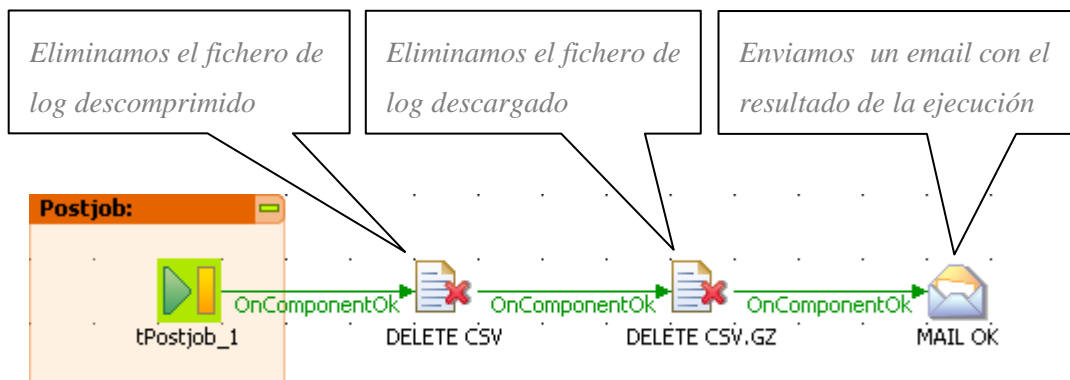
Expression	Column
row1.Download_ID	Download_ID
row1.End_Timestamp	End_Timestamp
row1.Total_Download_Duration	Total_Download_Dur...
row1.Protohash	Protohash
row1.Total_Bytes_Received	Total_Bytes_Received
row1.Peer_Endpoint	Peer_Endpoint
row1.Peer_ID	Peer_ID
row1.User_Agent	User_Agent
row1.Pieces_in_File	Pieces_in_File
row1.Initial_Pieces	Initial_Pieces
row1.Final_Pieces	Final_Pieces
row1.Start_Timestamp	Start_Timestamp
row1.Download_Sessions	Download_Sessions
row1.Cache_Bytes_Downloaded	Cache_Bytes_Downl...
TalendDate.parseDate("yyyyMMdd",co... context.anno	FechaDatosDownload
context.mes.shortValue()	MesDatosDownload
context.dia.shortValue()	DiaDatosDownload
GuardianUtils.mapAsn(row1.Peer_Endp...	ASN

Formateamos la fecha origen para tener
un formato de este estilo 'YYYYMMDD'

Función para obtener el código ASN a
partir de la IP origen

(Ver ANEXO, Código de mapeo de IP's con sus ASNS correspondientes)

Por último, eliminaremos el fichero de log descargado, y el fichero de log descomprimido que se usó para la carga de datos. Además se enviará un correo con el resultado de la ejecución de todo el proceso.



Eliminamos el fichero de
log descomprimido

Eliminamos el fichero de
log descargado

Enviamos un email con el
resultado de la ejecución

4.1.2.2.- Proceso de gestión y planificación de recarga

Objetivo:

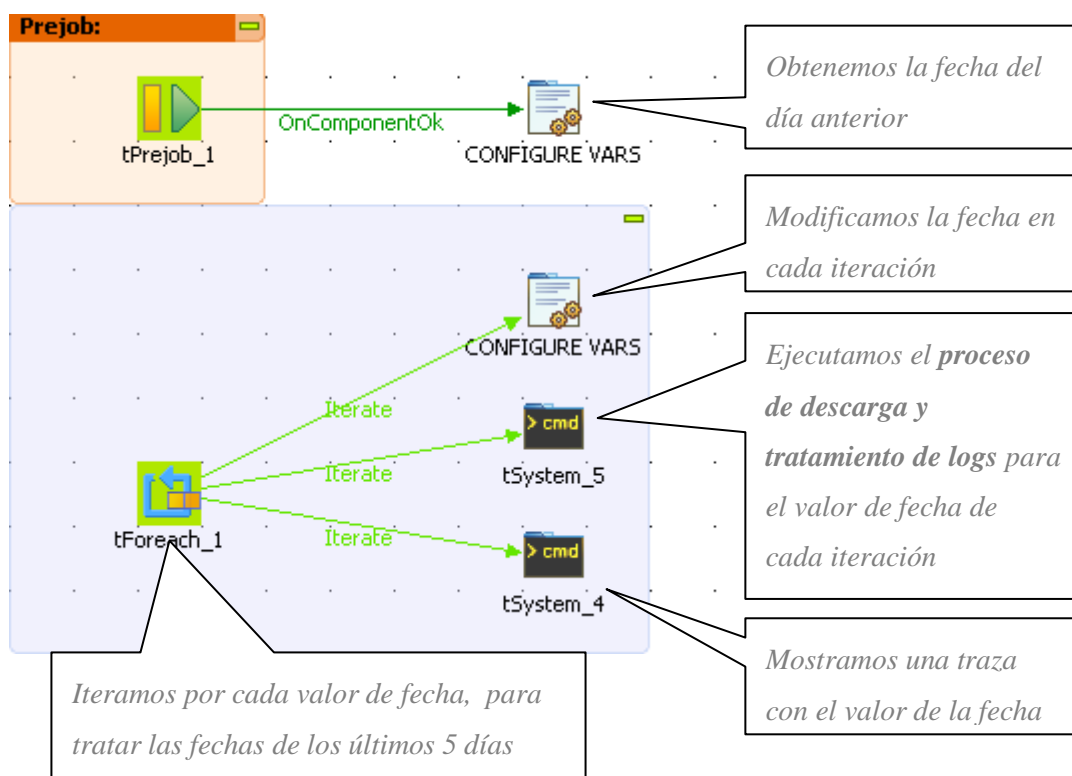
Este proceso es el responsable de coordinar las descargas y procesamiento de logs, con la recarga de datos de los informes QlikView.

Los principales objetivos de este proceso son:

- Ejecutar diariamente el **proceso de descarga y tratamiento de logs** (anteriormente comentado).
- Ejecutar la **creación de .qvd's diarios**, para mejorar la eficiencia en la recarga de datos en los informes QlikView.
- **Recargar informe QlikView** correspondiente al mes actual (también recargaremos el informe del mes anterior durante los primeros días del mes siguiente).
- **Gestionar los directorios** con los informes y .qvd's mensuales.

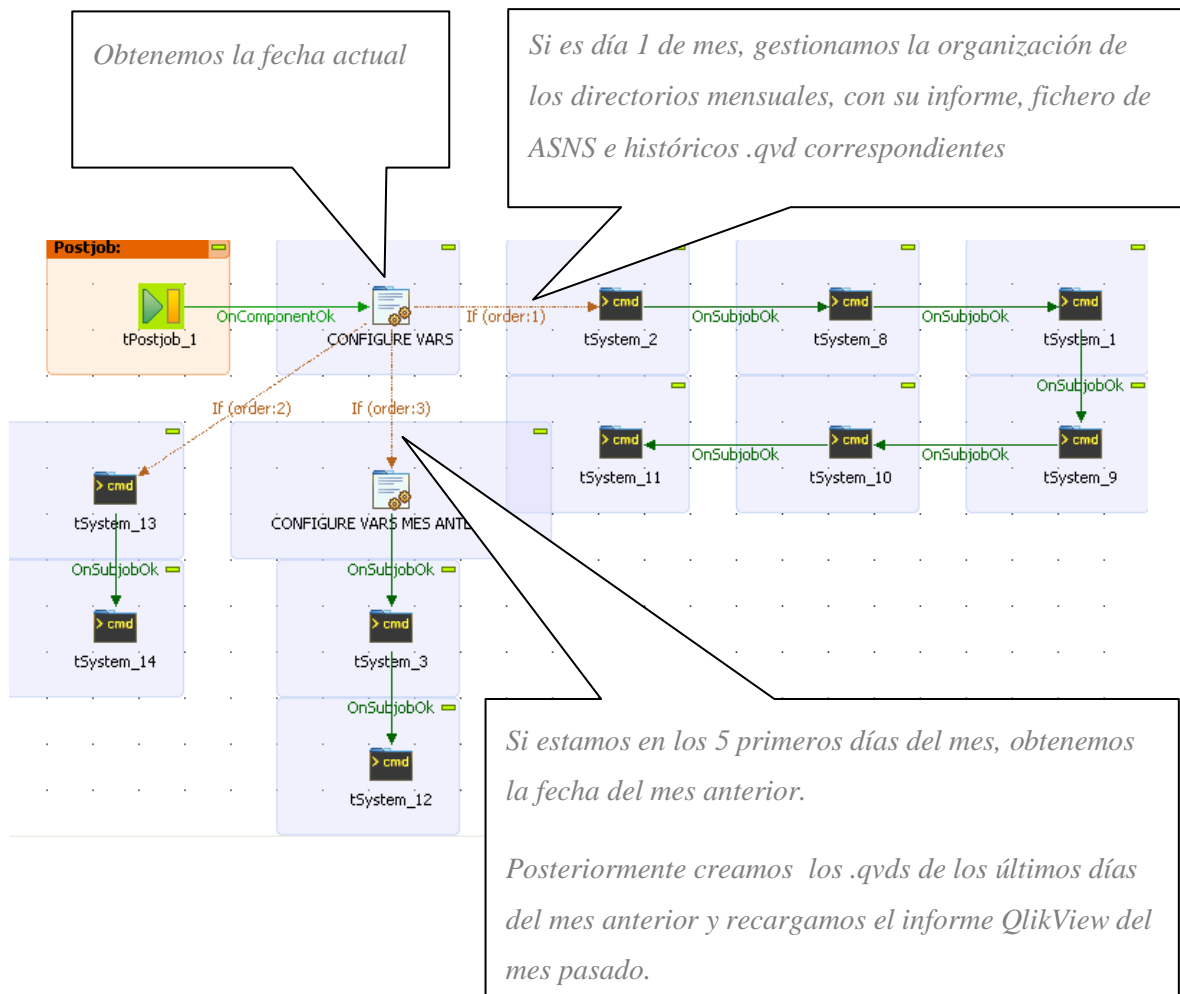
Detalle del proceso:

En primer lugar, nos ocuparemos de mantener lo más actualizada posible la información de las descargas de vídeos. Para ello ejecutaremos el proceso de descarga y tratamiento de logs para los últimos 5 días.

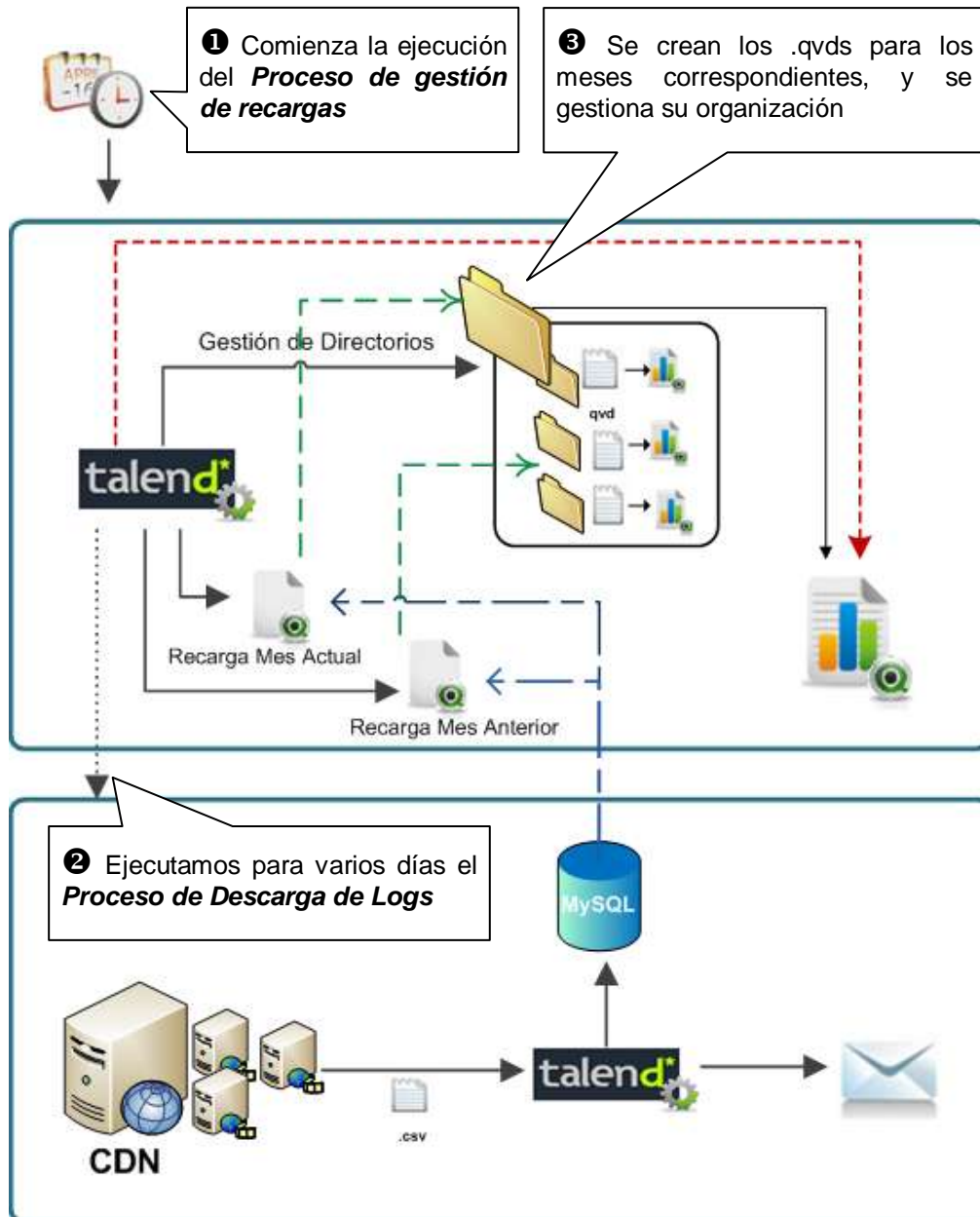


A continuación, el proceso ejecutará la creación de .qvd's, y recargaremos los datos del informe QlikView correspondiente al mes actual (también se realizará para el mes anterior si estamos en los 5 primeros días del mes).

Debido al gran volumen de datos, organizaremos por meses los .qvd's y los informes QlikView. La organización de estos ficheros se llevará a cabo automáticamente, el día 1 de cada mes se creará un directorio (en formato YYYY_MM) donde estará el informe QlikView, y un subdirectorio ('Historicos') con los .qvd's de ese mes.



El esquema que mostramos a continuación nos dará una visión más general del funcionamiento de estos dos procesos, y la relación entre ellos.



4.1.3.- Creación de ficheros .qvd

Los ficheros con extensión .qvd, son ficheros que pueden ser usados por QlikView para almacenar información comprimida. Dichos ficheros son un XML con la estructura necesaria para poder ser procesados muy rápidamente desde un informe QlikView.

Mediante el uso de este tipo de ficheros hemos logrado mejorar la eficiencia del proceso de recarga de datos en los informes (se han reducido más de la mitad tiempos de recarga de datos), actualmente los tiempos de recarga de los informes mensuales completos suelen estar entorno a 35 - 40 minutos, cuando anteriormente los tiempos nunca eran inferiores a 1'5 horas.

En general, las principales ventajas que este tipo de ficheros nos ha aportado son:

- **Gran rapidez de lectura y procesamiento de datos.** Entre 10 y 100 veces más rápido que leer desde otras fuentes de datos, dependiendo de: Numero de campos por registro, Tipos de datos de cada campo, Velocidad del Disco Duro, Velocidad de la CPU, Transformaciones y operaciones con los datos, etc. La velocidad de lectura desde un fichero de log en formato .csv, es aproximadamente de 3.000 registros por segundo, en cambio la velocidad de lectura desde un fichero .qvd con el mismo contenido sería de 200.000 registros por segundo.
- **Menor tamaño de los ficheros que contienen información.** Una tabla con información de los logs de un mes de descargas puede ocupar varios GBytes (*La tabla de descargas de vídeos de Enero del 2010 ocupa 8,1 GBytes*) al tener esta información en ficheros .qvds diarios mejoramos sustancialmente no solo su rapidez de lectura, sino que reducimos el volumen de datos. (*La suma del volumen de los .qvds correspondientes a las descargas de videos de Enero del 2010 es de 1,63 GBytes*)

A continuación mostramos un ejemplo del tamaño de diferentes tipos de fichero con la misma información:



(1) Fichero de logs comprimido, (2) Fichero de logs Descomprimido en .csv, (3) Fichero QVD con toda la información de fichero de logs, (4) Fichero QlikView con toda la información de fichero de logs.

- **Fácil manejo de este tipo de ficheros desde el editor de script de QlikView.**

Para la creación de los ficheros .qvd hemos usado dos informes QlikView, cuyo único cometido es comprimir en ficheros .qvd la información diaria obtenida desde los logs.

En caso de fallar la ejecución de alguno de ellos, podremos volver a generar los .qvd ejecutando manualmente el informe que corresponda.

Los informes son: *Recarga_QVDS_MesAnterior* y *Recarga_QVDS_MesActual*

4.1.3.1.- Recarga de ficheros .qvd del mes anterior

El objetivo de este informe es actualizar la información de los .qvd's correspondientes a los últimos días del mes anterior (solo si estamos en los primeros 5 días del mes actual). Esto es debido a que la información de los logs no viene completa hasta pasados 4 o 5 días, es por ello que nos aseguramos de recargar los .qvds correspondientes a varios días atrás, para que contengan la información lo más actualizada posible.

Mediante el script de dicho informe, obtendremos la información correspondiente a cada día del mes anterior que queremos almacenar mediante ficheros .qvd.

Podemos ver el script en el Anexo (Codigo: Recarga_QVDS_MesAnterior)

La información origen para crear los .qvd's la obtendremos desde la tabla correspondiente en la Base de Datos de Historicos (MySQL). La sentencia que usaremos en los script de QlikView para crear cada fichero .qvd será algo de este estilo:

```
store Download into Descargas_y_Proveedores\$(AñoMesAnterior)_$(MesAnterior)\Historicos\H_Downloads$(AñoMesAnterior)_$(MesAnterior)_$(i).qvd;
```

4.1.3.2.- Recarga de ficheros .qvd del mes actual

De la misma manera que en el informe anterior, mediante el informe correspondiente creamos los .qvds correspondientes al mes actual.

Si estamos entre los 5 primeros días del mes, crearemos los .qvds de dichos días (los .qvds de los últimos días del mes anterior se actualizarán mediante el informe que vimos anteriormente), en caso contrario actualizaremos los .qvds de los últimos 6 días desde la fecha actual.

Al igual que en el informe anterior, la información correspondiente a las descargas de vídeos en los días del mes actual la obtendremos desde el script de dicho informe, obteniendo los datos desde la Base de Datos de Históricos.

Podemos ver el script de dicho informe en el Anexo (Codigo: Recarga_QVDS_MesActual)

La sentencia que usaremos en los script de QlikView para crear cada fichero .qvd, será algo de este estilo:

```
store Download into Historicos\H_Downloads$(Año)_$(Mes)_$(i).qvd;
```

4.1.4.- Planificación diaria de ejecución de los procesos

Para planificar la ejecución de los procesos ETL, simplemente necesitaremos un ejecutable que nos permita arrancar dichos procesos, y una herramienta que nos permita configurar a qué hora y qué días se realizará su ejecución.

Desde la interfaz de Talend, tenemos la opción de exportar a un fichero un proyecto determinado. Al exportar cada proyecto obtendremos un fichero comprimido, el cual internamente estará compuesto por varios directorios, los cuales contendrán las librerías, ficheros de propiedades y elementos necesarios para poder ejecutar el proceso.

Entre los diferentes ficheros que obtenemos al descomprimir el proyecto exportado, prestaremos especial atención a tres de ellos:

- Un fichero .jar, con las clases java creadas internamente.
- Un fichero .sh, para ejecutar el proceso java desde Linux.
- Un fichero .bat, para llamar al proceso java desde MS-DOS.

Será el ejecutable de MS-DOS, mediante el cual ejecutaremos diariamente dichos procesos. Para ello, se realizará internamente una llamada de este estilo:

```
java -Xms256M -Xmx1024M -cp
./lib/javacsv.jar;./lib/systemRoutines.jar;./lib/userRoutines.jar;.;re
proceso_y_recarga_velocix_1_0.jar;./lib;
guardian.reproceso_y_recarga_velocix_1_0.REPROCESO_y_RECARGA_Velocix --
context=Default %*
```

La herramienta que usaremos para la planificación de dichos procesos, será el propio Task Scheduler (‘Planificador de Tareas’) que viene incorporado en Windows.

Los procesos actualmente están configurados para ejecutarse diariamente a las 03:00, y su duración variará entre 35 y 40 minutos (aproximadamente), en función del volumen de datos a tratar.

4.2.- DISEÑO DE INFORMES DINÁMICOS

En esta fase nos centraremos en detallar los pasos necesarios para generar un informe QlikView, y en particular para generar nuestro informe de Descargas de Vídeos. Trataremos desde la creación del script y modelo de datos, hasta el diseño y construcción de los medidores gráficos.

4.2.1.- Script de recarga y creación del modelo de datos

La creación del script de recarga, es la parte más importante de los informes QlikView. El script de recarga es el responsable de conectar con las diferentes fuentes de información, y realizar las transformaciones de datos necesarias para posteriormente construir nuestro modelo de datos.

Mediante sentencias muy parecidas a las usadas en SQL (select, join, left join, inner join, drop table, etc...) en el script de recarga compondremos las diferentes tablas que formarán nuestro modelo, teniendo en cuenta que QlikView asocia los campos que tienen el mismo nombre.

La lógica asociativa usada por QlikView, será la encargada de propagar las selecciones realizadas en los componentes y gráficos de nuestro informe.

En cada recarga de un informe QlikView se eliminarán todos los datos de dicho informe, para posteriormente cargarlos desde cero mediante el script de recarga. Es por ello, que para poder realizar cargas incrementales debemos hacer uso de los ficheros .qvd.

Para realizar la recarga mediante línea de comandos, usaremos una instrucción de este estilo:

[Ruta donde se encuentra el ejecutable QV] /r [Ruta del fichero a recargar]

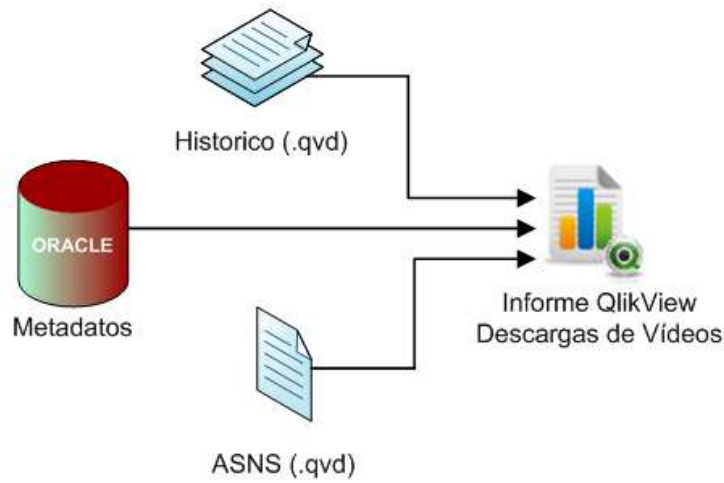
"C:\qlikview\qve.exe" /r "C:\qlikview\example\file.qvw"

Mediante el parámetro '/r' el informe QlikView se abrirá, cargará sus datos a partir del script, se guardarán sus datos, y posteriormente se cerrará automáticamente.

En nuestro caso el script debe obtener los datos desde diferentes orígenes, estos son:

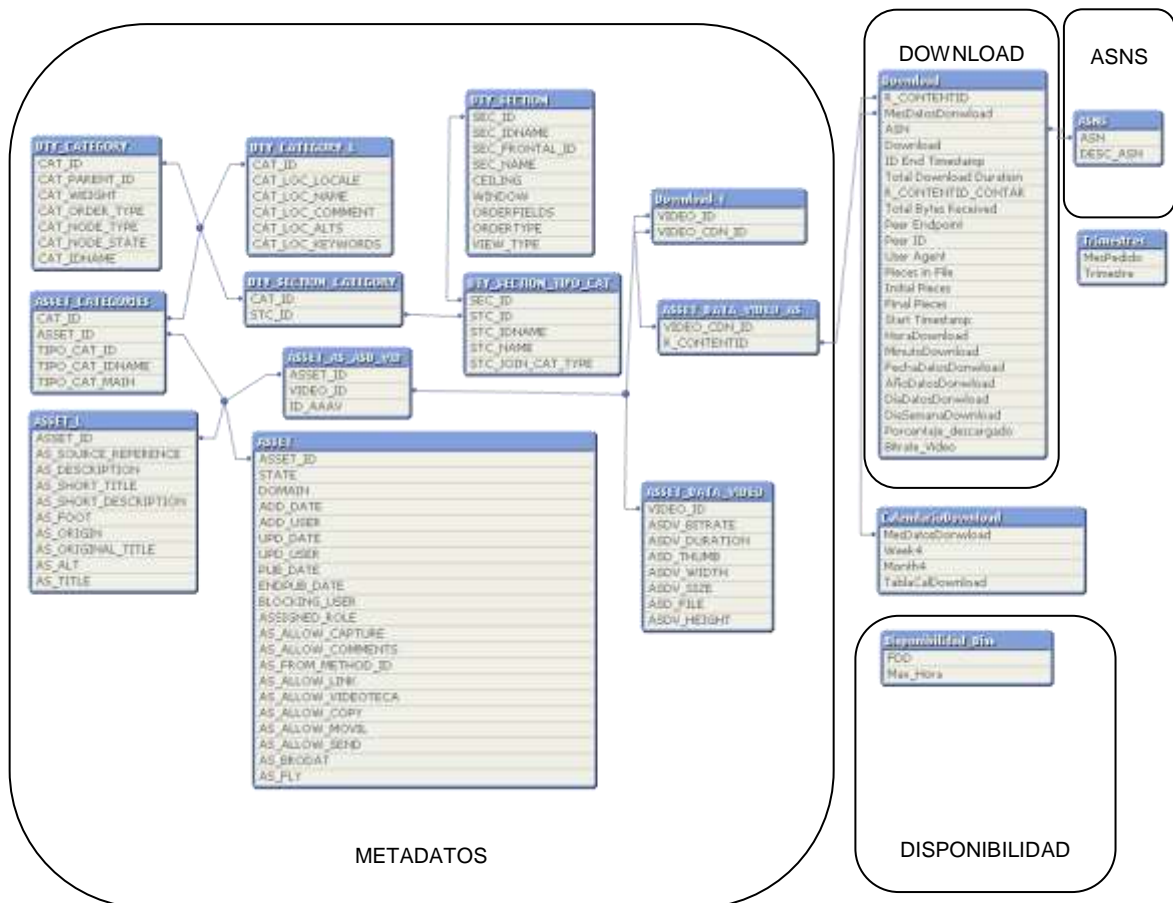
- **Ficheros .qvd con información de Descargas.** Los cuales fueron creados mediante los procesos Talend comentados en la Fase de Extracción, Transformación y Carga de Datos.
- **Base de Datos con metadatos de los vídeos.** Se trata de una Base de Datos Oracle, en la cual están los metadatos asociados a cada vídeo. Esta Base de Datos corresponde a una aplicación externa mediante la cual los redactores suben los vídeos a la CDN.

- **Fichero .qvd de Proveedores.** En este fichero tenemos los códigos ASN y la Descripción de los diferentes operadores de red.



En el Anexo podremos ver las principales instrucciones que forman el script de recarga correspondiente a nuestro informe QlikView. **(Ver Anexo, Código: Descarga-Videos)**

A continuación mostraremos el modelo de datos resultante de ejecutar nuestro script.



La mayor parte de las tablas son las responsables de aportar todos los metadatos a los vídeos, dichas tablas las obtenemos desde una Base de datos Oracle, dichos datos son creados y gestionados desde otra aplicación.

La tabla **'Download'** contiene la información obtenida desde los ficheros .qvd (generados por los procesos Talend). Esta tabla estará asociada con la estructura de tablas de metadatos mediante el campo **'R_CONTENTID'**, el cual contiene el Protohash del vídeo.

Además la tabla Download estará asociada con la tabla de **ASNS**, en la cual tenemos el código ASN y la Descripción de los diferentes Proveedores.

Se ha creado una tabla independiente para mostrarnos la disponibilidad de los datos (Nombre de la tabla: **'Disponibilidad_Dias'**), dicha tabla refleja por cada día cargado hasta qué hora contiene datos.

4.2.2.- Creación y Diseño de los Gráficos

Para el diseño de informes en QlikView partimos de un documento en blanco, en el cual vamos insertando objetos de diferentes tipos para componer nuestro informe.

Los objetos más importantes en este tipo de informes son dos:

Cuadro de Lista  .

Este tipo de objeto contiene un listado de los diferentes valores posibles de un determinado campo. El campo seleccionado será cualquiera de los diferentes campos que componen las tablas el modelo de datos.

Las selecciones que se realicen sobre este objeto, se propagarán internamente a todas las tablas que estén asociadas con el campo asignado.

En nuestro proyecto se utilizarán varios Cuadros de Lista, los cuales hemos duplicado por cada hoja de nuestro informe. Dichos Cuadros de Lista podemos agruparlos en dos categorías:

- *Los que forman parte de la selección de fechas (Año, Mes, Día, Día de la Semana, Hora)*
- *Los que forman parte de los metadatos de los vídeos. (Sección, Categoría, Título, Descripción, etc...)*

Objeto Gráfico



En este caso tenemos la posibilidad de elegir entre diferentes objetos: Gráfico de Barras, Lineal, Combinado, Tipo Radar, Tipo Indicador, Tabla Pivotante, etc., mediante los cuales podremos ver de una manera muy visual y atractiva la representación de nuestra información, y realizar cálculos personalizados.

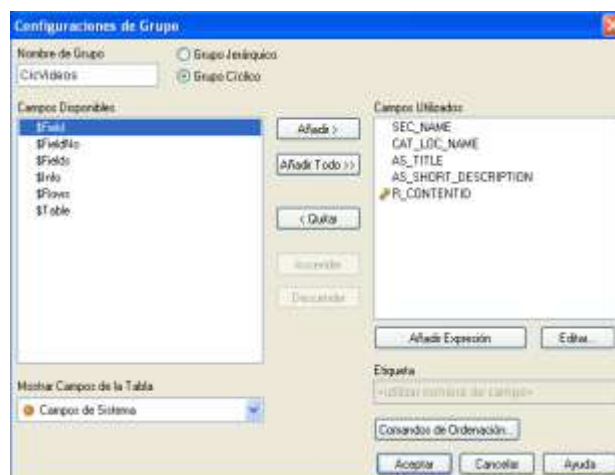


En la configuración de la mayor parte de los gráficos podemos elegir una o varias Dimensiones, en función de las cuales se realizarán los cálculos definidos en la pestaña de Expresiones.

Para obtener por ejemplo el número de Descargas de Vídeos que se han realizado desde cada Proveedor, tendremos que:

1. Entrar en las Propiedades de un Gráfico de Barras.
2. Seleccionar en Dimensiones el campo que contiene el nombre de todos los Proveedores.
3. Posteriormente en la pestaña de Expresiones introducimos el calculo que queremos hacer al agrupar por cada Proveedor. (en nuestro caso, será simplemente una expresión que permita contar el número de descargas **COUNT(Download)**).

Una característica muy útil que hemos usado en nuestros gráficos, es la capacidad de poder crear grupos formados por diferentes campos o expresiones, estos grupos pueden clasificarse en **Jerárquicos**  ó **Cíclicos** .



Un **grupo jerárquico**, será aquel compuesto por varias expresiones o campos ordenados de menor a mayor detalle, de manera que al usar este tipo de grupo en la dimensión de un gráfico, nos dará la sensación de profundidad en el análisis de los datos al realizar selecciones sobre él.

Por ejemplo: Un gráfico cuya Expresión a calcular es el Total de Descargas, y cuya Dimensión es un grupo jerárquico compuesto por Sección, Título de Vídeo y Partes del vídeo.



Un **grupo cíclico**, será aquel compuesto por varios campos o expresiones formando una especie de lista circular, y así poder seleccionar en esta lista el campo según el cual queremos analizar los datos.

Por ejemplo: Un gráfico cuya Expresión sea un grupo cíclico compuesto por dos expresiones, una de ellas para obtener el total de descargas, y otra para obtener el total de GB servidos, siendo la Dimensión las diferentes Secciones.



Estructura de los informes

El diseño de los informes es totalmente libre, no tenemos ninguna limitación en lo que a la disposición de los elementos se refiere. En nuestro caso, todas las hojas de nuestro informe compartirán la misma disposición de sus elementos, los cuales podemos agruparlos en cuatro zonas básicas:

- **Zona de selección de Metadatos.** Corresponde al conjunto de listas de selección situadas en la parte izquierda del documento. En esta zona encontraremos la información correspondiente a la Sección, Categoría, Título y Descripción de los diferentes vídeos descargados.
- **Zona de Gráficos y Medidores.** En la parte superior de cada una de las hojas de nuestro informe, encontraremos numerosos gráficos y medidores minimizados, los cuales podremos visualizar haciendo doble click sobre ellos.
- **Zona de Visualización.** La parte central de nuestro informe será donde visualicemos cada uno de los gráficos y medidores que queramos analizar.
- **Zona de Calendario.** En la parte inferior de nuestro informe podremos seleccionar el rango de fechas que queremos analizar. Tenemos la posibilidad analizar los resultados por meses, semanas, días, horas, minutos, etc...



4.3.- ANÁLISIS DE INFORMES DINÁMICOS

En esta última fase comentaremos donde hemos preparado el entorno para alojar y servir los informes QlikView, y además mostraremos algunos ejemplos de cómo podemos analizar la información disponible en nuestro informe mediante consultas dinámicas.

4.3.1.- Entorno para el análisis de los Informes

Los informes que hemos desarrollado en este proyecto trabajan con un gran volumen de información, tal y como comentamos anteriormente este tipo de informes dinámicos utilizan gran parte de la memoria RAM, ya que es allí donde se alojarán todos los datos que estos contienen.

Los procesos de recarga de nuestros informes, y las operaciones que estos realizan internamente (operaciones en algunos gráficos, tablas, medidores, etc...) requerirán un elevado uso de CPU en momentos puntuales.

Nuestros informes deben de poder ser consultados por cualquier usuario interno, sin necesidad de que estos dispongan de sofisticados equipos ni un gran volumen de memoria RAM, es por esto que nuestros informes serán alojados en un servidor dedicado única y exclusivamente a procesar, almacenar y servir los datos e informes realizados.

Estas son algunas de las características de nuestro servidor:



Sistema Operativo: Microsoft Windows Server
2003 Estándar x64 Edition

Procesador: Intel(R)Xenon(R) E5450 3.00GH

Mamoria RAM: 16 GB

Principales Unidades:

C: (Para el S.O y programas básicos) 15GB

Z: (Para la Base de Datos MySQL) 200GB

D: (Para alojar los informes) 900GB

Para poder mostrar los informes QlikView, deberemos instalar QlikView Server en nuestro servidor, y disponer de un determinado número de licencias de usuario (nos delimitarán cuantos usuarios pueden consultar los informes de manera concurrente).

Cuando cualquier cliente (usuario interno) quiera consultar un informe, este será servido por QlikView Server desde el servidor, y serán los recursos de dicho servidor los que se verán afectados al mostrar el informe e interactuar con él.

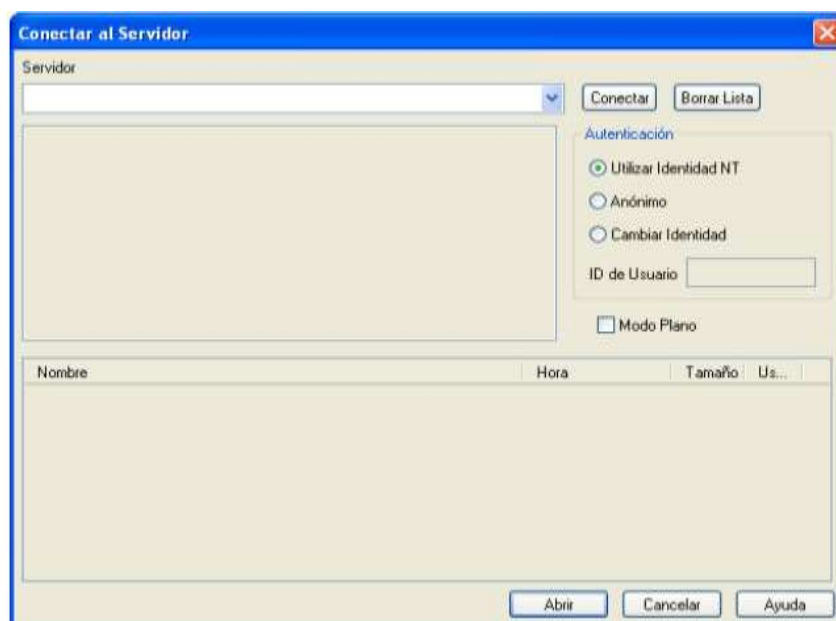
4.3.2.- Acceso a los informes

Actualmente los usuarios pueden acceder a los informes dinámicos de dos maneras:

- Desde QlikView, usando la opción **Archivo → Abrir en el Servidor**.
- Desde Internet Explorer, instalando un pequeño plugin que permite mostrarnos este tipo de informes.

Mediante la primera forma, simplemente tendremos que indicar el nombre o la URL de la máquina desde la cual estamos sirviendo los informes con QlikView Server. Posteriormente al pulsar el botón conectar, podremos ver el listado de informes qlikview a los cuales tenemos acceso.

En nuestro caso la autenticación estará a nivel de Windows.



Para la segunda opción hemos creado una sencilla página web, desde la cual podremos abrir el informe de Descargas de Vídeos de un mes concreto, y además tendremos un enlace al Plugin que debemos instalar para poder ver los informes desde Internet Explorer.



Para las dos formas de acceso disponibles nos aparecerá un mensaje de petición de usuario y contraseña, justo antes de abrir el informe seleccionado. El usuario y contraseña solicitados serán los mismos para cualquier informe que queramos abrir. Todos ellos se encontrarán en un directorio previamente configurado mediante QlikView Server.

La seguridad de nuestros informes está configurada para que sea a nivel de Windows, esto es debido a que no es compatible la seguridad en QlikView a nivel de informe con la recarga automática de los mismos.

Mediante QlikView podemos configurar la seguridad de cada informe, es decir, podemos crear diferentes cuentas (usuario y contraseña) para posteriormente ocultar o visualizar determinados objetos en función del usuario. Al abrir un informe QlikView con opciones de seguridad configuradas, será necesario introducir un usuario y contraseña, no solo si abrimos el informe manualmente, sino también al ejecutar una recarga de sus datos desde línea de comandos. Es por esto que se decidió usar la seguridad a nivel de Windows.

El acceso a los informes será en modo lectura, ningún cambio que realicen podrá afectar al informe original. Solo los administradores que tengan acceso a al servidor, podrán modificar los informes.

4.3.3.- Análisis de la Información

En la mayoría de los casos, cuando se quiere representar gráficamente los resultados de una actividad concreta, se suelen realizar informes estáticos mediante algún programa o herramienta, es decir, informes en los cuales el usuario no puede alterar la forma ni el conjunto de información a analizar.

Normalmente, cuando estamos analizando un informe estático, la información mostrada por un determinado gráfico está basada en una algún tipo de consulta ya definida sobre los datos origen.

En nuestro caso, el usuario encargado de analizar los informes QlikView tendrá la posibilidad de interactuar con ellos. Mediante diferentes objetos de selección, el usuario podrá alterar la información mostrada en los diferentes gráficos y medidores.

Las selecciones realizadas serán una especie de filtro, mediante el cual nuestro informe obtendrá los resultados correspondientes sobre todos los datos que disponemos.

La propagación de las selecciones dinámicas que hagamos en nuestro documento, nos permitirán obtener información de una forma inmediata, esto es debido a que sus datos se encuentran en memoria y no tenemos que esperar a que se ejecute ningún tipo de query contra ninguna Base de Datos externa.

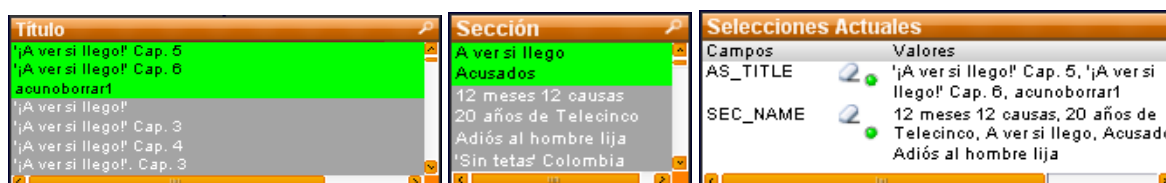
Algo que sí podría ralentizar un poco la obtención de datos son las operaciones o cálculos que tengan que hacer internamente nuestros gráficos o tablas, para así obtener determinados resultados. El tiempo de respuesta dependerá del volumen de datos a tratar, de la complejidad de las operaciones a realizar, y de la capacidad de la máquina donde se estén ejecutando los informes.

4.3.3.1.- Interacción con los Informes Dinámicos

Antes de comenzar el análisis de nuestro informe debemos tener en cuenta cómo podemos interactuar con ellos.

Nuestro informe puede dividirse en cuatro zonas básicas:

Zona de Selección de Metadatos



Nos permite seleccionar sobre qué tipo de vídeos queremos realizar nuestro análisis.

Zona de Calendario

2010														ene								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15								
lun	mar			mié			jue		vie		sáb		dom									
00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00

Podemos obtener información sobre de las descargas en función de un alto nivel de detalle, en esta zona podremos filtrar los datos en función del año, mes, día, e incluso hora correspondiente a las descargas.

Por ejemplo, podríamos obtener información sobre cuáles son los vídeos más descargados en un día concreto, en un rango de días, a unas horas determinadas, etc... Tenemos múltiples posibilidades de filtrar nuestros datos, lo que nos permite una gran flexibilidad a la hora de hacer nuestro análisis dinámico.

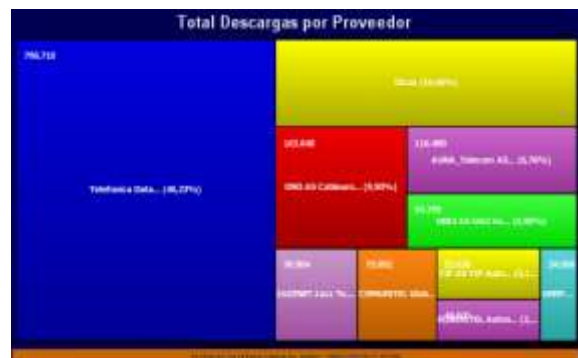
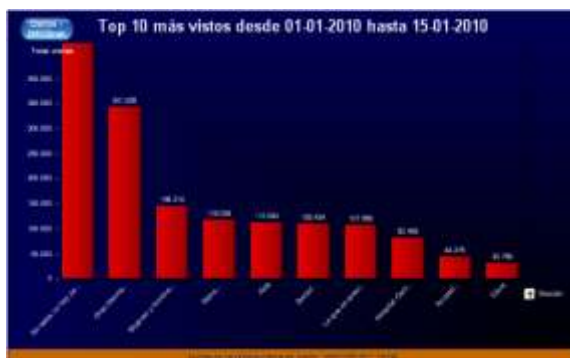
Zona de Gráficos y Medidores

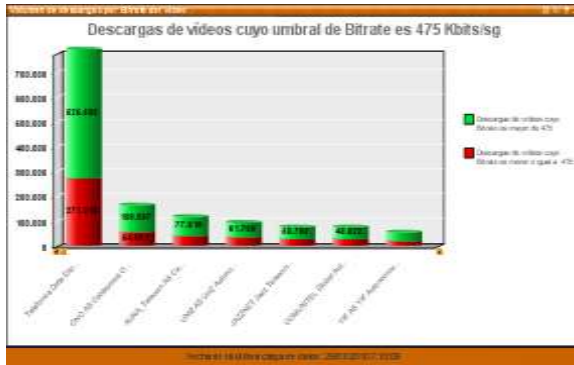


En esta zona se encuentran minimizados los diferentes gráficos que podemos analizar en la zona de visualización (para visualizarlos haremos click sobre ellos).

El icono mostrado al tener los objetos minimizados nos dará una idea del tipo de gráfico que se ha usado en cada caso (Gráfico de Sectores, Gráfico de Barras, Gráfico de Líneas, Gráfico de Bloques, Gráfico tipo Indicador, Gráfico tipo Radar, Tablas, etc...).

Zona de Visualización





Estos son algunos ejemplos de los gráficos disponibles en nuestro informe, mediante los cuales podemos hacer un análisis dinámico realizando selecciones sobre ellos, y así analizar un rango de información determinada.

Todas las selecciones que realicemos en las diferentes zonas de selección afectarán a todo nuestro informe a la hora de visualizar los datos y obtener resultados basados en operaciones lógicas o matemáticas.

Dichas selecciones quedarán reflejadas en un objeto denominado ‘Cuadro de selecciones actuales’, el cual es muy importante, ya que nos muestra las diferentes selecciones que actualmente están afectando a todos los datos de nuestro informe. Mediante dicho objeto, podemos también desactivar alguna selección concreta, y si lo que realmente queremos es volver al estado inicial del informe, entonces deberemos hacer click en la opción borrar.



Selecciones Actuales

Campos	Valores
Año Datos Download	2010
Día Datos Download	1, 2, 3, 4, 5
Month4	ene

4.3.3.2.- Ejemplos de Análisis

A continuación mostraremos algunos ejemplos sobre como interactuar con los informes para poder analizar en detalle la información disponible:

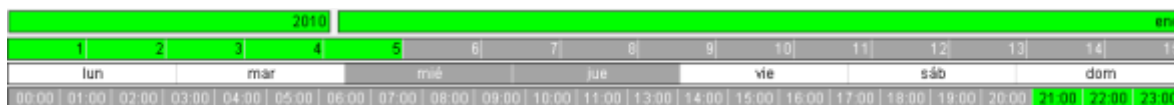
Ejemplo1:

“Queremos analizar cuál es el título del vídeo más descargado durante los 5 primeros días de Enero del 2010 desde las 21:00, y además saber cuántas descargas ha tenido y cuál ha sido el volumen descargado”

Para obtener dicha información debemos seguir los siguientes pasos:

1º Realizar las selecciones necesarias

En este caso seleccionaremos en nuestra Zona Calendario el año, mes, días y horas sobre los cuales queremos hacer nuestro análisis



2º Localizar en qué gráfico podemos obtener la información que buscamos

Para obtener la información sobre el vídeo más descargado, podemos hacerlo analizando la información mediante algún gráfico de barras, en el cual podamos ver cada título de vídeo ordenado descendientemente por el número de descargas que ha tenido.



En la configuración de nuestro gráfico insertaremos la Expresión o Expresiones que se calcularán en función de la Dimensión seleccionada, es decir, en la Dimensión seleccionaremos sobre qué contenidos se calcularán las Expresiones definidas.

Para obtener los datos que queremos, deberemos seleccionar en nuestro gráfico ‘**Títulos de Vídeos**’ como Dimensión, y ‘**Descargas de Vídeos**’ como la Expresión a evaluar en el gráfico.



El valor obtenido en la parte superior, se corresponderá con el total de descargas del vídeo más descargado en el rango de fechas seleccionado. Para obtener el volumen descargado por el vídeo con más descargas, simplemente haremos click sobre el gráfico en la parte correspondiente al vídeo a analizar.



A partir de este momento, todos los gráficos y tablas mostrarán su información y realizarán sus cálculos en base al vídeo seleccionado.

Finalmente para obtener el valor buscado podemos modificar la Expresión del Gráfico, seleccionando la opción '**Volumen total Descargado (GB)**' en el icono de la parte superior.



Otra opción para obtener la información anterior sería mediante la visualización de tablas de datos, que podemos diseñar a medida, y mediante las cuales podemos consultar información detallada de los vídeos descargados.

Detalle de Vídeos

Sección	TITULO	MB recibidos	Peso Video (MB)	Descargas	(Peso Video) X (Descargas) MB	Porcentaje Medio Descargado	Duración total descargada	Nivel Medio Descargado	Rango
Becarios	'La bella y El bestia'	1.013.323	15	94.518	1.430.333	70,77%	2.428,1 h		BT: 7a8 974 87

Ejemplo2:

“Queremos analizar cuáles son es las tres secciones con más descargas, y ver la evolución de vídeos descargados en el máximo rango de días disponible”

Para obtener dicha información debemos seguir los siguientes pasos:

1º Realizar las selecciones necesarias

Por defecto nuestro informe utiliza todos los datos disponibles, a no ser que limitemos a un rango concreto los datos que queramos analizar, ya sea mediante la selección de un rango de fechas de un tipo de vídeo, de un título concreto, o mediante cualquier otro tipo de selecciones (*nuestro ejemplo ha sido realizado sobre un informe cuyos datos disponibles corresponden a los días del 1 al 15 de Enero 2010*), por tanto no realizaremos ninguna selección en la Zona de Calendario.

En el gráfico que aparece por defecto en la pestaña ‘Descargas’ de nuestro informe mostramos un gráfico de barras con las secciones con más descargas, ordenadas descendientemente por este valor. Desde este mismo gráfico podemos seleccionar las tres secciones con más descargadas.

Dicho gráfico nos permite analizar los datos en función de las Secciones, Categorías o Títulos correspondientes a los vídeos descargados (*según seleccionemos en el icono de la parte inferior derecha*), y además nos permite modificar el tipo de agrupación (Cíclica o Jerárquica) que tendrá la Dimensión de dicho gráfico.

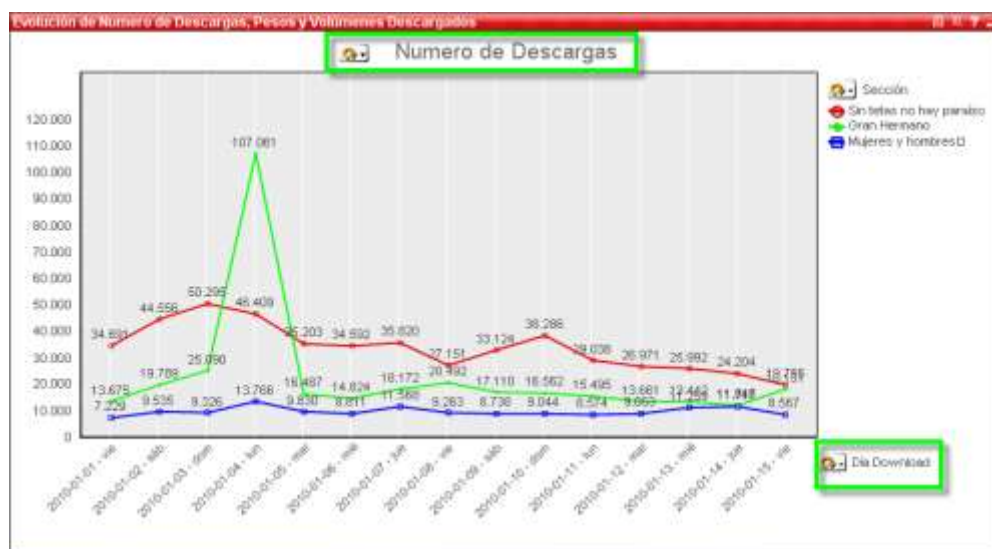


Una vez seleccionadas las tres columnas correspondientes a las Secciones con más descargas, dicha selección se quedará reflejada automáticamente en el componente de ‘Selecciones Actuales’.

2º Localizar en qué gráfico podemos obtener la información que buscamos

Para ver la evolución de descargas en las Secciones elegidas anteriormente, visualizaremos el correspondiente gráfico lineal, mediante el cual tendremos una línea por cada Sección a lo largo de todas las fechas a analizar.

Evolución de Numero de Descargas, Peso...



Mediante el icono de la parte inferior derecha de nuestro gráfico, podremos elegir entre analizar los datos a nivel de Semana, Día, Hora, e incluso Minuto.

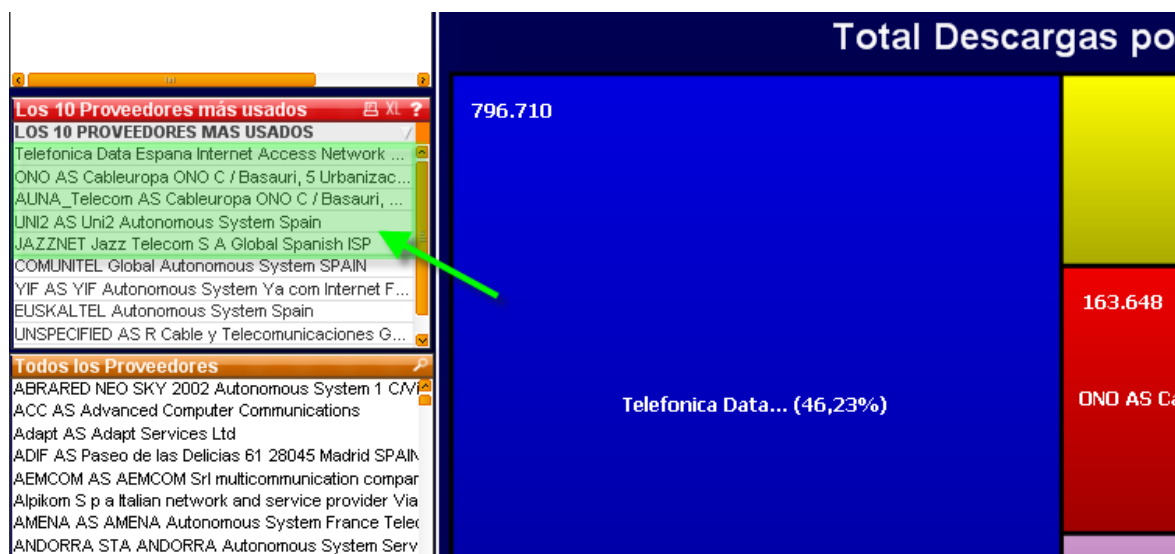
Ejemplo3:

“Queremos analizar cuáles son los cinco proveedores desde los que se realizan más descargas, y comparar la velocidad media en las descargas realizadas por los usuarios de dichos proveedores”

Para obtener dicha información debemos seguir los siguientes pasos:

1º Realizar las selecciones necesarias

Para este ejemplo debemos situarnos en la pestaña ‘Proveedores’ de nuestro informe, una vez allí, podemos seleccionar los 5 proveedores directamente en el gráfico de bloques que aparece por defecto, o bien realizar la misma selección en el listado de ‘Los 10 Proveedores más usados’ situado en la parte izquierda.



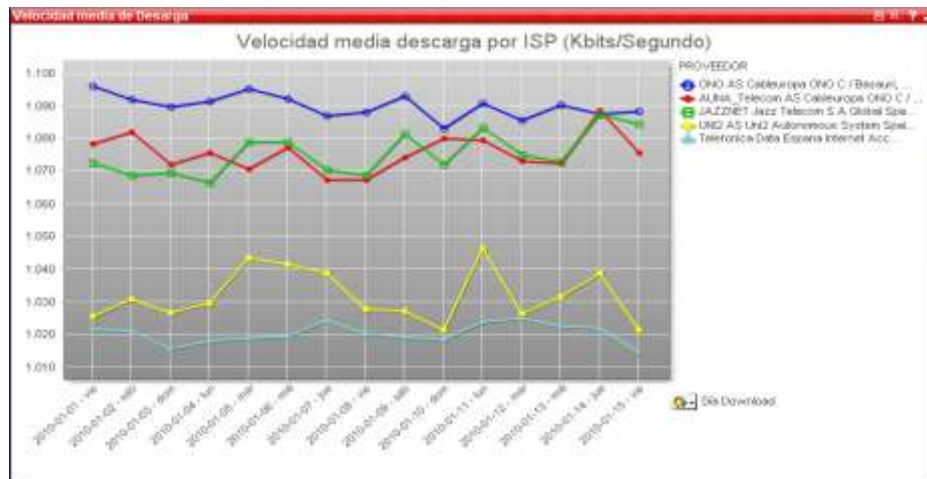
La información mostrada en nuestro informe variará en función de la selección realizada anteriormente.

2º Localizar en qué gráfico podemos obtener la información que buscamos

Visualizamos en gráfico lineal correspondiente a la evolución de la Velocidad Media de Descarga.

Velocidad media de Descarga

En este gráfico podemos observar como varía la velocidad de descarga de los proveedores seleccionados (para cada uno de los días disponibles).



5.- DEFINICIONES

ASN: Autonomous System Number (ASN por sus siglas en inglés). Es un código numérico único, el cual identifica a un conjunto de redes de direcciones IP que son administradas por un grupo de uno o más operadores de red.

Prothash: Es el protocolo mediante el cual se identifica únicamente a un objeto en la CDN. En el caso de las descargas de vídeos se usará el formato BitTorrent que es “BT:” seguido del ‘info_hash’ del vídeo.

Ejemplo: BT:0a2c08816b4cca36b6f36bfafd8f9b2179e0eedc

6.- CONCLUSIONES

Este proyecto nos ha mostrado un ejemplo de cómo se puede llevar a cabo el desarrollo completo de un sistema de Business Intelligence, cuyo objetivo es el análisis de contenidos web.

Se ha intentado dar una visión general sobre cuál es el entorno sobre el que se ha realizado este proyecto, así como un acercamiento a sus diferentes fases, para finalmente mostrar algunos ejemplos de uso, mediante los cuales podemos hacernos un idea de las grandes posibilidades de análisis que podría tener un proyecto similar aplicado a otros entornos.

Algunos de los logros que hemos obtenido con el desarrollo de este proyecto son:

- Implantar un sistema de Business Intelligence basado en el uso de Informes Dinámicos con recarga automática e histórico de datos.
- Conocer en detalle QlikView como herramienta de Business Intelligence, lo cual nos ha permitido adelantarnos a las exigencias del cliente, ofreciendo la posibilidad de diferentes análisis de datos mediante nuevos indicadores.
- Mediante la herramienta Talend se han suplido algunas carencias de QlikView en cuanto a funcionalidad como ETL (en el paso previo al diseño de los informes dinámicos).
- Mejora en la eficiencia de los procesos de recarga de datos, mediante la creación de ficheros .qvd, los cuales ofrecen una velocidad de lectura entre 10 y 100 veces más rápido que otras fuentes de datos.

Como trabajo futuro, es posible que esté orientado a realizar este tipo de informes en análisis de:

- Información de facturación.
- Alarmas sobre el estado del servicio ofrecido.
- Experiencia de usuario.

En el desarrollo completo de nuestro sistema de Análisis se han empleado 750 horas aproximadamente, de las cuales el 70% se han dedicado a llevar a cabo la primera fase de nuestro desarrollo, y a las tareas relacionadas con el Análisis sobre el entorno y modo de implantación del sistema. El 30 % restante ha sido empleado básicamente en el diseño y creación de los informes, así como en la evaluación y comprobación de los resultados.

Como conclusión final, me gustaría resaltar cómo algunas de las capacidades adquiridas durante el transcurso de mi carrera en la Universidad Rey Juan Carlos (tales como: la capacidad de análisis, y la adaptabilidad al desarrollo en nuevos entornos), han sido claves para éxito del trabajo realizado.

La experiencia que este proyecto me ha proporcionado en cuanto a herramientas de análisis y procesamiento de datos, tales como QlikView y Talend, han complementado notablemente los conocimientos teóricos y prácticos adquiridos en la carrera de Ingeniería Técnica en Informática de Sistemas.

7.- BIBLIOGRAFÍA

- <http://www.qlikview.com/> (Web de QlikView)

Versión 8.5 para Microsoft Windows® 1ª edición, Madrid, España, Mayo 2008
Autor: QlikTech International AB /HIC/KHN/JNN/MSJ/MMG (Manuales de QlikView)

- <http://talend.com> (Web de talend)
- Documento CacheLogic <http://www.velocix.com>
- Wikipedia, <http://en.wikipedia.org>

ANEXO

Código de mapeo de IP's con su ASNS correspondiente

PARA MAPEAR LA IP OBTENIENDO CODIGO ASN

```
package routines;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class GuardianUtils {

    private static boolean mapperStarted = false;
    private static List<CIDRMap> maps;

    public static class CIDRMap {
        private long mask;
        private Map<Long, Long> blocks_Asn;

        public CIDRMap(long mask, Map<Long, Long> blocks_Asn) {
            this.mask = mask;
            this.blocks_Asn = blocks_Asn;
        }

        public long getMask() {
            return mask;
        }

        public void setMask(int mask) {
```

```
        this.mask = mask;
    }

    public Map<Long, Long> getBlocks_Asn() {
        return blocks_Asn;
    }

    public void setBlocks_Asn(Map<Long, Long> blocks_Asn) {
        this.blocks_Asn = blocks_Asn;
    }
}

private static long IpToLong( String ip ) {
    String direccion = ip.split("/")[0];
    String[] subred = direccion.split("\\.");
    long a = Long.parseLong(subred[0]);
    long b = Long.parseLong(subred[1]);
    long c = Long.parseLong(subred[2]);
    long d = Long.parseLong(subred[3]);
    long valor = d | (c << 8) | (b << 16) | (a << 24);
    return valor;
}

// private static long MaskToLong( String ip ) {
//     String mascara = ip.split("/")[1];
//     return Long.parseLong(mascara);
// }

private static void startAsnMapper() throws Exception {
    java.sql.Connection conn = null;
```

```
String url_tMysqlOutput_1 =
"jdbc:mysql://localhost:3306/guardian?noDatetimeStringSync=true";

String dbUser_tMysqlOutput_1 = "root";

String dbPwd_tMysqlOutput_1 = "admin";

java.lang.Class.forName("org.gjt.mm.mysql.Driver");

conn =
java.sql.DriverManager.getConnection(url_tMysqlOutput_1,
dbUser_tMysqlOutput_1, dbPwd_tMysqlOutput_1);

conn.setAutoCommit(false);

String query1 = "SELECT MASK_N FROM prefixlist GROUP BY
MASK_N ORDER BY MASK_N DESC";

String query2 = "SELECT BLOCK_N, ASN, MASK_N FROM prefixlist
WHERE MASK_N=?";

java.sql.PreparedStatement pstmt1 =
conn.prepareStatement(query1);

java.sql.PreparedStatement pstmt2 =
conn.prepareStatement(query2);

maps = new ArrayList<CIDRMap>();
java.sql.ResultSet rs1 = pstmt1.executeQuery();
while (rs1.next()) {
    long mask = rs1.getLong(1);
    final Map<Long,Long> mapa = new HashMap<Long,Long>();
    pstmt2.setLong(1, mask);
    java.sql.ResultSet rs2 = pstmt2.executeQuery();
    while (rs2.next()) {
        long block = rs2.getLong(1);
        Long asn = rs2.getLong(2);
        mapa.put(block, asn);
    }
    maps.add(new CIDRMap(mask, mapa));
}
```

```
        pstmt1.close();
        pstmt2.close();
        conn.close();
        mapperStarted = true;
    }

    public static Long mapAsn(String peerEndpoint) throws Exception {
        if (!mapperStarted) {
            startAsnMapper();
        }
        // Obtenemos la representación numérica de la IP
        try {
            String ip = peerEndpoint.split(":")[0];
            long valor = IpToLong(ip);
            // Recorremos los mapas
            for (CIDRMap map : maps) {
                long mask = map.getMask();
                Map<Long, Long> blocks = map.getBlocks_Asn();
                long buscar = valor >> (32 - mask);
                Long asn = (Long) blocks.get(buscar);
                if (asn != null) {
                    return asn;
                }
            }
        } catch (RuntimeException e) {
            System.err.println(e.getMessage());
        }
        return new Long(0);
    }
}
```

Código Script QlikView (Recarga_QVDS_MesAnterior)

```
SET ThousandSep='.';
SET DecimalSep=',';
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';
SET TimeFormat='h:mm:ss';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff]';
SET MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';

/* Para Actualizar el Listado de .qvds del Directorio Historicos
del mes anterior, si el dia de ayer es menor de dia 6 */
ODBC CONNECT TO HistoricoDownloads (XUserId is MPEUCZdNPDdGF,
XPassword is IdWQbQRMNLacGCD);

SET DateFormat='YYYY/MM/DD';
LET Año = Year(Today());
LET Mes = Num(Month(Today()));
LET Dia = day(Today());

IF Mes = 1 THEN
    LET MesAnterior = 12;
    LET AñoMesAnterior = $(Año)-1;
ELSE
    LET MesAnterior = $(Mes)-1;
    LET AñoMesAnterior = $(Año);
END IF

LET inicio= day(Today()-5);

IF Dia < 6 THEN
for i=$(inicio) to 31

Download:
LOAD `Download_ID` as Download,
`End_Timestamp` as [ID End Timestamp],
`Total_Download_Duration` as [Total Download Duration],
`Protohash` as R_CONTENTID,
`Protohash` as R_CONTENTID_CONTAR,
`Total_Bytes_Received` as [Total Bytes Received],
`Peer_Endpoint` as [Peer Endpoint],
`Peer_ID` as [Peer ID],
`User_Agent` as [User Agent],
`Pieces_in_File` as [Pieces in File],
`Initial_Pieces` as [Initial Pieces],
`Final_Pieces` as [Final Pieces],
`Start_Timestamp` as [Start Timestamp],
HoraDownload,
MinutoDownload,
FechaDatosDownload as FechaDatosDonwload,
```



```
AnnoDatosDownload as AñoDatosDonwload,  
MesDatosDownload as MesDatosDonwload,  
DiaDatosDownload as DíaDatosDonwload,  
ASN,  
WeekDay(FechaDatosDownload) as DiaSemanaDownload;  
  
SQL SELECT *, DATE_FORMAT(Start_Timestamp,'%H:00') AS HoraDownload,  
DATE_FORMAT(Start_Timestamp,'%H:%i') AS MinutoDownload  
  
FROM guardian.downloads$(AñoMesAnterior)_$(MesAnterior)  
WHERE (DiaDatosDownload = $(i));  
  
store Download into  
Descargas_y_Proveedores\$(AñoMesAnterior)_$(MesAnterior)\Historicos  
\H_Downloads$(AñoMesAnterior)_$(MesAnterior)_$(i).qvd;  
  
drop table Download;  
  
next  
  
END IF
```

Código Script QlikView (Recarga_QVDS_MesActual)

```
SET ThousandSep='.';  
SET DecimalSep=',';  
SET MoneyThousandSep='.';  
SET MoneyDecimalSep=',';  
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';  
SET TimeFormat='h:mm:ss';  
SET DateFormat='DD/MM/YYYY';  
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff]';  
SET MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';  
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';  
  
ODBC CONNECT TO HistoricoDownloads (XUserId is MPeUCZdNPdGF,  
XPassword is IdWQbQRMNLacGCD);  
  
SET DateFormat='YYYY/MM/DD';  
LET Año = Year(Today()-1);  
LET Mes = Num(Month(Today()-1));  
LET Dia = day(Today()-1);  
LET DiaHoy = day(Today());  
  
IF DiaHoy > 1 THEN
```

```
IF Dia < 6 THEN

  for i=1 to Dia

    Download:
    LOAD `Download_ID` as Download,
    `End_Timestamp` as [ID End Timestamp],
    `Total_Download_Duration` as [Total Download Duration],
    `Protohash` as R_CONTENTID,
    `Protohash` as R_CONTENTID_CONTAR,
    `Total_Bytes_Received` as [Total Bytes Received],
    `Peer_Endpoint` as [Peer Endpoint],
    `Peer_ID` as [Peer ID],
    `User_Agent` as [User Agent],
    `Pieces_in_File` as [Pieces in File],
    `Initial_Pieces` as [Initial Pieces],
    `Final_Pieces` as [Final Pieces],
    `Start_Timestamp` as [Start Timestamp],
    HoraDownload,
    MinutoDownload,
    FechaDatosDownload as FechaDatosDownload,
    AnnoDatosDownload as AñoDatosDownload,
    MesDatosDownload as MesDatosDownload,
    DiaDatosDownload as DíaDatosDownload,
    ASN,
    WeekDay(FechaDatosDownload) as DiaSemanaDownload;
    SQL SELECT *, DATE_FORMAT(Start_Timestamp,'%H:00') AS
    HoraDownload, DATE_FORMAT(Start_Timestamp,'%H:%i') AS
    MinutoDownload
    FROM guardian.downloads$(Año)_$(Mes)
    WHERE (DiaDatosDownload = $(i));

    store Download into
    Historicos\H_Downloads$(Año)_$(Mes)_$(i).qvd;

    drop table Download;

  next

ELSE
LET inicio=day(Today()-5);
  For i=$(inicio) to Dia
    Download:
    LOAD `Download_ID` as Download,
    `End_Timestamp` as [ID End Timestamp],
    `Total_Download_Duration` as [Total Download Duration],
    `Protohash` as R_CONTENTID,
    `Protohash` as R_CONTENTID_CONTAR,
    `Total_Bytes_Received` as [Total Bytes Received],
    `Peer_Endpoint` as [Peer Endpoint],
    `Peer_ID` as [Peer ID],
    `User_Agent` as [User Agent],
    `Pieces_in_File` as [Pieces in File],
    `Initial_Pieces` as [Initial Pieces],
    `Final_Pieces` as [Final Pieces],
    `Start_Timestamp` as [Start Timestamp],
```

```
HoraDownload,  
MinutoDownload,  
FechaDatosDownload as FechaDatosDonwload,  
AnnoDatosDownload as AñoDatosDonwload,  
MesDatosDownload as MesDatosDonwload,  
DiaDatosDownload as DíaDatosDonwload,  
ASN,  
WeekDay(FechaDatosDownload) as DiaSemanaDownload;  
SQL SELECT *, DATE_FORMAT(Start_Timestamp,'%H:00') AS  
HoraDownload, DATE_FORMAT(Start_Timestamp,'%H:%i') AS  
MinutoDownload  
FROM guardian.downloads$(Año)_$(Mes)  
WHERE (DiaDatosDownload = $(i));
```

```
store Download into Historicos\H_Downloads$(Año)_$(Mes)_$(i).qvd;  
drop table Download;
```

```
next
```

```
END IF
```

```
END IF
```

Código Script QlikView (Descarga-Videos)

```
SET ThousandSep='.';  
SET DecimalSep=',';  
SET MoneyThousandSep='.';  
SET MoneyDecimalSep=',';  
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';  
SET TimeFormat='h:mm:ss';  
SET DateFormat='DD/MM/YYYY';  
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff]';  
SET MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';  
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';
```

```
Directory;
```

```
//InputField CF.PlataformaWeb;
```

```
InputField CF.Mes;
```

```
InputField CF.Resto2008;
```

```
InputField CF.2009;
```

```
ODBC CONNECT TO [MiTele;DBQ=T5PRO ] (XUserId is XXXXXX, XPassword  
is XXXXXX);
```

```
set ErrorMode=0;
```

```
Download:
```

```
load *,if( [Pieces in File] = 0 , 0 , Num(([Final Pieces]/[Pieces  
in File]),'#.##0,00')) as Porcentaje_descargado  
from Historicos\H_Downloads*.qvd (qvd);
```

```
if ScriptError=7 then  
exit script;  
//sin archivo;  
end if
```

```
inner join
```

```
LOAD  
"R_CONTENTID" as R_CONTENTID,  
"ASDV_BITRATE" as Bitrate_Video;  
SQL SELECT DISTINCT "R_CONTENTID", "ASDV_BITRATE"  
FROM CMS."ASSET_DATA_VIDEO" A, CMS."ASSET_DATA_VIDEO_ASD_CDN" B,  
CMS."ASSET_DATA_VIDEO_ASD_CDN_L" C  
WHERE ((A.ID=B."FROM_ASSET_DATA_VIDEO") AND (B.ID=C."REL_ID"));  
//*****
```

```
LOAD ID as VIDEO_CDN_ID,  
"FROM_ASSET_DATA_VIDEO" as VIDEO_ID;  
SQL SELECT *  
FROM CMS."ASSET_DATA_VIDEO_ASD_CDN";  
//*****
```

```
LOAD "REL_ID" as VIDEO_CDN_ID,  
"R_CONTENTID";  
SQL SELECT *  
FROM CMS."ASSET_DATA_VIDEO_ASD_CDN_L";  
//*****
```

```
LOAD ID as VIDEO_ID,  
"ASDV_BITRATE",  
"ASDV_DURATION",  
"ASD_THUMB",  
"ASDV_WIDTH",  
"ASDV_SIZE",  
"ASD_FILE",  
"ASDV_HEIGHT";  
SQL SELECT *  
FROM CMS."ASSET_DATA_VIDEO";  
//*****
```

```
LOAD ID AS ID_AAAV,  
"FROM_ASSET" as ASSET_ID,  
"TO_ASSET_DATA_VIDEO" as VIDEO_ID;  
SQL SELECT *  
FROM CMS."ASSET_AS_ASD_VID";  
//*****
```

```
LOAD ID as ASSET_ID,  
STATE,
```

```
DOMAIN,  
"ADD_DATE",  
"ADD_USER",  
"UPD_DATE",  
"UPD_USER",  
"PUB_DATE",  
"ENDPUB_DATE",  
"BLOCKING_USER",  
"ASSIGNED_ROLE",  
"AS_ALLOW_CAPTURE",  
"AS_ALLOW_COMMENTS",  
"AS_FROM_METHOD_ID",  
"AS_ALLOW_LINK",  
"AS_ALLOW_VIDEOTECA",  
"AS_ALLOW_COPY",  
"AS_ALLOW_MOVIL",  
"AS_ALLOW_SEND",  
"AS_BRODAT",  
"AS_FLY";  
SQL SELECT *  
FROM CMS.ASSET;  
//*****  
  
LOAD "CONTENT_ID" as ASSET_ID,  
"AS_SOURCE_REFERENCE",  
"AS_DESCRIPTION",  
"AS_SHORT_TITLE",  
"AS_SHORT_DESCRIPTION",  
"AS_FOOT",  
"AS_ORIGIN",  
"AS_ORIGINAL_TITLE",  
"AS_ALT",  
"AS_TITLE";  
SQL SELECT *  
FROM CMS."ASSET_L";  
//*****  
  
LOAD "TIPO_CAT_ID",  
"TIPO_CAT_CONT_ID" as ASSET_ID,  
"TIPO_CAT_CAT_ID" as CAT_ID,  
"TIPO_CAT_IDNAME",  
"TIPO_CAT_MAIN";  
SQL SELECT *  
FROM CMS."ASSET_CATEGORIES"  
WHERE (TIPO_CAT_CAT_ID<>128) AND (TIPO_CAT_CAT_ID<>433) AND  
(TIPO_CAT_CAT_ID<>133);  
//*****  
  
LOAD "CAT_ID",  
"CAT_PARENT_ID",  
"CAT_WEIGHT",  
"CAT_ORDER_TYPE",  
"CAT_NODE_TYPE",  
"CAT_NODE_STATE",  
"CAT_IDNAME";  
SQL SELECT *
```

```
FROM CMS."DTY_CATEGORY"  
WHERE (CAT_ID<>128) AND (CAT_ID<>433) AND (CAT_ID<>133);  
//*****  
  
LOAD "CAT_LOC_CAT_ID" as CAT_ID,  
"CAT_LOC_LOCALE",  
"CAT_LOC_NAME",  
"CAT_LOC_COMMENT",  
"CAT_LOC_ALTS",  
"CAT_LOC_KEYWORDS";  
SQL SELECT *  
FROM CMS."DTY_CATEGORY_L"  
WHERE (CAT_LOC_CAT_ID<>128) AND (CAT_LOC_CAT_ID<>433) AND  
(CAT_LOC_CAT_ID<>133);  
//*****  
  
LOAD "SC_TIPOCAT_ID" as STC_ID,  
"SC_CATEGORY_ID" as CAT_ID;  
SQL SELECT *  
FROM CMS."DTY_SECTION_CATEGORY"  
WHERE (SC_CATEGORY_ID<>128) AND (SC_CATEGORY_ID<>433) AND  
(SC_CATEGORY_ID<>133);  
//*****  
  
LOAD "STC_ID",  
"STC_SECTION_ID" as SEC_ID,  
"STC_IDNAME",  
"STC_NAME",  
"STC_JOIN_CAT_TYPE";  
SQL SELECT *  
FROM CMS."DTY_SECTION_TIPO_CAT";  
//*****  
  
LOAD "SEC_ID",  
"SEC_IDNAME",  
"SEC_FRONTAL_ID",  
"SEC_NAME",  
CEILING,  
WINDOW,  
ORDERFIELDS,  
ORDERTYPE,  
"VIEW_TYPE";  
SQL SELECT *  
FROM CMS."DTY_SECTION";  
//*****
```

```
ASNS:  
Load * from ASNS.qvd (qvd);
```

```
Disponibilidad_Dias:  
LOAD  
FechaDatosDownload as FDD, MAX(hour([Start Timestamp])) as Max_Hora
```

resident Download
group by FechaDatosDownload;