

Inteligencia de Negocio en centros terapéuticos especializados en trastornos alimentarios

Javier Vázquez Bodi

Trabajo de final de máster Universitat Oberta de Catalunya

6-3-2016

Título del trabajo	<i>Inteligencia de Negocio en centros terapéuticos especializados en trastornos alimentarios</i>
Nombre del autor	<i>Javier Vázquez Bodi</i>
Nombre del consultor	<i>David Amorós Alcaraz</i>
Fecha de entrega	<i>06/2016</i>
Área del Trabajo Final	<i>Business Intelligence</i>
Titulación	<i>Master en Ingeniería Informática</i>
Resumen del trabajo	
<p>Desde la revolución industrial hasta nuestros días, las organizaciones van incorporando en sus procesos descubrimientos que sacan rendimiento a cosas que eran anteriormente eran impensables, desde el trabajo en cadena, pasando por el reciclaje de residuos hasta el caso que ahora nos ocupa consistente en explotar y analizar los datos producidos por la propia organización en su día a día.</p> <p>Es tal la relevancia y el grado de madurez que han alcanzado estas técnicas y procedimientos, que a ciertas áreas de esta especialidad se les llama minería de datos y otras Inteligencia del negocio, ya cada vez son más en cantidad y más complejas las decisiones que se logran tomar con ellas.</p> <p>Este proyecto tiene como fin, crear un almacén de datos para poder explotar y analizar los datos obtenidos de una serie de pacientes con trastornos alimentarios enviados a los centros terapéuticos distribuidos en diferentes puntos geográficos. En su contenido, después de un periodo de análisis, se propone una solución basada en un gestor de base de datos MySQL, una ETL con Pentaho Integration y una solución de Bussiness Intelligence con Pentaho BI server como herramienta de análisis y generación de informes.</p> <p>Utilizando dicha solución se logra responder a una serie de preguntas planteadas al inicio en forma de requisitos en las cuales se busca encontrar relaciones como los estados emocionales del paciente o sus actividades realizadas y los episodios que sufre.</p>	
Abstract	
<p>From Industrial Revolution to nowadays, organizations are including on its processes some findings, that, take profit of things that were unthinkable before, such as the assembly-line work, recicyng and also as the one we are refering to now, that consist in take profit and analise date produced by the organization in its day-to-day.</p> <p>Such is the relevance, and the maturity reached by this processes that some áreas of this specialty are called data mining or bussines intelligence. Everytime is greatest the</p>	

amount and also more complex the decisions taken by using such techniques .

The target of this Project is to create a datawarehouse to take profit and analyse the data produced by the patients with eating disorders and sent to the therapeutic centers distributed geographically. In the content of this Project, and after an analysis period, it's is proposed a solution based on a Mysql as database manager and Pentaho BI Server as the tool to analyse and generate reports.

By using this solution we get the answers to a set of questions seted out as requirements at the begining, where we try to find the relation between emotional states or activities performed and the attacks suffered.

Contenido

Contenido	4
1. Introducción	7
1.1. Contextualización del proyecto	7
1.2. Particularidades, restricciones, límites, asunciones y reglas de negocio	8
1.3. Objetivos	8
1.4. Definición de requisitos	9
1.5. Metodologías y estándares de trabajo	9
1.6. Fases del proyecto	10
1.6.1. Definición de objetivos.....	10
1.6.2. Definición de requerimientos, límites, restricciones y asunciones.....	10
1.6.3. Diseño y modelización.....	10
1.6.4. Implementación	10
1.6.5. Control de calidad	11
1.6.6. Revisión	11
1.6.7. Gestión del proyecto	12
2. Planificación	12
2.1. Participantes del proyecto	12
2.2. Tareas	12
2.3. Fechas clave e hitos	13
2.4. Cronograma	13
2.5. Riesgos	13
2.6. Análisis de costes	14
3. Diseño	15
3.1. Arquitectura propuesta	18
3.2. Asunciones	20
3.3. Estándares de nomenclatura	20
3.4. Selección de herramientas a utilizar	21
3.5. Estructura de datos de origen	27
3.6. Modelo conceptual de Datawarehouse	29
3.6.1. Elementos del modelo: dimensiones y hechos.	30
3.7. Decisiones tomadas sobre el diseño	33
3.8. Herramienta de calidad de datos	33
4. Implementación	34

4.1.	Configuración del servidor de MySQL	34
4.2.	Configuración del servidor de ETL	34
4.3.	Modelo físico	35
4.4.	Creación de la base datos	40
4.5.	Creación de las tablas	40
4.6.	Creación de las vistas del datamarts	40
4.7.	ETL	42
4.8.	Cubos	54
4.9.	Los informes	63
4.9.1.	Informe 1.- Relacion_actividad_episodio.....	63
4.9.2.	Informe 2.- Relación episodios y estados de animo	64
4.9.3.	Informe 2.- Relación episodios y estados de ánimo (inversa).....	65
4.9.4.	Informe 3.- Relación episodio entorno - inversa totales.....	66
4.9.5.	Informe 3.b.- Relación episodio entorno (ciudad)	67
4.9.6.	Informe 4.d- Evolución paciente desde datamart (evolución pacientes críticos).....	68
4.9.7.	Informe 4.d- Evolución paciente desde datamart (evolución todos)	69
4.9.8.	Informe 4.- Evolución paciente (agrupado)	70
4.9.9.	Informe 4.- Evolución paciente (detalle).....	71
4.9.10.	Informe 4.- Evolución paciente (temporal)	72
4.9.11.	Informe 5.b.- Evolución de pacientes comparativa dos terapeutas	73
4.9.12.	Informe 5.b.- Evolución de pacientes todos los terapeuta	74
4.9.13.	Informe 5.b.- Evolución de pacientes todos los terapeuta por episodio	75
4.9.14.	Informe 5.c. Evolución de episodios de paciente en centros médicos	76
4.9.15.	Informe 5.e.- Evolución de pacientes comparativa dos centros.....	77
4.9.16.	Informe 6.b.- Relación episodio y tiempo – franja.....	78
4.9.17.	Informe 6.c.- Relación episodio y tiempo – mes	79
4.9.18.	Informe 6.d.- Relación episodio y tiempo - día_semana	80
4.9.19.	Informe 6.d.- Relación episodio y tiempo - día_semana – total	81
4.9.20.	Informe 7.- Análisis de episodios que actividades se hacían (agrupados).....	82
4.9.21.	Informe 7.- Análisis de episodios que actividades se hacían (agrupados).....	83
4.9.22.	Informe 7.- Evolución de actividad de un episodio a siguiente episodio.....	84
5.	Explotación de datos	85
6.	Conclusiones	88
7.	Apéndices	90

8. Puntos de mejora	107
9. Referencias	108

1. Introducción

El proyecto que a continuación se presenta forma parte se ha realizado como entregable de la asignatura de Trabajo de final de máster de la Universitat Oberta de Catalunya, realizado por mi persona: Javier Vázquez Bodi.

En ella, se requiere del alumno la planificación, el diseño y la implementación de un almacén de datos (Datawarehouse) y la explotación de la inteligencia de negocio, utilizando los datos procedentes de diferentes centros, este almacén de datos se utilizará como base para construcción de herramientas de reporte que exploten esos datos.

Debido a que actualmente estoy involucrado laboralmente en la construcción de un Datawarehouse, me gustaría que este proyecto me aportase conocimiento acerca de áreas de la Bussiness Inteligence que, por el momento, no he tenido la oportunidad de adquirir, como es el área de Reporting, puesto que la gran parte del tiempo me he involucrado en la fase de ETL's y Modelaje de datos, pero no he frecuentado demasiado proyectos en donde poder profundizar en el Reporting. También me gustaría conocer herramientas de Open Source de este ámbito, puesto que, las que actualmente conozco son básicamente las comerciales como Informática PowerCenter, IBM DB2, Microstrategy, Wherescape RED, etc.

1.1. Contextualización del proyecto

Los trastornos de la conducta alimentaria (TCA) engloban diversas enfermedades crónicas y progresivas que, a pesar de que se manifiestan a través de la conducta alimentaria, en realidad consisten en una gama muy complejas de síntomas entre los que prevalece una alteración o distorsión de la auto-imagen corporal, un gran temor en subir de peso y la adquisición de una serie de valores a través de la imagen corporal.

Durante los últimos 30 años, los TCA han surgido como enfermedades cada vez más frecuentes, sobretodo dentro de los adolescentes. Los estudios sobre la frecuencia de estos trastornos muestran un aumento preocupante, principalmente dentro de la población de las mujeres. Estos trastornos se muestran en una población de 10 a 1 de mujeres respecto a los hombres, aunque en los últimos años se ha incrementado el número de hombres que padecen estas enfermedades. Generalmente, este incremento se ha dado entre los 14 años y los 20.

En la explosión de la creación de apps para el entorno clínico, son varias las propuestas que intentan ofrecer una solución para el seguimiento de pacientes con este tipo de enfermedades. Este tipo de apps ofrecen, a través de un entorno atractivo, la posibilidad de que el paciente pueda tener una herramienta y le dé soporte en todo momento y que le permita informar de sus actividades, así como de los éxitos y recaídas durante el tratamiento o terapia.

Eso ha provocado la existencia de numerosos sistemas que, desde hace algunos años, recogen y almacenan información sobre los hábitos, actividades, estados de ánimo que los pacientes con este tipos de enfermedades generan.

Estos sistemas actualmente están formados por una app con la cual se genera una serie de información sobre la actividad del paciente y un back office, que es utilizado por los terapeutas para consultar a la actividad de los pacientes, enviar mensajes, etc.

1.2. Particularidades, restricciones, límites, asunciones y reglas de negocio

El entorno en el cual el sistema se va a implantar se rige por las siguientes restricciones o reglas:

Terapeutas

- Un terapeuta en principio solo trabaja para un centro médico. Se quiere tener información geográfica para los diferentes centros médicos.

Tiempo

- Los pacientes utilizan su App para enviar la información de cualquier momento, y no, necesariamente, en el mismo momento, ni a la misma hora.
- Es posible que un día se hagan muchos registros y otro ninguno.
- EL nivel de agrupación mínimo de toda la información será un día.

Pacientes

- No necesariamente los pacientes tienen el centro médico en su lugar de residencia. Se quiere tener toda la información geográfica a nivel población y región.

Indicadores

- Los principales indicadores que son necesarios para poder hacer un seguimiento preciso de los pacientes son las diferentes actividades que realizan durante el día, los episodios de crisis el estado de ánimo y las diferentes comidas.

1.3. Objetivos

El objetivo de este trabajo es el diseño y la implementación de un sistema de Business Intelligence que facilite la adquisición del almacén y la explotación de los datos asociados a los pacientes con enfermedades de trastornos alimenticios provenientes de diferentes centros médicos. Por lo tanto, el trabajo tendrá los siguientes objetivos:

- Diseñar el almacén de datos que permita almacenar la información adquirida desde los diferentes orígenes de datos situados en cada centro médico. Teniendo en cuenta que cada centro médico estará formado por un grupo de terapeutas que tendrán asignados un cierto número de pacientes.
- Implementar este almacén de datos y programar los procesos de ETL (Extracción, transformación y carga) que permitan alimentar el Datawarehouse a partir de los ficheros base facilitados.
- Analizar las diferentes plataformas BI Open Source disponibles en el mercado que nos permitan explotar la información almacenada.

- Seleccionar e implantar una de estas herramientas Open Source de tal forma que se disponga de una capa de aplicación para el análisis de la información.

1.4. Definición de requisitos

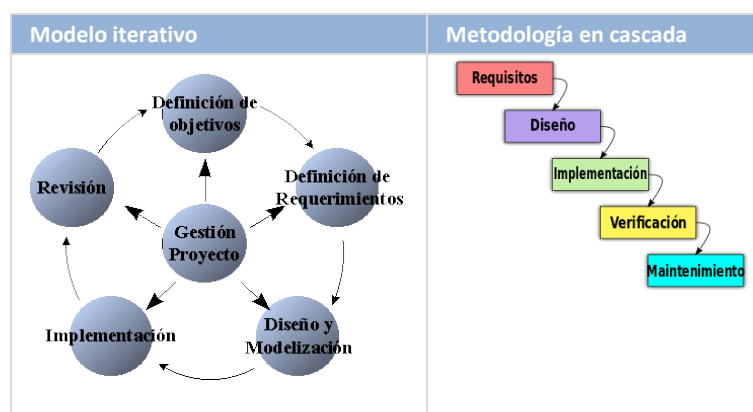
El sistema a implementar debería satisfacer las siguientes necesidades:

- Averiguar qué relación hay entre las actividades realizadas y los episodios de crisis.
- Averiguar si se puede establecer algún tipo de relación entre los valores de los diferentes estados de ánimo y los episodios de crisis.
- Averiguar si se puede establecer alguna relación a nivel geográfico, por ejemplo el entorno rural o urbano.
- Averiguar cuál ha sido la evolución de los diferentes pacientes a lo largo de tiempo.
- Conocer si hay alguna diferencia de la evolución de pacientes a nivel de centro médico o terapeuta.
- Conocer si se puede establecer alguna relación entre episodios de crisis y el momento del día o de la semana o del año.
- La granularidad mínima debería ser un día.
- Saber si la realización de actividades mejora o empeora el estado de ánimo de los pacientes.

1.5. Metodologías y estándares de trabajo

Para la realización de este proyecto se considera apropiado seguir un modelo clásico de desarrollo en cascada aprovechando en ciertas circunstancias ciertas particularidades de los modelos ágiles y del desarrollo evolutivo.

El modelo de gestión de calidad estará basado en las pruebas, con un fase de pruebas de integración y otra de aceptación de usuario posterior a la implementación, adaptado, evidentemente, al equipo humano del que se dispone en el proyecto y aplicados sobre un único entorno específico para las pruebas de sistema y las pruebas de usuario.



1.6. Fases del proyecto

Como resultado de lo anteriormente expuesto se han detectado que serán necesarias las siguientes fases durante el proyecto:

1.6.1. Definición de objetivos

Durante esta fase se plantearán una serie de necesidades por parte de los interesados y se planteará una posible solución a muy alto nivel para dar solución a dichas necesidades, se plantearán las metas y las expectativas del proyecto.

1.6.2. Definición de requerimientos, límites, restricciones y asunciones

En este momento se analizará el escenario donde el producto tiene que asentarse, y concretarán cada una de esas necesidades y demandas, marcando los límites del alcance, las restricciones y asunciones para cada una de la lista de requerimientos a cumplir.

Durante esta fase se hará una selección de herramientas y plataformas sobre las cuales se va a desarrollar el producto.

1.6.3. Diseño y modelización

Los requerimientos de información identificados durante la anterior fase proporcionarán las bases para realizar el diseño y la modelización del DataWarehouse.

En esta fase se identificarán las fuentes de los datos (sistema operacional, fuentes externas,..) y las transformaciones necesarias para, a partir de dichas fuentes, obtener el modelo lógico de datos del DataWarehouse. Este modelo estará formado por entidades y relaciones que permitirán resolver las necesidades de negocio de la organización.

El modelo lógico se traducirá posteriormente en el modelo físico de datos que se almacenará en el DataWarehouse y que definirá la arquitectura de almacenamiento del DataWarehouse adaptándose al tipo de explotación que se realice del mismo.

Se identificarán las tablas de hechos y las dimensiones, así como las métricas necesarias para poder satisfacer las exigencias de los diferentes tipos de reportings necesarios, y los niveles de agrupación requeridos.

1.6.4. Implementación

La implantación de un DataWarehouse lleva implícitos los siguientes pasos:

- Extracción de los datos del sistema operacional y transformación de los mismos.

- Carga de los datos validados en el DataWarehouse. Esta carga deberá ser planificada con una periodicidad que se adaptará a las necesidades de refresco detectadas durante las fases de diseño del nuevo sistema.
- Explotación del Data Warehouse mediante diversas técnicas dependiendo del tipo de aplicación que se dé a los datos:
 - Query & Reporting
 - On-line analytical processing (OLAP)
 - Executive Information System (EIS) ó Información de gestión
 - Decision Support Systems (DSS)
 - Visualización de la información
 - Data Mining ó Minería de Datos, etc.

Parte de la información necesaria para mantener el control sobre la calidad y carga de los datos se almacenará en metadatos técnicos (metadatos que informan sobre el origen, o la forma en la que los datos se han cargado, que no aportan información sobre el negocio).

Con la finalización de esta fase se obtendrá un Data Warehouse y un sistema de Business Intelligence disponible para ser explotado por los analistas médicos.

La implementación se desarrollara sobre un entorno de Desarrollo exclusivo.

El despliegue de la aplicación se realizará en un entorno exclusivo de Producción.

1.6.5. Control de calidad

El control de calidad estará basado en pruebas planificadas y se dividirán en 3 tipos:

- Pruebas unitarias.
- Pruebas integradas.
- Pruebas de aceptación de usuario.

En estas pruebas se definirá un criterio para la aceptación o rechazo del software/componentes que se estará testeando en cada una de ellas.

Las pruebas unitarias se realizaran en desarrollo, sin embargo, las pruebas de sistema e integración y de usuario se realizaran en el entorno producción.

1.6.6. Revisión

Esta fase, aunque está fuera del tiempo del trabajo del máster, y siguiendo la hoja de ruta de un proyecto evolutivo, se recomienda hacer una revisión después de alrededor de 6 meses tras la puesta en marcha del funcionamiento del sistema y la funcionalidad, los puntos fuertes a potenciar y posibles mejoras a desarrollar y se traza un plan para llevar a cabo esta iniciativa.

1.6.7. Gestión del proyecto

Durante el ciclo de vida del proyecto serán necesarias diferentes tareas de coordinación, representación, gestión y planificación de los diferentes recursos y tiempos, que se deberán desempeñar bajo el rol de gestor del proyecto.

2. Planificación

2.1. Participantes del proyecto

Los diferentes roles del proyecto y la persona que los llevara a cabo serán los siguientes:

- Javier Vázquez Bodi como director del proyecto.
- Javier Vázquez Bodi como analista desarrollador del proyecto.
- Javier Vázquez Bodi como Tester
- Javier Vázquez Bodi como Arquitecto

La dedicación diaria será de 1,5-2,5 horas.

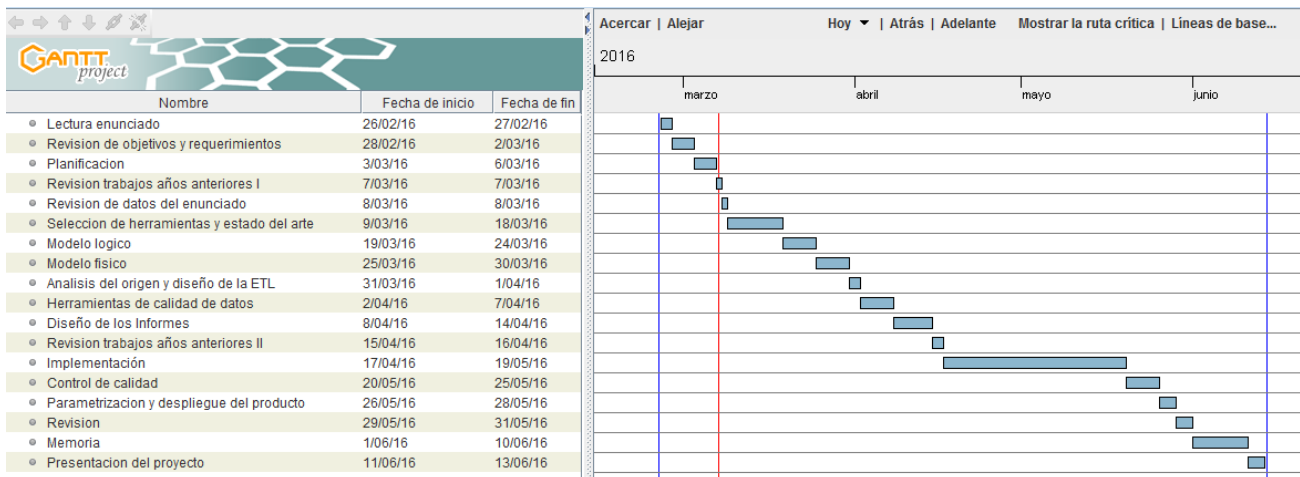
2.2. Tareas

Paquete	Nombre de la tarea	Coste	Responsable	Fecha de inicio	Fecha fin
PAC1	Lectura del enunciado	3h	Analista	26/02/2016	27/02/2016
PAC1	Revisión de objetivos y requerimientos	6h	Analista	28/02/2016	02/03/2016
PAC1	Planificación	6h	Director proyecto	03/03/2016	06/03/2016
PAC1	Revisión trabajos años anteriores I	2h	Analista	07/03/2016	07/03/2016
PAC2	Revisión datos del enunciado	2h	Analista	08/03/2016	08/03/2016
PAC2	Selección de herramientas y estado del arte	20h	Arquitecto	09/03/2016	18/03/2016
PAC2	Modelo lógico	12h	Analista	19/03/2016	24/03/2016
PAC2	Modelo físico	12h	Analista	25/03/2016	30/03/2016
PAC2	Análisis del origen y Diseño de la ETL	5h	Analista	31/03/2016	01/04/2016
PAC2	Herramientas de calidad de datos	12h	Analista	02/04/2016	07/04/2016
PAC2	Diseño de los informes	12h	Analista	08/04/2016	14/04/2016
PAC2	Revisión trabajos años anteriores II	4h	Analista	15/04/2016	16/04/2016
PAC3	Implementación	50h	Implementador	17/04/2016	19/05/2016
PAC3	Control de calidad	9h	Tester	20/05/2016	25/05/2016
PAC3	Parametrización y despliegue del producto	5h	Implementador	26/05/2016	28/05/2016
PAC4	Revisión	5h	Analista/Director proyecto	29/05/2016	31/05/2016
PAC4	Memoria	16h	Analista	01/06/2016	10/06/2016
PRES	Presentación del proyecto	8h	Director proyecto	11/06/2016	13/06/2016

2.3. Fechas clave e hitos

Titulo	Inicio	Fin
PAC 1 - Planificación	25/02/2016	07/03/2016
PAC 2 – Diseño	08/03/2016	11/04/2016
PAC 3 – Implementación	12/04/2016	09/05/2016
PAC 4 – Memoria	10/05/2016	13/06/2016
Presentación del proyecto	10/05/2016	13/06/2016

2.4. Cronograma



2.5. Riesgos

A continuación se describen los posibles riesgos que pueden aparecer durante el transcurso del tiempo en el que el proyecto se va a llevar a cabo:

Actualmente, a nivel laboral, estoy involucrado en un proyecto internacional en el que durante el año tengo que realizar algunos viajes y esto me restará tiempo disponible para dedicar al proyecto.

A la vez, estoy cumpliendo con la función de presidente de su comunidad de vecinos, cuyos trámites y preparativos me restarán tiempo libre.

Además, formo parte del Comité de Empresa y en la actualidad nos encontramos en un proceso de conflicto colectivo, y debido a ello también mi dedicación se verá afectada por la cantidad de imprevistos que se produzcan durante este proceso judicial.

Paralelamente estoy cursando la asignatura de sistemas distribuidos de gran escala como última asignatura pendiente en este mismo máster, la cual cosa disminuirá aún más de tiempo disponible.

No obstante y a pesar de las dificultades de tiempo, intentaré cumplir con los objetivos sin perjuicio en el nivel de detalle y la calidad del producto.

Aunque soy actualmente miembro del BICC en la organización para la que trabajo, no siempre durante mi carrera profesional he estado dedicado a tareas relacionadas con el Data Management y mi perfil técnico es más bien transversal, es por esto, que es posible, que no tenga demasiado bagaje en ciertas partes de las arquitecturas, como el reporting, y eso puede resultar un posible riesgo, no obstante, con esfuerzo intentaré subsanar esa carencia de experiencia.

2.6. Análisis de costes

Los costes detallados a continuación solo incluyen los costes de desarrollo e implantación del sistema de Bussiness intelligence y excluye el resto del sistema, como por ejemplo, la app de usuario y resto de sistema informático.

Software

- Coste por Licencia de Mysql Community versión: 0 euros.
- Coste por soporte de Pentaho Community versión: 0 euros.

Hardware

- Ordenador Intel core i5 vpro con 8GB de memoria RAM
- 1TB de disco duro SATA
- Sistema operativo Windows 7 64bits

Coste: 900 €

Personal

	Precio hora	Número de horas	Total coste
Director de proyecto	36 €	19 h	684 €
Ingeniero de calidad	30 €	9 h	270 €
Arquitecto DWH	35 €	20 h	700 €
Analista	30 €	92 h	2.760 €
Implementador	25 €	55 h	1.375 €
Total coste en personal			5.789 €

3. Diseño

Como primer paso voy a intentar aclarar una serie de puntos que ayudarán a entender que hay que desarrollar, de qué forma y qué se espera. Entendiendo esto resultara más fácil entender cómo voy a ir tomando decisiones, así como los paradigmas, procedimientos, metodologías y técnicas se van a utilizar.

¿Qué es un Datawarehouse?

Un Datawarehouse (almacén de datos) es una colección de datos de una organización con una permanencia determinada, que se caracteriza por integrar información desde diferentes orígenes de datos de la organización, y depurar información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta para explotarla en determinado ámbito.

Su fin es reflejar la variación de los datos de la organización a lo largo del tiempo, para ello, se suele crear una base de datos separada de la base de datos de transacciones actuales de la organización, en la que se guarde un histórico de todos los estados por los que ha pasado un registro previamente, en vez de guardar tan solo el último estado modificado.

Llegando a acumular cantidades muy grandes de datos, por motivos de rendimiento, seguridad, a menudo sus estructuras se deben dividir y aislar en otras más pequeñas dependiendo del uso que se le vaya a dar, dependiendo de ello también se suelen agrupar en Datamarts temáticos.

Utilizando la paralización los datos y sin afectar a la operativa del sistema, se extraen, transforman, limpian y cargan desde el origen al Datawarehouse a través del proceso llamado ETL.

Asimismo durante el proceso de ETL los datos pasan por diferentes áreas temporales y cargan en diferentes tipos de destinos:

- **Staging área:** Es un área temporal donde se recogen los datos que se necesitan de los sistemas origen. Se recogen los datos estrictamente necesarios para las cargas, y se aplica el mínimo de transformaciones a los mismos. No se aplican restricciones de integridad ni se utilizan claves, los datos se tratan como si las tablas fueran ficheros planos. De esta manera se minimiza la afectación a los sistemas origen, la carga es lo más rápida posible para minimizar la ventana horaria necesaria, y se reduce también al mínimo la posibilidad de error.

- ODS: En algunos Datawarehouse es necesario almacenar un snapshot de algunas tablas del sistema de origen sin ninguna transformación, para su posterior utilización por los procesos de carga o consulta.
- El almacén de datos corporativo: es el almacén en sí de datos corporativos historificados, la procedencia de los datos suele ser desde la Staging área o del dataMining, es un esquema parecido en un modelo normalizado, pero a él se le aplican técnicas de desnormalización y el resultado suele ser un número menor de tablas junto con un campos de control y vigencia del registro.
- Los Datamarts: es un subconjunto del almacén orientado a un tema de análisis, normalmente asociado a un departamento de la empresa en donde cada objeto de análisis es una tabla de hechos enlazada con diversas tablas de dimensiones.
- Si se diseñan siguiendo el Modelo en Estrella habrá prácticamente una tabla para cada dimensión, es la versión más desnormalizada. Si se sigue un modelo de Copo de Nieve las tablas de dimensiones estarán menos desnormalizadas y para cada dimensión se podrán utilizar varias tablas enlazadas jerárquicamente.
- Capa de usuario: Suele ser un conjunto de herramientas que aíslan al usuario de la capa técnica de la base de datos y facilitan su consulta y la generación de reportes sin necesidad de conocimiento técnico.

En cuanto a las tipologías de carga existen 2 tipologías:

- Acumulación Simple. La acumulación simple es, sin duda, la más sencilla y común, y consiste en realizar una somatización o resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el DW.
- Rolling. El proceso de Rolling por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo.

A través de una serie de metadatos se guarda la información relativa a los propios datos, cómo se han cargado, cuándo, como se relacionan y agrupan,...

En la actualidad se Incluyen las herramientas de análisis (Inteligencia de negocio) las de gestionar y recuperar metadatos.

Para explotar los datos y analizar los datos se generan un estructuras especiales llamadas cubos OLAP (cubos de proceso analítico en línea) que son unas estructuras multidimensionales que a menudo se suelen ver como una evolución con varias dimensiones de las hojas, también cada atributo dimensional se puede desglosar

jerárquicamente en otra sub-dimensión, por ejemplo si una dimensión de un reporte de ventas fuese la región donde se produce la venta, si esa región se pudiese desglosar en provincias.

A través de las herramientas de inteligencia de negocio se puede acceder a los cubos OLAP que se hacen visibles al usuario a través una serie de tablas dinámicas con el que el usuario puede “jugar” con los datos utilizando las dimensiones y las jerarquías para ver cómo se comportan aquellos atributos de información llamados variables o indicadores, todos ellos se agrupan en un tipo de tabla especial que es el objetivo que se está sometiendo a análisis, la tabla de hechos, a través de los sistemas de soporte a la toma de decisiones, cuadros de mandos y sistemas de Información Ejecutiva Sistemas de Soporte a la Decisión por acceso directo a los datos por parte de los usuarios o reportes generados automáticamente.

3.1. Arquitectura propuesta

La arquitectura propuesta está influenciada por mi bagaje profesional y los diferentes paradigmas y metodologías de almacenamiento y minería de datos con las que he trabajado.

Se trata de una arquitectura de Datawarehouse multicapa híbrido (top-down+bottom up), que no tiene necesidad de implementar un ODS, que utilice un tipo de carga de actualización simple y con posibilidad de implementar rolling en algunas entidades.

El modelo central o almacén de datos será del tipo copo de nieve, que, aunque tiene un impacto sobre el rendimiento y complica la consulta, puesto se incrementa en número de JOINS cuando la consulta implica más de una tabla, tiene como contrapartida que la información está organizada de una forma más optimiza.

Para paliar la complejidad de las consultas que el copo de nieve pueda ocasionar, en el caso de que el usuario necesite acceder a consultas que requieran cruzar datos de varias tablas, se prepararán una capa de vistas que hará transparente al usuario esta complejidad.

Los objetos formaran parte de un solo Datamart que será utilizado por todos los usuarios, no siendo requerido fragmentación alguna.

En el siguiente grafico se puede ver la arquitectura propuesta:

ARQUITECTURA DE LA SOLUCION

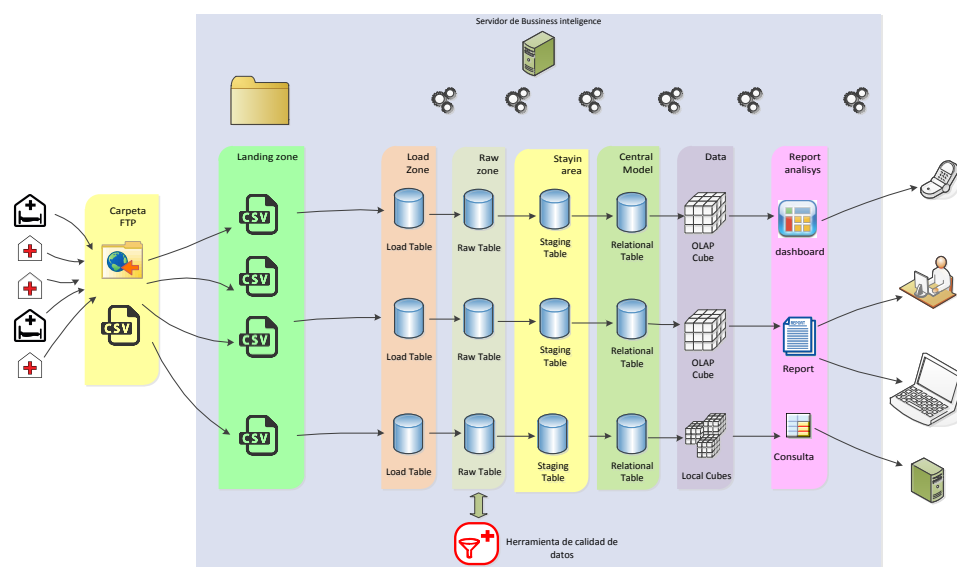


Ilustración 1 - Diagrama conceptual de la arquitectura propuesta

Para dar servicio a las necesidades de la aplicación se proponen 3 áreas:

- Servidor FTP (fuera del alcance del proyecto): encargado de recibir los envíos de datos de los centros terapéuticos.
- Servidor de BBDD: encargado de la gestión de la base de datos.
Compuesta de 5 grandes capas de datos.

- **Landing zone:** Los ficheros se recibirán por en un sitio ftp y posteriormente se trasladarán a este área de trabajo donde a través de comandos de sistema, el fichero original Excel recibido, se separara en diferentes ficheros CSV, cada uno de los cuales será una de las hojas Excel del fichero original.

Además en esta zona se adecuarán los ficheros para que todos los valores estén separados por el carácter “;” y delimitados por comillas doble.

Esta operación se ejecutará a través del Script anexo 1

- **Load zone:** Es la primera zona de carga, las tablas son una imagen de los ficheros CSV pero en formato tabla SQL con todos los campos formateados como LONGTEXT de 255 bytes.
 - **Raw zone:** Zona de filtrado de datos, formatos incorrectos, uniones de tablas y filtros por volumen. Se marcan los registros erróneos y se separan de los que están correctos, además se añaden los campos de auditoría: meta_upload_time (fecha de entrada) y meta_upload_process (proceso de creación de registro normalmente LOAD para aquellos registros procedentes de la carga del fichero), meta_rejected (registro rechazado por la herramienta de calidad).
 - **Staying área:** Zona de conversión de formato de datos al formato final, aseguramiento de la integridad referencial, se alimenta de los registros correctos de la raw zone.
 - **Modelo central:** Modelo de base de datos relacional final con estructura de copo de nieve con las correspondientes tablas de hechos y dimensiones.
- Servidor de bussiness intelligence para la parte analítica y reporting.
 - Área Multidimensional: Almacén de cubos OLAP

- Área de reporting: zona donde se guardan y se ejecutan los informes sobre el modelo central

3.2. Asunciones

- El volumen de usuarios será de alrededor: 25 usuarios.
- Se trata de un sistema de dimensiones medianas. No se trata de un sistema gigante en donde ninguna tabla superará los 2TB y la historificación requerida será de acorde. Esta asunción y este límite no será nunca superado.
- Los usuarios son terapeutas que no tienen conocimiento técnico pero si de médico.
- No existen problemas de espacio para albergar la solución propuesta ni su crecimiento vegetativo en un corto/medio plazo.

3.3. Estándares de nomenclatura

- El nombre de los componentes, ficheros y campos será siempre en minúscula, en la mayoría de casos, el nombre los componentes, estará formado por un prefijo que indicará a qué capa pertenece el componente antes de su nombre, y en algunos casos por algún prefijo que indicará una instancia especial del componente con un uso particular y el prefijo indicará dicho uso especial.
 - Ejemplo: `stg_centro_terapeutico_tipo`: componente de la staging área, que extrae exclusivamente la información sobre el tipo de centro terapéutico de la entidad.
- Todas las partes de los nombres de los componentes, campos y ficheros estarán separados por el símbolo de guion bajo “_”.
- **load_**: Todos las tablas, vistas y scripts, cuyo fin sea insertar datos, actualizar, truncar, crear componentes de la capa de load utilizarán dicho prefijo en el nombre.
- **raw_**: Todos las tablas, vistas y scripts cuyo fin sea insertar datos, actualizar, truncar, crear componentes de la capa de datos raw utilizarán dicho prefijo en el nombre.
- **stg_**: Todos las tablas, vistas y scripts cuyo fin sea insertar datos, actualizar, truncar, crear componentes de la staying área utilizarán dicho prefijo en el nombre.
- **fact_**: Todos las tablas, vistas y scripts cuyo fin sea insertar datos, actualizar, truncar, crear componentes del tipo hecho utilizarán dicho prefijo en el nombre.

- **dim_**: Todos las tablas, vistas y scripts cuyo fin sea insertar datos, actualizar, truncar, crear componentes del tipo dimensión utilizarán dicho prefijo en el nombre.
- **meta_**: Todos aquellos campos artificiales y técnicos creados con el único fin de guardar algún tipo de información creada exclusivamente para mejorar el proceso de carga, pero que no aportan ninguna información funcional sobre la entidad que la contiene, por ejemplo:
 - o Ejemplo meta_upload_time: nombre del campo técnico de un registro, que contendrá la fecha y hora de carga en el Datawarehouse.
- **_01, _02, _NN**: Aquellas tablas previas de carga si es necesario se utilizara un sufijo número, siendo la tabla sin sufijo la tabla final, es decir, si la carga entre la raw y la staging de la entidad terapeuta requiere varias fases de carga se generarán las tablas intermedias

stg_terapeuta_01 → stg_terapeuta_02, → ... → stg_terapeuta_nn → stg_terapeuta

- **id**: Cada tabla tendrá una clave artificial con el nombre ID, que será la clave foránea en las tablas dependientes de ella.
- **fk_**: Los campos foreign key tendrán este prefijo.

3.4. Selección de herramientas a utilizar

Framework Bussiness Inteligence

Analizando los requerimientos se llega a la conclusión de que se necesitan herramientas que satisfagan las áreas de Almacén de datos, Reporting, Análisis, Cuadro de mandos y control de procesos.

En el mercado existen diferentes herramientas que nos ofrecen todos estos servicios en forma de herramientas específicas para cada área, que más tarde habría que integrar para formar un sistema o bien optar por una suite que las integre a todas, se priorizará aquella que integren toda la funcionalidad para reducir complejidad, tiempo y riesgos de integración.

Después de una búsqueda en diversas fuentes, descartar todas aquellas herramientas que considero en un grado de madurez baja. Para tomar la decisión de sobre que framework desarrollar se han tenido en cuenta las siguientes plataformas: JasperReports, Microstrategy, Oracle BI Standard Edition, SAP Bussiness Objets, IBM Cognos, Talent, Eclipse Birt, Pentaho BI, Informática Powercenter, Palo BI suite, Mondrian, jPivot, Tableau,...

Como requisito se establece que la plataforma sea open-source, por lo que se descarta de las anteriores aquellas que exclusivamente se distribuyen bajo licencia comercial y

propietaria. El hecho de escoger herramientas Opensource, supone una reducción en el coste del sistema a implementar de una forma exponencial llegando a unas cotas de reducción de costes de entre el 90%/100% en software.

Siempre y cuando el número de dado que el coste por licencia va ser 0 el coste real vendría marcado por el coste de soporte, con un máximo de 25 usuarios en el sistema se ha calculado que el coste por soporte es de 1.097 euros, el coste total será de 27.425 euros en el peor de los casos.

Tras un tiempo consultando información en diversos formatos, se llega a la conclusión de que en la mayor parte de las fuentes consultadas se aconseja, sobre todo, el uso de Talent o Pentaho y, aunque Talent SUITE destaca por su generador de código tanto JAVA como PERL, que además integra con las suites SpagoBI o Jaspersoft, teniendo capacidad para tratar los formatos de entrada CSV, Excel, XML como diversos conectores a CRM's, permite creación de nuevas funciones.

Aunque en esta diversidad de plataformas tan atractivas destacaría a Eclipse BIRT que se distribuye bajo licencia Eclipse, que tiene como uno de sus atractivos la cantidad de opciones de agregación, o, su capacidad para integrarse con librerías de usuario, y, aunque en el 2015 fue fruto de muchas revisiones. He centrado mi análisis en las dos más recomendadas como son Pentaho y JasperReports, a continuación podemos ver la comparativa.

Pentaho Community	JasperSoft BI Suite Community
Características	
<ul style="list-style-type: none"> • Integración con diversas bases de datos: Oracle, Sybase , MySQL, Informix, SQLServer • Posibilidad de compartición de objetos entre desarrollos. • Formatos de entrada: Texto, XML, Excel • Detector de presencia de archivos • Tratamiento de datos mediante procesos ETL • Cubos MOLAP • Informes, dashboards y score Cards. • Conexión con LDAP/LDIF • Capacidades de movilidad • Interface Drag & Drop • Licencia mixta • Compatible con MacOs, Linux, Windows. 	<ul style="list-style-type: none"> • iReport/Jasper Studio como IDE de escritorio stand-alone para desarrollo de reports and Professional Dashboard designer en el servidor. • Compatible con portátiles y equipos de escritorio. • Licencia GPL.

• Ventajas y puntos fuertes	
<ul style="list-style-type: none"> • Gran número de conectores • Ventajas propias del open-source. • Posibilidad de múltiples conversiones, uniones filtros. • Utilización de javascript. • Desarrollado utilizando las ultimas tecnología. • Comunidad grande. • Formado por diferentes piezas. • Basada en eclipse. • Chart designer • Report designer • Java friendly • Ventajas del open-source • Multiplataforma • Es escalable • Está en activo y continua evolucionando • El diseño del reporting es dinámica y ad-hoc por el propio usuario. • Rápido de implantar • Ágil en la creación mantenimiento • El más aconsejado de las plataformas open-source • Facilidad de uso • Ampliable con una larga lista de plugins • GUI Amigable • Los clientes están mayoritariamente satisfechos. • Documentación y ejemplos muy accesible • Gran número de lenguajes de scripting. 	<ul style="list-style-type: none"> • Se pueden generar tablas en los reports • Gran número de formatos de salida html, pdf, Excel,xml, flash, ... • Bastantes conectores • Informes multi columnas • Mejor integración con herramientas ofimáticas. • Gran número de diagramas. • Gráficos interactivos con hipervínculos.
Desventajas	
<ul style="list-style-type: none"> • No estar familiarizado con el framework. • No tiene planificador propio se delega en el sistema operativo o software específico dicha función. • Algunos complementos tienen licencia privada. • No es la opción con mayor número de diagramas. 	<ul style="list-style-type: none"> • No es compatible con ipad y móvil. • La comunidad no es de las más activas.

Una característica interesante es su capacidad de creación y gestión de un modelo empresarial y conexión con diversas bases de datos. Sin embargo, en el cuadrante mágico de Gartner, tan solo hace mención de un Pentaho bien posicionado y compitiendo con el resto de plataformas.

Opto pues por Pentaho como plataforma de Business intelligence y reporting, ya que parece que es la que cumple de mejor forma con los requisitos.

SGBD

Tal y como ha pasado con la Suite de BI he tenido en cuenta, a la hora de seleccionar las herramientas SGBD Cliente/Servidor, las más distribuidas del momento, que se integran bien con Pentaho, son: MySQL, PostgreSQL, Microsoft SQL Server, Oracle.

Tras descartar las que no son opensource, las opciones son PostgreSQL y MySQL.

A continuación expongo los datos que he conseguido de ambas, cómo he realizado la comparativa y cuál es la opción seleccionada.

MySQL Community Edition	PostgreSQL
Características	
<ul style="list-style-type: none"> • Licencia GPL o licencia comercial. Incluyendo la biblioteca cliente, se puede comercializar o distribuir un sistema que se conecte a MySQL con el cliente oficial, se debe pagar a Oracle o liberar el código GPL compatible. • Lo Usan Wikipedia,2 Google3 4 (aunque no para búsquedas), Facebook,5 6 7 Twitter,8 Flickr,9 y YouTube.10 • Es un proyecto opensource de una marca comercial • La versión community es gratuita • Es más modular. • Los trigger automáticos en cascada de las claves foráneas dan problemas y no está bien integrado. • Tiene un clúster de alta estabilidad y disponibilidad. • Tiene un buen sistema de replicación. • Menor tipo de datos, además No soporta JSON. • El lenguaje de los stored procedures no está suficiente maduro. 	<ul style="list-style-type: none"> • Licencia Mit (software libre y se pueden distribuir versiones modificadas sin requerir el código). • Es un proyecto de software libre auspiciado por la comunidad. • Cumple más estándares de SQL • Su SQL Esta mejor integrado con el almacén de datos. • Es menos escalable y no asegura tanto la disponibilidad. • Es más monolítico. • Tiene un buen sistema de replicación. • Soporta diversos tipos de, direcciones ip, enums dinámicos, direcciones ip, arrays de cualquier tipo y XML, JSON. • Soporta diversos lenguajes, además de un lenguaje propio basado en plsql. • Se puede poner una función como valor por defecto de una columna. • El pgModeler, Open System Architect, pgadmin no es tan completo. • Comunidad excelente entusiasta además de profesionales de pago. • Usan postgres McAfee, Trend Micro,

<ul style="list-style-type: none"> • No se puede poner una función como valor por defecto de una columna. • El mySql workbench como herramienta grafica funciona bien. • Comunidad excelente entusiasta además de profesionales de pago. • Licencia GPL o licencia comercial. Incluyendo la biblioteca cliente, se puede comercializar o distribuir un sistema que se conecte a Mysql con el cliente oficial, se debe pagar a Oracle o liberar el código GPL compatible. • Lo Usan Wikipedia,2 Google3 4 (aunque no para búsquedas), Facebook,5 6 7 Twitter,8 Flickr,9 y YouTube.10 • Es un proyecto opensource de una marca comercial • Versión community es gratuita 	LastFM, Skype, ...
---	--------------------

Ambas soluciones permiten a nivel licencia implementar el sistema que queremos desarrollar, si bien, proyectos han mejorado considerablemente desde que se formaron inicialmente estas impresiones, y una comparación es más difícil de hacer que antes, las bases técnicas y modos de pensar de los desarrolladores que informaron estas percepciones iniciales continúan siendo una influencia dominante en ambos proyectos.

PostgreSQL se ha enfocado tradicionalmente en la fiabilidad, integridad de datos y características integradas enfocadas al desarrollador. Tiene un planificador de consultas extremadamente sofisticado, que es capaz de unir cantidades relativamente grandes de tablas eficientemente.

MySQL, por otra parte, se ha enfocado tradicionalmente en aplicaciones web de lectura mayormente, usualmente escritas en PHP, donde la principal preocupación es la optimización de consultas sencillas. Pero MySQL asegura A.C.I.D. a través de InnoDB.

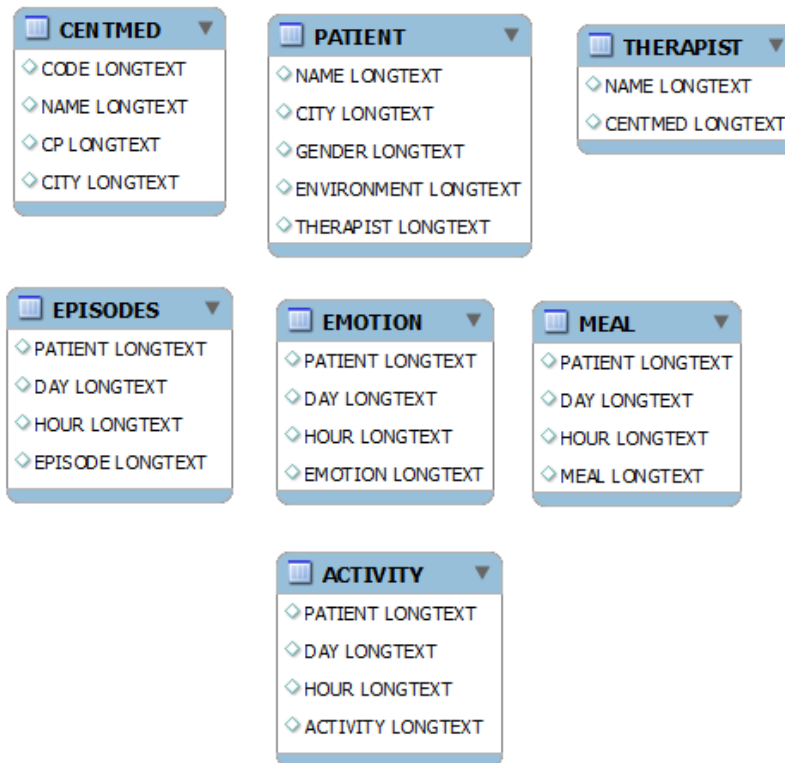
Personalmente tengo experiencia con ambas, aunque PostgreSQL parece más recomendada para crear Datawarehouses, mi experiencia personal me ha demostrado que, para almacenes de datos del volumen y complejidad que se plantean en esta solución, es suficiente Mysql con InnoDB, en este escenario de baja complejidad, la sofisticación de las herramientas de soporte de PostGreSQL pasa inadvertida y la curva de aprendizaje más baja del MySQL workbench ante igualdad condiciones me hace decantarme finalmente por el MySQL.

Finalmente tomo la determinación de utilizar:

- MySQL Community Edition server 5.7
- MySQL workbench 5.2.44 CE
- Pentaho Data Integration 4.4.0
- Pentaho Reporting Tool

3.5. Estructura de datos de origen

El fichero de origen es un Excel con varias hojas de cálculo (cada uno representa una entidad) que se recibe como salida del sistema operación:



Las entidades

- CENTMED: Contiene los centros médicos o terapéuticos que trabajan en el piloto.
- THERAPIST: Lista de los terapeutas i el centro en el que trabaja.
- PATIENT: Lista de pacientes con la información sobre sexo, su localidad y su entorno urbano o rural, y el terapeuta asignado.
- EPISODES: Contiene un registro de los episodios de crisis ocurridos durante el día según la franja horaria: madrugada-mañana, mediodía, tarde-noche. Los episodios puede restrictivos (no comer), vómitos o atracones.
- EMOTION: Registro de los estados de ánimo durante las franjas horarias. Pueden ser Relajado, feliz, triste, culpable, angustiado o frustrado.
- MEAL: Registro de ingestas: Nada, 1, 2,3 tercios, etc.
- ACTIVITY: Registre de las actividades realizadas durante las diferentes franjas: Estudiar/Trabajar, Deporte, Familia, Amigo o Sol

Otros:

- Para simplificar se han dividido el día en 3 franjas.
- El indicador de entorno es aproximado, habrá entornos más urbanos, menos rurales, etc.
- Las actividades registradas no ocupan toda la franja, pero es la actividad principal realizada.

3.6. Modelo conceptual de Datawarehouse

En cuanto al modelo de esquema de Datawarehouse se opta por un modelo de copo de nieve, que como ventaja nos encontramos una base de datos más comprensible y con un espacio más optimizado, ordenada y clara, pero un poco más costosa y compleja durante la fase de consulta. Se generarán para intentar aprovechar al máximo el número de dimensiones que el sistema permita, con el fin de ofrecer el máximo número de perspectivas a la hora del análisis.

A través de vistas, consultas y tablas sumariadas, a modo de DataMars se adecuará el modelo a un modelo más sencillo focalizado en cada utilidad que se quiera dar al modelo en estrella.

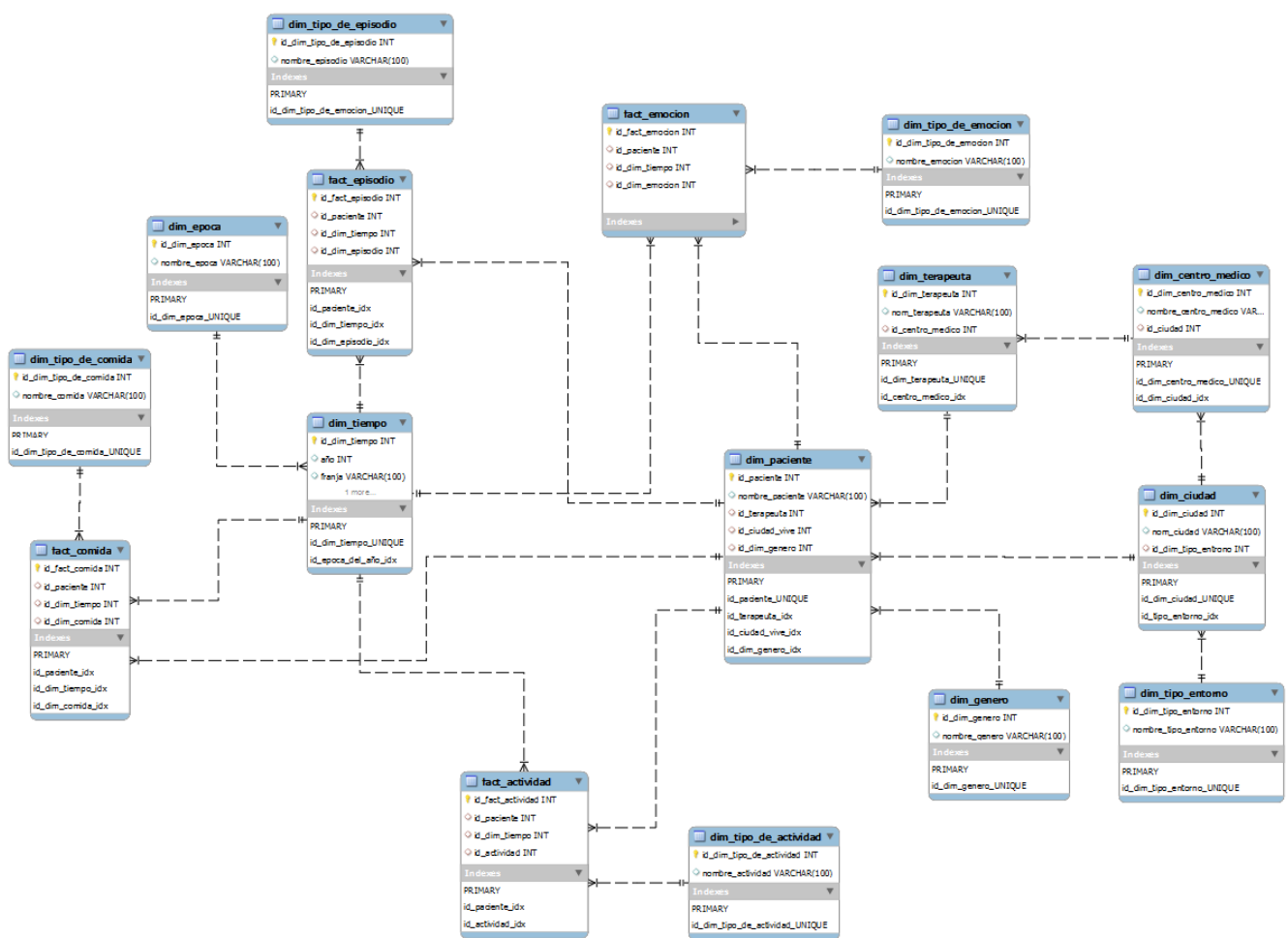


Ilustración 2.- Modelo conceptual de datawarehouse

3.6.1. Elementos del modelo: dimensiones y hechos.

Hechos

Nombre	fact_episodio
Descripción	Contiene los diferentes episodios que los pacientes padecen a lo largo del tiempo.
Dimensiones	dim_tipo_de_episodio dim_tiempo dim_paciente
Atributos	id_fact_episodio
Indicadores	Numero de episodios sufridos por un paciente a lo largo del día.

Nombre	fact_comida
Descripción	Contiene las diferentes comidas que el paciente hace a lo largo del tiempo.
Dimensiones	dim_tipo_de_comida dim_tiempo dim_paciente
Atributos	id_fact_comida
Indicadores	Numero comidas que realiza el paciente a lo largo del día.

Nombre	fact_emocion
Descripción	Contiene el registro de emociones que el paciente sufre a lo largo del tiempo.
Dimensiones	dim_tipo_de_emocion dim_tiempo dim_paciente
Atributos	id_fact_emocion
Indicadores	Número de veces que el paciente siente las diversas emociones a lo largo del día.

Nombre	fact_actividad
Descripción	Contiene un registro de las actividades que el paciente realiza
Dimensiones	dim_tipo_de_actividad dim_tiempo dim_paciente
Atributos	id_fact_actividad
Indicadores	Número de veces que el paciente practica las diferentes actividades a lo largo del día.

Dimensiones

Nombre	dim_tipo_de_episodio
Descripción	Contiene los tipos de episodios que puede tener un paciente
Atributos	id_dim_tipo_de_episodio nombre_episodio
Jerarquías	Nivel 0: episodio

Nombre	dim_tiempo
Descripción	Contiene los tipos de episodios que puede tener un paciente
Atributos	id_dim_epoca nombre_epoca
Jerarquías	nivel 0 – franja horaria nivel 1 – día nivel 2 – mes nivel 3 – trimestre nivel 4 – estación nivel 5 – semestre nivel 6 - año

Nombre	dim_tipo_de_comida
Descripción	Contiene los tipos de episodios que puede tener un paciente
Atributos	id_dim_tipo_de_comida nombre_tipo_de_comida
Jerarquías	nivel 0 – tipo de comida

Nombre	dim_tipo_de_actividad
Descripción	Contiene los tipos de episodios que puede tener un paciente
Atributos	id_dim_tipo_de_actividad nombre_actividad
Jerarquías	Nivel 0 – actividad

Nombre	dim_paciente
Descripción	Contiene los pacientes que están en tratamiento
Atributos	id_paciente nombre_paciente id_terapeuta id_ciudad_vive id_dim_genero nombre_epoca
Jerarquías	Nivel 0 – paciente

Nombre	dim_tipo_de_emocion
Descripción	Contiene los tipos de emociones que puede tener un paciente
Atributos	id_dim_tipo_de_emocion nombre_emocion
Jerarquías	Nivel 0 – Emoción

Nombre	dim_terapeuta
Descripción	Contiene los terapeutas de los centros médicos
Atributos	id_dim_terapeuta nombre_terapeuta id_centro_medico
Jerarquías	Nivel 0 - Terapeuta

Nombre	dim_centro_medico
Descripción	Contiene los centros médicos
Atributos	id_dim_centro_medico nombre_centro_medico id_ciudad
Jerarquías	Nivel 0 – Centro medico

Nombre	dim_ciudad
Descripción	Contiene las ciudades del territorio
Atributos	id_dim_epoca nombre_epoca
Jerarquías	Nivel 0 – ciudad Nivel 1 – Tipo entorno

Nombre	dim_tipo_entorno
Descripción	Contiene los tipos de entorno que existen
Atributos	id_dim_tipo_entorno nombre_entorno
Jerarquías	Nivel 0 – tipo de entorno

Nombre	dim_genero
Descripción	Contiene los géneros sexuales
Atributos	id_dim_genero nombre_genero

Jerarquías Nivel 0 – genero**Medidas**

La principal medida que se va a tener en cuenta es la ocurrencia de los diferentes hechos: episodio, comida, emoción y actividad a lo largo del día.

Granularidad

Aunque el requisito de granularidad está definido como diario, con el fin de no perder capacidad de análisis, la información se almacenará por timestamp y en los diferentes informes se ajustará dicha granularidad temporal a lo requerido.

Por otro lado la granularidad máxima temporal será una agrupación total de las ocurrencias desde diferentes puntos de vista, del total de datos del sistema.

3.7. Decisiones tomadas sobre el diseño

- Las claves primarias originales no se propagarán, por tanto, se generará una clave artificial para cada entidad a modo de identificador, ello resultará un poco más complicado a la hora de generar una consulta, pero el rendimiento se verá mejorado, como medida paliativa se crearán queries en un datamart que facilitarán el uso de los datos si fuera necesario.
- Todos los identificadores serán del tipo entero de 11 posiciones autoincrementales.
- Aunque es requisito que la granularidad sea diaria, se conservará la granularidad original, para no perder detalle, si fuera necesario en los informes y en las queries adecuadas se agruparán los datos con la granularidad deseada.

3.8. Herramienta de calidad de datos

La herramienta de calidad de datos será una herramienta encargada de marcar aquellos del área raw que no cumplan los requisitos para ser cargados en la staging área. Cada tabla de la Raw área tendrá un campo técnico llamado meta_rejected, se ejecutará la herramienta que comprobará que cada uno de los datos de la tabla y aquellos registros que no cumplan los requisitos para la carga de ese dato en el respectivo campo de la staging, se marcará el registro como meta_rejected = 1, en la carga de la staging no se cargarán aquellos registros con meta_rejected = 1.

La herramienta constará de un script y una tabla de chequeos SQL a ejecutar en cada ETL.

4. Implementación

La implementación del datawarehouse se divide en 4 grandes partes:

- Configuración del sistema.
- Creación de la base datos.
- ETL
- Definición de los reportes.

4.1. Configuración del servidor de MySQL

Para la configuración del servidor de base de datos fue necesario descargar el instalador del servidor MySQL server 5.7, crear una instancia de servidor corriendo en el puerto 3306 en el local host.

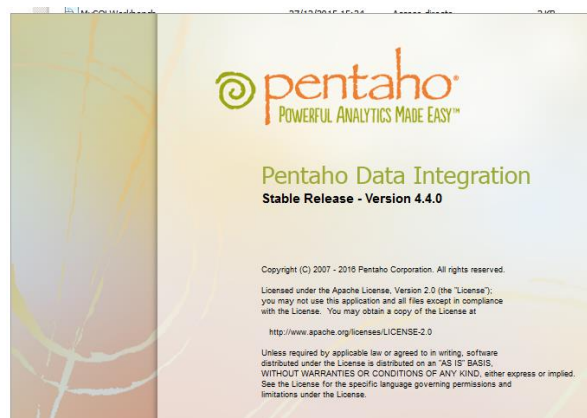
Utilizando el usuario root se ha creado la base datos TFM_BI_TCA que son las siglas de: “Trabajo de Final de Master de Bussiness Inteligence para Análisis de Trastornos alimentarios”.

Además se instaló el MySQLWorkbench como IDE de usuario para gestionar la base de datos.

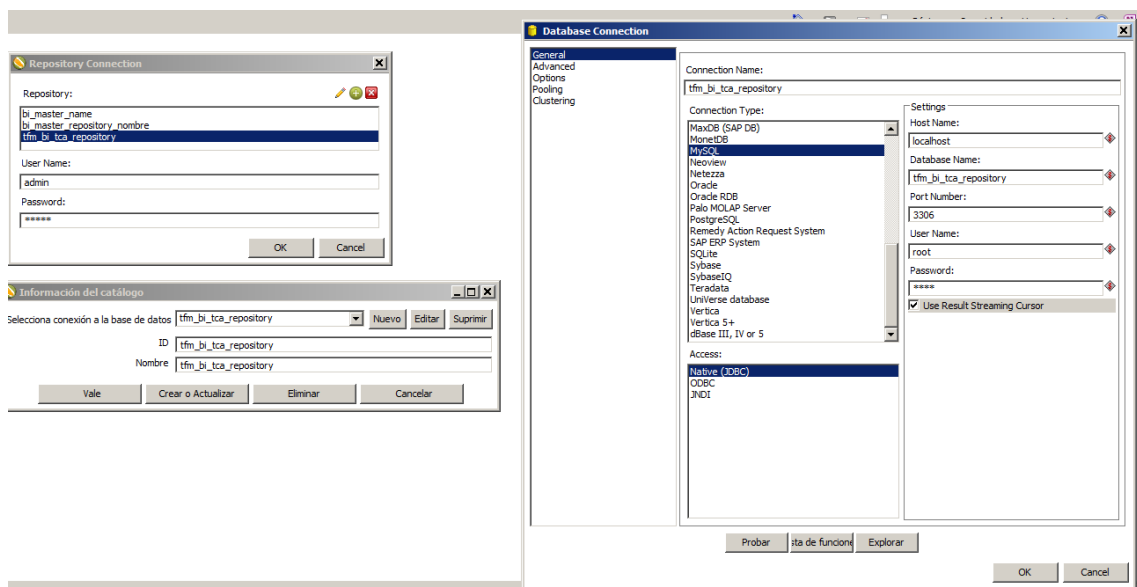
4.2. Configuración del servidor de ETL

El paso previo es la creación de la base de datos del repositorio del integration server en el servidor de base de datos, en nuestro caso se crea la base de datos: TFM_BI_TCA_REPOSITORY.

La utilidad seleccionada para gestionar la ETL es el Pentaho Integration, este servidor requiere que se genere una base de datos para almacenar los objetos de esta utilidad, por lo que se crea una base de datos cuyo nombre es TFM_BI_TCA_REPOSITORY en el mismo servidor que la base de datos del datawarehouse.



Como preparación antes de empezar es necesario crear una conexión con el repositorio a la base de datos que previamente hemos generado, en nuestro caso esta sería la definición de la conexión:



4.3. Modelo físico

El modelo físico a crear para albergar, dará cabida a las diferentes áreas del datawarehouse:

Área Load: se creará una tabla por cada hoja de cálculo con datos recibidos en la hoja Excel original, la estructura de tablas ya se ha mencionado anteriormente en la sección anterior, todos los campos son LONGTEXT para asegurar que no se pierde ningún dato.

Área Raw: El modelo físico creado para albergar el área de raw es el siguiente:

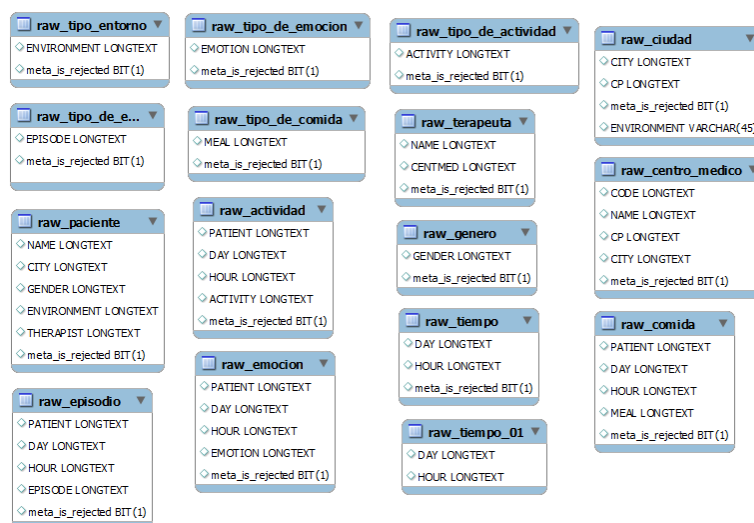


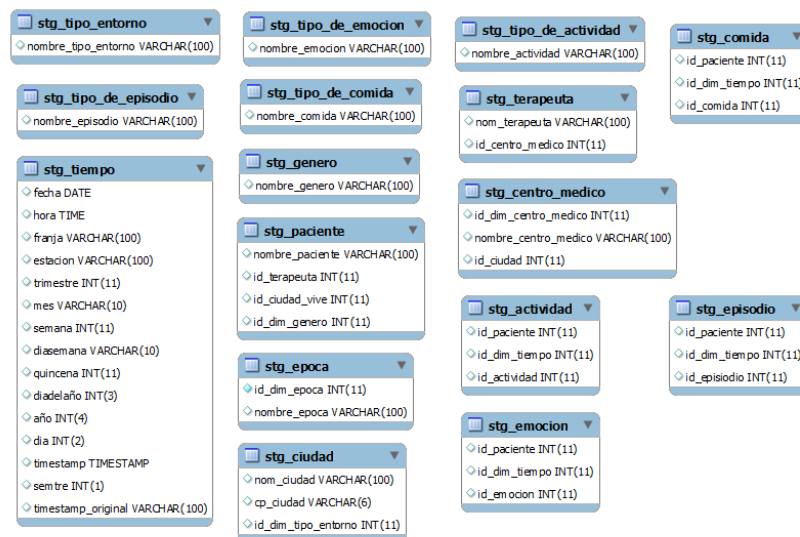
Ilustración 3 Modelo físico Área RAW

La estructura es muy parecida al área de load, pero, se añade un campo de control de calidad `meta_is_rejected`, el cual esta inicializado a 0 por defecto, además se crean las tablas de género, ciudades, entorno, actividades, episodios, comidas, emociones y tiempo.

La tabla `raw_tiempo` se genera de la unión de los timestamps extraídos de las tablas `raw_episodio`, `raw_comida`, `raw_emocion`, `raw_actividad`, además se asocia las franjas horarias a el timestamp y la franja pasa a ser un atributo más del registro de tiempo, ya que se puede deducir dado un timestamp concreto.

En el caso de la `raw_ciudad`, se obtiene de la unión de ciudades extraídas de la ubicación de los diferentes centros médicos y la ciudad donde viven los pacientes.

Área Stg: El modelo físico que vamos a crear para albergar el área de staging es el siguiente:



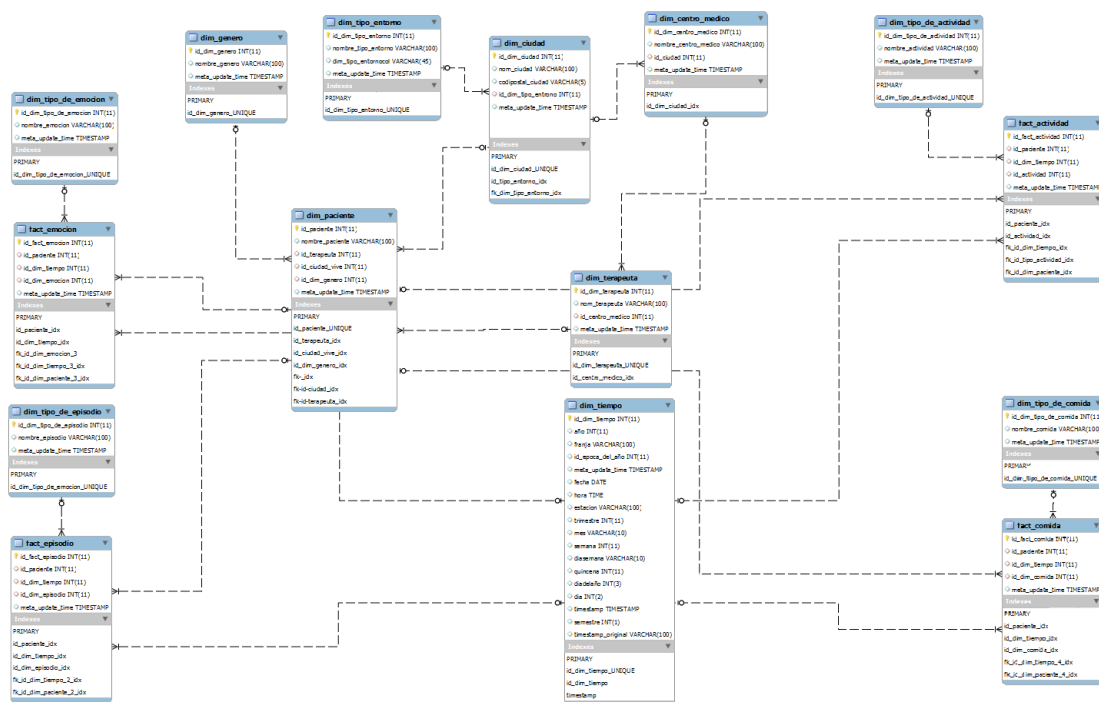
Las particularidades de esta estructura es que se genera la tabla de `stg_tiempo` en la que se generan los nuevos campos de tiempo calculados en base el timestamp, estos nuevos campos calculados:

- Año: Año en formato aaaa
- franja
- meta_update_time
- fecha: Fecha en formato dd/mm/aaaa
- hora: Hora en formato HH:MM:SS
- estación: hace referencia a las estaciones del año {P: primavera, O: Otoño, I:Invierno, V:Verano}
- trimestre: Numero de trimestre dentro del año {1,2,3,4}
- mes: Numero de mes dentro del año {1,2,3,4,5,6,7,8,9,10,11,12}
- semana: Numero de semana dentro del año {1 .. 52}
- diasemana: Numero de día de la semana: {1:Lunes, 2:Martes, 3:Miercoles, 4:Jueves, 5: Viernes, 6: Sábado, 7: Domingo}
- quincena: Numero de quincena dentro del año {1..27}

- diadelañ: Numero de día dentro del año {1...365}
- día: día del mes en formato dd
- timestamp: timestamp en formato MySQL
- semestre: Semestre dentro del año {1,2}
- timestamp_original: Timestamp original procedente de las tablas de origen.

Además, en este área solo se cargan aquellos registros que pasan las reglas de calidad.

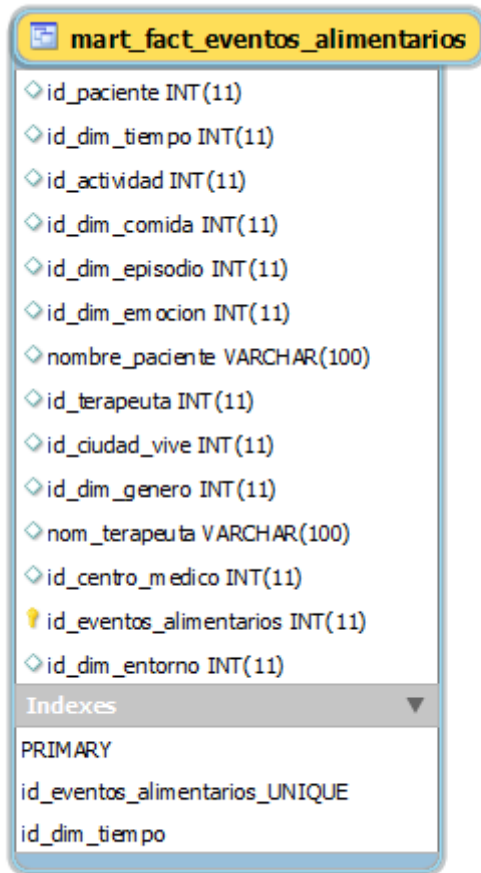
El área del datawarehouse tiene un modelo físico que vamos a crear para albergar el datawarehouse es el siguiente:



Las principales particularidades de este modelo son las foreing keys correspondientes, la creación de un campo de control de carga meta_upload_time, y la generación de una clave primaria por entidad que será un auto numérico.

El datamart: El datamart estará compuesto por 1 tabla y 2 vistas, que nos ayudaran a realizar mejor las consultas y los cubos.

Tabla fact_eventos_alimentarios como fusión de todas las tablas de hechos.



mart_relacion_episodios_con_actividades_previas: a través de esta consulta vemos sumariada de episodios con su actividad y con la actividad del registro anterior.

	previa_actividad	actividad_actual	episodio_actual	cantidad_ocurrencias_episodio
▶	SPORTS	ALONE	VOMITED	1993
	FRIENDS	ALONE	VOMITED	1190
	FAMILY	ALONE	VOMITED	992
	SPORTS	FRIENDS	VOMITED	982
	ALONE	ALONE	VOMITED	821
	FRIENDS	FRIENDS	VOMITED	550
	FAMILY	FRIENDS	VOMITED	490
	ALONE	FRIENDS	VOMITED	396
	SPORTS	ALONE	RESTRICTION	2138

mart_relacion_actividades_con_mejora_o_empeora: a través de esta consulta vemos sumariada de episodios con su actividad, y el episodio anterior con la actividad que estaba haciendo en el momento del episodio anterior.

previa_actividad	episodio_previo	actividad_actual	episodio_actual	cantidad_ocurrencias_episodio
ALONE	VOMITED	ALONE	BINGE	453
FRIENDS	VOMITED	ALONE	BINGE	205
ALONE	VOMITED	ALONE	RESTRICTION	195
ALONE	VOMITED	ALONE	VOMITED	153
FRIENDS	VOMITED	ALONE	RESTRICTION	87
FRIENDS	VOMITED	ALONE	VOMITED	83
ALONE	VOMITED	FRIENDS	RESTRICTION	81
ALONE	VOMITED	FRIENDS	VOMITED	76
FRIENDS	VOMITED	FRIENDS	RESTRICTION	36
FRIENDS	VOMITED	FRIENDS	VOMITED	35

4.4. Creación de la base datos

Con el siguiente script procederemos a crear físicamente la base de datos para albergar todo el proceso de ETL y Datawarehouse:

```
CREATE DATABASE `tfm_bi_tca` /*!40100 DEFAULT CHARACTER SET latin1 COLLATE latin1_spanish_ci */;$
```

4.5. Creación de las tablas

Seguidamente con el script referenciado en el Apéndice1 se crean todas las tablas anteriormente referenciadas.

4.6. Creación de las vistas del datamarts

Como último paso de generación del modelo físico, crearemos las diferentes vistas del datamart, que nos ayudaran a realizar las consultas para luego generar los reports, consistirá en una tabla donde se aglutinara todas las tablas de hechos y se fusionara por timestamp y paciente, y 2 vistas una de ellas servirá para el estudio de la evolución del estado del paciente teniendo en cuenta la actividad y episodio previo.

```
delimiter $$
CREATE TABLE `fact_eventos_alimentarios` (
  `id_paciente` int(11) DEFAULT NULL,
  `id_dim_tiempo` int(11) DEFAULT NULL,
  `id_actividad` int(11) DEFAULT NULL,
  `id_dim_comida` int(11) DEFAULT NULL,
  `id_dim_episodio` int(11) DEFAULT NULL,
  `id_dim_emocion` int(11) DEFAULT NULL,
  `nombre_paciente` varchar(100) COLLATE latin1_spanish_ci
  DEFAULT NULL,
  `id_terapeuta` int(11) DEFAULT NULL,
  `id_ciudad_vive` int(11) DEFAULT NULL,
  `id_dim_genero` int(11) DEFAULT NULL,
  `nom_terapeuta` varchar(100) COLLATE latin1_spanish_ci DEFAULT
  NULL,
  `id_centro_medico` int(11) DEFAULT NULL,
  `id_eventos_alimentarios` int(11) NOT NULL AUTO_INCREMENT,
  `id_dim_entorno` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_eventos_alimentarios`),
  UNIQUE KEY `id_eventos_alimentarios_UNIQUE`
  (`id_eventos_alimentarios`),
  KEY `id_dim_tiempo` (`id_dim_tiempo`)
) ENGINE=InnoDB AUTO_INCREMENT=6463 DEFAULT
CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```



```

delimiter $$

CREATE VIEW `mart_paciente_evolucion` AS

select `b`.`nombre_paciente` AS `nombre_paciente`,`c`.`año` AS `año`,`c`.`mes` AS
`mes`,`a`.`id_dim_episodio` AS `id_dim_episodio` from ((`fact_eventos_alimentarios` `a` left join
`dim_paciente` `b` on((`a`.`id_paciente` = `b`.`id_paciente`))) left join `dim_tiempo` `c`
on((`a`.`id_dim_tiempo` = `c`.`id_dim_tiempo`)))$$

CREATE VIEW `mart_relacion_episodios_con_actividades_previas` AS

select `z`.`nombre_actividad` AS `previa_actividad`,`x`.`nombre_actividad` AS
`actividad_actual`,`u`.`nombre_episodio` AS `episodio_actual`,count(0) AS `cantidad_ocurrencias_episodio`
from ((((`tfm_bi_tca`.`fact_eventos_alimentarios` `m` join `tfm_bi_tca`.`dim_tiempo` `s`
on((`m`.`id_dim_tiempo` = `s`.`id_dim_tiempo`))) join (select `g`.`id_actividad` AS `actividad`,`w`.`timestamp`
AS `tiempo`,`g`.`id_dim_episodio` AS `episodio`,(select max(`t`.`timestamp`) AS `key_timestamp` from
`tfm_bi_tca`.`dim_tiempo` `t` where (`t`.`timestamp` < `w`.`timestamp`)) AS `previo_time` from
(`tfm_bi_tca`.`fact_eventos_alimentarios` `g` join `tfm_bi_tca`.`dim_tiempo` `w` on((`g`.`id_dim_tiempo` =
`w`.`id_dim_tiempo`))) `n` on((`n`.`previo_time` = `s`.`timestamp`))) left join
`tfm_bi_tca`.`dim_tipo_de_actividad` `z` on((`z`.`id_dim_tipo_de_actividad` = `m`.`id_actividad`))) left join
`tfm_bi_tca`.`dim_tipo_de_actividad` `x` on((`x`.`id_dim_tipo_de_actividad` = `n`.`actividad`))) left join
`tfm_bi_tca`.`dim_tipo_de_episodio` `u` on((`u`.`id_dim_tipo_de_episodio` = `n`.`episodio`))) where
((`n`.`actividad` is not null) and (`n`.`episodio` is not null) and (`m`.`id_actividad` is not null)) group by
`m`.`id_actividad`,`n`.`actividad`,`n`.`episodio` order by `n`.`episodio`,count(0) desc$$

```

4.7. ETL

Como primer paso de la parte del proceso de carga hay un proceso anterior al la manipulación, que es la importación de datos desde el fichero Excel, este proceso se hace en dos fases, primero se manipula el fichero Excel TCADATAPFM.xlsx y se divide en diferentes ficheros CSV (out_THERAPIST.csv, out_PATIENT.csv, out_MEAL.csv, out_EPISODES.csv, out_EMOTION.csv, out_CENTMED.csv, out_ACTIVITY.csv) y más adelante se carga en las tablas load correspondientes.

Para ello se ejecuta un fichero cmd Apendice2 que llama a un script visual basic Apendice3.

A continuación, me centraré en la cadena de carga, que como ya se ha comentado anteriormente, la cadena ETL está diseñada de carga completa, es decir, siempre se borran todos los datos y se cargan de nuevo sin mantener los datos anteriormente cargados.

La carga incremental o carga de ficheros delta sería una mejora recomendable y posible si este proyecto evolucionara en el futuro, pero está fuera del alcance de este proyecto, por lo tanto, la cadena de carga y transformación está compuesta por los siguientes pasos:

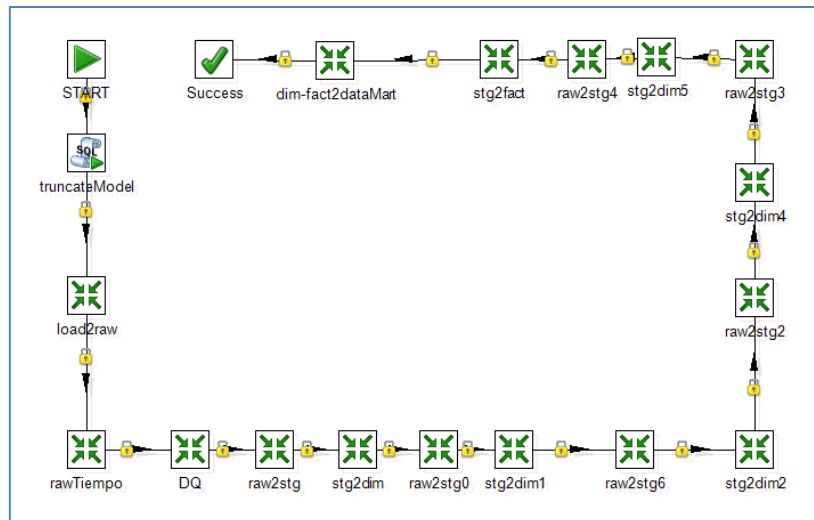


Ilustración 4 Cadena de carga y transformación

- **TruncateModel:** Paso encargado de vaciar todas las tablas del modelo (ver ilustración) (ver script).

```
delete from dim_centro_medico;
delete from dim_ciudad;
delete from dim_epoca;
delete from dim_genero;
delete from dim_paciente;
delete from dim_terapeuta;
delete from dim_tiempo;
delete from dim_tipo_de_actividad;
delete from dim_tipo_de_comida;
delete from dim_tipo_de_emocion;
delete from dim_tipo_de_episodio;
delete from dim_tipo_entorno;
delete from fact_actividad;
delete from fact_comida;
delete from fact_emocion;
delete from fact_episodio;
delete from raw_actividad;
delete from raw_centro_medico;
delete from raw_ciudad;
delete from raw_comida;
delete from raw_emocion;
delete from raw_episodio;
delete from raw_genero;
delete from raw_paciente;
delete from raw_terapeuta;
delete from raw_tiempo;
delete from raw_tiempo_01;
delete from raw_tipo_de_actividad;
delete from raw_tipo_de_comida;
delete from raw_tipo_de_emocion;
delete from raw_tipo_de_episodio;
delete from raw_tipo_entorno;
delete from stg_actividad;
delete from stg_centro_medico;
delete from stg_ciudad;
delete from stg_comida;
delete from stg_emocion;
delete from stg_episodio;
delete from stg_epoca;
delete from stg_genero;
delete from stg_paciente;
delete from stg_terapeuta;
delete from stg_tiempo;
delete from stg_tipo_de_actividad;
delete from stg_tipo_de_comida;
delete from stg_tipo_de_emocion;
delete from stg_tipo_de_episodio;
delete from stg_tipo_entorno;
```

Load2raw: Paso encargado de la extracción y transformación desde la load a la raw
(ver ilustraciones)

- Ilustración 5 ETL load2raw (parte 1), Ilustración 6 ETL load2raw (parte 2), Ilustración 7 ETL load2raw (parte 3)).

rawTiempo: Paso encargado de filtrar tiempos repetidos y separar la parte de fecha y hora del timestamp de origen (ver

- Ilustración 8 ETL raw_tiempo).

```
SELECT
    CONCAT(
        SUBSTRING_INDEX(DAY, ' ', 1), ' ',
        SUBSTRING_INDEX( SUBSTRING_INDEX(DAY, ' ', -1), ':', 1), ':',
        SUBSTRING_INDEX( SUBSTRING_INDEX(DAY, ' ', -1), ':', -1)) DAY,
    HOUR
FROM tfm_bi_tca.raw_tiempo_01
GROUP BY
    CONCAT(
        SUBSTRING_INDEX(DAY, ' ', 1), ' ',
        SUBSTRING_INDEX( SUBSTRING_INDEX(DAY, ' ', -1), ':', 1), ':',
        SUBSTRING_INDEX( SUBSTRING_INDEX(DAY, ' ', -1), ':', -1))
    , HOUR
```

- **DQ:** Paso encargado de ejecutar las reglas de calidad y marcar aquellos registros que no cumplen poniendo meta_rejected=1 en el registro de la tabla raw que se esté analizando que no cumpla las reglas. (ver script)

```
update `tfm_bi_tca`.`raw_actividad` set meta_is_rejected = 1 where
length(PATIENT) > 3;
update `tfm_bi_tca`.`raw_actividad` set meta_is_rejected = 1 where length(HOUR) >
13;
update `tfm_bi_tca`.`raw_actividad` set meta_is_rejected = 1 where
length(ACTIVITY) > 13;
update `tfm_bi_tca`.`raw_actividad` set meta_is_rejected = 1 where
STR_TO_DATE(concat(substring(DAY,1,2), substring(DAY,4,2) , substring(DAY,7,4)
), '%d%m%Y') IS NULL;
update `tfm_bi_tca`.`raw_actividad` set meta_is_rejected = 1 where
STR_TO_DATE(SUBSTRING_INDEX(DAY, ' ',-1), '%H:%i:%S') IS NULL AND DAY IS NOT
NULL;

update `tfm_bi_tca`.`raw_centro_medico` set meta_is_rejected = 1 where
length(CODE) > 3;
update `tfm_bi_tca`.`raw_centro_medico` set meta_is_rejected = 1 where
length(NAME) > 100;
update `tfm_bi_tca`.`raw_centro_medico` set meta_is_rejected = 1 where length(CP)
> 5;
update `tfm_bi_tca`.`raw_centro_medico` set meta_is_rejected = 1 where
length(CITY) > 100;

update `tfm_bi_tca`.`raw_ciudad` set meta_is_rejected = 1 where length(CP) > 5;
update `tfm_bi_tca`.`raw_ciudad` set meta_is_rejected = 1 where length(CITY) >
100;

update `tfm_bi_tca`.`raw_comida` set meta_is_rejected = 1 where length(PATIENT) >
3;
update `tfm_bi_tca`.`raw_comida` set meta_is_rejected = 1 where length(HOUR) >
13;
```

```

update `tfm_bi_tca`.`raw_comida` set meta_is_rejected = 1 where
STR_TO_DATE(concat(substring(DAY,1,2), substring(DAY,4,2) , substring(DAY,7,4) ),
'%d%m%Y') IS NULL;
update `tfm_bi_tca`.`raw_comida` set meta_is_rejected = 1 where
STR_TO_DATE(SUBSTRING_INDEX(DAY, ' ',-1), '%H:%i:%S') IS NULL AND DAY IS NOT NULL;
update `tfm_bi_tca`.`raw_comida` set meta_is_rejected = 1 where length(MEAL) > 10;

update `tfm_bi_tca`.`raw_emocion` set meta_is_rejected = 1 where length(PATIENT) > 3;
update `tfm_bi_tca`.`raw_emocion` set meta_is_rejected = 1 where length(HOUR) > 13;
update `tfm_bi_tca`.`raw_emocion` set meta_is_rejected = 1 where
STR_TO_DATE(concat(substring(DAY,1,2), substring(DAY,4,2) , substring(DAY,7,4) ),
'%d%m%Y') IS NULL;
update `tfm_bi_tca`.`raw_emocion` set meta_is_rejected = 1 where
STR_TO_DATE(SUBSTRING_INDEX(DAY, ' ',-1), '%H:%i:%S') IS NULL AND DAY IS NOT NULL;
update `tfm_bi_tca`.`raw_emocion` set meta_is_rejected = 1 where length(EMOTION) > 10;

update `tfm_bi_tca`.`raw_episodio` set meta_is_rejected = 1 where length(PATIENT) > 3;
update `tfm_bi_tca`.`raw_episodio` set meta_is_rejected = 1 where length(HOUR) > 13;
update `tfm_bi_tca`.`raw_episodio` set meta_is_rejected = 1 where
STR_TO_DATE(concat(substring(DAY,1,2), substring(DAY,4,2) , substring(DAY,7,4) ),
'%d%m%Y') IS NULL;
update `tfm_bi_tca`.`raw_episodio` set meta_is_rejected = 1 where
STR_TO_DATE(SUBSTRING_INDEX(DAY, ' ',-1), '%H:%i:%S') IS NULL AND DAY IS NOT NULL;
update `tfm_bi_tca`.`raw_episodio` set meta_is_rejected = 1 where length(EPIISODE) >
11;

update `tfm_bi_tca`.`raw_genero` set meta_is_rejected = 1 where length(GENDER) > 1;

update `tfm_bi_tca`.`raw_paciente` set meta_is_rejected = 1 where length(NAME) > 5;
update `tfm_bi_tca`.`raw_paciente` set meta_is_rejected = 1 where length(CITY) > 100;
update `tfm_bi_tca`.`raw_paciente` set meta_is_rejected = 1 where length(GENDER) > 1;
update `tfm_bi_tca`.`raw_paciente` set meta_is_rejected = 1 where length(ENVIRONMENT)
> 5;
update `tfm_bi_tca`.`raw_paciente` set meta_is_rejected = 1 where length(THERAPIST) >
5;

update `tfm_bi_tca`.`raw_terapeuta` set meta_is_rejected = 1 where length(NAME) > 5;
update `tfm_bi_tca`.`raw_terapeuta` set meta_is_rejected = 1 where CONVERT(CENTMED,
SIGNED INTEGER) IS NULL;

update `tfm_bi_tca`.`raw_tiempo` set meta_is_rejected = 1 where length(HOUR) > 13;

update `tfm_bi_tca`.`raw_tiempo` set meta_is_rejected = 1 where
STR_TO_DATE(concat(substring(DAY,1,2), substring(DAY,4,2) , substring(DAY,7,4) ),
'%d%m%Y') IS NULL;
update `tfm_bi_tca`.`raw_tiempo` set meta_is_rejected = 1 where
STR_TO_DATE(SUBSTRING_INDEX(DAY, ' ',-1), '%H:%i:%S') IS NULL AND DAY IS NOT NULL;

update `tfm_bi_tca`.`raw_tipo_de_actividad` set meta_is_rejected = 1 where
length(ACTIVITY) > 13;

update `tfm_bi_tca`.`raw_tipo_de_comida` set meta_is_rejected = 1 where length(MEAL) >
10;

update `tfm_bi_tca`.`raw_tipo_de_emocion` set meta_is_rejected = 1 where
length(EMOTION) > 10;

update `tfm_bi_tca`.`raw_tipo_de_episodio` set meta_is_rejected = 1 where
length(EPIISODE) > 11;

update `tfm_bi_tca`.`raw_tipo_entorno` set meta_is_rejected = 1 where
length(ENVIRONMENT) > 5;

```

- **raw2stg:** Paso encargado de pasar de algunas raws que no tienen dependencias con otras a su área de staging, además realiza toda transformación para obtener los campos calculados de tiempo (ver Ilustración 9 ETL raw2stg)

```
SELECT
  GENDER
, meta_is_rejected
FROM raw_genero
where meta_is_rejected=0
```

```
select
  STR_TO_DATE(concat(substring(DAY,1,2),'-' ,substring(DAY,4,2) , '-'
, substring(DAY,7,4) ), '%d-%m-%Y') as fecha,
  substring(DAY,7,4) as año,
  substring(DAY,4,2) as mes,
  substring(DAY,1,2) as dia,
  HOUR franja,
  concat(STR_TO_DATE(concat(substring(DAY,7,4) , '-' ,substring(DAY,4,2)
, '-' ,substring(DAY,1,2)), '%Y-%m-%d')
, ' ' ,SUBSTRING_INDEX(DAY,' ',-1) ) as timestamp_calc,

  DAYOFWEEK(STR_TO_DATE(concat(substring(DAY,1,2),'-' ,substring(DAY,4,2)
, '-' ,substring(DAY,7,4) ), '%d-%m-%Y') ) dia_semana,
  case
    when (substring(DAY,4,2) >0 and substring(DAY,4,2) <4) then 1
    when (substring(DAY,4,2) >=4 and substring(DAY,4,2) <7) then 2
    when substring(DAY,4,2) >=7 and substring(DAY,4,2) <10 then 3
    when substring(DAY,4,2) >=10 and substring(DAY,4,2) <13 then 4
  end as trimestre,
  case
    when (substring(DAY,4,2) >0 and substring(DAY,4,2) <7) then 1
    when (substring(DAY,4,2) >=6 and substring(DAY,4,2) <13) then 2
  end as semestre,
  WEEKOFYEAR(STR_TO_DATE(concat(substring(DAY,1,2),'-' ,substring(DAY,4,2) , '-'
, substring(DAY,7,4) ), '%d-%m-%Y')) semana_del_año,
  case
    when concat( substring(DAY,4,2) ,substring(DAY,1,2)) >= 101 and
concat( substring(DAY,4,2) ,substring(DAY,1,2))<=320 then 'I'
    when concat( substring(DAY,4,2) ,substring(DAY,1,2)) >= 321 and
concat( substring(DAY,4,2) ,substring(DAY,1,2))<=620 then 'P'
    when concat( substring(DAY,4,2) ,substring(DAY,1,2)) >= 621 and
concat( substring(DAY,4,2) ,substring(DAY,1,2))<=920 then 'V'
    when concat( substring(DAY,4,2) ,substring(DAY,1,2)) >=921 and concat(
substring(DAY,4,2) ,substring(DAY,1,2))<=1220 then 'O'
    when concat( substring(DAY,4,2) ,substring(DAY,1,2)) >= 1221 and concat(
substring(DAY,4,2) ,substring(DAY,1,2))<=1231 then 'I'
  end as estacion,
  DAYOFYEAR(STR_TO_DATE(concat(substring(DAY,1,2),'-' ,substring(DAY,4,2) , '-'
, substring(DAY,7,4) ), '%d-%m-%Y')) dia_del_año,
  ceiling(DAYOFYEAR(STR_TO_DATE(concat(substring(DAY,1,2),'-'
, substring(DAY,4,2) , '-' ,substring(DAY,7,4) ), '%d-%m-%Y')) /15) quincena,
  SUBSTRING_INDEX(DAY,' ',-1) hora,
  DAY timestamp_original
from `tfm_bi_tca`.`raw_tiempo`
where meta_is_rejected=0
```

- **stg2dim:** Para las mismas entidades que no dependen de ninguna otra en este paso se cargan las tablas de dimensión, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (**ver**

- Ilustración 10 ETL stg2dim y ver script)

```
SELECT nombre_tipo_entorno,  
       current_timestamp as  
       meta_update_time  
FROM stg_tipo_entorno
```

- **raw2stg0:** En este paso se carga la stg_ciudad con la raw table que le corresponde (ver ilustración)
- **stg2dim1:** Paso que lleva los datos del área de staying a su correspondiente tabla de dimensión dim_ciudad, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (ver ilustración).
- **raw2stg6:** En este paso se carga la stg_centro_medico con la raw table que le corresponde (ver ilustración)
- **stg2dim2:** Paso que lleva los datos del área de staying a su correspondiente tabla de dimensión dim_centro_medico, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (ver ilustración).
- **raw2stg2** En este paso se carga la stg_terapeuta la raw table que le corresponde (ver ilustración)
- **stg2dim4** Paso que lleva los datos del área de staying a su correspondiente tabla de dimensión dim_terapeuta, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (ver ilustración).
- **raw2stg3** En este paso se carga la stg_paciente con la raw table que le corresponde (ver ilustración)
- **stg2dim5** Paso que lleva los datos del área de staying a su correspondiente tabla de dimensión dim_paciente, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (ver ilustración).
- **raw2stg4** En este paso se cargan las staying de los hechos, la stg_episodio, stg_emocion, stg_actividad, stg_comida con la raw table que le corresponde (ver ilustración)
- **stg2fact** Paso que lleva los datos del área de staying a su correspondiente tabla de dimensión dim_paciente, además se añade el timestamp de carga utilizando el campo *current_timestamp* SQL. (ver ilustración).

- **dimfact2dataMart:** En este paso se carga la tabla sumariada de del datamart fact_eventos_alimentarios (ver script).

```

select
  `a`.`id_paciente` AS `id_paciente`,
  `a`.`id_dim_tiempo` AS `id_dim_tiempo`,
  `a`.`id_actividad` AS `id_actividad`,
  `b`.`id_dim_comida` AS `id_dim_comida`,
  `c`.`id_dim_episodio` AS `id_dim_episodio`,
  `d`.`id_dim_emocion` AS `id_dim_emocion`,
  `e`.`nombre_paciente` AS `nombre_paciente`,
  `e`.`id_terapeuta` AS `id_terapeuta`,
  `e`.`id_ciudad_vive` AS `id_ciudad_vive`,
  `e`.`id_dim_genero` AS `id_dim_genero`,
  `f`.`nom_terapeuta` AS `nom_terapeuta`,
  `f`.`id_centro_medico` AS `id_centro_medico`,
  `h`.`id_dim_tipo_entorno` AS `id_tipo_entorno`
from
  ((((`tfm_bi_tca`.`fact_actividad` `a`
  join `tfm_bi_tca`.`fact_comida` `b` ON
  (((`a`.`id_paciente` = `b`.`id_paciente`)
  and (`a`.`id_dim_tiempo` = `b`.`id_dim_tiempo`))))
  join `tfm_bi_tca`.`fact_episodio` `c` ON
  (((`a`.`id_paciente` = `c`.`id_paciente`)
  and (`a`.`id_dim_tiempo` = `c`.`id_dim_tiempo`))))
  join `tfm_bi_tca`.`fact_emocion` `d` ON
  (((`a`.`id_paciente` = `d`.`id_paciente`)
  and (`a`.`id_dim_tiempo` = `d`.`id_dim_tiempo`))))
  left join `tfm_bi_tca`.`dim_paciente` `e` ON
  ((`a`.`id_paciente` = `e`.`id_paciente`))
  left join `tfm_bi_tca`.`dim_terapeuta` `f` ON
  ((`f`.`id_dim_terapeuta` = `e`.`id_terapeuta`))
  left join `tfm_bi_tca`.`dim_ciudad` `g`
  ON (e.id_ciudad_vive=g.id_dim_ciudad)
  left join `tfm_bi_tca`.`dim_tipo_entorno` `h`
  ON
  (g.`id_dim_tipo_entorno`=h.id_dim_tipo_entorno)

```

Ilustración 5 ETL load2raw (parte 1)

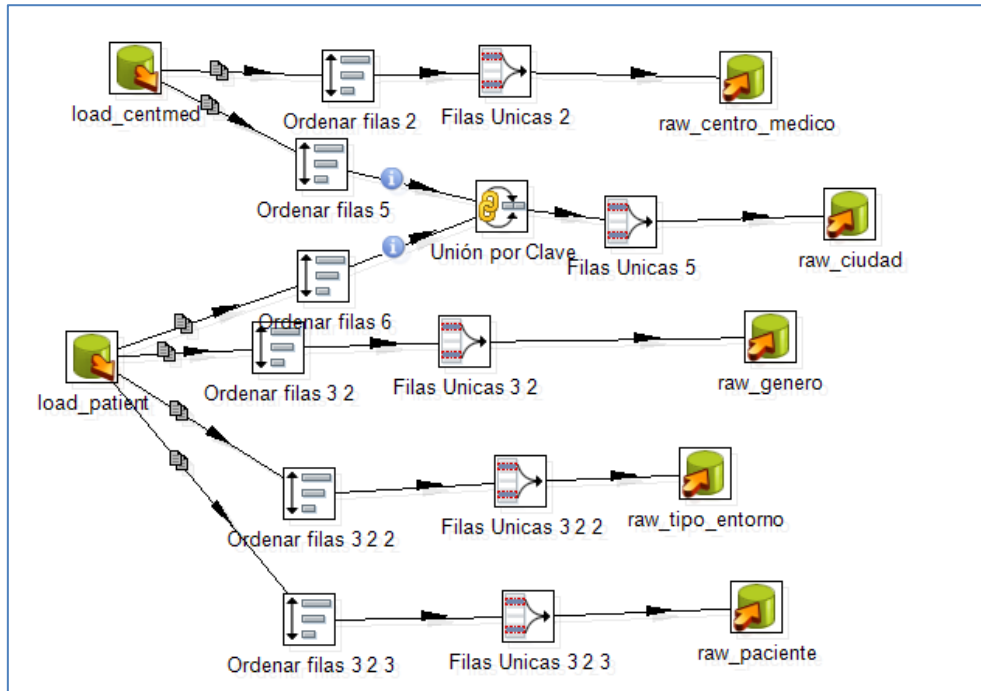


Ilustración 6 ETL load2raw (parte 2)

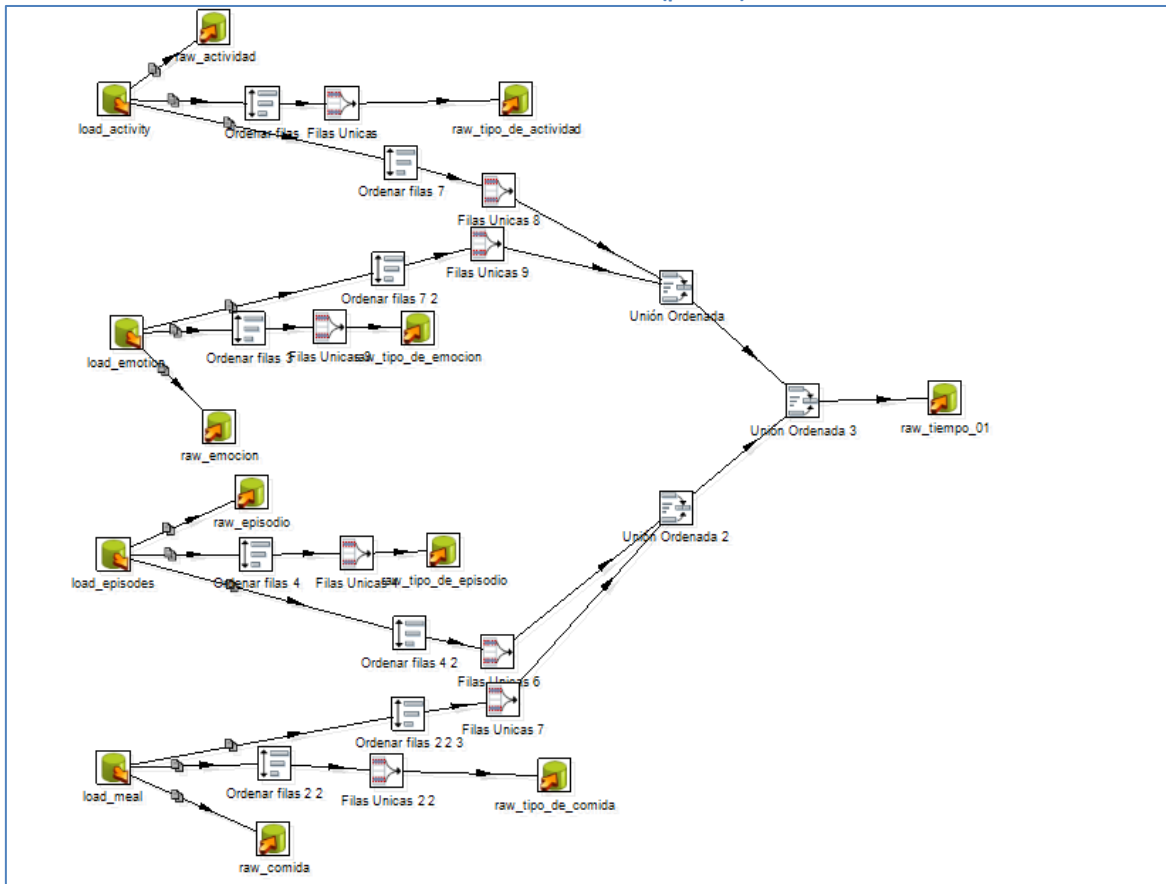


Ilustración 7 ETL load2raw (parte 3)

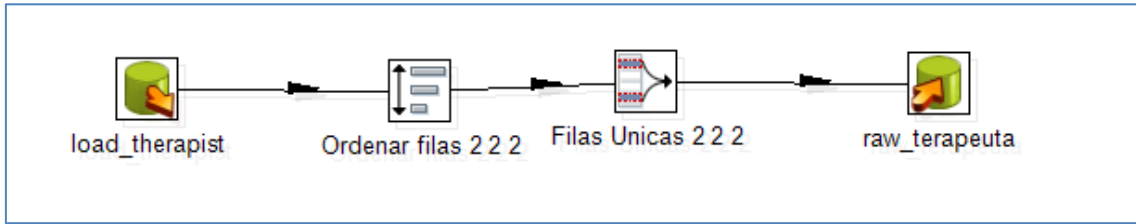


Ilustración 8 ETL raw_tiempo

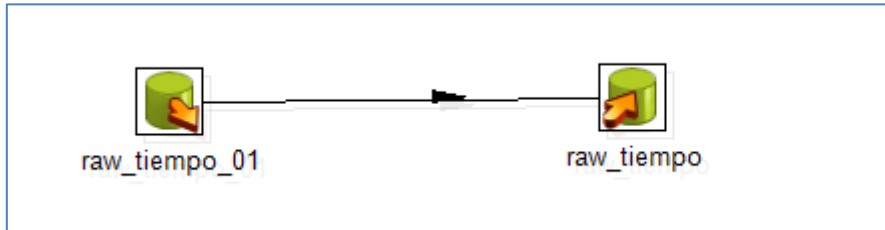


Ilustración 9 ETL raw2stg

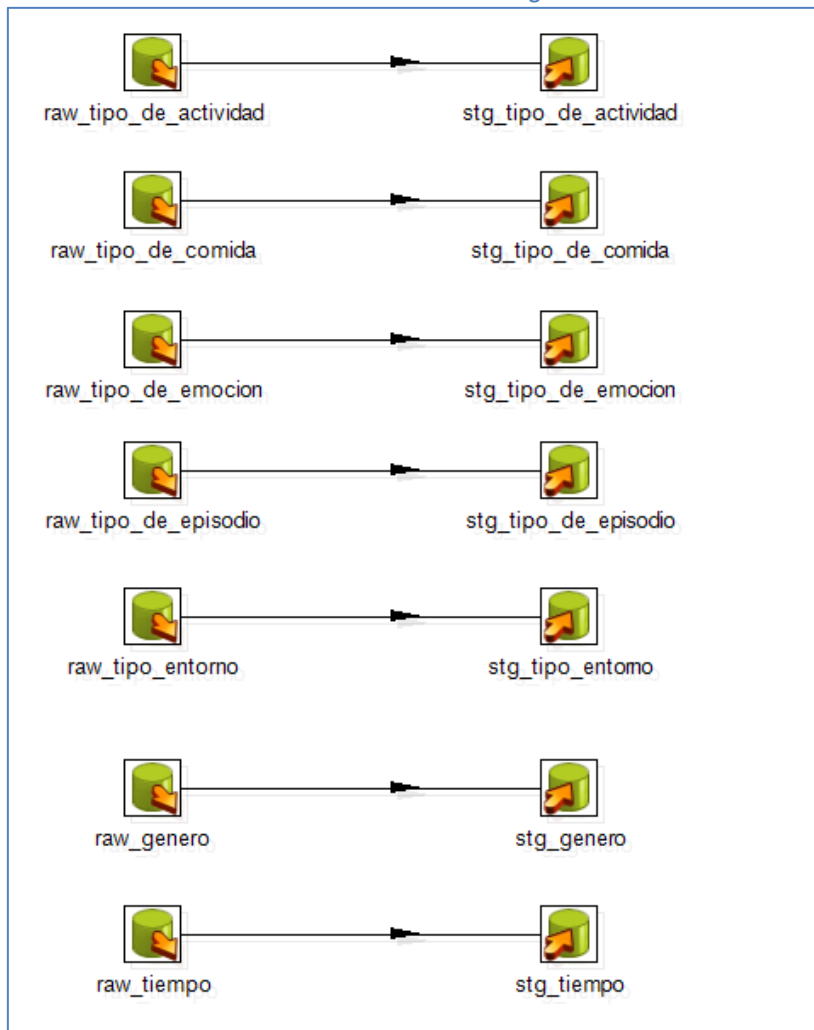


Ilustración 10 ETL stg2dim

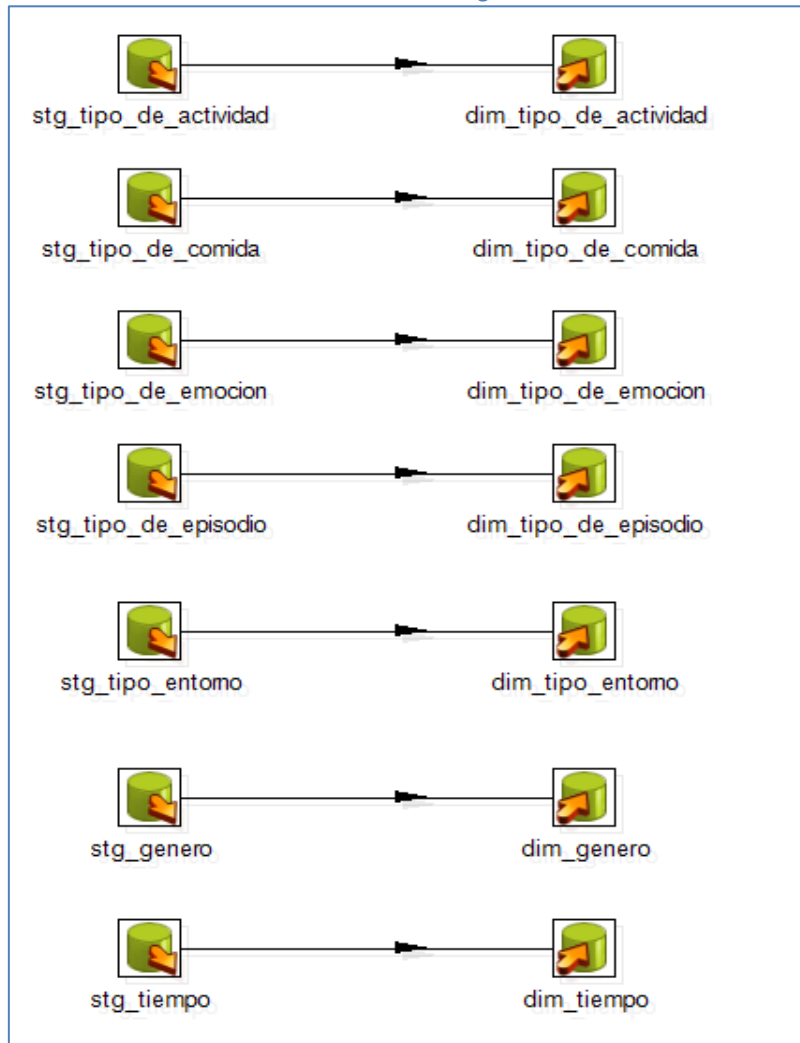


Ilustración 11 ETL raw2stg0



Ilustración 12 stg2dim1

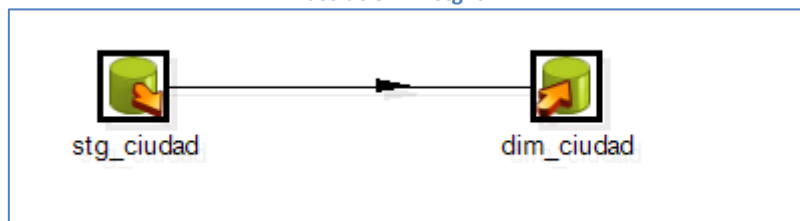


Ilustración 13 raw2stg6

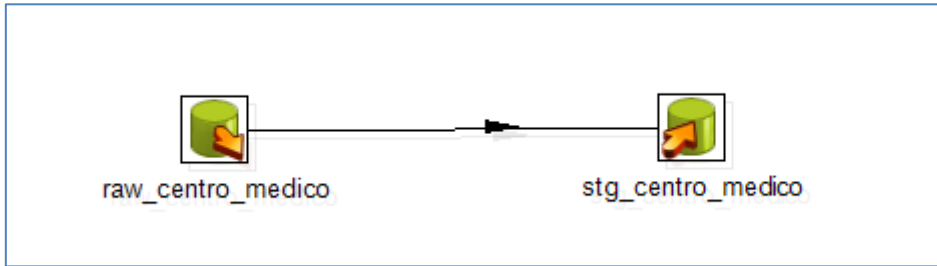


Ilustración 14 ETL stg2dim2



Ilustración 15 ETL raw2stg2

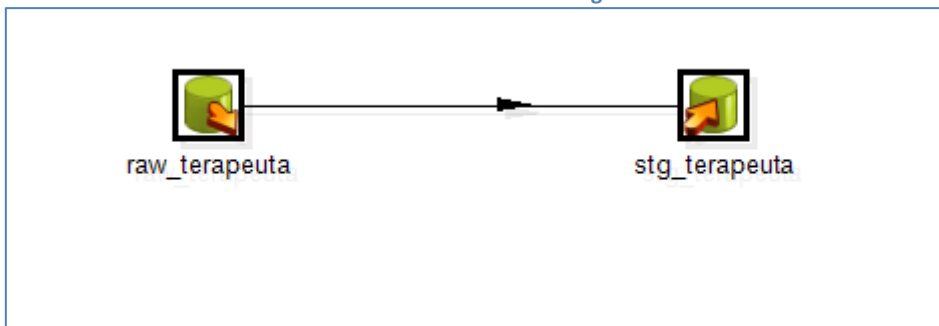


Ilustración 16 ETL stg2dim4

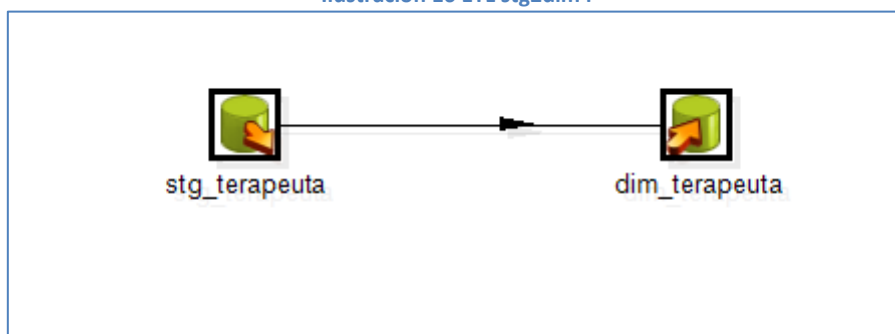


Ilustración 17 ETL raw2stg3

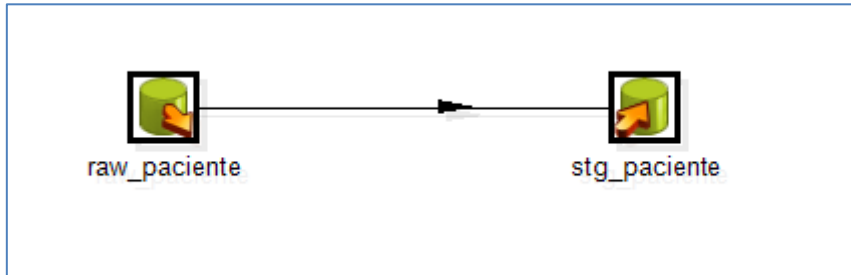


Ilustración 18 ETL stg2dim5

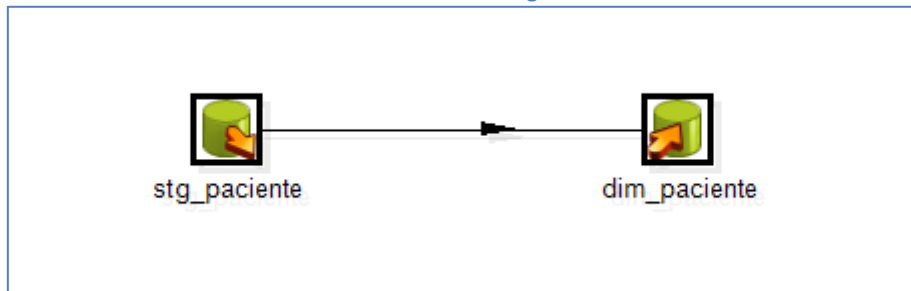


Ilustración 19 ETL raw2stg4

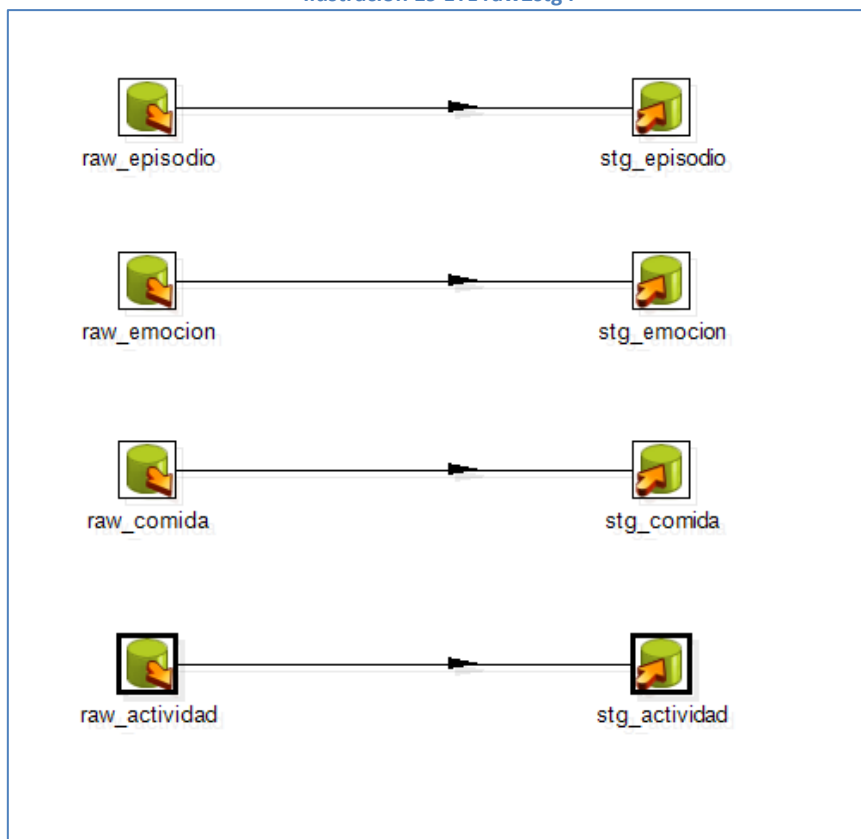


Ilustración 20 ETL stg2fact

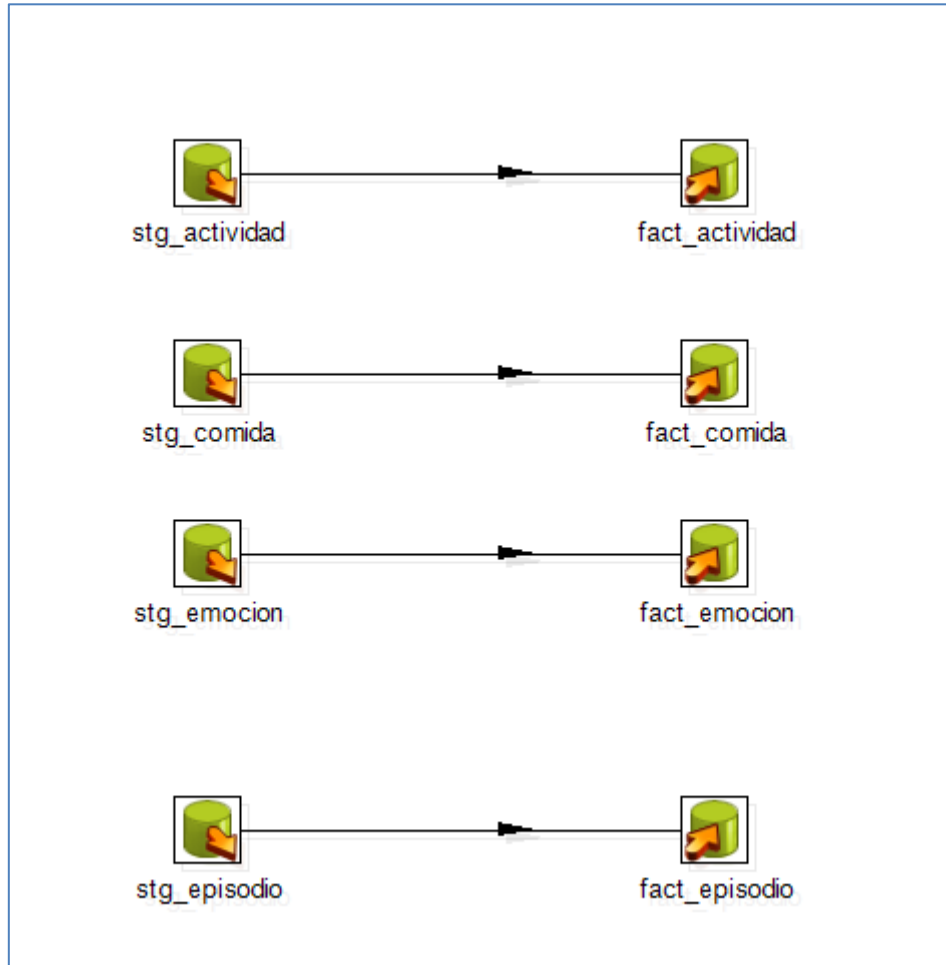
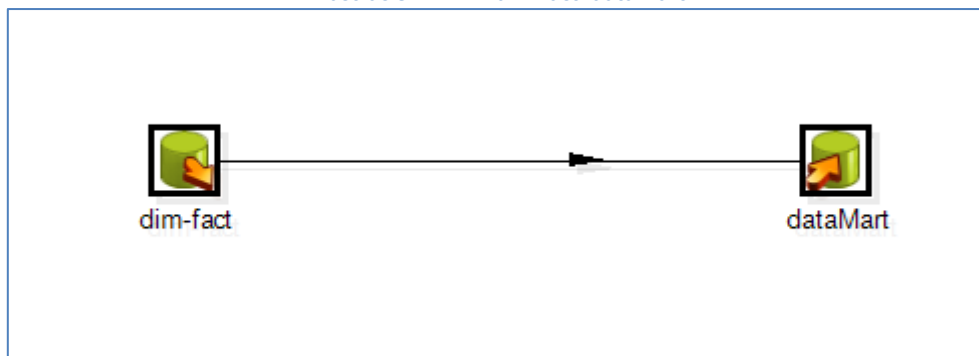


Ilustración 21 ETL dim-fact2dataMart



4.8. Cubos

Es un paso necesario generar los cubos OLAP para poder hacer un análisis multidimensional, estos son los cubos que he generado:

```

<Schema name="Terapeuta">
  <Cube name="emocion_episodio" caption="emocion_episodio" visible="true" cache="true"
enabled="true">

    <Table name="fact_eventos_alimentarios">
    </Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_emocion"
highCardinality="false" name="tipo_emocion">
      <Hierarchy name="emocion" visible="true" hasAll="true">
        <Table name="dim_tipo_de_emocion">
        </Table>
        <Level name="emocion" visible="true" table="dim_tipo_de_emocion"
column="id_dim_tipo_de_emocion" nameColumn="nombre_emocion" type="String"
uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
      <Hierarchy name="episodio" visible="true" hasAll="true">
        <Table name="dim_tipo_de_episodio">
        </Table>
        <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" name="tipo_entorno">
      <Hierarchy name="entorno" visible="true" hasAll="true">
        <Table name="dim_tipo_entorno" alias="">
        </Table>
        <Level name="entorno" visible="true" table="dim_tipo_entorno"
column="id_dim_tipo_entorno" nameColumn="nombre_tipo_entorno"
uniqueMembers="false">
        </Level>
      </Hierarchy>
    </Dimension>
    <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">

```

```

</Measure>
<Measure name="cuenta_eventos_porcentaje" column="id_eventos_alimentarios"
datatype="Numeric" aggregator="distinct count" visible="true">
</Measure>
</Cube>
<Cube name="emocion_episodio_entorno" caption="emocion_episodio_entorno"
visible="true" cache="true" enabled="true">
<Table name="fact_eventos_alimentarios">
</Table>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
<Hierarchy name="episodio" visible="true" hasAll="true">
<Table name="dim_tipo_de_episodio">
</Table>
<Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
</Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_entorno"
highCardinality="false" name="tipo_entorno">
<Hierarchy name="entorno" visible="true" hasAll="true">
<Table name="dim_tipo_entorno">
</Table>
<Level name="entorno" visible="true" table="dim_tipo_entorno"
column="id_dim_tipo_entorno" nameColumn="nombre_tipo_entorno" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
</Level>
</Hierarchy>
</Dimension>
<Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
</Measure>
<Measure name="cuenta_eventos_porcentaje" column="id_eventos_alimentarios"
datatype="Numeric" aggregator="distinct count" visible="true">
</Measure>
</Cube>
<Cube name="2.-Relacion-emocion-episodio" caption="2.-Relacion-emocion_episodio"
visible="true" cache="true" enabled="true">
<Table name="fact_eventos_alimentarios">
</Table>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_emocion"
highCardinality="false" name="tipo_emocion">
<Hierarchy name="emocion" visible="true" hasAll="true">

```

```

    <Table name="dim_tipo_de_emocion">
    </Table>
    <Level name="emocion" visible="true" table="dim_tipo_de_emocion"
column="id_dim_tipo_de_emocion" nameColumn="nombre_emocion" type="String"
uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
    <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
    </Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
<Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
</Measure>
</Cube>
<Cube name="3.- Relacion episodio entorno" caption="3.- Relacion episodio entorno"
visible="true" cache="true" enabled="true">
    <Table name="fact_eventos_alimentarios">
    </Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
    <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
    </Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_entorno"
highCardinality="false" name="tipo_entorno">
    <Hierarchy name="entorno" visible="true" hasAll="true">
    <Table name="dim_tipo_entorno">
    </Table>

```

```

    <Level name="entorno" visible="true" table="dim_tipo_entorno"
column="id_dim_tipo_entorno" nameColumn="nombre_tipo_entorno" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
<Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
</Measure>
</Cube>
<Cube name="3.b.- Relacion episodio ciudad" caption="3.b - Relacion episodio ciudad"
visible="true" cache="true" enabled="true">
    <Table name="fact_eventos_alimentarios">
</Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
    <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
</Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_ciudad_vive"
highCardinality="false" name="ciudad">
    <Hierarchy name="ciudad" visible="true" hasAll="true">
    <Table name="dim_ciudad">
</Table>
    <Level name="ciudad" visible="true" table="dim_ciudad" column="id_dim_ciudad"
nameColumn="nom_ciudad" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
    </Level>
</Hierarchy>
</Dimension>
    <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
</Measure>
</Cube>
<Cube name="4.-d Evolucion paciente" visible="true" cache="true" enabled="true">
    <Table name="mart_paciente_evolucion" alias="">
</Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="nombre_paciente"
name="paciente">

```

```

<Hierarchy name="paciente" visible="true" hasAll="true">
  <Table name="dim_paciente" alias="">
  </Table>
  <Level name="paciente" visible="true" column="nombre_paciente"
nameColumn="nombre_paciente" uniqueMembers="false">
  </Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
name="episodio">
  <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio" alias="">
    </Table>
    <Level name="episodio" visible="true" column="id_dim_tipo_de_episodio"
nameColumn="nombre_episodio" uniqueMembers="false">
    </Level>
  </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="a&#241;o"
name="a&#241;o">
  <Hierarchy name="a&#241;o" visible="true" hasAll="true">
    <Table name="dim_tiempo" alias="">
    </Table>
    <Level name="a&#241;o" visible="true" column="a&#241;o" nameColumn="a&#241;o"
uniqueMembers="false">
    </Level>
  </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="mes" name="mes">
  <Hierarchy name="mes" visible="true" hasAll="true">
    <Table name="dim_tiempo" alias="">
    </Table>
    <Level name="mes" visible="true" column="mes" nameColumn="mes"
uniqueMembers="false">
    </Level>
  </Hierarchy>
</Dimension>
<Measure name="cuenta_episodio" column="id_dim_episodio" aggregator="count"
visible="true">
  </Measure>
</Cube>
<Cube name="6.- Relacion_episodio_tiempo_estacion_del_anyo" caption="6.-
Relacion_episodio_tiempo_estacion_del_anyo" visible="false" cache="true" enabled="true">
  <Table name="fact_eventos_alimentarios">
  </Table>

```

```

    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
    <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
    </Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_tiempo"
highCardinality="false" name="tiempo">
    <Hierarchy name="estaciondeLa" visible="true" hasAll="true">
    <Table name="dim_tiempo">
    </Table>
    <Level name="estacion" visible="true" table="dim_tiempo" column="id_dim_tiempo"
nameColumn="estacion" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
    </Level>
    </Hierarchy>
</Dimension>
    <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
    </Measure>
</Cube>
    <Cube name="6.b- Relacion_episodio_tiempo_franja" caption="6.b-
Relacion_episodio_tiempo_franja" visible="true" cache="true" enabled="true">
    <Table name="fact_eventos_alimentarios">
    </Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
    <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
    </Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_tiempo"
highCardinality="false" name="tiempo">
    <Hierarchy name="franja" visible="true" hasAll="true">
    <Table name="dim_tiempo">

```

```

</Table>
  <Level name="franja" visible="true" table="dim_tiempo" column="id_dim_tiempo"
nameColumn="franja" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
  </Level>
</Hierarchy>
</Dimension>
  <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
  </Measure>
</Cube>
  <Cube name="6.c- Relacion_episodio_tiempo_mes" caption="6.b-
Relacion_episodio_tiempo_mes" visible="true" cache="true" enabled="true">
  <Table name="fact_eventos_alimentarios">
  </Table>
  <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
  <Hierarchy name="episodio" visible="true" hasAll="true">
  <Table name="dim_tipo_de_episodio">
  </Table>
  <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
  </Level>
  </Hierarchy>
</Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_tiempo"
highCardinality="false" name="tiempo">
  <Hierarchy name="mes" visible="true" hasAll="true">
  <Table name="dim_tiempo">
  </Table>
  <Level name="mes" visible="true" table="dim_tiempo" column="id_dim_tiempo"
nameColumn="mes" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
  </Level>
  </Hierarchy>
</Dimension>
  <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
  </Measure>
</Cube>
  <Cube name="6.d- Relacion_episodio_tiempo_dia_semana" caption="6.d-
Relacion_episodio_tiempo_dia_semana" visible="true" cache="true" enabled="true">
  <Table name="fact_eventos_alimentarios">
  </Table>

```

```

<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
highCardinality="false" name="tipo_episodio">
  <Hierarchy name="episodio" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio">
    </Table>
    <Level name="episodios" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_tiempo"
highCardinality="false" name="tiempo">
  <Hierarchy name="dia_semana" visible="true" hasAll="true">
    <Table name="dim_tiempo">
    </Table>
    <Level name="dia_semana" visible="true" table="dim_tiempo" column="id_dim_tiempo"
nameColumn="diasemana" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
<Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
</Measure>
</Cube>
<Cube name="evolucion_centro_medico" caption="evolucion_centro_medico" visible="true"
cache="true" enabled="true">
  <Dimension type="StandardDimension" visible="true" foreignKey="id_paciente"
highCardinality="false" name="paciente">
    <Hierarchy name="paciente" visible="true" hasAll="true">
      <Table name="dim_paciente">
      </Table>
      <Level name="paciente" visible="true" table="dim_paciente" column="id_paciente"
nameColumn="nombre_paciente" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="id_dim_tiempo"
highCardinality="false" name="tiempo">
    <Hierarchy name="tiempo" visible="true" hasAll="true">
      <Table name="dim_tiempo" alias="">
      </Table>

```



```
<Level name="timestamp" visible="true" table="dim_tiempo" column="id_dim_tiempo"
nameColumn="fecha" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
  </Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="id_dim_episodio"
name="episodio">
  <Hierarchy name="episodios" visible="true" hasAll="true">
    <Table name="dim_tipo_de_episodio" alias="">
      </Table>
    <Level name="episodio" visible="true" table="dim_tipo_de_episodio"
column="id_dim_tipo_de_episodio" nameColumn="nombre_episodio"
uniqueMembers="false">
      </Level>
    </Hierarchy>
  </Dimension>
  <Measure name="cuenta_eventos" column="id_eventos_alimentarios" datatype="Numeric"
aggregator="count" visible="true">
    </Measure>
  </Cube>
</Schema>
```

4.9. Los informes

Se han definido una serie de reports en el servidor de Business Intelligence utilizando el plugin JPivot disponible en el Marketplace, que si bien no es el más sofisticado, sí es el más fácil de instalar, más estable y está en versiones más maduras que otras opciones como Pivot4J o Saiku, la interfaz de este plugin es bastante sencilla y fácil de aprender.

A continuación veremos, uno a uno, cada uno de los informes creados para cubrir las necesidades de este trabajo:

4.9.1. Informe 1.- Relacion_actividad_episodio

En este informe de dos dimensiones, actividad y episodio y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa actividad y episodio. En él se puede ver claramente la relación que hay entre la actividad que se estaba haciendo cuando se ha producido el episodio correspondiente.

		Measures
actividad	episodio	● cuenta_eventos
SPORTS	VOMITED	
	RESTRICTION	1
	BINGE	1
FRIENDS	VOMITED	104
	RESTRICTION	107
	BINGE	5
FAMILY	VOMITED	
	RESTRICTION	1
	BINGE	
ALONE	VOMITED	213
	RESTRICTION	237
	BINGE	392

Figura.- 1.- Cuadro de episodios por actividad con ocurrencias totales

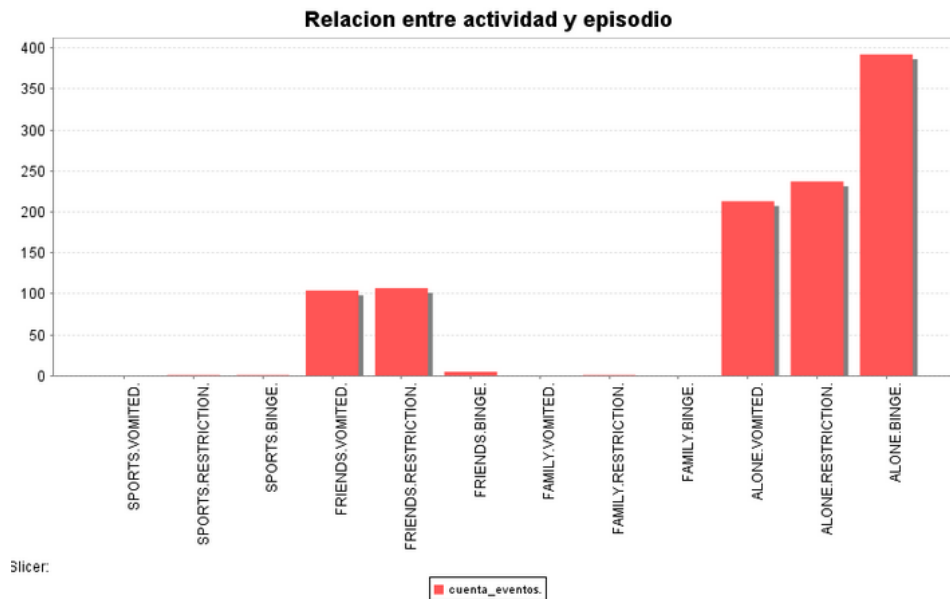


Figura.- 2 .- Grafico de episodios por actividad con ocurrencias totales

4.9.2. Informe 2.- Relación episodios y estados de animo

En este informe de dos dimensiones, estado de ánimo y episodio y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de ese estado de ánimo y episodio. En él se puede ver claramente la relación que hay entre el estado de ánimo que se estaba experimentando cuando se ha producido el episodio correspondiente.

	Measures			
	emocion			
episodio	● DISTRESSED	● FUSTRATED	● GUILTY	● SAD
BINGE	104	94	96	104
RESTRICTION	84	101	88	73
VOMITED	78	76	87	76

Slicer:



Slicer:



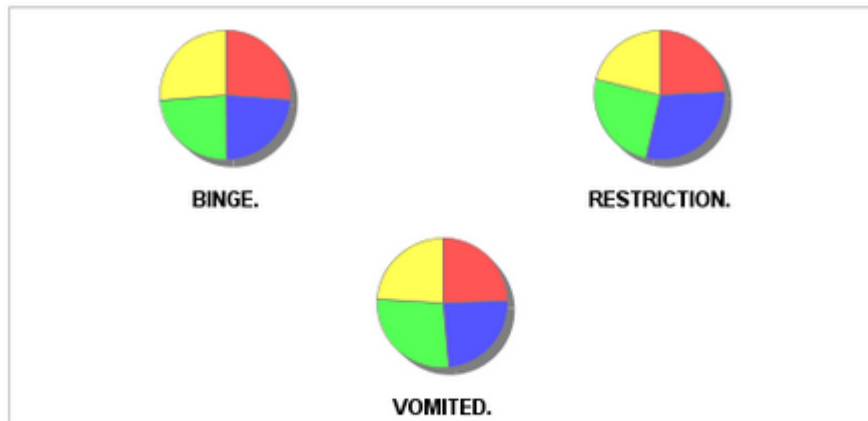
Figura.- 3 - Tabla y grafico de episodios por estado de animo

4.9.3. Informe 2.- Relación episodios y estados de ánimo (inversa)

En este informe de dos dimensiones, estado de ánimo y episodio y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de ese estado de ánimo y episodio. En él se puede ver claramente la relación que hay entre el estado de ánimo que se estaba experimentando cuando se ha producido el episodio correspondiente. Es parecido al informe anterior pero con las columnas transpuestas por las filas.

		episodio		
Measures	emocion	BINGE	RESTRICTION	VOMITED
cuenta_eventos	DISTRESSED	104	84	78
	FUSTRATED	94	101	76
	GUILTY	96	88	87
	SAD	104	73	76

Slicer:



Slicer:

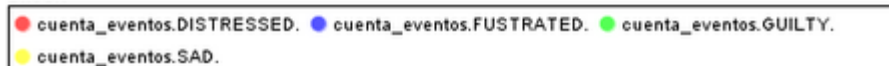


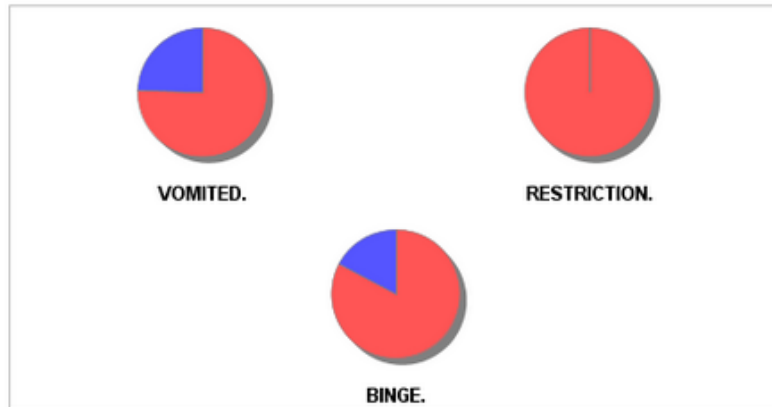
Figura.- 4 Tabla y grafico Relación entre episodios por estados de ánimo

4.9.4. Informe 3.- Relación episodio entorno - inversa totales

En este informe de dos dimensiones, entorno geográfico y episodio y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de ese entorno geográfico y episodio. En él se puede ver claramente la relación que hay entre el entorno geográfico en el que el paciente vive y el episodio correspondiente.

episodio	Measures	
	cuenta_eventos	
	entorno	
	• URBAN	• RURAL
VOMITED	163	53
RESTRICTION	81	
BINGE	124	26

Slicer:



Slicer:



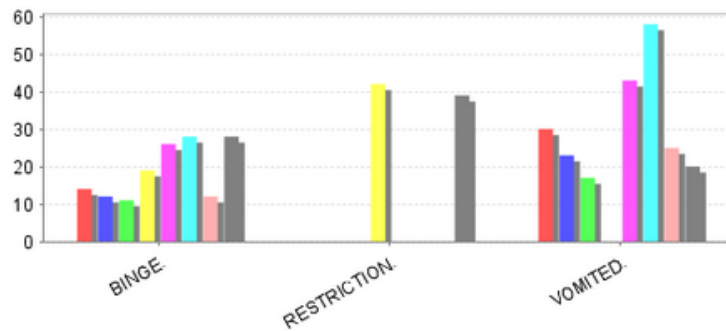
Figura.- 5 Tabla y grafico Relación entre episodios y entorno

4.9.5. Informe 3.b.- Relación episodio entorno (ciudad)

En este informe de dos dimensiones, ciudad donde vive y episodio y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa ciudad y episodio. En él se puede ver claramente la relación que hay entre las ciudades en el que el paciente vive y el episodio correspondiente.

		Measures							
		cuenta_eventos							
		ciudad							
episodio		• MERIDA	• TORTOSA	• A CORUÑA	• BARCELONA	• BILBAO	• GRANADA	• MADRID	• SEVILLA
BINGE		14	12	11	19	26	28	12	28
RESTRICTION					42				39
VOMITED		30	23	17		43	58	25	20

Slicer:



Slicer:



Figura.- 6 Tabla y grafico Relación entre episodios y la ciudad donde habita el paciente

4.9.6. Informe 4.d- Evolución paciente desde datamart (evolución pacientes críticos)

En este informe de tres dimensiones, paciente, fecha y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para ese paciente, de ese episodio. En él se puede ver una selección de los pacientes con más episodios, en cada mes, para ver qué evolución tienen.

			Measures			
			Id dim episodio			
			Id dim episodio			
Nombre paciente	Año	Mes	200	199	198	#null
P16	2015	11	5	4		
		12	5	14		
	2016	01	3	21		
P20	2015	11	4		3	
		12	8		9	
	2016	01	9		14	
P27	2015	11	3	8		
		12	6	11		
	2016	01	6	14		

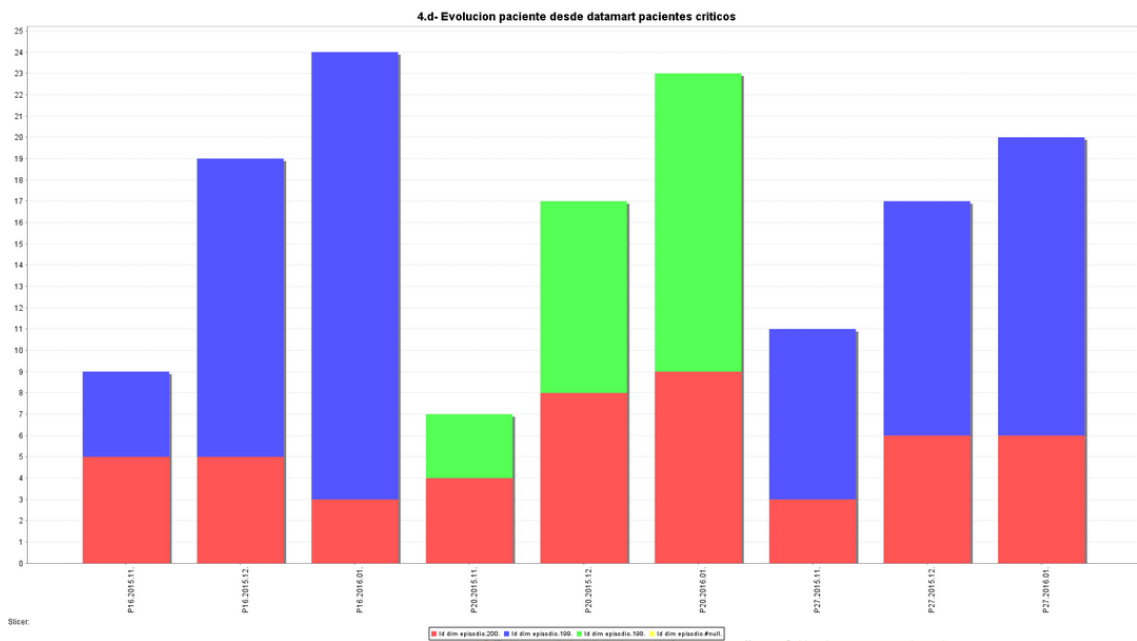


Figura.- 7 Tabla y grafico de evolución de los pacientes más críticos

4.9.7. Informe 4.d- Evolución paciente desde datamart (evolución todos)

En este informe de tres dimensiones, paciente, fecha y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para ese paciente, de ese episodio. En él se puede ver una selección de los pacientes con más episodios, en cada mes, para ver qué evolución tienen. Similar al informe anterior, pero agrupando todos los pacientes para ver la evolución general.

			Measures			
			Id dim episodio			
			Id dim episodio			
Nombre paciente	Año	Mes	● 200	● 199	● 198	● #null
+ All Nombre pacientes	2015	11	177	134	126	
		12	115	93	84	
	2016	01	106	119	107	

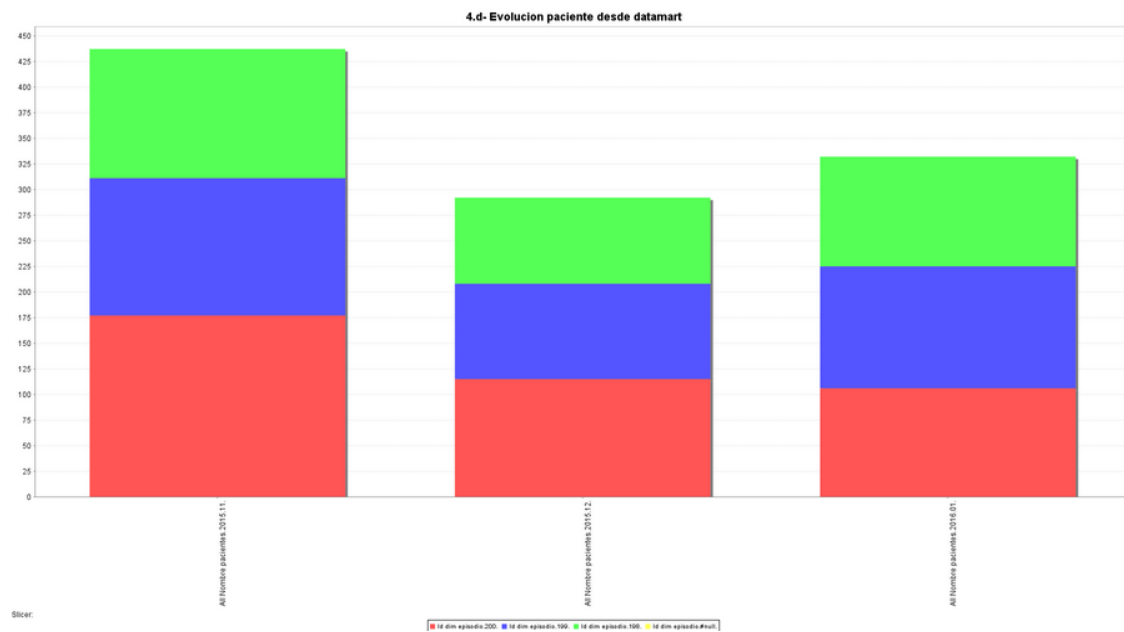


Figura.- 8 Tabla y grafico de evolución de todos los pacientes

4.9.8. Informe 4.- Evolución paciente (agrupado)

En este informe de dos dimensiones, paciente, fecha, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para ese paciente. En él se puede ver una el detalle de todas las fechas en el que cada paciente tiene episodios y se puede ver claramente qué pacientes son los que tienen más episodios.

Measures	paciente	tiempo				
		2016-01-01	2015-11-01	2015-12-01	2016-01-02	2015-11-02
cuenta_eventos	P1					
	P15		1			
	P16				1	
	P17					
	P22		1			
	P25		1			1
	P28					
	P29		1	1		
	P3				1	
	P8					
	P10		1			
	P13					
	P2					1
	P18					
	P19	1				
	P20					
	P21			1		1
	P23	1	1		1	
	P24		1			1
	P26		1			1
P27				1		
P30	1					
P4		1				
P5					1	
P6					1	
P7		1				
P9		1				
P11		1				
P12					1	
P14					1	

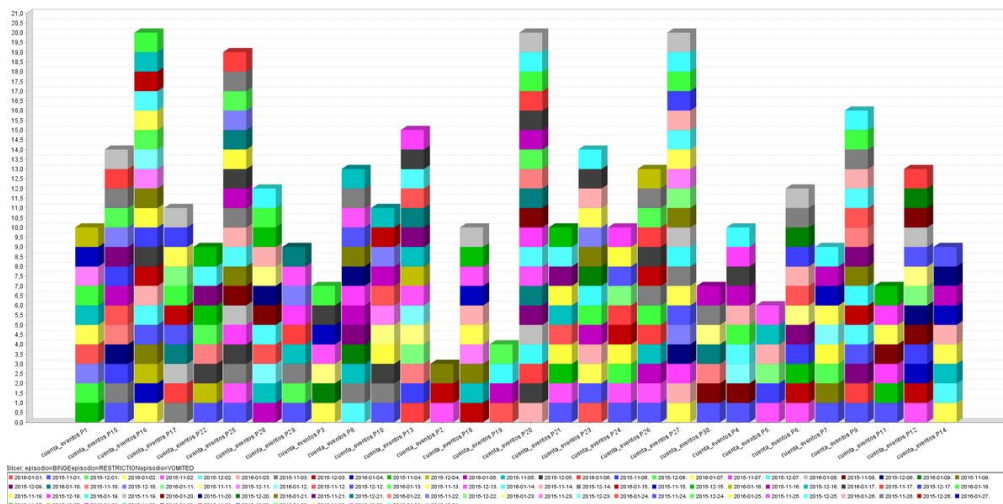


Figura.- 9 Tabla y grafico los episodios de todos los pacientes

4.9.9. Informe 4.- Evolución paciente (detalle)

En este informe de tres dimensiones, paciente, fecha y tipo de episodio, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, con ese episodio y paciente. En él se puede ver una el detalle de todos los episodios que han sufrido cada paciente. Se puede ver claramente qué pacientes son los que tienen más episodios.

		tiempo						
		2016-01-01			2015-11-01			
		episodios						
Measures	paciente	VOMITED	RESTRICTION	BINGE	RESTRICTION	BINGE	VOMITED	RESTRIC
cuenta_eventos	P1							
	P15						1	
	P16					1		
	P17							
	P22							1
	P25							1
	P28							
	P29							1
	P3							
	P8							
	P10							
	P13				1			
	P2				1			
	P18							
	P19	1					1	
	P20							
	P21							1
	P23			1				
	P24							
	P26							
	P27							
	P30							
	P4							
	P5							
	P6							
	P7							1
	P9						1	
	P11							1
	P12							
	P14							

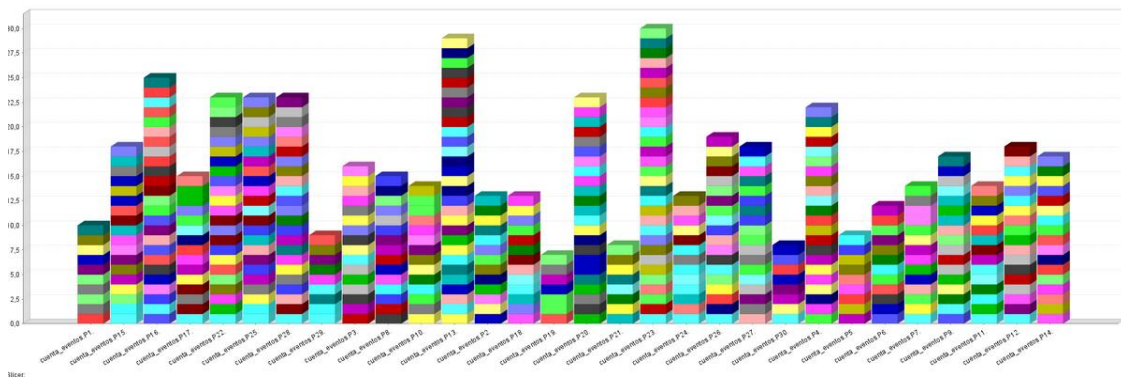


Figura.- 10 Evolución paciente, cantidad de episodios

4.9.10. Informe 4.- Evolución paciente (temporal)

En este informe de tres dimensiones, paciente, fecha y tipo de episodio, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, con ese episodio y paciente, está filtrado por un paciente en concreto. En él se pueden ver los episodios que ha tenido el paciente a lo largo del tiempo.

Measures	episodios					
	tiempo	paciente	VOMITED	BINGE	RESTRICTION	#null
cuenta_eventos	2016-01-01	P16				
					1	
	2015-11-01	P16				
				1		
	2015-12-01	P16				
	2016-01-02	P16				1
2015-11-02	P16					
2015-12-02	P16					
2016-01-03	P16					
2015-11-03	P16					1

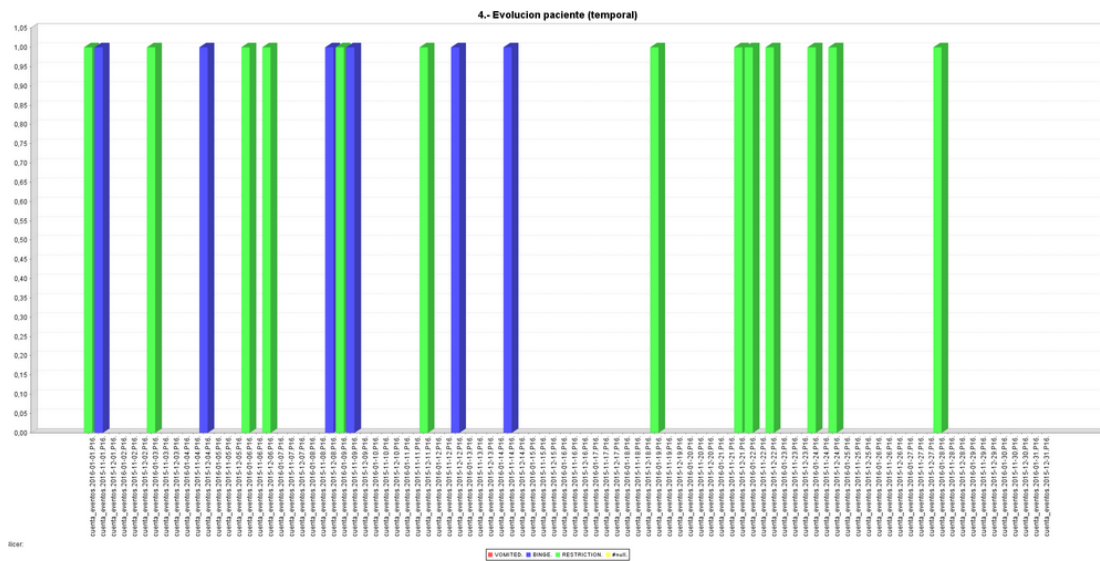


Figura.- 11 Tabla y grafico del detalle de los episodios de un paciente concreto

4.9.11. Informe 5.b.- Evolución de pacientes comparativa dos terapeutas

En este informe de dos dimensiones, terapeuta y fecha, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, con ese terapeuta, está filtrado por dos terapeutas en concreto para poder comparar la evolución temporal en el número de episodios entre ambos terapeutas.

tiempo	Measures	
	cuenta_eventos	
	todos_los_terapeutas	
	● T1	● T9
2015-11-01	1	2
2015-11-02	2	2
2015-11-03	2	2
2015-11-04	1	1
2015-11-05	1	2
2015-11-06	1	2
2015-11-07	1	2
2015-11-08	2	2
2015-11-09	2	1
2015-11-10	1	2
2015-11-11	1	1
2015-11-12	2	2
2015-11-13	2	1
2015-11-14	2	2
2015-11-15	2	2
2015-11-16	2	1
2015-11-17	2	1

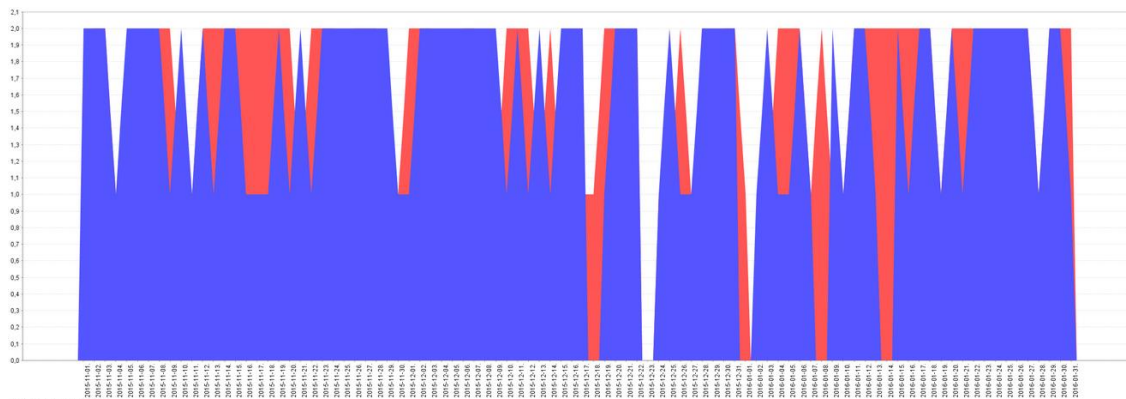


Figura.- 12 Tabla y grafico de la comparativa entre dos terapeutas

4.9.12. Informe 5.b.- Evolución de pacientes todos los terapeuta

En este informe de dos dimensiones, el total de los terapeutas y fecha, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia esa fecha, con el total de los terapeutas, se puede desglosar una fecha para ver la distribución de los episodios por terapeuta además de ver la evolución temporal del los episodios del global de los terapeutas así como también desglosado por terapeuta en fechas concretas.

Measures	
tiempo	• cuenta_eventos
2015-11-01	26
2015-11-02	25
2015-11-03	26
2015-11-04	23
2015-11-05	19
2015-11-06	20
2015-11-07	26
2015-11-08	27
2015-11-09	25
2015-11-10	25
2015-11-11	21
2015-11-12	25
2015-11-13	23
2015-11-14	25
2015-11-15	23
2015-11-16	21
2015-11-17	21
2015-11-18	20
2015-11-19	23
2015-11-20	23
2015-11-21	23
2015-11-22	23
2015-11-23	26
2015-11-24	27
2015-11-25	24
2015-11-26	23
2015-11-27	23
2015-11-28	25
2015-11-29	26
2015-11-30	22
2015-12-01	23

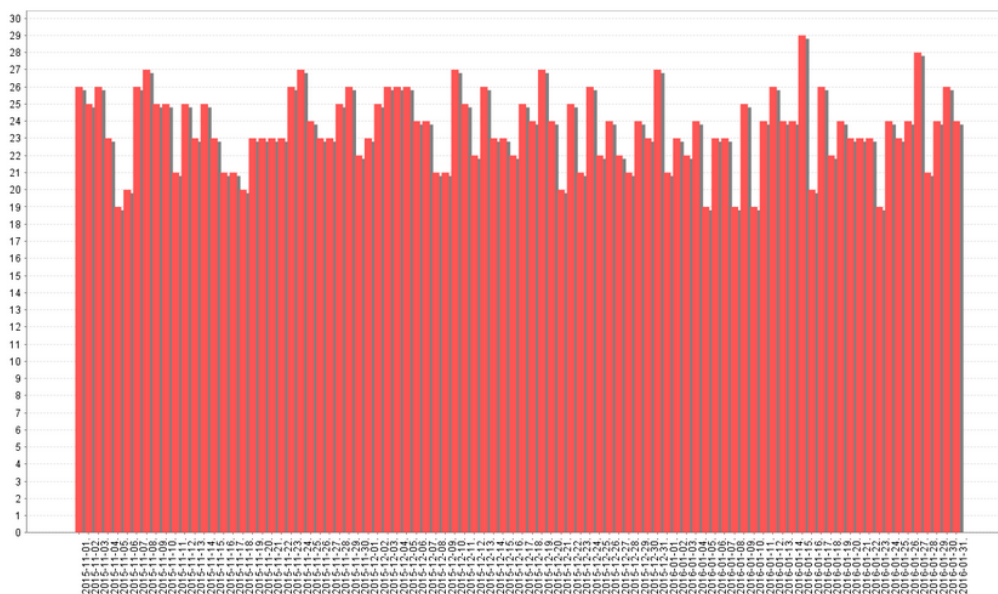
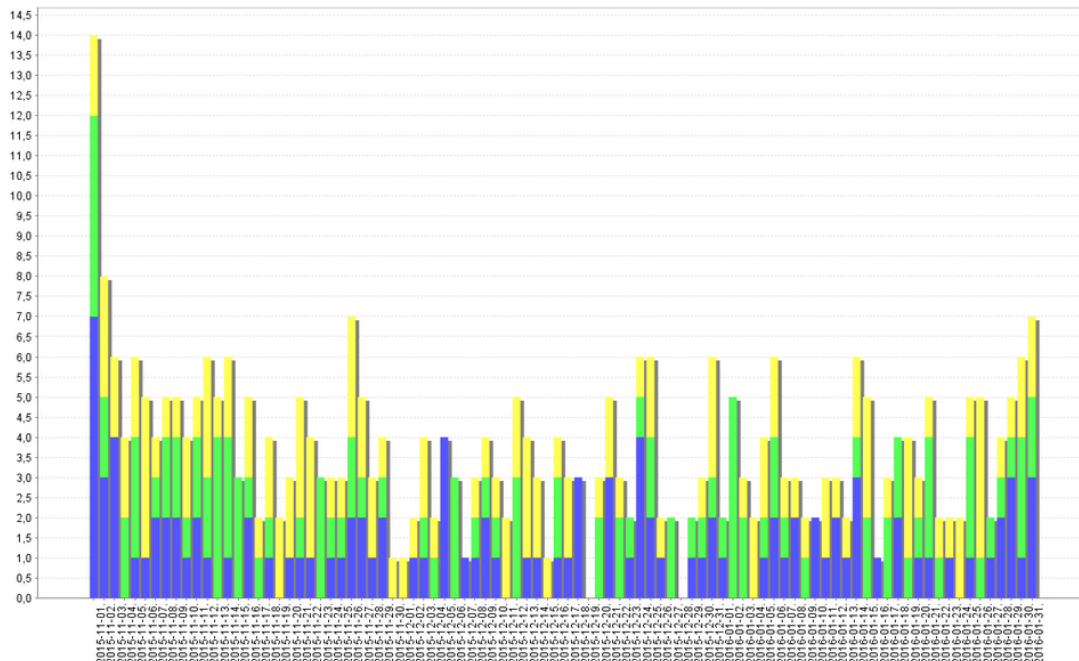


Figura.- 13 Tabla y grafico de la evolución del global de todos los terapeutas

4.9.13. Informe 5.b.- Evolución de pacientes todos los terapeuta por episodio

En este informe de dos dimensiones, episodio y fecha, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia esa fecha, con el total de los pacientes desglosado por tipo de episodio.

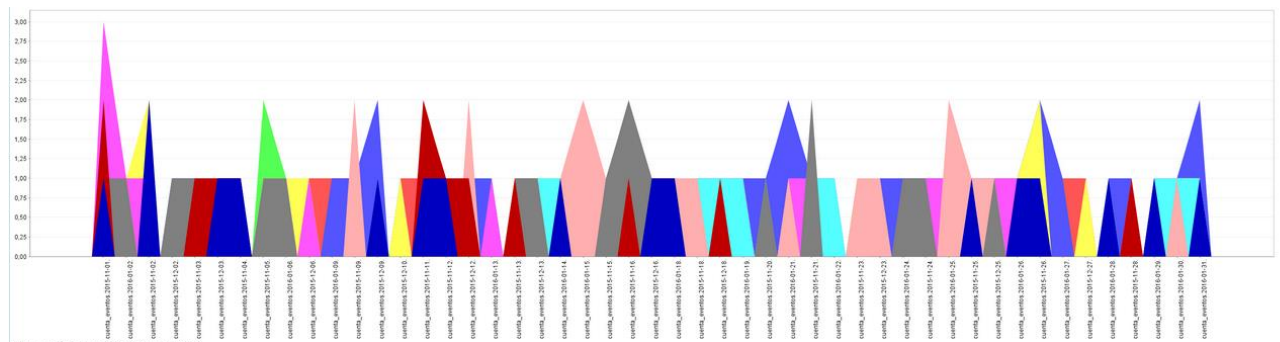
Measures				
cuenta_eventos				
episodios				
tiempo	#null	VOMITED	RESTRICTION	BINGE
2015-11-01		7	5	2
2015-11-02		3	2	3
2015-11-03		4		2
2015-11-04			2	2
2015-11-05		1	3	2
2015-11-06		1		4
2015-11-07		2	1	1
2015-11-08		2	2	1
2015-11-09		2	2	1
2015-11-10		1	1	2
2015-11-11		2	2	1
2015-11-12		1	2	3
2015-11-13			4	1
2015-11-14		1	3	2
2015-11-15			3	
2015-11-16		2	1	2
2015-11-17			1	1
2015-11-18		1	1	2
2015-11-19				2
2015-11-20		1		2
2015-11-21		1	1	3
2015-11-22		1		3
2015-11-23			3	
2015-11-24		1	1	1
2015-11-25		1	1	1
2015-11-26		2	2	3
2015-11-27		2	1	2



4.9.14. Informe 5.c. Evolución de episodios de paciente en centros médicos

En este informe de tres dimensiones, tiempo, centro terapéutico y episodio, y, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para ese centro médico. En él se puede ver la evolución temporal de los episodios por centro médico.

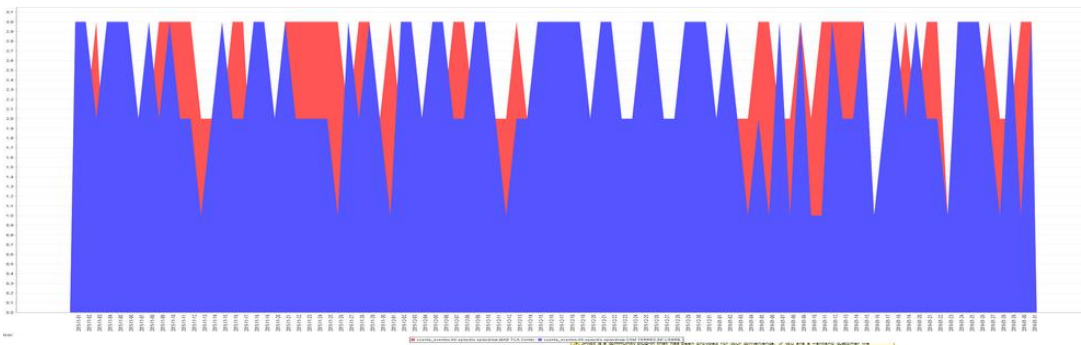
Measures	tiempo	todos_los_centros													
		BCN TCA THERAPY	MAD TCA Center	OSASUNA MUNGIA	CTA MERIDA	CSM TERRES DE L'EBRE	ARANDA TERAPEUTAS	CT SEVILLA	GRN FOOD Disorder	GAL Eating Disorder	TRG THERAPY CENIR				
cuanta_eventos	2015-11-01		2		2		3	1			1				1
	2016-01-02	1					1		1		1				
	2015-11-02	1				2		1					1		2
	2015-12-02		1								1				
	2015-11-03		1		1	1				1		1			
	2015-12-03	1							1		1				1
	2015-11-04	1							1				1		1
	2015-11-05		1		2	1			1		1				
	2016-01-06		1		1	1			1	1	1				
	2015-12-06	1				1		1							
	2016-01-09	1	1												
	2015-11-09	1	1			1				2					
	2015-12-09			2					1						1
	2015-12-10	1			1	1									
	2015-11-11	1									1		2		1
	2015-11-12	1			1				1		1		1		1
	2015-12-12		1			1				2		1			
	2016-01-13		1						1						
	2015-11-13	1				1				1		1			
	2015-12-13	1			1					1		1			
	2016-01-14		1		1	1				1		1			1
	2016-01-15		1			1			1		2				
	2015-11-15	1								1		1			
	2015-11-16		1				1				2		1		
	2015-12-16	1							1			1			1
	2016-01-18	1						1		1					1
	2015-11-18		1			1			1	1					



4.9.15. Informe 5.e.- Evolución de pacientes comparativa dos centros

En este informe de tres dimensiones, tiempo, centro terapéutico y episodio, y, como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para ese centro médico. En él se puede ver la evolución temporal de los episodios por centro médico. Similar al anterior pero están filtrados dos centros médicos, para que sea más fácil hacer una comparativa entre ellos.

		Measures	
		cuenta_eventos	
		episodios	
		+ All episodio.episodioss	
		todos_los_centros	
tiempo		● MAD TCA Center	● CSM TERRES DE L'EBRE
2015-11-01		3	3
2015-11-02		2	3
2015-11-03		3	2
2015-11-04		1	3
2015-11-05		2	3
2015-11-06		1	3
2015-11-07		2	2
2015-11-08		2	3
2015-11-09		3	2
2015-11-10		3	3
2015-11-11		3	2
2015-11-12		3	2
2015-11-13		2	1
2015-11-14		2	2
2015-11-15		2	3

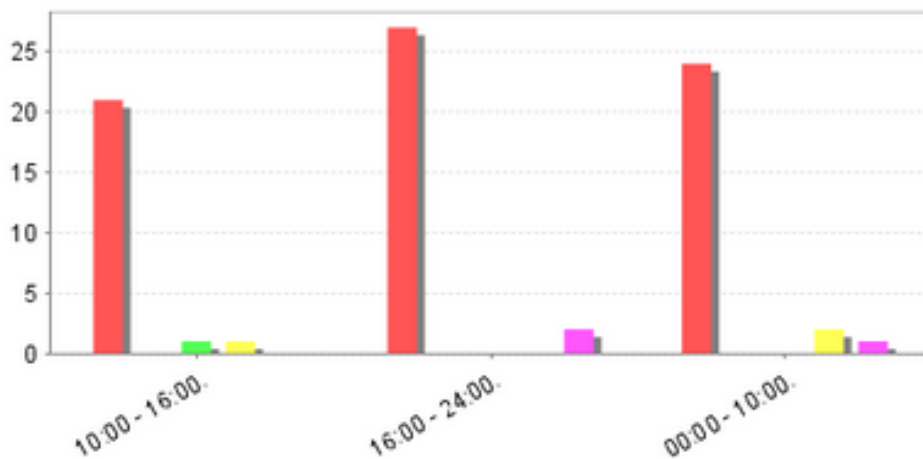


4.9.16. Informe 6.b.- Relación episodio y tiempo – franja

En este informe de dos dimensiones tiempo y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para una franja concreta de cada episodio. En él se puede ver las franjas con más episodios y comprobar si éstas tienen alguna relación con los diferentes episodios.

		Measures				
		cuenta_eventos				
		episodio				
franja	<input checked="" type="checkbox"/> All tipo_episodio.episodios	<input checked="" type="checkbox"/> #null	<input checked="" type="checkbox"/> VOMITED	<input checked="" type="checkbox"/> RESTRICTION	<input checked="" type="checkbox"/> BINGE	
10:00 - 16:00	21		1	1		
16:00 - 24:00	27				2	
00:00 - 10:00	24			2	1	

Slicer:



Slicer:

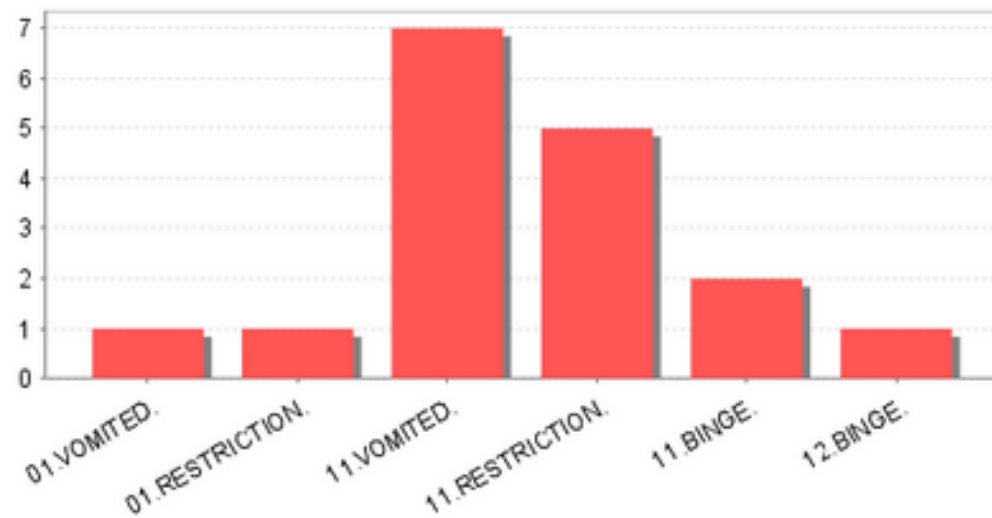
■ cuenta_eventos.All tipo_episodio.episodios	■ cuenta_eventos.#null	■ cuenta_eventos.VOMITED
■ cuenta_eventos.RESTRICTION	■ cuenta_eventos.BINGE	

4.9.17. Informe 6.c.- Relación episodio y tiempo – mes

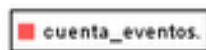
En este informe de dos dimensiones tiempo y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para un mes concreto de cada episodio. En él se puede ver los meses con más episodios y comprobar si éstos tienen alguna relación con los diferentes episodios.

		Measures
mes	episodio	• cuenta_eventos
01	VOMITED	1
	RESTRICTION	1
11	VOMITED	7
	RESTRICTION	5
	BINGE	2
12	BINGE	1

Slicer:



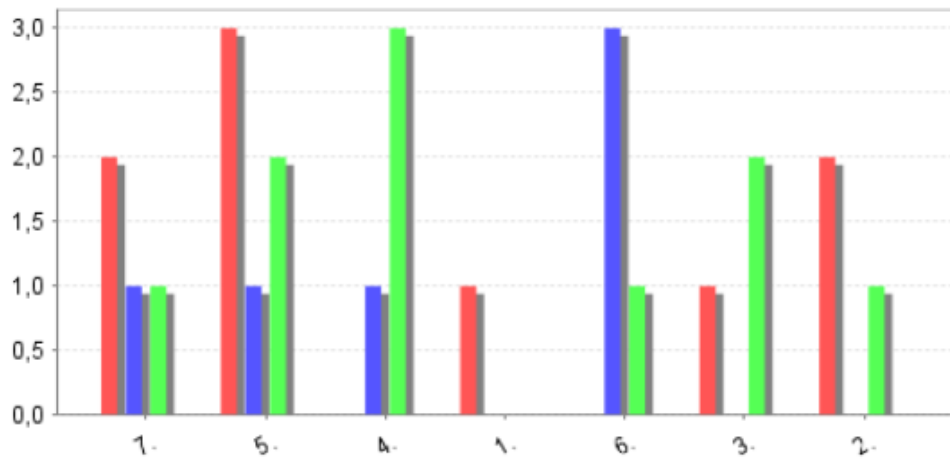
Slicer:



4.9.18. Informe 6.d.- Relación episodio y tiempo - dia_semana

En este informe de dos dimensiones tiempo y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para un día de la semana concreto de cada episodio. En él se puede ver los días de la semana con más episodios y comprobar si éstos tienen alguna relación con los diferentes episodios.

		Measures			
		cuenta_eventos			
		episodio			
dia_semana	episodio	VOMITED	RESTRICTION	BINGE	#null
7	1		5		
	2		1	1	
	3	2		2	
5	1	1	1	2	
	2	1	1	2	
	3	2	1		
	4	1	3	2	
	5	2	1	3	
	6	3	1	2	

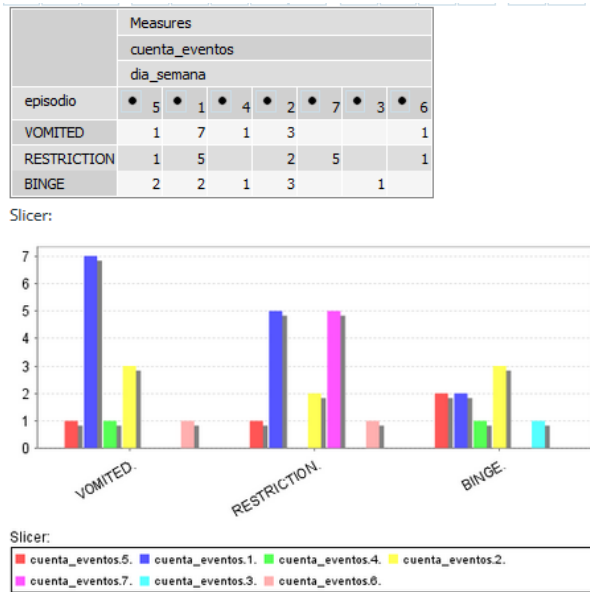


Slicer:

■ cuenta_eventos.VOMITED.	■ cuenta_eventos.RESTRICTION.	■ cuenta_eventos.BINGE.
■ cuenta_eventos.#null.		

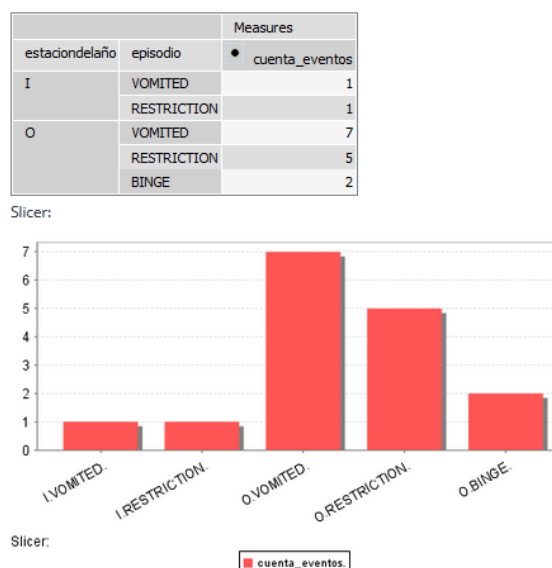
4.9.19. Informe 6.d.- Relación episodio y tiempo - dia_semana – total

En este informe de dos dimensiones tiempo y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para un día de la semana concreto de cada episodio. En él se puede ver los días de la semana con más episodios y comprobar si éstos tienen alguna relación con los diferentes episodios. Similar al anterior pero con las columnas y las filas transpuestas.



Informe 6.- Relación episodio y tiempo - estación del año

En este informe de dos dimensiones tiempo y episodio, y como medida la cantidad de ocurrencias en el que un timestamp dado hay una ocurrencia de esa fecha, para una estación concreta del año de cada episodio. En él se puede ver las estaciones del año con más episodios y comprobar si éstas tienen alguna relación con los diferentes episodios.

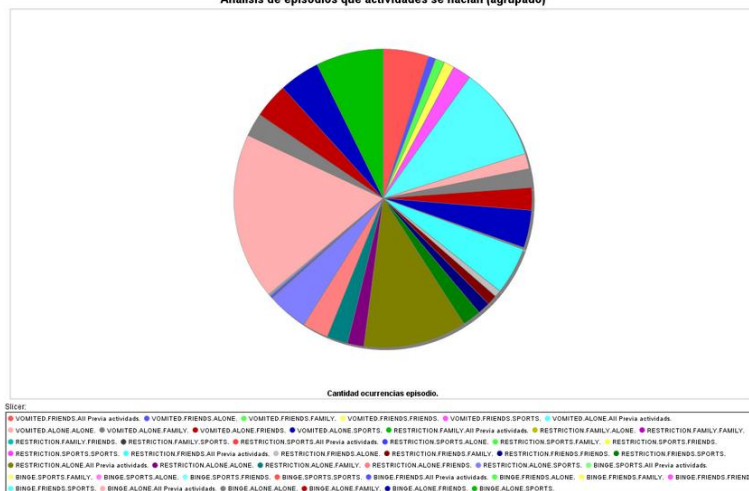


4.9.20. Informe 7.- Análisis de episodios que actividades se hacían (agrupados)

En este informe de tres dimensiones, que son episodio actual y la actividad que actualmente se está realizando, y la actividad de la lectura anterior, y como medida la cantidad de ocurrencias en el que un timestamp dado, hay una ocurrencia con esas características. En él se puede ver si durante una actividad practicada varias veces desemboca mayoritariamente en algún episodio y ver si esa actividad mejora o empeora al paciente, el cuadro contiene totales.

Episodio actual	Actividad actual	Previa actividad	Measures		
			● Cantidad ocurrencias episodio		
VOMITED	FRIENDS	All Previa actividades	2418		
		ALONE	396		
		FAMILY	490		
		FRIENDS	550		
		SPORTS	982		
		ALONE	4996		
	ALONE	All Previa actividades	4996		
		ALONE	821		
		FAMILY	992		
		FRIENDS	1190		
		SPORTS	1993		
		RESTRICTION	FAMILY	All Previa actividades	19
				ALONE	3
				FAMILY	3
FRIENDS	5				
SPORTS	8				
SPORTS	All Previa actividades		24		
	ALONE		3		
	FAMILY		3		
	FRIENDS		5		
	SPORTS		13		

Análisis de episodios que actividades se hacían (agrupado)

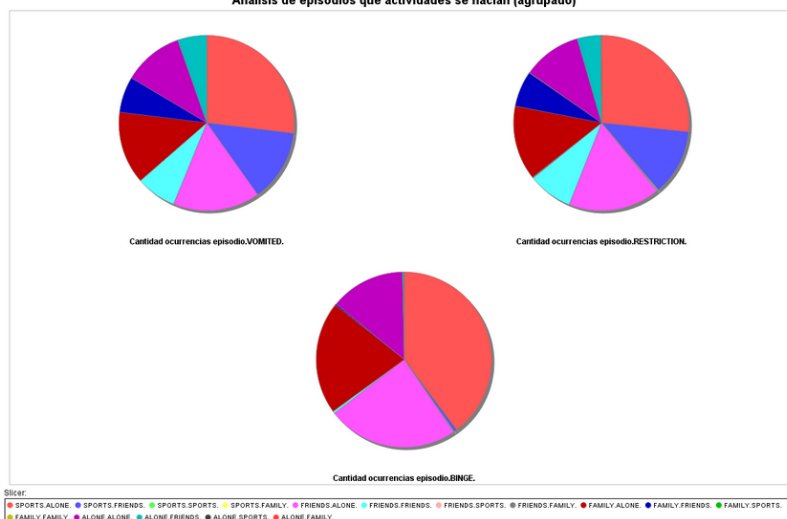


4.9.21. Informe 7.- Análisis de episodios que actividades se hacían (agrupados)

En este informe de tres dimensiones, que son episodio actual y la actividad que actualmente se está realizando, y la actividad de la lectura anterior, y como medida la cantidad de ocurrencias en el que un timestamp dado, hay una ocurrencia con esas características. En él se puede ver si durante una actividad practicada varias veces desemboca mayoritariamente en algún episodio y ver si esa actividad mejora o empeora al paciente, la tabla no contiene totales.

		Measures		
		Cantidad ocurrencias episodio		
		Episodio actual		
Previa actividad	Actividad actual	VOMITED	RESTRICTION	BINGE
SPORTS	ALONE	1993	2138	3564
	FRIENDS	982	979	50
	SPORTS		13	11
	FAMILY		8	
FRIENDS	ALONE	1190	1361	2160
	FRIENDS	550	662	33
	SPORTS		5	6
	FAMILY		5	
FAMILY	ALONE	992	1105	1857
	FRIENDS	490	527	21
	SPORTS		3	1
	FAMILY		3	
ALONE	ALONE	821	870	1237
	FRIENDS	396	349	20
	SPORTS		3	3
	FAMILY		3	

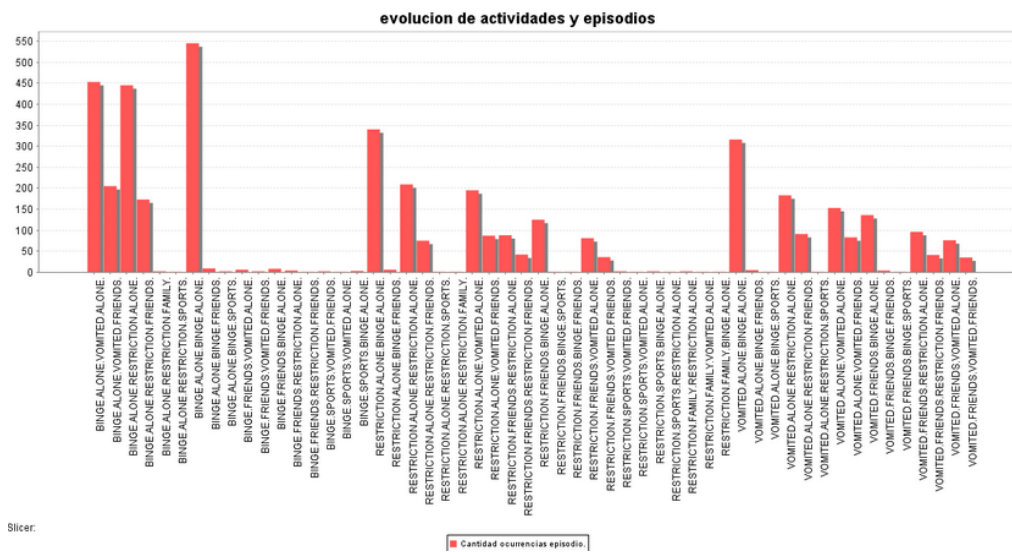
Análisis de episodios que actividades se hacían (agrupado)



4.9.22. Informe 7.- Evolución de actividad de un episodio a siguiente episodio

En este informe de cuatro dimensiones, que son episodio actual y la actividad que actualmente se está realizando, y la actividad de la lectura anterior junto con el episodio sufrido, y como medida, la cantidad de ocurrencias en el que un timestamp dado, hay una ocurrencia con esas características. En él se puede ver si durante una actividad practicada varias veces desemboca mayoritariamente en algún episodio y ver si esa actividad mejora o empeora al paciente.

Episodio actual	Actividad actual	Episodio previo	Previa actividad	Measures				
				●	Cantidad ocurrencias episodio			
BINGE	ALONE	VOMITED	ALONE	453				
			FRIENDS	205				
			RESTRICTION	445				
			FRIENDS	173				
			FAMILY	2				
			SPORTS	1				
		BINGE	ALONE	545				
			FRIENDS	9				
			SPORTS	2				
			FRIENDS	6				
			BINGE	8				
			RESTRICTION	4				
	FRIENDS	VOMITED	ALONE	2				
			FRIENDS	2				
			BINGE	8				
			RESTRICTION	4				
			FRIENDS	1				
			SPORTS	2				
		SPORTS	VOMITED	FRIENDS	2			
				ALONE	1			
				ALONE	3			
			BINGE	ALONE	340			
				FRIENDS	6			
				RESTRICTION	209			
RESTRICTION	ALONE	BINGE	ALONE	75				
			SPORTS	1				
			FAMILY	1				
			VOMITED	195				
			FRIENDS	87				
			RESTRICTION	88				
		FRIENDS	RESTRICTION	FRIENDS	42			
				ALONE	125			
				SPORTS	1			
			VOMITED	FRIENDS	1			
				ALONE	81			
				FRIENDS	36			
	SPORTS	VOMITED	FRIENDS	2				
			ALONE	1				
			BINGE	2				
			RESTRICTION	1				
			FAMILY	2				
			RESTRICTION	1				
		FAMILY	RESTRICTION	ALONE	2			
				VOMITED	1			
				BINGE	1			
			VOMITED	ALONE	BINGE	ALONE	1	
						FRIENDS	1	
						RESTRICTION	1	
FRIENDS	RESTRICTION	ALONE			1			
		FRIENDS			1			
		SPORTS			1			
	SPORTS	RESTRICTION		ALONE	1			
				FRIENDS	1			
				SPORTS	1			



5. Explotación de datos

Contestando las preguntas

El propósito de este trabajo era contestar una serie de preguntas que se planteaban en el enunciado, para ello se observan los diferentes gráficos, tablas e informes generados. A continuación se dará una breve respuesta a cada una de ellas.

1.- ¿Qué relación hay entre las actividades realizadas y los episodios de crisis?

La respuesta sí que hay relación con la actividad y el episodio, vemos muy claramente que la mayoría de los episodios se suceden en periodos donde el sujeto estaba solo con mucha diferencia. A destacar los episodios de borrachera en solitario que podemos ver en el informe *1.- Relacion_actividad_episodio* en donde se ven 392 de este situación, también podemos ver que el grupo de deportes tiene muy pocos episodios de crisis así como el estar en familia, con casi ninguna ocurrencia, por otro lado los casos de vómitos y restricciones con amigos también hay que destacarlos.

2.- ¿Se puede establecer algún tipo de relación entre los valores de los diferentes estados de ánimos y los estados de crisis?

Vemos según los informes *2.- Relación episodios y estados de ánimo* y *2.- Relación episodios y estados de ánimo (inversa)* y si nos basamos en el mayor número de ocurrencias, vemos que la mayor parte de episodios de borrachera se dan cuando el individuo estaba afligido o triste, la restricción en el caso de que estuviese experimentando frustración y los vómitos en el caso del sentimiento de culpabilidad.

3.- ¿Se puede establecer algún tipo de relación entre el nivel geográfico de tipo entorno urbano o rural con los episodios de crisis?

Pues si, como se puede ver en el informe *3.- Relación episodio entorno - inversa totales*, más del 75% de los casos se producen en entorno urbano, siendo los vómitos el episodio que más ocurre en un entorno rural, en el caso de la ciudad hay más del triple, para hacer un buen análisis, habría que considerar el número de pacientes que viven en un entorno rural.

4.- ¿Cuál ha sido la evolución de los pacientes a lo largo del tiempo?

Tal y como podemos ver en los reports *4.- Evolución paciente (agrupado)* la evolución de los pacientes ha sido desigual, así como el paciente P16, P20 y P27 han tenido bastantes episodios del orden de 20, hay otros como el paciente P2 o P19 que no han tenido más de 2 o 3 episodios a lo largo de este tiempo.

Se puede apreciar que en el informe sumariado mensual *4.d- Evolución paciente desde datamart (evolución todos)* hay una mejoría del global de pacientes durante el mes de diciembre de 2015 respecto noviembre y un ligero empeoramiento durante el mes de enero del 2016.

Si nos fijamos en la evolución mensual de los 3 pacientes con más episodios P16,P20 Y P27 utilizando el report *4.d- Evolución paciente desde datamart (evolución pacientes críticos)* vemos que estos han empeorado y la cantidad de episodios se ha duplicado.

En el caso del paciente P16 podemos observar que sus episodios de restricción alimentaria han aumentado considerablemente.

5.- ¿Hay diferencia de la evolución de pacientes a nivel centro médico? ¿Y por terapeuta?

Según el informe *5.c. Evolución de episodios de paciente en centros médicos* podemos ver que en cuanto a episodios se refiere la evolución, el nivel de episodios en cada lectura ha mejorado en general, en donde en las primeras lecturas noviembre del 2015 por ejemplo en Terres de l'Ebre era de 3 episodios por lectura y la mayoría de lecturas daban 2 episodios, pero a finales de enero del 2016 las lecturas daban lecturas de un episodio y esporádicamente un máximo de dos por lectura.

Si utilizamos el report *5.e.- Evolución de pacientes comparativa dos centros* en el que comparamos por ejemplo los centros de Terres de l'Ebre y Madrid y vemos que dentro de unos parámetros de entre 1 episodio y 3 episodios por lectura, el centro de Terres de l'Ebre durante el 2016 ha bajado el número de episodios mientras que el de Madrid salvo en situaciones muy puntuales que ha bajado a un episodio prácticamente llega a 3 episodios muy a menudo.

En cuanto a terapeutas, vemos que la situación que destacan mas por su actividad el terapeuta T1, T9 y utilizando el informe *5.b.- Evolución pacientes comparativa 2 terapeutas* vemos el mismo comportamiento que durante el 2016, aunque de estos dos terapeutas no han bajado los máximos, si vemos mas periodos sin episodios a partir de la segunda quincena de diciembre del 2015 y primera quincena de enero del 2016.

6.- ¿Se puede establecer una relación entre los episodios de crisis y el momento del día, de la semana o del año?

En cuando a momento del día la franja horaria con más episodios es desde las 16 hasta las 24 horas, por otro lado durante el otoño se producen la mayor parte de los episodios desde que hay registros, con los vómitos en primera posición.

En cuanto al momento de la semana, se ve que episodios de vómitos y restricciones se producen mayoritariamente durante el domingo y el sábado.

En cuanto a mes de año se ve claramente la progresión de los episodios va de mayor a menor desde noviembre del 2015 a enero del 2016. Para poder observar estos comportamientos se recomienda consultar los siguientes informes:

- *6.d.- Relación episodio y tiempo - dia_semana – total*
- *6.d.- Relación episodio y tiempo – franja*
- *6.d.- Relación episodio y tiempo – mes*
- *6.d.- Relación episodio y tiempo – estación del año*

7.- ¿La realización de actividades físicas mejora o empeora el estado de ánimo de los pacientes?

Para ello se creó una consulta en el datamart *relacion_actividades_con_mejora_o_empeora* en la que se calculaban la actividad que se estaba realizando en la lectura anterior a la lectura del episodio actual, en ella, junto con el informe *7.- Análisis de episodios que actividades se hacían (agrupados)* se ve claramente que el estar **solo** es un factor importante en los trastornos, también se ve que mantener un deporte regularmente es una actividad que baja los episodios así como combinarlo con estar en familia.

Aquellos casos con mayor número de episodios se dan al pasar de estar solo o con amigos, y los más graves son los que previamente estaban haciendo deporte o con amigos, sin embargo y curiosamente al revés, no se produce, es decir, al pasar de soledad a hacer deportes o amigos, en el momento del deporte no se reducen los episodios.

6. Conclusiones

Al final del desarrollo de este trabajo de final de máster y después de haber superado bastantes dificultades en cuanto disponibilidad de tiempo e incompatibilidades de software con los sistemas de los que dispongo, puedo decir que me ha resultado interesante, he disfrutado haciéndolo, pero que me he quedado con el gusanillo de avanzar un poco más, y de esta forma ver alguna de las herramientas más de las que dispone pentaho Integration y explotar algunos plugins del Pentaho server para análisis, o los dashboards.

He visto que este área tecnológica en sí, es un mundo con sus propios procedimientos y diversas técnicas, unión de dos mundos como la estadística y la informática, y que no es apto para aquel que no disfrute trabajando con las bases de datos.

Me ha dejado evidencia de lo fundamental que es un buen almacén de datos en una organización, de lo estratégico que es una buena arquitectura de datos y de ETL, y, de que si se siguen las técnicas adecuadas de análisis y diseño, se obtiene un buen producto y un activo vital para una organización moderna y actual.

Veo que en el mundo de las organizaciones no se acaba de explotar de forma adecuada ni suficiente, por diversos motivos, como que los expertos de negocio no tienen conocimiento de tecnología y los tecnólogos no tienen contacto directo con lo que analizan, como en muchos aspectos sociales, los intereses de los actores también resulta una dificultad en la implantación.

También, he podido poner en práctica algunos conocimientos aprendidos a lo largo de mi experiencia laboral, he tenido la oportunidad de construir un almacén de datos desde cero ya que siempre me ha tocado bregar con almacenes de datos ya hechos.

Al ser mi primera experiencia íntegra con herramientas opensource, al principio era un poco escéptico respecto a éstas, pero, una vez finalizado me quedo convencido de que estas herramientas pueden dar un servicio adecuado.

La que más me ha gustado ha sido la herramienta de integración de pentaho, una herramienta completa y madura; el JPivot aunque es sencillo considero que le faltan asistentes para poderlo utilizar de una forma ágil, el mismo caso es el schema workbench. Pentaho CDE Dashboards he creado un dashboard pero no he logrado hacerlo funcionar.

Actualmente a nivel profesional estoy involucrado en algunos proyectos relacionados con el área de datos, y es por ello que tengo claro debo seguir mi formación en este área, y también que a partir de ahora intentaré aplicar en mis decisiones lo que he aprendido en este trabajo.

Los objetivos se han logrado. Se podían haber generado muchos más informes ya que hay atributos que no se han explotado.

7. Apéndices

Apendice1

delimiter \$\$

```
CREATE DATABASE `tfm_bi_tca` /*!40100 DEFAULT CHARACTER SET latin1 COLLATE latin1_spanish_ci */$$
```

```
CREATE TABLE `dim_centro_medico` (
  `id_dim_centro_medico` int(11) NOT NULL,
  `nombre_centro_medico` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_ciudad` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_dim_centro_medico`),
  KEY `id_dim_ciudad_idx` (`id_ciudad`),
  CONSTRAINT `fk_id_dim_ciudad` FOREIGN KEY (`id_ciudad`) REFERENCES `dim_ciudad` (`id_dim_ciudad`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_ciudad` (
  `id_dim_ciudad` int(11) NOT NULL AUTO_INCREMENT,
  `nom_ciudad` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `codipostal_ciudad` varchar(5) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_dim_tipo_entrono` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_dim_ciudad`),
  UNIQUE KEY `id_dim_ciudad_UNIQUE` (`id_dim_ciudad`),
  KEY `id_tipo_entorno_idx` (`id_dim_tipo_entrono`),
  KEY `fk_dim_tipo_entorno_idx` (`id_dim_tipo_entrono`),
  CONSTRAINT `fk_dim_tipo_entorno` FOREIGN KEY (`id_dim_tipo_entrono`) REFERENCES `dim_tipo_entorno` (`id_dim_tipo_entorno`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=368 DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_epoca` (
  `id_dim_epoca` int(11) NOT NULL,
  `nombre_epoca` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_dim_epoca`),
  UNIQUE KEY `id_dim_epoca_UNIQUE` (`id_dim_epoca`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_genero` (
  `id_dim_genero` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_genero` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_dim_genero`),
  UNIQUE KEY `id_dim_genero_UNIQUE` (`id_dim_genero`)
) ENGINE=InnoDB AUTO_INCREMENT=67 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_paciente` (
  `id_paciente` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_paciente` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_terapeuta` int(11) DEFAULT NULL,
  `id_ciudad_vive` int(11) DEFAULT NULL,
  `id_dim_genero` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_paciente`),
  UNIQUE KEY `id_paciente_UNIQUE` (`id_paciente`),
  KEY `id_terapeuta_idx` (`id_terapeuta`),
  KEY `id_ciudad_vive_idx` (`id_ciudad_vive`),
  KEY `id_dim_genero_idx` (`id_dim_genero`),
  KEY `fk_idx` (`id_dim_genero`),
  KEY `fk-id-ciudad_idx` (`id_ciudad_vive`),
  KEY `fk-id-terapeuta_idx` (`id_terapeuta`),
  CONSTRAINT `fk-id-ciudad` FOREIGN KEY (`id_ciudad_vive`) REFERENCES `dim_ciudad`
(`id_dim_ciudad`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk-id-genero` FOREIGN KEY (`id_dim_genero`) REFERENCES
`dim_genero` (`id_dim_genero`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk-id-terapeuta` FOREIGN KEY (`id_terapeuta`) REFERENCES
`dim_terapeuta` (`id_dim_terapeuta`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1291 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_terapeuta` (
  `id_dim_terapeuta` int(11) NOT NULL AUTO_INCREMENT,
  `nom_terapeuta` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_centro_medico` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
```

```

PRIMARY KEY (`id_dim_terapeuta`),
UNIQUE KEY `id_dim_terapeuta_UNIQUE` (`id_dim_terapeuta`),
KEY `id_centro_medico_idx` (`id_centro_medico`),
CONSTRAINT `fk_id_centro_medico` FOREIGN KEY (`id_centro_medico`) REFERENCES
`dim_centro_medico` (`id_dim_centro_medico`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `dim_tiempo` (
  `id_dim_tiempo` int(11) NOT NULL AUTO_INCREMENT,
  `año` int(11) DEFAULT NULL,
  `franja` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_epoca_del_año` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  `fecha` date DEFAULT NULL,
  `hora` time DEFAULT NULL,
  `estacion` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `trimestre` int(11) DEFAULT NULL,
  `mes` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,
  `semana` int(11) DEFAULT NULL,
  `diasemana` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,
  `quincena` int(11) DEFAULT NULL,
  `diadel año` int(3) DEFAULT NULL,
  `dia` int(2) DEFAULT NULL,
  `timestamp` timestamp NULL DEFAULT NULL,
  `semestre` int(1) DEFAULT NULL,
  `timestamp_original` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`id_dim_tiempo`),
  UNIQUE KEY `id_dim_tiempo_UNIQUE` (`id_dim_tiempo`),
  KEY `id_dim_tiempo` (`id_dim_tiempo`) USING BTREE,
  KEY `timestamp` (`timestamp`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=10385 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `dim_tipo_de_actividad` (
  `id_dim_tipo_de_actividad` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_actividad` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_dim_tipo_de_actividad`),
  UNIQUE KEY `id_dim_tipo_de_actividad_UNIQUE` (`id_dim_tipo_de_actividad`)

```

```
) ENGINE=InnoDB AUTO_INCREMENT=154 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_tipo_de_comida` (  
  `id_dim_tipo_de_comida` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_comida` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `meta_update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_dim_tipo_de_comida`),  
  UNIQUE KEY `id_dim_tipo_de_comida_UNIQUE` (`id_dim_tipo_de_comida`)  
) ENGINE=InnoDB AUTO_INCREMENT=327 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_tipo_de_emocion` (  
  `id_dim_tipo_de_emocion` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_emocion` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `meta_update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_dim_tipo_de_emocion`),  
  UNIQUE KEY `id_dim_tipo_de_emocion_UNIQUE` (`id_dim_tipo_de_emocion`)  
) ENGINE=InnoDB AUTO_INCREMENT=389 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_tipo_de_episodio` (  
  `id_dim_tipo_de_episodio` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_episodio` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `meta_update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_dim_tipo_de_episodio`),  
  UNIQUE KEY `id_dim_tipo_de_emocion_UNIQUE` (`id_dim_tipo_de_episodio`)  
) ENGINE=InnoDB AUTO_INCREMENT=201 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `dim_tipo_entorno` (  
  `id_dim_tipo_entorno` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_tipo_entorno` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `dim_tipo_entornocol` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `meta_update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_dim_tipo_entorno`),  
  UNIQUE KEY `id_dim_tipo_entorno_UNIQUE` (`id_dim_tipo_entorno`)  
) ENGINE=InnoDB AUTO_INCREMENT=120 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `fact_actividad` (  
  `id_fact_actividad` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_actividad` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `meta_update_time` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id_fact_actividad`),  
  UNIQUE KEY `id_fact_actividad_UNIQUE` (`id_fact_actividad`)  
) ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci$$
```



```

`id_fact_actividad` int(11) NOT NULL AUTO_INCREMENT,
`id_paciente` int(11) DEFAULT NULL,
`id_dim_tiempo` int(11) DEFAULT NULL,
`id_actividad` int(11) DEFAULT NULL,
`meta_update_time` timestamp NULL DEFAULT NULL,
PRIMARY KEY (`id_fact_actividad`),
KEY `id_paciente_idx` (`id_paciente`),
KEY `id_actividad_idx` (`id_actividad`),
KEY `fk_id_dim_tiempo_idx` (`id_dim_tiempo`),
KEY `fk_id_tipo_actividad_idx` (`id_actividad`),
KEY `fk_id_dim_paciente_idx` (`id_paciente`),
CONSTRAINT `fk_id_dim_paciente` FOREIGN KEY (`id_paciente`) REFERENCES
`dim_paciente` (`id_paciente`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_id_dim_tiempo` FOREIGN KEY (`id_dim_tiempo`) REFERENCES
`dim_tiempo` (`id_dim_tiempo`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_id_tipo_actividad` FOREIGN KEY (`id_actividad`) REFERENCES
`dim_tipo_de_actividad` (`id_dim_tipo_de_actividad`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=6463 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `fact_comida` (
  `id_fact_comida` int(11) NOT NULL AUTO_INCREMENT,
  `id_paciente` int(11) DEFAULT NULL,
  `id_dim_tiempo` int(11) DEFAULT NULL,
  `id_dim_comida` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_fact_comida`),
  KEY `id_paciente_idx` (`id_paciente`),
  KEY `id_dim_tiempo_idx` (`id_dim_tiempo`),
  KEY `id_dim_comida_idx` (`id_dim_comida`),
  KEY `fk_id_dim_tiempo_4_idx` (`id_dim_tiempo`),
  KEY `fk_id_dim_paciente_4_idx` (`id_paciente`),
  CONSTRAINT `fk_id_dim_comida_4` FOREIGN KEY (`id_dim_comida`) REFERENCES
  `dim_tipo_de_comida` (`id_dim_tipo_de_comida`) ON DELETE CASCADE ON UPDATE
  CASCADE,
  CONSTRAINT `fk_id_dim_paciente_4` FOREIGN KEY (`id_paciente`) REFERENCES
  `dim_paciente` (`id_paciente`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_id_dim_tiempo_4` FOREIGN KEY (`id_dim_tiempo`) REFERENCES
  `dim_tiempo` (`id_dim_tiempo`) ON DELETE CASCADE ON UPDATE CASCADE

```

```
) ENGINE=InnoDB AUTO_INCREMENT=8281 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `fact_emocion` (
  `id_fact_emocion` int(11) NOT NULL AUTO_INCREMENT,
  `id_paciente` int(11) DEFAULT NULL,
  `id_dim_tiempo` int(11) DEFAULT NULL,
  `id_dim_emocion` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_fact_emocion`),
  KEY `id_paciente_idx` (`id_paciente`),
  KEY `id_dim_tiempo_idx` (`id_dim_tiempo`),
  KEY `fk_id_dim_emocion_3` (`id_dim_emocion`),
  KEY `fk_id_dim_tiempo_3_idx` (`id_dim_tiempo`),
  KEY `fk_id_dim_paciente_3_idx` (`id_paciente`),
  CONSTRAINT `fk_id_dim_emocion_3` FOREIGN KEY (`id_dim_emocion`) REFERENCES
`dim_tipo_de_emocion` (`id_dim_tipo_de_emocion`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `fk_id_dim_paciente_3` FOREIGN KEY (`id_paciente`) REFERENCES
`dim_paciente` (`id_paciente`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_id_dim_tiempo_3` FOREIGN KEY (`id_dim_tiempo`) REFERENCES
`dim_tiempo` (`id_dim_tiempo`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8281 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `fact_episodio` (
  `id_fact_episodio` int(11) NOT NULL AUTO_INCREMENT,
  `id_paciente` int(11) DEFAULT NULL,
  `id_dim_tiempo` int(11) DEFAULT NULL,
  `id_dim_episodio` int(11) DEFAULT NULL,
  `meta_update_time` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id_fact_episodio`),
  KEY `id_paciente_idx` (`id_paciente`),
  KEY `id_dim_tiempo_idx` (`id_dim_tiempo`),
  KEY `id_dim_episodio_idx` (`id_dim_episodio`),
  KEY `fk_id_dim_tiempo_2_idx` (`id_dim_tiempo`),
  KEY `fk_id_dim_paciente_2_idx` (`id_paciente`),
  CONSTRAINT `fk_id_dim_episodio_2` FOREIGN KEY (`id_dim_episodio`) REFERENCES
`dim_tipo_de_episodio` (`id_dim_tipo_de_episodio`) ON DELETE CASCADE ON UPDATE
CASCADE,
```

```

CONSTRAINT `fk_id_dim_paciente_2` FOREIGN KEY (`id_paciente`) REFERENCES
`dim_paciente` (`id_paciente`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_id_dim_tiempo_2` FOREIGN KEY (`id_dim_tiempo`) REFERENCES
`dim_tiempo` (`id_dim_tiempo`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8281 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `fact_eventos_alimentarios` (
  `id_paciente` int(11) DEFAULT NULL,
  `id_dim_tiempo` int(11) DEFAULT NULL,
  `id_actividad` int(11) DEFAULT NULL,
  `id_dim_comida` int(11) DEFAULT NULL,
  `id_dim_episodio` int(11) DEFAULT NULL,
  `id_dim_emocion` int(11) DEFAULT NULL,
  `nombre_paciente` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_terapeuta` int(11) DEFAULT NULL,
  `id_ciudad_vive` int(11) DEFAULT NULL,
  `id_dim_genero` int(11) DEFAULT NULL,
  `nom_terapeuta` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,
  `id_centro_medico` int(11) DEFAULT NULL,
  `id_eventos_alimentarios` int(11) NOT NULL AUTO_INCREMENT,
  `id_dim_entorno` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_eventos_alimentarios`),
  UNIQUE KEY `id_eventos_alimentarios_UNIQUE` (`id_eventos_alimentarios`),
  KEY `id_dim_tiempo` (`id_dim_tiempo`)
) ENGINE=InnoDB AUTO_INCREMENT=6463 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `load_activity` (
  `PATIENT` longtext COLLATE latin1_spanish_ci,
  `DAY` longtext COLLATE latin1_spanish_ci,
  `HOUR` longtext COLLATE latin1_spanish_ci,
  `ACTIVITY` longtext COLLATE latin1_spanish_ci
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$

```

```

CREATE TABLE `load_centmed` (
  `CODE` longtext COLLATE latin1_spanish_ci,
  `NAME` longtext COLLATE latin1_spanish_ci,
  `CP` longtext COLLATE latin1_spanish_ci,
  `CITY` longtext COLLATE latin1_spanish_ci
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$

```

```
CREATE TABLE `load_emotion` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `EMOTION` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `load_episodes` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `EPISODE` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `load_meal` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `MEAL` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `load_patient` (  
  `NAME` longtext COLLATE latin1_spanish_ci,  
  `CITY` longtext COLLATE latin1_spanish_ci,  
  `GENDER` longtext COLLATE latin1_spanish_ci,  
  `ENVIRONMENT` longtext COLLATE latin1_spanish_ci,  
  `THERAPIST` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `load_therapist` (  
  `NAME` longtext COLLATE latin1_spanish_ci,  
  `CENTMED` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_actividad` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `ACTIVITY` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_centro_medico` (  
  `CODE` longtext COLLATE latin1_spanish_ci,  
  `NAME` longtext COLLATE latin1_spanish_ci,  
  `CP` longtext COLLATE latin1_spanish_ci,  
  `CITY` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_ciudad` (  
  `CITY` longtext COLLATE latin1_spanish_ci,  
  `CP` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0',  
  `ENVIRONMENT` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_comida` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `MEAL` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_emocion` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `EMOTION` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_episodio` (  
  `PATIENT` longtext COLLATE latin1_spanish_ci,  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `EPISODE` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_genero` (  
  `GENDER` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_paciente` (  
  `NAME` longtext COLLATE latin1_spanish_ci,  
  `CITY` longtext COLLATE latin1_spanish_ci,  
  `GENDER` longtext COLLATE latin1_spanish_ci,  
  `ENVIRONMENT` longtext COLLATE latin1_spanish_ci,  
  `THERAPIST` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_terapeuta` (  
  `NAME` longtext COLLATE latin1_spanish_ci,  
  `CENTMED` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tiempo` (  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tiempo_01` (  
  `DAY` longtext COLLATE latin1_spanish_ci,  
  `HOUR` longtext COLLATE latin1_spanish_ci  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tipo_de_actividad` (  
  `ACTIVITY` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tipo_de_comida` (  
  `MEAL` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tipo_de_emocion` (  
  `EMOTION` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tipo_de_episodio` (  
  `EPISODE` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `raw_tipo_entorno` (  
  `ENVIRONMENT` longtext COLLATE latin1_spanish_ci,  
  `meta_is_rejected` bit(1) DEFAULT b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_actividad` (  
  `id_paciente` int(11) DEFAULT NULL,  
  `id_dim_tiempo` int(11) DEFAULT NULL,  
  `id_actividad` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_centro_medico` (  
  `id_dim_centro_medico` int(11) DEFAULT NULL,  
  `nombre_centro_medico` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `id_ciudad` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_ciudad` (  
  `nom_ciudad` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `cp_ciudad` varchar(6) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `id_dim_tipo_entorno` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_comida` (  
  `id_paciente` int(11) DEFAULT NULL,  
  `id_dim_tiempo` int(11) DEFAULT NULL,  
  `id_comida` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_emocion` (  
  `id_paciente` int(11) DEFAULT NULL,  
  `id_dim_tiempo` int(11) DEFAULT NULL,  
  `id_emocion` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_episodio` (  
  `id_paciente` int(11) DEFAULT NULL,  
  `id_dim_tiempo` int(11) DEFAULT NULL,  
  `id_episodio` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_epoca` (  
  `id_dim_epoca` int(11) NOT NULL,  
  `nombre_epoca` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_genero` (  
  `nombre_genero` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_paciente` (  
  `nombre_paciente` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `id_terapeuta` int(11) DEFAULT NULL,  
  `id_ciudad_vive` int(11) DEFAULT NULL,  
  `id_dim_genero` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_terapeuta` (  
  `nom_terapeuta` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `id_centro_medico` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tiempo` (  
  `fecha` date DEFAULT NULL,  
  `hora` time DEFAULT NULL,  
  `franja` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `estacion` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `trimestre` int(11) DEFAULT NULL,
```



```
`mes` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
`semana` int(11) DEFAULT NULL,  
`diasemana` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
`quincena` int(11) DEFAULT NULL,  
`diadel año` int(3) DEFAULT NULL,  
`año` int(4) DEFAULT NULL,  
`dia` int(2) DEFAULT NULL,  
`timestamp` timestamp NULL DEFAULT NULL,  
`semestre` int(1) DEFAULT NULL,  
`timestamp_original` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tipo_de_actividad` (  
  `nombre_actividad` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tipo_de_comida` (  
  `nombre_comida` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tipo_de_emocion` (  
  `nombre_emocion` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tipo_de_episodio` (  
  `nombre_episodio` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

```
CREATE TABLE `stg_tipo_entorno` (  
  `nombre_tipo_entorno` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci$$
```

Apéndice 2: Fichero cmd convert_file.cmd

```

@echo off
cls
set ComSpec=%SystemRoot%\system32\cmd.exe
set PATH=%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;
set PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
set TEMP=%SystemRoot%\TEMP
set TMP=%SystemRoot%\TEMP
set windir=%SystemRoot%

echo '-----'

set currentFolder=%cd%
set commandToExecute=""
echo %commandToExecute%
cd "C:\Users\eszjvx\Desktop\local
files\personal\formacion\master\trabajo_final_de_master_2016\04_convert_xls_csv_script"
set sourceFile='C:\Users\eszjvx\Desktop\local files\personal\formacion\master\Trabajo final de
master 2\02_fichero_de_datos\TCADATAPFM.xlsx'
set destinFile='C:\Users\eszjvx\Desktop\local files\personal\formacion\master\Trabajo final de
master 2\03_csv_salida\'
set commandToExecute=convertXLSX2CSV.vbs %sourceFile% %destinFile%
echo "command to execute"
echo %commandToExecute%

call %commantToExecute%

```

MANUAL EXECUTION

```

convertXLSX2CSV.vbs 'C:\\Users\\eszjvx\\Desktop\\local
files\\personal\\formacion\\máster\\Trabajo final de máster
2\\02_fichero_de_datos\\TCADATAPFM.xlsx' 'C:\\Users\\eszjvx\\Desktop\\local
files\\personal\\formacion\\máster\\Trabajo final de máster 2\\03_csv_salida\\'

```

Apéndices 3: Script Visual Basic convertXLSX2CSV.vbs

```
Set objFSO = CreateObject ("Scripting.FileSystemObject")
SourceFile="C:\Users\eszjvx\Desktop\local
files\personal\formacion\master\trabajo_final_de_master_2016\02_fichero_de_datos\TCADA
TAPFM.xlsx"
DestinFile="C:\Users\eszjvx\Desktop\local
files\personal\formacion\master\trabajo_final_de_master_2016\03_csv_salida\"

if WScript.Arguments.Count < 3 Then
    WScript.Echo "Please specify the sheet, the source, the destination files. Usage: ExcelToCsv
<sheetName> <xls/xlsx source file> <csv destination file>"
    Wscript.Quit
End If

csv_format = 6

dim src_file
dim dest_file
src_file =sourceFile
dest_file=destinFile

Dim oExcel
Set oExcel = CreateObject("Excel.Application")

Dim oBook
    WScript.Echo "source file " & src_file
    WScript.Echo "destin file " & dest_file
Set oBook = oExcel.Workbooks.Open (src_file)

oBook.activate

Dim hoja
ruta = dest_file

CreateCSV src_file,ruta, oBook, "CENTMED"
CreateCSV src_file,ruta, oBook, "THERAPIST"
CreateCSV src_file,ruta, oBook, "PATIENT"
CreateCSV src_file, ruta, oBook, "EPISODES"
CreateCSV src_file, ruta, oBook, "EMOTION"
CreateCSV src_file, ruta, oBook, "MEAL"
CreateCSV src_file, ruta, oBook, "ACTIVITY"

oBook.Close False
```

```
oExcel.Application.Quit
oExcel.Quit
WScript.Quit
```

```
Sub CreateCSV (sFname, ruta, oBook, hojaName)
```

```
    Dim rCell
```

```
    Dim rRow
```

```
    Dim sOutput
```

```
    Dim lFnum
```

```
    Dim ficheroSalida
```

```
    dim filesys, filetxt
```

```
        Dim hojaActiva
```

```
        oBook.Sheets(hojaName).Select
```

```
        oBook.Worksheets(hojaName).Activate
```

```
        oBook.Sheets(hojaName).Activate
```

```
        set hojaActiva=oBook.Worksheets(hojaName)
```

```
        Dim nombreArreglado
```

```
        nombreArreglado = CleanInput(hojaActiva.Name)
```

```
        ficheroSalida= "out_" & nombreArreglado & ".csv"
```

```
        Const ForReading = 1, ForWriting = 2, ForAppending = 8
```

```
        Set filesys = CreateObject("Scripting.FileSystemObject")
```

```
        Set filetxt = filesys.OpenTextFile(ruta & ficheroSalida, ForWriting, True)
```

```
'Open a text file to write
```

```
'sFname = "C:\MyCsv.csv"
```

```
'Loop through the rows'
```

```
For Each rRow In hojaActiva.UsedRange.Rows
```

```
    'Loop through the cells in the rows'
```

```
    For Each rCell In rRow.Cells
```

```
        sOutput = sOutput & """" & rCell.Value & ""","
```

```
    Next
```

```
    'remove the last comma'
```

```
sOutput = Left(sOutput, Len(sOutput) - 1)
```

```
'write to the file and reinitialize the variables'
```

```
        filetxt.WriteLine(sOutput)
    sOutput = ""
Next

'Close the file'
    filetxt.Close

End Sub

Function CleanInput(myString)
    myString=Replace(myString,"`", "O")
    myString=Replace(myString," ", "_")
    CleanInput=myString
End Function
```

8. Puntos de mejora

En esta sección quería enumerar algunos de los puntos que hubieran resultado interesantes crear y que no son objetivo del proyecto, pero que, bien están fuera del alcance del proyecto, o bien por motivos de tiempo no ha sido posible.

- Incluir porcentajes en los informes.
- Configurar Saiku para mejorar la presentación de los informes.
- Crear un tutorial de publicación del cubo.
- Crear un tutorial de la publicación de un informe.
- Finalizar la creación de un cuadro de mandos.
- Gestionar la carga de ficheros delta en vez de cargas completas siempre.
- Creación de un glosario en la memoria.

9. Referencias

- <https://sites.google.com/site/pentahobisuite/home/ventajas>
- <http://www.cognus.biz/reduciendo-el-coste-de-business-intelligence-con-open-source/>
- <http://www.slideshare.net/oktopuslu/bi-comparison-of-open-source-and-traditional-vendor-4327259>
- <https://www.gartner.com/doc/reprints?id=1-2Y77IIA&ct=160208&st=sb>
- <http://www.dataprix.com/arquitectura-data-warehouse-areas-datos-nuestro-almacen-corporativo>
- https://es.wikipedia.org/wiki/Cubo_OLAP
- https://es.wikipedia.org/wiki/OLAP#Tipos_de_sistemas_OLAP
- https://es.wikipedia.org/wiki/Inteligencia_empresarial
- https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos
- <http://www.adictosaltrabajo.com/tutoriales/Datawarehouse-4/>
- <https://es.wikipedia.org/wiki/MySQL>
- <http://joseantoniosaiz.es/blog/cuando-adquirir-una-licencia-comercial-de-mysql/>
- <http://www.amazon.com/Dell-Latitude-E6440-DVD-Writer-Graphics/dp/B00JH11ITU>
- <http://2ndquadrant.com/es/postgresql/postgresql-vs-mysql/>
- <http://db-engines.com/en/ranking>
- <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- <https://danielpecos.com/documents/postgresql-vs-mysql/#AEN14>
- <http://www.bisente.com/documentos/mysql-postgres.html>
- Trabajos UOC: TFG: Antonio Carlos Vera Garcia Análisis de herramientas BI en el mercado actual
- <http://www.innoventolutions.com/comparison-matrix.html>
- <http://pentaho-bi-suite.blogspot.in/2014/06/pentaho-opensource-vs-jasper.html>
- <https://enfoquepractico.com/2010/01/15/la-dimension-tiempo/>
- <https://haciendocubos.com/2007/07/02/la-dimensin-tiempo/>