



Repositorio de Datos Sanitarios para Investigación

Desarrollo de Aplicaciones

Sergio Cervero Benedí

Consultor UOC: Gregorio Robles Martínez

Tutor Externo: Elisa Gracia Grijota

Enero 2015

Licencia de publicación



Esta obra está sujeta a la licencia Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Resumen

Este proyecto tiene su origen en la oportunidad que supone para la investigación el análisis de los datos de salud de los pacientes que reciben asistencia sanitaria pública en Aragón. Cada origen de datos se estudia de forma independiente, valorando la viabilidad de su integración en el sistema y, si su incorporación resulta viable, se genera una interfase de importación de datos utilizando herramientas de extracción, transformación y carga (ETLs), de forma que se obtiene un proceso automatizado que inserta nuevos datos de forma periódica.

El sistema resultante de la integración permite consultar la información de una forma unificada, permitiendo acceder a los datos de diferentes sistemas ya relacionados y extraerlos para analizarlos con fines de investigación.

En la fase de diseño se ha tenido en cuenta la necesidad de que el sistema permita la inclusión de nuevos orígenes de datos en un futuro, por lo que se ha utilizado la herencia como mecanismo que permite definir una estructura básica común a cada origen de datos y que se puede extender con las peculiaridades de cada uno de ellos. Así mismo, dado que es un sistema de gran volumen, se han utilizado tecnologías que favorecen el rendimiento del acceso a datos, como las vistas materializadas.

Índice de contenido

1. Introducción.....	5
1.1. Antecedentes y situación actual.....	5
1.2. Motivación.....	6
1.3. Estado del arte.....	7
1.4. Tecnologías empleadas.....	8
1.4.1. Red Hat Enterprise Linux.....	8
1.4.2. PostgreSQL.....	8
1.4.3. Pentaho Data Integration.....	9
1.5. Estructura de la memoria.....	10
2. Objetivos y planificación.....	11
2.1. Objetivos.....	11
2.2. Planificación.....	12
2.2.1. Diagrama de Gantt.....	14
3. Requisitos e implementación.....	15
3.1. Estudio de viabilidad.....	15
3.2. Análisis de requisitos.....	17
3.2.1. Sistemas Origen.....	17
3.2.2. Sistema Integrado.....	24
3.3. Diseño.....	25
3.3.1. Arquitectura del sistema.....	25
3.3.2. Software.....	27
3.3.3. Diseño lógico de la base de datos.....	28
3.3.4. Tablas auxiliares.....	34
3.3.5. Vistas de consulta.....	36
3.4. Implementación.....	38
3.4.1. Instrucciones ETL.....	38
3.4.2. Perfiles de Acceso.....	41
3.5. Pruebas.....	41
3.5.1. Pruebas de carga de datos.....	41
3.5.2. Pruebas de rendimiento.....	42
4. Resultados.....	43
5. Conclusiones.....	45
5.1. Objetivos alcanzados.....	45
5.2. Conclusiones personales.....	46
5.3. Trabajos futuros.....	47
6. Referencias.....	49
7. Anexos.....	51
7.1. Anexo 1: Diagrama de la Base de Datos.....	52
7.2. Anexo 2: Script de creación de la Base de Datos.....	53
7.3. Anexo 3: Licencia.....	64

Glosario

ATC: El código ATC es el código de clasificación anatómica-terapéutica-química asignado por el *Collaborating Centre for Drug Statistics Methodology* de la OMS

BDU: Base de Datos de Usuario

CEX: Sistema de consultas externas

CIAP: Clasificación Internacional de Atención Primaria (en inglés, ICPC)

CIE: Clasificación Internacional de Enfermedades (en inglés, ICD) estandarizada por la Organización Mundial de la Salud. La última versión publicada es la 10, aunque en España está muy extendido el uso de la versión 9 hasta el año 2016, en el cual se prevé realizar un cambio coordinado a la 10.

CMBD: Conjunto Mínimo Básico de Datos. Conjunto de datos recogidos en un alta hospitalaria.

OMI/AP: software de gestión clínica de las consultas de atención primaria

PCH Urgencias: sistema de consultas de urgencias

UCI: Unidad de Cuidados Intensivos

1. Introducción

La introducción detalla cuales son los antecedentes del proyecto, otros proyectos similares y la motivación para su desarrollo. Así mismo, se describen las tecnologías que se van a utilizar y se explica la estructura de la memoria.

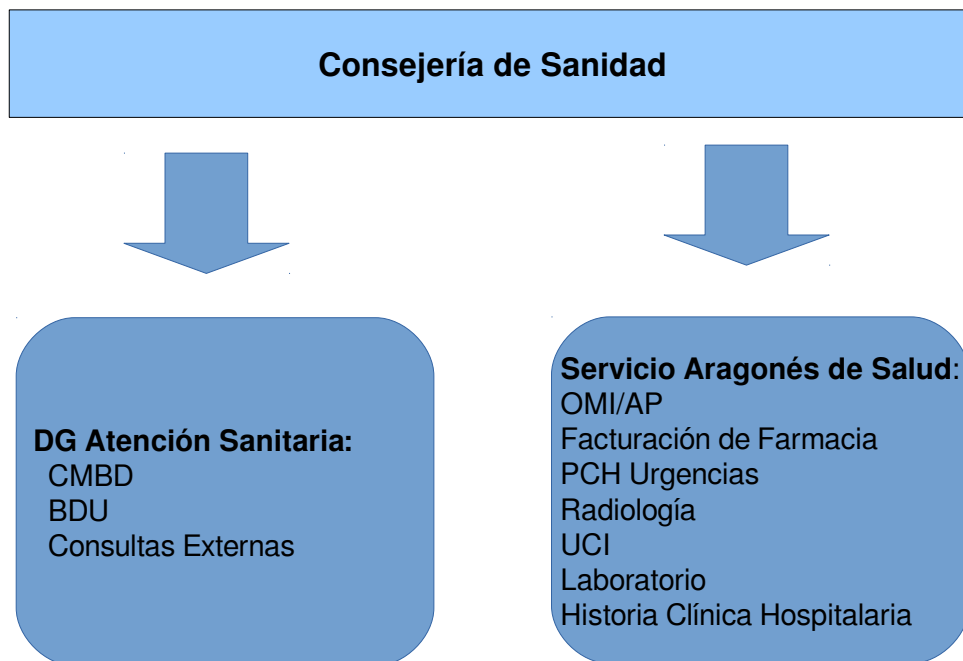
1.1. Antecedentes y situación actual

La alta capacidad de los sistemas actuales para recoger y almacenar grandes cantidades de datos ha posibilitado realizar análisis sobre ingentes cantidades de datos y obtener unos resultados que imposibles de adquirir hasta ahora. En el campo de la investigación en ciencias de la salud es crucial disponer de la mayor cantidad posible de datos sobre la que trabajar, a fin de que los estudios tengan una potencia suficiente que les permita tener validez estadística. El diseño de un protocolo de investigación y su puesta en marcha necesita recursos, los cuales son limitados, por lo que se tiene que ajustar la muestra de pacientes sobre la que recolectar la información. Sin embargo, durante la actividad asistencial diaria de un hospital o de un centro de salud se recoge una cantidad ingente de información que se podría reutilizar en el ámbito investigador. Disponer de esa información estructurada e interrelacionada permitiría llevar a cabo estudios de gran potencia que buscaran relaciones entre factores de salud que pasan desapercibidas en una muestra normal.

El interés de la comunidad investigadora en salud de Aragón ha motivado que la Consejería de Sanidad de la comunidad haya publicado una Orden de creación de un repositorio de datos sanitarios con fines de investigación, en el que se integrarán de forma interrelacionada los datos provenientes de los diferentes sistemas asistenciales. La gestión de dicho repositorio está a cargo del Instituto Aragonés de Ciencias de la Salud, un ente de derecho público dependiente de la citada Consejería de Sanidad. La Orden resalta la necesidad de cumplir con la legislación vigente en materia de protección de datos, por lo que establece que, si bien los datos de un mismo paciente provenientes de diferentes sistemas se podrán relacionar, en ningún caso se podrá conocer la identidad del paciente, por lo que los datos del repositorio deberán estar anonimizados.

Situación actual

En la actualidad, para la práctica asistencial, la información de salud está distribuida entre varios sistemas que no están conectados de forma directa entre ellos, aunque están gestionados únicamente por dos organismos diferentes, ambos dependientes de la Consejería de Sanidad: la Dirección General de Asistencia Sanitaria y el Servicio Aragonés de Salud. En la siguiente figura se ve como se agrupan los sistemas en función de la entidad que los gestiona:



Sistemas de información de la Consejería de Salud.

En todos los sistemas se almacenan datos que permiten relacionar la información entre sí, aunque también nos permitirían identificar al paciente. El principal campo de interconexión es la Tarjeta de Identificación Sanitaria (TIS), un código que identifica a cada ciudadano como usuario de los servicios de salud. Este es el código que se utiliza cuando se accede a una consulta de atención primaria, a una hospitalización o se expide una receta. Sin embargo el enlace entre el código TIS y los datos personales de la usuario (nombre, apellidos, DNI, ...) es una información accesible a todos los profesionales que trabajan en sanidad en Aragón, por lo que no es un dato que pueda incluirse en el repositorio, ya que identificaría a los pacientes.

1.2. Motivación

Mi trayectoria profesional ha estado muy relacionada con el sector de la investigación en salud, en la que he prestado apoyo a los investigadores en el desarrollo de sus proyectos de investigación. En los últimos años he comprobado como el movimiento *OpenData* abría nuevas posibilidades en el mundo de la investigación, siendo cada más los científicos del mundo sanitario interesados en el manejo de grandes cantidades de datos para hacer análisis y sacar conclusiones. Sin embargo, este interés chocaba con la falta de infraestructuras que pusieran a su disposición los datos, ya que la prioridad de los sistemas de información de salud y los técnicos que trabajan en ellos es la asistencia sanitaria, por lo que cualquier petición de datos para investigación provoca una sobrecarga de trabajo. A esto se suma que los datos de salud son tremendamente sensibles, y su uso siempre provoca recelo, entendiéndose que se utilicen para mejorar la salud de las personas

a través de la investigación, pero con cierto miedo de que se utilicen para obtener un beneficio comercial.

En este contexto, la Orden de creación del Repositorio de Datos para Investigación supuso un hito, ya que creaba una infraestructura cuyo principal objetivo es la investigación, y en la que yo vi una oportunidad de desarrollo personal y profesional. Dado que estaba realizando el máster de Software Libre propuse que todo el desarrollo se basara en aplicaciones con licencias libres, teniendo la oportunidad de demostrar que sus funcionalidades son equiparables a las del software propietario.

A lo anterior hay que añadir, que el proyecto me permitía adquirir experiencia en las herramientas ETL, un campo que siempre me ha interesado porque permiten interconectar sistemas de una forma sencilla, sin tener que programar una interfaz específica.

1.3. Estado del arte

El sector público genera en su funcionamiento diario una gran cantidad de datos. Esta situación ha generado un debate centrado en la pregunta ¿de quién son los datos públicos? Indudablemente su titularidad es de cada estado, pero es algo lógico que esos datos puedan ser reutilizados en beneficio de los ciudadanos. Las instituciones se han empezado a mover en esta dirección, como por ejemplo la Unión Europea, que ha aprobado formalmente en 2014 la estrategia sobre *Big Data*, que pretende impulsar la economía con el aprovechamiento de datos públicos. Derivada de esta iniciativa de reutilización, se ha creado el movimiento *Open Data*, que aboga por la publicación de datos en formatos y estándares abiertos para que libremente puedan ser utilizados por terceros, incluso con fines comerciales. Por otra parte, hay datos que están disponibles, pero bajo pago de una tasa, contribuyendo de esta forma al mantenimiento de la infraestructura necesaria para su publicación.

Sin embargo, este movimiento también está generando controversia, ya que colectivos que se cuestionan hasta que punto los datos públicos deberían generar beneficios en el campo privado, especialmente cuando se trata de datos tan sensibles como los de salud. La Agència de Qualitat i Avaluació Sanitàries de Catalunya (AQuAS) ha puesto en marcha el proyecto VISC+, que tiene por objetivo relacionar la información de salud que se genera en Cataluña de una manera totalmente anonimizada y segura para impulsar y facilitar la investigación y la innovación en medicina y las ciencias de la salud. Es un proyecto ambicioso y con una orientación clara a la reutilización de datos públicos, pero que está encontrando con una gran resistencia por la posibilidad de venta de los datos de salud a compañías privadas para uso en beneficio propio.

El repositorio de datos que se plantea en este proyecto no contempla la posibilidad de ceder los

datos a personas externas al entorno público. En la Orden del Consejero de Sanidad de creación se restringe el uso de datos a la investigación pública, exigiendo para su cesión que exista un protocolo de investigación que cuente con las autorizaciones legalmente exigidas a los proyectos de investigación en salud.

1.4. Tecnologías empleadas

En este apartado se describen brevemente las tecnologías empleadas en este proyecto, que son las siguientes:

- Red Hat Enterprise Linux
- PostgreSQL
- Pentaho Data Integration

1.4.1. Red Hat Enterprise Linux

Red Hat Enterprise Linux (RHEL) es una distribución Linux de carácter comercial desarrollada por Red Hat. Es la distribución que nos recomienda utilizar el proveedor de servicios TIC de la Diputación General de Aragón, AST, ya que tienen experiencia con la tecnología y un contrato de mantenimiento. Aceptando la propuesta AST se responsabiliza de monitorizar el servidor de forma remota, avisándonos de las situaciones anormales que se detecten, como por ejemplo un alto número de intentos de conexión al servidor o una carga de trabajo excesiva.

La instalación del sistema operativo es estándar, recayendo sobre nosotros la responsabilidad la instalación del software y configuración adicionales necesario para el proyecto.

1.4.2. PostgreSQL

Los inicios de la base de datos PostgreSQL se remontan al año 1982 en la Universidad de California, Berkley, aunque la versión 1.0 no vio la luz hasta 1989. La última versión estable disponible en la actualidad es la 9.4, aunque la versión 9.5 está disponible para su descarga como versión experimental. Se distribuye a través de la licencia *PostgreSQL License*, que es una licencia libre permisiva.

Cada base de datos que se crea en el servidor, contiene uno o varios esquemas que agrupan el resto de objetos que se creen. Esto posibilita que varios objetos tengan el mismo nombre, siempre y cuando se encuentren en diferentes esquemas.

Tiene una amplia variedad de tipos de datos, pudiéndose definir tipos personalizados que incluyen restricciones adicionales a los tipos base. Las instrucciones SQL son compatibles con el estándar

SQL:2011.

PostgreSQL implementa un mecanismo de herencia, por el cual las tablas pueden heredar las características de una tabla padre, siguiendo vinculadas tras su creación. Si se añade una nueva columna a la tabla padre, ésta es heredada por las tablas derivadas, que la incorporan a su definición automáticamente. Cuando se realiza una consulta sobre los datos de la tabla padre, también se muestran los datos de las tablas derivadas.

Entre sus características también se incluyen las restricciones por clave ajena, restricciones por columna y cláusulas *check* que limitan los valores posibles de un campo. Acepta objetos *blob* y permite agrupar las tablas en *tablespaces*, de forma que se pueda controlar en que unidad se almacena cada uno.

Por último, hay que destacar que desde la versión 9.3 soporta vistas materializadas. Este tipo de vistas se rellenan con datos en el momento de su creación, por lo que no tienen se ejecutan cada vez que se consultan. Por contra, para que se actualicen los datos que contienen hay que ejecutar el comando *REFRESH MATERIALIZED VIEW*.

1.4.3. Pentaho Data Integration

Pentaho Data Integration, también conocido como *Kettle*, es una herramienta de la empresa Pentaho para apoyar en los procesos de ETL. Dispone de una versión *community* que se distribuye con la licencia libre LGPL. Dispone de una interfaz gráfica (Spoon) en la que se diseñan las transformaciones y jobs, y se almacenan en XML o en un catálogo específico *Kettle* para una posterior ejecución. La ejecución se realiza con las herramientas *pan* para transformaciones o *kettle* para jobs, las cuales leen la definición del diseño previo y lo ejecutan pasándole los parámetros que se han especificado en la llamada.

Las principales características de la herramienta que se adaptan a lo necesitado en este proyecto son las siguientes:

- Fácil instalación
- Al estar desarrollado en Java es multiplataforma
- Entorno gráfico que facilita el diseño de las ETL
- Objetos que permiten la conexión a un gran número de bases de datos, así como la transformación de los datos.
- Arquitectura que permite trabajar con flujos de datos de gran volumen

1.5. Estructura de la memoria

La memoria se ha estructurado en siete apartados principales:

- **Introducción:** se justifican las razones que motivan la realización del proyecto detallando los antecedentes y la situación actual. También se realiza una breve descripción de las tecnologías empleadas.
- **Objetivos y planificación:** enumera los objetivos y el plan de trabajo temporal inicialmente contemplado.
- **Requisitos e implementación:** contiene la descripción técnica de la memoria, incluyendo el estudio de viabilidad, los análisis de requisitos, el diseño y la implementación del sistema.
- **Resultados:** se explica las características del sistema una vez finalizado y tras la carga inicial de datos, detallando la ventaja que supone frente a los sistemas anteriores.
- **Conclusiones:** se valora la consecución de los objetivos inicialmente planteados, se añade una valoración personal del proyecto y se exponen posibles líneas futuras de desarrollo.

En el apartado de anexos se va a incluir la siguiente información:

- Diagramas de la base de datos
- Script de creación de la base de datos
- Licencia del documento

2. Objetivos y planificación

El presente apartado muestra los objetivos del proyecto y la la planificación temporal propuesta inicialmente para alcanzarlos. La planificación se ha dividido en tareas, y se explica el alcance de cada una. Al final de este capítulo se ha incluido un diagrama de Gantt para tener una referencia visual del calendario de desarrollo.

2.1. Objetivos

El proyecto tiene un único objetivo principal, que se divide en varios objetivos intermedios:

Objetivo principal

- Crear un sistema repositorio de datos para investigación que integre los datos de los diferentes sistemas asistenciales

Objetivos intermedios

- Realizar un estudio para definir las herramientas de software libre a utilizar
- Analizar las fuentes de datos a integrar
- Diseñar la base de datos integrando los orígenes de datos
- Definir y programar la ejecución periódica de las ETLs
- Definir usuarios y niveles de seguridad
- Optimizar la base de datos

2.2. Planificación

La planificación inicial del proyecto ha sido la siguiente:

1. **Definición requisitos hardware para el servidor:** el servidor se albergará sobre la plataforma de servidores virtuales del Gobierno de Aragón. Sin embargo, hay que definir los requerimientos en cuanto a memoria y espacio de almacenamiento en disco de los que será necesario disponer, atendiendo a criterios de eficiencia que permitan ofrecer un buen servicio sin incurrir en un coste demasiado elevado.

Fecha prevista para la finalización de la tarea: 25 de septiembre

2. **Instalación y configuración de servidor:** de acuerdo al Plan de Sistemas del Gobierno de Aragón, el sistema operativo a instalar será Red Hat Enterprise Linux. Una vez instalado habrá que configurarlo contemplando especialmente la tarea que va a desarrollar como servidor de base de datos y crear los diferentes usuarios que podrán acceder a trabajar en la máquina.

Fecha prevista para la finalización de la tarea: 2 de octubre

3. **Estudio de herramientas software para la carga y almacenamiento de datos:** se estudiarán distintas las distintas alternativas disponibles de base de datos para albergar los datos del repositorio y para realizar la extracción, manipulación y carga de los datos provenientes de los sistemas origen.

Fecha prevista para la finalización de la tarea: 16 de octubre

4. **Instalación y configuración del servidor de base de datos:** tras la selección del motor de base de datos a utilizar se procederá a su instalación y configuración atendiendo a las características específicas del repositorio.

Fecha prevista para la finalización de la tarea: 23 de octubre.

5. **Análisis de fuentes de datos:** se realizará un análisis de las fuentes de datos con el fin de detallar que información concreta se transfiere desde cada sistema así como de evitar duplicidades en campos existentes en varios sistemas.

Fecha prevista para la finalización de la tarea: 30 de octubre

6. **Diseño lógico de la base de datos integrando todos los orígenes de datos y generación de los scripts de creación:** partiendo de las diferentes fuentes de información, se generará una única base de datos que integrará la información proveniente de todos los sistemas relacionándolos por el código identificador de cada uno de ellos.

Fecha prevista para la finalización de la tarea: 13 de noviembre.

7. **Definición y programación de la ejecución periódica de las ETLs:** para cada origen de datos se diseñará e implementará un proceso de carga utilizando la herramienta de extracción, transformación y carga seleccionada en el punto 3. Asimismo, se programará para su ejecución automática de acuerdo a la periodicidad definida para cada sistema.

Fecha prevista para la finalización de la tarea: 27 de noviembre.

8. **Definición de usuarios y niveles de seguridad:** se crearán diferentes usuarios con diferentes niveles de acceso a los datos del repositorio.

Fecha prevista para la finalización de la tarea: 11 de diciembre.

9. **Optimización de la base de datos (creación de índices, definición de segmentos...):** tras la puesta en marcha inicial del sistema se realizará una tarea de optimización de la base de datos en base a su patrón de uso.

Fecha prevista para la finalización de la tarea: 23 de diciembre.

3. Requisitos e implementación

El apartado presente incluye la descripción técnica del proyecto. Se ha dividido en los siguientes subapartados:

- Estudio de viabilidad: se comprueba que sistemas cumplen las condiciones para integrarse en el repositorio
- Análisis de requisitos: se hace un estudio de los orígenes de datos
- Diseño: se detalla la estructura datos del repositorio
- Implementación: se describe las instrucciones que importan de forma periódica los datos en el repositorio.
- Pruebas: se explican las pruebas que se han realizado para asegurar el correcto funcionamiento del sistema.

3.1. Estudio de viabilidad

En el estudio de viabilidad se va a realizar un estudio de que sistemas se van a integrar en el repositorio. Para ello se han definido los siguientes condicionantes

Hay diferentes condicionantes a tener en cuenta antes de decidir que sistemas integrar:

- Datos personales: los datos de los sistemas de origen no tienen que permitir identificar a los pacientes. Por esta razón no se traspasarán datos personales (DNI, nombre, apellidos, ...), ni datos que permitan una identificación de forma indirecta (p. ej: calle y número).
- Información estructurada: para que la información pueda ser utilizada en investigación, ésta tiene que estar estructurada. Por ello, no se importarán sistemas que tengan la información en soporte documental o con imágenes.
- Posibilidad de relacionar la información: la información a traspasar tiene que estar anonimizada, pero de forma que se puedan relacionar varios orígenes de datos. Eso implica que se tiene que poder implementar el mismo algoritmo de anonimización en el sistema de origen.
- Volumen de datos a incorporar: aunque la mayoría de sistemas tienen una gran cantidad de información, hay algunos de ellos que tienen una cantidad desmesurada de información, únicamente aprovechable para estudios muy concretos. Un ejemplo de ello

son los datos recogidos en una UCI, donde cada décima de segundo se recogen varios parámetros de un paciente.

- Interés para investigación: es imprescindible que el sistema tenga interés para investigación. Esto se verificará consultado a los grupos de investigación del Instituto.

De acuerdo a los parámetros expuestos, se han seleccionado los siguientes sistemas para integrar:

- CMBD
- OMI/AP
- BDU
- Consultas Externas
- Facturación de Farmacia
- PCH Urgencias

Por contra, se ha decidido excluir los siguientes sistemas:

- Radiología: se almacena la radiografía en alta calidad, por lo que la información no está estructurada, además de ser un sistema de un volumen de datos excesivo.
- UCI: la cantidad de datos recogidos de cada paciente un día es muy grande. Cada décima de segundo se toman varios parámetros, como el pulso, la saturación de oxígeno o la monitorización del electrocardiograma.
- Laboratorio: por cada muestra analizada se genera un pdf con los resultados, por lo que no es información estructurada.
- Historia Clínica Electrónica: se compone de varios sistemas (HP Doctor -para facultativos- y Gacela -para enfermería-), pero aún no están desplegados completamente, conservándose gran parte de la historia hospitalaria en papel.

3.2. Análisis de requisitos

Una vez decididos los sistemas que se van a integrar en el repositorio, se va a realizar un análisis pormenorizado de cada sistema origen.

3.2.1. Sistemas Origen

Para cada uno de los siguientes sistemas, se va a estudiar el propósito, los datos que lo componen, la frecuencia de actualización y el volumen de datos:

- CMBD
- OMI/AP
- BDU
- Consultas Externas
- Facturación de Farmacia
- PCH Urgencias

Todos los sistemas tienen un campo común: CIA (Código de Identificación Anonimizado) que va permitir relacionarlos entre sí.

Conjunto Mínimo Básico de Datos

El Conjunto Mínimo Básico de Datos (CMBD) es un sistema de información gestionado por la Dirección General de Atención Sanitaria que recoge todas las altas tras hospitalización que se producen en los centros sanitarios. Cada hospital tiene un CMBD propio pero con una estructura común que al final de año se envía a la Dirección General de Atención Sanitaria. Los técnicos de la Dirección General realizan la tarea de integración y consolidación de datos de los diferentes centros. El resultado es una base de datos que recoge todas las altas hospitalarias realizadas durante el año.

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
CIA	Identifica al paciente de forma anónima
hospital	Código del hospital de ingreso
des_hospital	Nombre del hospital
tipasis	Tipo de asistencia
cias_pro	<i>Sin uso actual</i>
cias_tar	<i>Sin uso actual</i>
Fecnac	Fecha de nacimiento del paciente
edad	Edad del paciente en el momento del alta
sexo	Sexo
pais	País, codificado de acuerdo a la norma ISO 3166-1
prov	Provincia de residencia
mres	Municipio de residencia
codpos	Código postal
regcon	<i>Sin uso actual</i>
fecing	Fecha de ingreso
tiping	Tipo de ingreso
fecalt	Fecha de alta
servalt	Servicio que da el alta
secalt	Sección del servicio que da el alta
medicoalt	Médico que da el alta
tipalt	Tipo de alta
des_tipalt	Descripción del tipo de alta
trashospital	Hospital al que se le traslada
ver_cie	Versión CIE utilizada para la codificación

d1..d15	Campos de diagnóstico
cm1, cm2 y cm3	<i>Sin uso actual</i>
p1..p15	Campos de procedimientos aplicados
t1..t10	Campos de terapias aplicadas
tipint	<i>Sin uso actual</i>
fecint1	<i>Sin uso actual</i>
reingreso	Indica si es un reingreso
Exitus	Indica si el paciente ha fallecido en el centro
poa_01..poa15	<i>Sin uso actual</i>
tipo_actividad	<i>Sin uso actual</i>
grdver	Versión de GRD utilizada para codificar
cod_grd	Código de GRD (Grupos Relacionados de Diagnóstico)
des_grd	Descripción del GRD
severidad	Severidad del diagnóstico
mortalidad	Mortalidad del diagnóstico
zbs	Zona de salud
zbs_des	Denominación de la zona de salud
sector	Sector sanitario
sector_des	Denominación del sector sanitario

Volumen

El volumen de datos anual, una vez que los datos están integrados y consolidados es de unos 350 MB.

Base de Datos de Usuario

El derecho al acceso a los servicios de salud se materializa con la emisión de la tarjeta sanitaria. La Base de Datos de Usuario (BDU) contiene información administrativa relativa a los usuarios del Sistema de Salud, existiendo un registro por cada tarjeta sanitaria emitida. Este registro se actualiza cada vez que hay una variación en los datos administrativos del usuario. Las actualizaciones realizadas por los operadores surten efecto de forma inmediata, aunque mensualmente se realiza un análisis de los datos para detectar errores y registros duplicados. Sin embargo, también se nutre de sistemas externos, como el Registro Nacional de Mortalidad. Los datos de fallecimiento provienen del registro anterior, pudiendo existir hasta tres meses de diferencia entre la fecha de fallecimiento y su actualización efectiva en la BDU.

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
CIA	Identifica al paciente de forma anónima
sexo	Sexo: V: Varón – M: Mujer
fechanac	Fecha de nacimiento
nacionalidad	Codificada según ISO 3166-1
paisnac	Codificado según ISO 3166-1
alta_bdu	Fecha de alta en el sistema
zona	Zona de salud asignada
centro	Centro de salud asignado
activo	Indica si el registro está activo o dado de baja
fec_baja	Fecha de baja del registro
fec_fallece	En caso de fallecimiento, fecha en la que se produjo
tp_usuario	Tipo de usuario
situacion	Situación
ine11	Población de residencia, codificada de acuerdo al INE
cp	Código postal de residencia
zona_nombre	Nombre de la zona de salud asignada
sector	Sector sanitario
sector_nombre	Nombre del sector sanitario

Volumen

El volumen de cada extracción de la base de datos se estima en 300 Mb. Como se realizan extracciones mensuales, el total anual son 3,6 Gb.

Consultas Externas

Las visitas realizadas por los médicos especialistas a pacientes sin hospitalizar, se denomina Consulta Externa. La consulta puede ser solicitada por el médico de Atención Primaria, lo que se denomina primera visita, por el propio médico especialista que cita a un paciente para revisiones posteriores o por otro médico de otra especialidad que deriva la consulta.

La información relativa a la citación incluyendo su origen se almacena en el sistema de Consultas Externas (CEX), dependiente del Servicio Aragonés de Salud. Cada consulta se representa con un nuevo registro. Con periodicidad anual, la información se consolida y se a

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
--------------	--------------------

CIA	Identifica al paciente de forma anónima
codcentro	Código del centro en el que se realiza la consulta
codespecialidad	Código de la especialidad
especialidad	Nombre de la especialidad
fecha	Fecha de la consulta
origen	Origen de la petición de la consulta (Atención Primaria u otro especialista)
cias	Código anonimizado del facultativo que pasa la consulta
primera	Indica si es la primera consulta o no

Volumen

El volumen de cada anualidad de la base de datos se estima en 1 Gb.

PCH Urgencias

Las visitas realizadas por los pacientes a urgencias no responden a una programación previa. Se registran en el sistema PCH Urgencias, gestionado de forma independiente por cada hospital y que anualmente se centraliza fusionando la información de todos los centros.

Cada visita a urgencias queda como un registro en el sistema en el que se almacena información del paciente, su estado de salud y del resultado final de su atención: alta a domicilio, hospitalización, derivación a otro centro o fallecimiento.

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
CIA	Identifica al paciente de forma anónima
Fecha atención	Fecha en la que el paciente acudió a urgencias
Tipo de consulta	Motivo que origina la visita: accidente casual, accidente de tráfico, intoxicación o enfermedad común
Código diagnóstico_1	Diagnóstico codificado según CIE-9
Diagnostico_1	Descripción del diagnóstico
Código diagnóstico_2	Diagnóstico codificado según CIE-9
Diagnostico_2	Descripción del diagnóstico
Código diagnóstico_3	Diagnóstico codificado según CIE-9
Diagnostico_3	Descripción del diagnóstico
Tipo de alta	Motivo de alta del paciente: Traslado a otro centro, domicilio, ingreso hospitalario, fuga, fallecimiento, o alta voluntaria

Volumen

El volumen anual de la base de datos se estima en 100 Mb.

OMI/AP

Las visitas realizadas a los centros de salud de Atención Primaria quedan registradas en el sistema OMI/AP. Queda registrado la fecha de visita, el diagnóstico y otros datos generales como los resultados de pruebas y datos clínicos del paciente (peso, talla, IMC..).

El sistema es gestionado por el Servicio Aragonés de Salud y con periodicidad anual se consolida.

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
CIA	Identifica al paciente de forma anónima
Ciap	Diagnóstico, codificado según CIAP
Descriptor	Descriptor del diagnóstico
Fecha_diag	Fecha en el que se ha hecho el diagnóstico
Fecha_cierre	Fecha de cierre del episodio diagnóstico
Fecha visita	Fecha de consulta
acto	Acto de consulta: Médico o Enfermera
especialidad	<i>Sin uso actual</i>

Volumen

El volumen anual de la base de datos se estima en 20 Gb.

Facturación de Farmacia

Tras la prescripción por un facultativo de un medicamento, los usuarios proceden a retirarlo en las oficinas de farmacia. Por cada medicamento retirado se genera un registro que vincula al usuario con la medicina, su porcentaje de aportación (variable en función de los ingresos) y el médico que lo ha prescrito. El sistema está gestionado por el Servicio Aragonés de Salud, que mensualmente traspasa la información a la Dirección General de Asistencia Sanitaria.

Campos a integrar

Se integrarán los siguientes campos:

Campo	Descripción
CIA	Identifica al paciente de forma anónima
Fecha_Receta	Fecha en la que se emite la receta
Grupo terapéutico	Grupo terapéutico al que pertenece el fármaco
Codificación ATC	Codificación del principio activo
Nombre comercial	Nombre del medicamento
Empresa	Empresa farmacéutica que lo comercializa
Administración	Tipo de administración: oral, rectal, vaginal, suspensión, tópica
Unidades	Número de unidades retiradas
Precio	Precio de cada unidad del medicamento
Total	Importe total
Aportación	Importe aportado por la administración
Visado	Indica si necesita autorización por parte del servicio de inspección farmacéutica
Cofinanciación	Porcentaje de cofinanciación

Volumen

El volumen anual de la base de datos se estima en 10 Gb.

3.2.2. Sistema Integrado

Los datos provenientes del origen tienen que estar integrados en un único sistema de destino. Para evitar inconsistencias se trabajará con datos consolidados, validados por la entidad que gestiona cada sistema de información origen. Aunque cada sistema es independiente, hay determinados datos que aparecen en más de un sistema. Para evitar redundancia, se incluirá el dato que provenga de la fuente más fiable. Por la frecuencia de actualización, BDU se considerará siempre como el sistema con mayor calidad de información. En caso de que haya duplicidad en otros sistemas, se realizará un estudio individual para determinar que sistema es más fiable en el dato concreto. En la siguiente tabla se han detectado los duplicados y se explica que sistema origen se ha seleccionado:

Dato	Sistemas	Observaciones
Fecha de nacimiento	OMI/AP, CMBD, BDU	Se selecciona el dato de BDU
Fecha nacimiento, edad	BDU, PCH Urgencias	Son datos recogidos en momentos diferentes: en BDU se tiene la edad administrativa registrada al dar al usuario de alta en el

		sistema, mientras que en PCH Urgencias la edad se recoge en el momento de acudir al centro
Zona de salud, Sector sanitario	CMBD, BDU, OMI/AP	Aunque el campo tiene la misma denominación, la información es diferente: - En BDU: es la zona y sector asignados por ser usuario del sistema - En CMBD y OMI/AP: es la zona y sector en la que se encuentra el centro en el que se ha recibido atención sanitaria
Provincia, municipio y código postal de residencia	CMBD, BDU	Se almacenan ambos datos: - En BDU indica el domicilio habitual del pacientes - En CMBD puede ser diferente, ya que no es inhabitual que temporalmente durante la hospitalización y posterior recuperación temporalmente se resida en el domicilio de algún familiar.
Pais	BDU, CMBD	Se selecciona el dato de BDU

3.3. Diseño

En este apartado se describe la arquitectura del sistema, así como las herramientas que se van a utilizar para desarrollar el proyecto.

3.3.1. Arquitectura del sistema

La elección de la arquitectura del sistema está condicionada a los requisitos exigidos por la entidad AST, encargada de la gestión de la informática en toda la Diputación General de Aragón. Además, siguiendo la línea de las políticas de la Diputación, se va a intentar utilizar herramientas de software libre siempre que sea posible.

Hardware

La entidad AST ofrece servidores virtuales configurados con las características requeridas. El coste varía según los requisitos contratados, por lo que es necesario solicitar un servidor con la potencia suficiente de responder a nuestras necesidades, pero evitando incurrir en costes demasiado elevados.

Alternativas contempladas

Se plantean dos alternativas en la arquitectura hardware:

- Dos máquinas: una para albergar la base de datos del repositorio y otra para la ejecución de las herramientas ETL.
- Una única máquina que albergue la base de datos del repositorio y sobre la que corran las

herramientas ETL.

La primera opción tiene un mejor rendimiento, ya que al ejecutar en distintos servidores las ETL y la base de datos, se consigue que el rendimiento de esta última sea mejor, impidiendo que se bloquee por la alta carga de trabajo de procesar las importaciones.

Por contra, la opción de tener una única máquina es más económica, y conseguiría aprovechar mejor los recursos. Dado que en esta fase la periodicidad de extracción y carga de los datos va a ser muy espaciada (los sistemas se actualizan como máximo una vez al mes), es más eficiente utilizar una única máquina para ejecutar tanto el gestor de la base de datos, como las herramientas ETL.

Justificación del hardware

Para calcular el espacio en disco necesario, se ha tenido en cuenta el volumen anual que representa cada base de datos integrada, sumando 35Gb. La carga inicial de datos va a comprender el periodo 2009-2014, 5 anualidades, por lo que el volumen inicial de disco necesario alcanza los 165 Gb, el cual incrementaremos en un 20% como margen de seguridad, totalizando 200 Gb. Añadiendo al dato anterior la proyección de necesidades de espacio en los próximos 5 años, totalizaremos 400 Gb, a los que habrá que añadir el espacio para el sistema operativo, aplicaciones y espacio temporal.

Respecto al dimensionamiento de RAM, las consultas incluyen cruces de tablas de varios millones de filas, lo cual implica que necesitaremos de toda la memoria de la que presupuestariamente podamos disponer.

La máquina virtual que se ha solicitado a AST, proveedor del servicio, tiene las siguientes características:

Descripción	Uds.	Importe / Ud.	Periodo	Importe/Año(*)
Procesadores	2	-	1 año	-
Almacenamiento	500 Gb	0,55€/Gb	1 año	275 €
Mmemoria RAM	8 Gb	419,00 €/Gb	1 año	3.352 €
Total (IVA INCLUIDO)				3.627 €

3.3.2. Software

Para el desarrollo se van a necesitar las siguientes herramientas de software:

- Sistema Operativo
- Base de datos
- Herramientas ETL

Hay que añadir que toda la documentación y el desarrollo del proyecto se va realizar en *github*, para de esta forma tener un sistema de control de versiones y que toda la documentación esté accesible desde cualquier ordenador con conexión a internet, sin tener que estar dentro de la red corporativa.

Sistema Operativo

Los requisitos sobre el software que hemos fijado anteriormente es seguir las líneas marcadas por el proveedor de servicios TIC de la Diputación General de Aragón, AST, así como intentar utilizar herramientas de software libre siempre que sea posible. Las recomendaciones de AST es utilizar un sistema Linux, siendo Red Hat Enterprise la distribución que se nos recomienda para el proyecto. Aceptando la propuesta AST se responsabiliza de monitorizar el servidor de forma remota, avisándonos de las situaciones anormales que se detecten, como por ejemplo un alto número de intentos de conexión al servidor o una carga de trabajo excesiva. En todo caso en el que se estime que el servidor pueda estar sufriendo un ataque, se nos informará para que lo verifiquemos.

Sistema Gestor de Base de Datos

Dado que vamos a prescindir de los servicios de AST para la gestión de la base de datos, no tenemos que seguir sus directrices en lo que respecta al sistema a usar. Para seleccionar el motor de base de datos relacional a instalar en el servidor, se va a realizar un estudio comparativo de los tres con mayor índice de popularidad y con licencia libre, de acuerdo a la web *db-engines.com*:

- MySQL
- PostgreSQL
- SQLite

	MySQL	PostgreSQL	SQLite
Versión para Linux	Sí	Sí	Sí
Tamaño máximo base de datos	Ilimitado	Ilimitado	128 TB
Tablas temporales	Sí	Sí	Sí
Vistas	Sí	Sí	Sí
Vistas Materializadas	No	Sí (desde versión 9.3)	No
Transmisión datos encriptados	Sí	Sí	No

Los tres SGBD muestran unas características similares, siendo todos válidos para el desarrollo del proyecto. Sin embargo, dado que el repositorio va a integrar un gran volumen de datos y que las consultas que se realicen van a cruzar una gran cantidad de registros, se considera que las vistas materializadas de *PostgreSQL* (existentes a partir de la versión 9.3) van a permitir tener un mejor rendimiento que las consultas. Tras cada importación, como parte del proceso de carga de datos se refrescarán las vistas materializadas, de forma que ya estén cargadas con datos y todas sus consultas posteriores sean más rápidas.

Herramienta ETL

Existen varias herramientas ETL en el mercado, algunas propietarias y otras de software libre. La herramienta que se ha seleccionado es *Pentaho* en su versión *Community*, ya que es ampliamente utilizada en AST para proyectos de integración, y nos pueden dar soporte en lo que respecta a su uso. Las principales características son:

- Fácil instalación
- Al estar desarrollado en Java es multiplataforma
- Entorno gráfico que facilita el diseño de las ETL
- Objetos que permiten la conexión a un gran número de bases de datos, así como la transformación de los datos.
- Arquitectura que permite trabajar con flujos de datos de gran volumen

3.3.3. Diseño lógico de la base de datos

El diseño de la base de datos se va a centrar en el paciente al que le suceden eventos de salud a lo largo de su vida. Cada evento de salud estará determinado por un identificador, la fecha en la que se produce, el tipo de evento y los datos específicos de éste. Los tipos de eventos serán los siguientes:

- Ingreso hospitalario
- Consulta especializada
- Consulta de Atención Primaria
- Atención de urgencia
- Prescripción fármaco

La tabla de pacientes recogerá los datos propios de la persona que hace uso del sistema sanitario, pero no se guardan los datos que se almacenan como eventos de salud.

Tablas principales

Aprovechando el la posibilidad de herencia que tiene el motor de base de datos seleccionado (PostgreSQL), se va a relacionar la tabla de *pacientes* con una tabla denominada *eventos* de la cual heredarán todas las tablas de eventos. Esta tabla contendrá un identificador del evento automáticamente asignado por el sistema, el código del paciente (CIA) y la fecha en la que se produce el evento.

Pacientes

En la tabla *Pacientes* hay un registro por cada usuario que ha sido dado de alta en el sistema. Además de sus datos personales como el sexo y la fecha de nacimiento, también se almacenan datos administrativos como la zona y el centro de salud que le atiende o el tipo de usuario que es (trabajador régimen general, pensionista, etc.). El campo activo indica si el usuario es válido o si ha sido dado de baja, bien sea por defunción (en este caso se rellena la fecha de fallecimiento) o por otras causas como un cambio de domicilio a otra comunidad autónoma (en este caso se rellena la fecha de baja).

Campo	Tipo	Descripción	Origen
<u>CIA</u>	Texto(10)	Código de identificación	BDU.CIA
Sexo	Texto(1)	V: Varón M: Mujer	BDU.Sexo
FechaNacimiento	Fecha		BDU.fechanac
Nacionalidad	Entero	Tabla auxiliar: <i>Países</i>	BDU.nacionalidad
PaisNacimiento	Entero	Tabla auxiliar: <i>Países</i>	BDU.paisnac
Alta	Fecha		BDU.alta_bdu
ZonaSalud	Texto	Tabla auxiliar: <i>Zonas</i>	BDU.zona
CentroSalud	Texto	Tabla auxiliar: <i>Centros</i>	BDU.centro
Activo	Entero		BDU.activo

FechaBaja	Fecha		BDU.fec_baja
FechaFallecimiento	Fecha		BDU.fec_fallece
TipoUsuario	Entero		BDU.tp_usuario
Situacion	Entero		BDU.situacion
Poblacion	Texto	Tabla auxiliar: <i>Poblaciones</i>	BDU.ine11
CP	Entero		BDU.cp
Sector	Texto	Tabla auxiliar: <i>Sectores</i>	BDU.sector

Hay algunos campos de la tabla *BDU* que dependen de otro campo de esa misma tabla, como *zona* y *zona_nombre*. La tabla *Pacientes* se ha normalizado dejando en en ella únicamente los códigos, y extrayendo el resto de información a la siguientes tablas auxiliares:

- Países
- Zonas
- Centros
- Sectores
- Poblaciones

Eventos

La tabla evento contiene la información común a cada tipo de evento registrado para cada paciente. Cada evento va a ser identificado por un identificador que se autoincrementará. Asimismo, cada eventos estará asignado a un paciente a través de su código anónimo CIA y tendrá una fecha en la que se produce.

Campo	Tipo	Descripción
<u>IDEvento</u>	Entero	Identificador del evento
CIA	Texto(10)	Identificación del paciente
FechaEvento	Fecha	Fecha del evento
FechaCarga	Timestamp	Fecha de carga del registro

IngresoHospitalario

La tabla de origen hospitalario almacena los datos que provienen del origen *CMBD*. Se omiten los campos de fecha de nacimiento, edad, sexo y país ya que es información que se carga desde el sistema de origen BDU.

Campo	Tipo	Descripción	Origen
<u>IDEvento</u>	Entero	Identificador del evento	Automático
CIA	Texto(10)		CMBD.CIA
FechaEvento	Fecha		CMBD.fecalt
Centro	Texto	Tabla auxiliar Centro	CMBD.hospital
TipoAsistencia	Texto		CMBD.tipasis
FechaIngreso	Fecha		CMBD.fecing
TipoIngreso	Texto		CMBD.tiping
ProvinciaResidencia	Entero		CMBD.prov
MunicipioResidencia	Texto		CMBD.mres
CP	Entero		CMBD.codpos
ServicioAlta	Texto		CMBD.servalt
SeccionAlta	Texto		CMBD.secalt
MedicoAlta	Texto		CMBD.medicoalt
TipoAlta	Entero	Tabla auxiliar TipoAlta	CMBD.tipalt
Traslado	Texto		CMBD.trashospital
VersionCIE	Entero		CMBD.ver_cie
Diagnostico1 ... Diagnostico15	Texto	Campos en los que recoger diagnósticos	CMBD.d1 ... CMBD.d15
Procedimiento1	Texto	Procedimientos aplicados	CMBD.p1

...			...
Procedimiento15			CMBD.p15
Terapia1	Texto	Terapias aplicadas	CMBD.t1
...			...
Terapia10			CMBD.t10
Reingreso	Caracter		CMBD.reingreso
Exitus	Caracter		CMBD.exitus
VersionGRD	Entero		CMBD.grdver
CodigoGRD	Texto		CMBD.cod_grd
Severidad	Entero		CMBD.severidad
Mortalidad	Entero		CMBD.mortalidad
Zona	Texto		CMBD.zbs
Sector	Texto		CMBD.sector

ConsultaEspecializada

La tabla *ConsultaEspecializada* deriva de Eventos y se nutrirá del origen de datos CEX. Ha sido normalizada eliminando el descriptor de la especialidad por lo que se ha dejado únicamente el código de la especialidad que se relaciona con la tabla *Especialidad*.

Campo	Tipo	Descripción	Origen
<u>IDEvento</u>	Entero	Identificador del evento	Automático
CIA	Texto(10)		CEX.CIA
FechaEvento	Fecha		CEX.fecha
Centro	Texto	Tabla auxiliar Centro	CEX.codcentro
Especialidad	Texto	Tabla auxiliar Especialidad	CEX.codespecialidad
Origen	Texto		CEX.origen
CIAS	Texto		CEX.cias
PrimeraConsulta	Entero		CEX.primera

ConsultaAP

La tabla *ConsultaAP* se normaliza creando la tabla *DiagnosticoCIAP*, en la que se almacena la correspondencia entre el código de diagnóstico CIAP y su descriptor.

Campo	Tipo	Descripción	Origen
<u>IDEvento</u>	Entero	Identificador del evento	Automático
CIA	Texto(10)		OMIAP.CIA
FechaEvento	Fecha		OMIAP.fecha_visita
DiganosticoCIAP	Texto	Tabla auxiliar: DiagnosticoCIAP	OMIAP.ciap
FechaDiagnostico	Fecha		OMIAP.fecha_diag
CierreDiagnostico	Fecha		OMIAP.fecha_cierre
Acto	Texto	Enfermería/Médico	OMIAP.acto

Urgencias

La tabla *Urgencias* se ha normalizado y se guarda únicamente el código del diagnóstico. Se utiliza una codificación diferente a la empleada en *ConsultaAP*, por lo que es necesario otra tabla auxiliar (*CIE*). La tabla *CIE* utiliza la versión 9 del estándar International Disease Classification. Dado que a principios del año 2016 se prevé cambiar a la versión 10 del estándar, se ha creado un campo que almacena la versión de CIE empleada. Este campo se rellenará automáticamente en la importación con un 9 (la versión actual) y se cambiará a 10 cuando la importación se realice con la nueva codificación. En la tabla auxiliar *CIE* se almacenarán ambas versiones.

Campo	Tipo	Descripción	Origen
<u>IDEvento</u>	Entero	Identificador del evento	Automático
CIA	Texto(10)		PCHUrgencias.CIA
FechaEvento	Fecha		PCHUrgencias.fechaatencion
TipoConsulta	Texto		PCHUrgencias.tipoconsulta
VersionCIE	Entero		Automático: 9
Diagnostico1	Texto	Tabla Auxiliar: <i>CIE</i>	PCHUrgencias.codigodiagnostico 1
Diagnostico2	Texto	Tabla Auxiliar: <i>CIE</i>	PCHUrgencias.codigodiagnostico 2
Diagnostico3	Texto	Tabla Auxiliar: <i>CIE</i>	PCHUrgencias.codigodiagnostico 2
TipoAlta	Entero		PCHUrgencias.tipoalta

Farmacia

La tabla *Farmacia* almacena todas las recetas emitidas a los pacientes. Los principios activos del

medicamento recetado se codifican guardando su código de acuerdo al estándar *ATC*. El descriptor completo del principio activo se almacena en la tabla auxiliar *ATC*.

Campo	Tipo	Descripción	Origen
<u>IDEvento</u>	Entero	Identificador del evento	Automático
CIA	Texto(10)		Farmacia.CIA
FechaEvento	Fecha		Farmacia.fechareceta
GrupoTerapeutico	Texto		Farmacia.grupoterapeutico
CodigoATC	Texto(7)	Tabla Auxiliar <i>ATC</i>	Farmacia.codificacionatc
NombreComercial	Texto		Farmacia.nombrecomercial
Empresa	Texto		Farmacia.empresa
Administracion	Texto		Farmacia.administracion
Unidades	Entero		Farmacia.unidades
Precio	Numérico(10, 2)		Farmacia.precio
Aportacion	Numérico(10, 2)		Farmacia.aportacion
Visado	Texto(1)		Farmacia.visado
Cofinanciacion	Entero		Farmacia.cofinancia

3.3.4. Tablas auxiliares

Países

Campo	Tipo
<u>Pais</u>	Entero
NombrePais	Texto

Zonas

Campo	Tipo
<u>Zona</u>	Texto
NombreZona	Texto

Centros

Campo	Tipo
<u>Centro</u>	Texto
NombreCentro	Texto

Poblaciones

Campo	Tipo
<u>Poblacion</u>	Texto(11)
Provincia	Entero
NombrePoblacion	Texto

Sectores

Campo	Tipo
<u>Sector</u>	Texto
NombreSector	Texto

CIAP

Campo	Tipo
<u>DiagnosticoCIAP</u>	Texto
DescriptorCIAP	Texto

CIE

Campo	Tipo
<u>DiagnosticoCIE</u>	Texto
<u>VersionCIE</u>	Entero
DescriptorCIE	Texto

ATC

Campo	Tipo
<u>ATC</u>	Texto(7)
DescriptorATC	Texto

TipoAlta

Campo	Tipo
<u>TipoAlta</u>	Entero
DescripcionAlta	Texto

Especialidad

Campo	Tipo
<u>Especialidad</u>	Texto
NombreEspecialidad	Texto

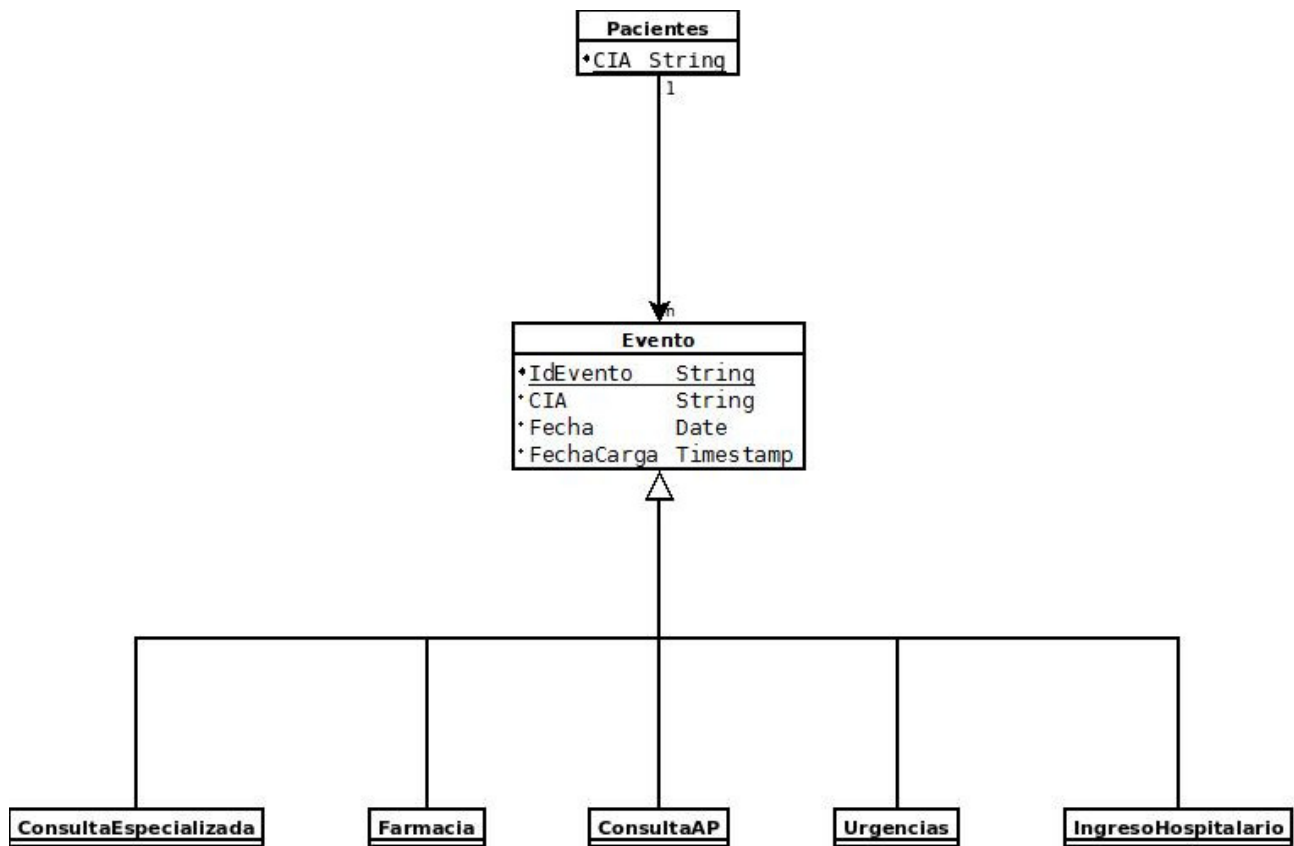
3.3.5. Vistas de consulta

Para la consulta de los datos una vez interrelacionados se van a generar una serie de vistas materializadas, a fin de que el rendimiento sea mejor:

- Vista de eventos: muestra los datos del paciente y todos los eventos que están registrados, con su fecha y tipo.
- Vista de ingresos: muestra los datos del paciente y toda la información de los eventos de tipo ingreso
- Vista de consulta especializada: muestra los datos del paciente y toda la información de los eventos de tipo consulta especializada
- Vista de consulta de Atención Primaria: muestra los datos del paciente y toda la información de los eventos de tipo consulta de Atención Primaria
- Vista de Urgencias: muestra los datos del paciente y toda la información de los eventos de tipo atención de urgencias
- Vista de Farmacia: muestra los datos del paciente y toda la información de los eventos de tipo prescripción farmacéutica.

Diagrama de Tablas

El diagrama muestra de forma visual la relación de la tabla pacientes con las tablas de eventos. En el anexo 1 se ha incluido el diagrama detallado con todas las tablas y los campos que las integran.



3.4. Implementación

La implementación del sistema ha conllevado la instalación del sistema operativo y la instalación de *ProFTP*, un servidor ftp al que se pueden conectar los sistemas de origen para transferir los ficheros de texto. Por cuestiones de seguridad, los sistemas se conectan con un usuario que tiene un acceso restringido al sistema, a través de ftp jailing. Con este sistema, únicamente se tiene acceso al directorio que se ha determinado como raíz en la configuración ftp, no pudiendo acceder ni si quiera con permisos de lectura a otros directorios que no dependan de éste.

3.4.1. Instrucciones ETL

Para la carga de datos en el sistema se va a utilizar *Pentaho Data Integration versión 5*. La utilidad tiene dos tipos de objetos principales:

- Transformaciones: realizan la conexión al origen de datos, transforman los datos y los insertan en el destino.
- Jobs: se ejecutan a más alto nivel y permiten controlar el flujo del proceso introduciendo pasos que verifican que se cumplen las condiciones necesarias para la ejecución de la transformación. Si estas se dan se ejecuta la transformación y, tras su finalización, se realizan las tareas post ejecución, como mover un archivo que hace de origen de datos a otro directorio. Adicionalmente, permite controlar los errores que se generan durante el proceso, enviando notificaciones de aviso a través de correo electrónico.

Previamente al diseño de transformaciones y jobs se ha creado un repositorio *PDI* en el que almacenar ambos objetos y otro objeto de conexión a la base de datos destino que será compartido por el resto de componentes. Esto va a permitir la ejecución programada de jobs a través de la herramienta *kitchen*, incluida en la suite de *Pentaho*.

Descripción de los jobs

Los *jobs* realizan una tarea común en todos los procesos de carga de datos:

- Comprobación de la existencia del origen de datos
- Ejecución de la transformación
- Traslado del fichero origen una vez terminado su proceso a un directorio histórico
- En el caso de que la transformación haya sido correcta, se refrescan las vistas materializadas afectadas.

Si se produce un error durante el proceso, se envía un aviso por email al operador con una breve descripción del mismo. El detalle del error se encuentra en el directorio de almacenamiento de logs.

Descripción de las transformaciones

Todas las transformaciones se componen de una serie secuenciales que se ejecutan para registro de datos encontrado en el origen. Si se produce un error al procesar un registro, se produce una entrada en el log que permite identificar el registro junto con información sobre el error producido, si es posible.

Carga de BDU

Los pasos ejecutados por la transformación son los siguientes:

- Conexión al fichero de texto plano que contiene los datos a importar
- Validación de las fechas y campos numéricos del registro
- Búsqueda del registro en la tabla de la base de datos
- Si existe se actualiza y se modifica la fecha de carga
- Si no existe se inserta el registro

Carga de CMBD

Los pasos ejecutados por la transformación son los siguientes:

- Conexión a la tabla en fichero MS Access que contiene los datos a importar
- Validación de las fechas y campos numéricos del registro que se encuentran en formato texto
- Cambio a fecha o entero de los datos que de este tipo que se encuentran en formato texto
- Se añade la constante de VersionCIE con el valor 9.
- Inserción del registro en la tabla.

Carga de OMI

Los pasos ejecutados por la transformación son los siguientes:

- Conexión al fichero de texto plano que contiene los datos a importar
- Selección de los campos a importar, renombrando aquellos con espacios para evitar

problemas en pasos posteriores

- Validación de las fechas y campos numéricos del registro
- Inserción del registro en la tabla destino

Carga de Facturación de Farmacia

Los pasos ejecutados por la transformación son los siguientes:

- Conexión al fichero de texto plano que contiene los datos a importar
- Selección de los campos a importar, renombrando aquellos con espacios para evitar problemas en pasos posteriores
- Reemplazar cadenas para que los separadores de decimales sean homogéneos
- Validación de las fechas y campos numéricos del registro
- Inserción del registro en la tabla destino

Carga de PCH Urgencias

Los pasos ejecutados por la transformación son los siguientes:

- Conexión al fichero excel que contiene los datos a importar
- Mapeo de valores para que la codificación sea homogénea con el resto de datos
- Selección de los campos a integrar, renombrando aquellos que tienen espacios para evitar posibles problemas durante la ejecución
- Validación de los datos, comprobando que los tipos de fecha y número son correctos
- Inserción en la tabla destino

Los errores

Carga de Consultas Externas

Los pasos ejecutados por la transformación son los siguientes:

- Conexión a la tabla en fichero MS Access que contiene los datos a importar
- Validación de las fechas y campos numéricos del registro que se encuentran en formato texto

- Cambio a fecha o entero de los datos que de este tipo que se encuentran en formato texto
- Uso de reemplazar cadenas para normalizar la información de origen
- Inserción del registro en la tabla

3.4.2. Perfiles de Acceso

Se han definido varios usuarios con diferentes permisos sobre los objetos de la base de datos, los cuales se pueden agrupar según las tareas que van a desarrollar de la siguiente forma:

- Administración: acceden a todos los objetos de la base de datos pudiendo modificar su estructura y sus datos.
- Importación: son usuarios que tienen permiso para acceder a las tablas y cargar datos en ellos. Los utilizan los procesos de importación de datos.
- Consulta: únicamente pueden consultar las vistas implementadas en la base de datos, no teniendo acceso directo a las tablas. Se crearán diversos usuarios de consulta, cada uno pudiendo acceder únicamente a una vista de eventos. El usuario con permiso de acceso a la Vista de Eventos tendrá acceso a todas las demás vistas.

3.5. Pruebas

Con el fin de comprobar el óptimo funcionamiento del sistema se van a diseñar dos tipos de pruebas: pruebas de carga de datos y pruebas de rendimiento. Las primeras tienen como objetivo comprobar que la importación de la información es correcta, mientras que las segundas se centran en analizar el rendimiento del sistema con el fin de buscar soluciones que permitan mejorarlo, como la creación de índices o el rediseño de las consultas.

3.5.1. Pruebas de carga de datos

La metodología de pruebas seguida para cada origen de datos ha sido la siguiente:

1. Ejecución de la transformación con un fichero reducido de datos correctos (100 registros). Se comprueba que todos los registros se cargan correctamente.
2. Ejecución de la transformación con un fichero con todos los registros correctos. Se comprueba que todos los registros se cargan correctamente y que se generan los mensajes de log apropiados.
3. Ejecución del job que llama a la transformación con un fichero con todos los registros correctos. Se comprueba que la llamada a la transformación se realiza correctamente, se

generan los logs apropiados y se envían los mensajes de correo correspondientes.

Tras las tres fases de pruebas realizadas con ficheros correctos, se realizaron comprobaciones con ficheros erróneos, para comprobar que el proceso se detenía o ignoraba los registros erróneos y que se generaban los logs adecuados:

1. Ejecución del job sin fichero de origen.
2. Ejecución del job con un fichero de tamaño 0.
3. Ejecución del job con un fichero con todas las filas erróneas.
4. Ejecución del job con un fichero con algunas filas erróneas.

3.5.2. Pruebas de rendimiento

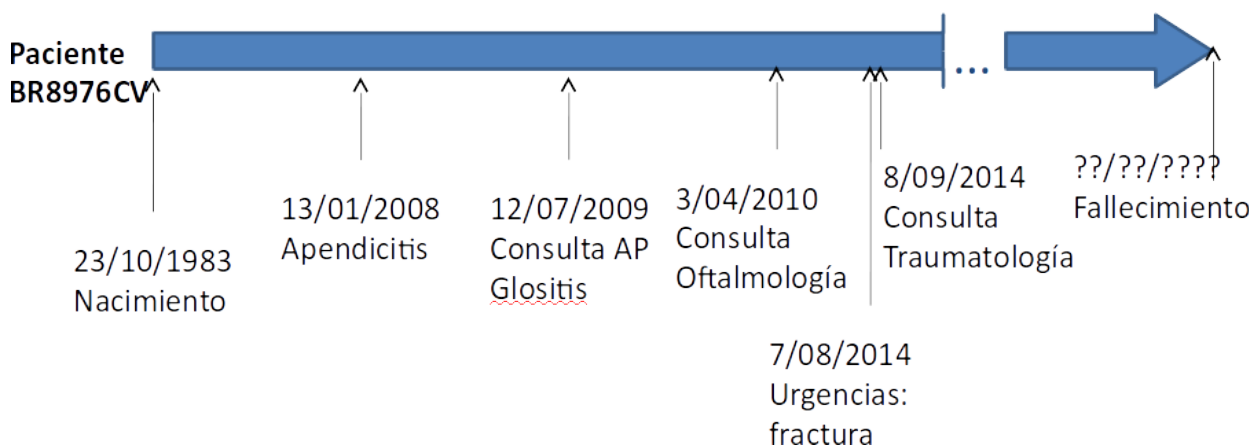
Para mejorar el rendimiento del sistema y la velocidad de acceso a los datos, se analizarán las consultas realizadas por los usuarios con el fin de identificar campos índice que permitan acelerarlas. Estas pruebas se realizarán de forma periódica.

4. Resultados

El resultado que se ha obtenido del proyecto es un servidor en producción al que conectarse a través de un cliente SQL para obtener la información interrelacionada. No hay información personal de los pacientes que pueda identificarlos, de forma que se cumple con la legislación en materia de protección de datos personales.

El periodo de información recogido varía según el tipo de sistema desde los 7 años en el caso de los ingresos hospitalarios a los datos de farmacia que se van a empezar a integrar con datos de 2016. Los datos del sistema ocupan 40Gb de espacio en disco y se componen de más de 26 millones de registros.

Respecto al avance que supone, antes del proyecto únicamente se podía consultar de forma aislada los datos de un paciente en un único sistema. En la actualidad se puede hacer un seguimiento temporal de la evolución de los pacientes y sus eventos de salud:



En el ejemplo anterior, el paciente identificado como BR8976CV, comprobamos que fue intervenido de apendicitis en el año 2008, ha tenido varias consultas a lo largo de su vida y en el año 2008 acude a urgencias por una fractura, yendo a consulta de traumatología un mes después. Estos dos últimos hechos están relacionados, la consulta de traumatología es consecuencia de la atención en urgencias, pero al ser información en sistemas diferentes no podía consultarse de forma relacionada. Los eventos de salud que tenga a lo largo de su vida se seguirán acumulando, hasta su fecha de fallecimiento.

Accediendo a las vistas de cada tipo de eventos se obtiene más información. Por ejemplo, si consultamos el evento de atención de urgencias de nuestro paciente, podemos saber si hay algún diagnóstico adicional a la fractura, como por ejemplo otra enfermedad que lo provoca.

El verdadero valor de tener un sistema así para investigación no está en el análisis individual de

cada paciente, sino en el análisis de grandes números de registros que permiten identificar relaciones causa efecto eliminando el efecto del azar.

Aunque los datos que se han importado en la carga inicial han sido de periodos anuales, se espera que la actualización de los mismos sea trimestral, salvo los datos de pacientes cuya carga se realizará mensualmente.

5. Conclusiones

5.1. Objetivos alcanzados

De los objetivos intermedios planteados inicialmente, se han alcanzado los siguientes:

- Realizar un estudio para definir las herramientas de software libre a utilizar: las herramientas a emplear han quedado claramente definidas y funcionalmente cubren todas las necesidades del proyecto. Asimismo, todo el software que se ha utilizado se distribuye a través de licencias de software libre.
- Analizar las fuentes de datos a integrar: se ha realizado un estudio de viabilidad de integración para cada sistema candidato a integrarse en el repositorio. En aquellos orígenes que han resultado viables se ha hecho un estudio detallado que analiza los requisitos que tendrá que tener el sistema a construir.
- Diseñar la base de datos integrando los orígenes de datos: se ha definido una nueva estructura de base de datos orientada a la explotación integrada de la información de los sistemas. A través del mecanismo de herencia de la base de datos se puede listar con una única consulta todos los eventos de salud de un paciente como si se tratara de una única tabla.
- Definir y programar la ejecución periódica de las ETLs: se ha diseñado una transformación y un job para la integración de cada origen de datos, de forma que se puedan ejecutar de forma automatizada. Se ha probado tanto con ficheros con formato correcto como con ficheros con registros erróneos.
- Definir usuarios y niveles de seguridad: además del administrador, se ha definido un rol para la importación y carga de los datos en el sistema, así como un perfil que únicamente permite la consulta de los datos.

Ha habido un objetivo que no se ha alcanzado completamente:

- Optimizar la base de datos: el uso de vistas materializadas aumenta el rendimiento de las consultas. Sin embargo, la realización de una amplia batería de pruebas ha consumido más tiempo del previsto inicialmente, no dejando suficiente tiempo para la creación de índices sobre las tablas que permitan una mayor velocidad de búsqueda cuando no se utilizan las vistas materializadas. No obstante, se ha desarrollado un plan basado en el análisis de las consultas que realicen los usuarios para revisarlas periódicamente e inferir

de esa información los índices que aportan una mayor utilidad al sistema.

De la consecución de los anteriores objetivos intermedios se concluye que el objetivo principal ha sido alcanzado:

- Crear un sistema repositorio de datos para investigación que integre los datos de los diferentes sistemas asistenciales

Se ha desarrollado un sistema que integra los datos de salud y permite la consulta de los mismos por parte de los usuarios.

5.2. Conclusiones personales

Las conclusiones personales tienen la intención de mostrar mi valoración subjetiva de los resultados del proyecto.

El proyecto se ha desarrollado sin contratiempos mayores y se ha conseguido cumplir casi completamente el plan de trabajo establecido al principio. La diversidad de los formatos de origen de datos representa una muestra de la necesidad de este proyecto, en el que me he adaptado todo lo posible al técnico que realizaba la extracción de cada sistema para facilitar su labor, aunque eso supusiera una mayor carga de trabajo para mí. Personalmente lo he vivido como un gran avance, ya que ahora existe una infraestructura nueva y en crecimiento que abre nuevas posibilidades a los grupos de investigación en salud de Aragón.

Respecto al diseño de la base de datos, me gustaría destacar el mecanismo de herencia que permite integrar nuevos sistemas con un coste de trabajo relativamente bajo, y facilita la integración de nuevos campos de los sistemas ya existentes. El rendimiento obtenido es el adecuado dentro de las posibilidades, teniendo en cuenta que en muchas consultas se cruzan tablas con varios millones de registros.

En cuanto al uso de la herramienta ETL, resaltar que es muy versátil con una curva de aprendizaje suave, lo cual anima a conocer cada vez en más profundidad su funcionamiento y posibilidades.

Conocimientos y competencias adquiridas

En mi desarrollo profesional, este proyecto me ha servido para adquirir experiencia en la utilización del servidor de software libre PostgreSQL. No es tan famoso como MySQL, pero desde mi punto de vista me ha parecido más potente a la vez que igual de sencillo de utilizar. También he aprendido como funcionan las herramientas ETL, comprobando que es una forma muy eficiente de interconectar sistemas y que una vez se ha trabajado con ellas es muy sencillo diseñar transformaciones para transportar datos entre bases de datos.

Relación con la formación del máster

En lo que respecta a la formación recibida en el máster, para este proyecto me ha sido muy útil las asignaturas de administración de servidores, que me han permitido saber configurar el servidor Linux. Igualmente, el conocer la existencia del gestor de base de datos PostgreSQL ha resultado muy importante para el proyecto, aunque yo creo que se le debería dedicar más tiempo en la asignatura. Por último, para la planificación y delimitación del proyecto, me ha sido muy útil la formación dada en la asignatura de implementación de proyectos de software libre.

5.3. Trabajos futuros

Este proyecto inicia una importante línea de trabajo, ya que es un campo muy amplio que hay que desarrollar. El repositorio es la base de un sistema que pone a disposición de los investigadores los datos, y sobre la cual se perfilan los siguientes trabajos:

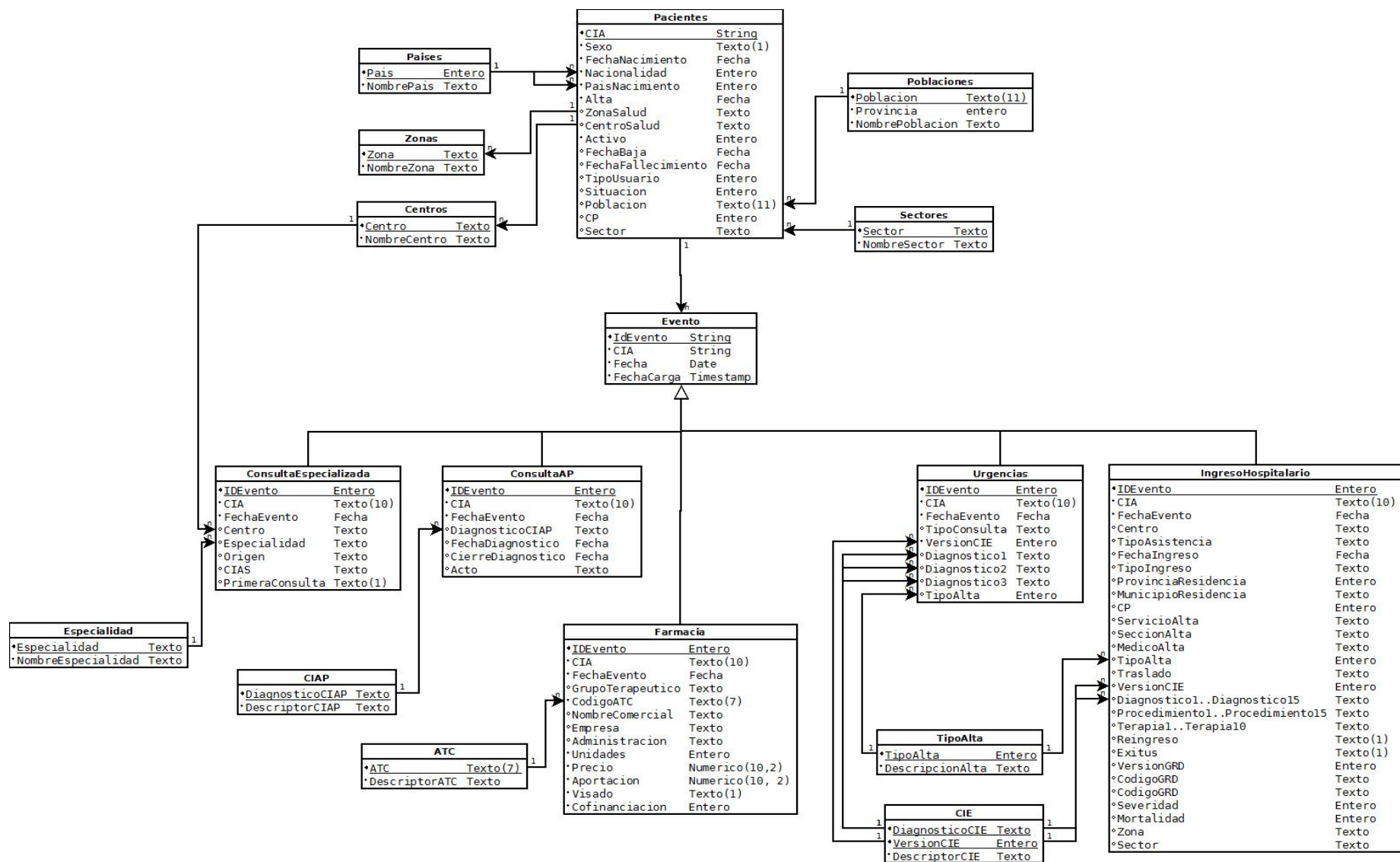
- Integración de otros sistemas de salud: estudio del desarrollo a realizar para la integración de aquellos sistemas excluidos durante el estudio de viabilidad.
- Integración de otros sistemas externos: incorporación de información proveniente de otros sistemas que no son de salud, como por ejemplo los datos meteorológicos, los cuales podrían relacionar el tiempo atmosférico con el desarrollo de algunas patologías.
- Desarrollo de una interfaz de usuario amigable: la consulta actual a la base de datos se realiza a través de SQL, por lo que mucho investigadores necesitan soporte por parte de un técnico para la extracción de los datos. Se contempla el desarrollo de un aplicativo que permita que las extracciones puedan realizarse de forma amigable.

6. Referencias

- Comparison of relational database management systems. https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems.
- DB-Engines Ranking. <http://db-engines.com/en/ranking>.
- El Proyecto Visc+ de datos sanitarios, el "petróleo del S.XXI". http://www.eldiario.es/catalunya/opinions/Proyecto-Visc-sanitarios-petroleo-SXXI_6_379072116.html.
- Free Open Source ETL Tools. <http://www.etltools.net/free-etl-tools.html>.
- Getting Started with PDI. <https://help.pentaho.com/Documentation/6.0/0J0/0C0>.
- Instituto Nacional de Estadística. www.ine.es.
- Linux Documentation – Die.net. <http://linux.die.net>.
- ORDEN de 1 de abril de 2013, del Departamento de Sanidad, Bienestar Social y Familia, por la que se crea el repositorio de datos sanitarios para la Investigación en el Instituto Aragonés de Ciencias de la Salud. BOA 88/2013. http://www.boa.aragon.es/cgi-bin/EBOA/BRSCGI?CMD=VERDOC&BASE=BZHT&PIECE=BOLE&DOCR=1&SEC=BUSQUEDA_AVANZADA&RNG=10&SORT=-PUBL&SEPARADOR=&&TITU=REPOSITORIO&SECC=C=BOA+O+DISPOSICIONES+O+PERSONAL+O+ACUERDOS+O+JUSTICIA+O+ANUNCIOS.
- PostgreSQL Wiki. <https://wiki.postgresql.org/wiki>.
- The ProFTPD project. <http://www.proftpd.org>.
- Top 15 Free Extract, Transform and Load ETL Software. <http://www.predictiveanalyticstoday.com/top-free-extract-transform-load-etl-software>.
- VISC+: más valor a la información de salud de Cataluña. <http://aguas.gencat.cat/es/projectes/visc>.

7. Anexos

7.1. Anexo 1: Diagrama de la Base de Datos



7.2. Anexo 2: Script de creación de la Base de Datos

```
-- Name: ATC; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "ATC" (
    "ATC" character varying(7) NOT NULL,
    "DescriptorATC" text NOT NULL
);
ALTER TABLE "ATC" OWNER TO repoadmin;

-- Name: CIAP; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "CIAP" (
    "DiagnosticoCIAP" text NOT NULL,
    "DescriptorCIAP" text NOT NULL
);
ALTER TABLE "CIAP" OWNER TO repoadmin;

-- Name: CIE; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "CIE" (
    "DiagnosticoCIE" text NOT NULL,
    "DescriptorCIE" text NOT NULL,
    "VersionCIE" integer DEFAULT 9 NOT NULL
);
ALTER TABLE "CIE" OWNER TO repoadmin;

-- Name: Centros; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Centros" (
    "Centro" text NOT NULL,
    "NombreCentro" text NOT NULL
);
ALTER TABLE "Centros" OWNER TO repoadmin;

-- Name: Evento; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Evento" (
    "IdEvento" integer NOT NULL,
    "CIA" text NOT NULL,
    "Fecha" date NOT NULL,
    "FechaCarga" timestamp without time zone DEFAULT now() NOT NULL
);
ALTER TABLE "Evento" OWNER TO repoadmin;

-- Name: ConsultaAP; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "ConsultaAP" (
    "DiagnosticoCIAP" text,
    "FechaDiagnostico" date,
    "CierreDiagnostico" date,
    "Acto" text
)
INHERITS ("Evento");
ALTER TABLE "ConsultaAP" OWNER TO repoadmin;

-- Name: ConsultaEspecializada; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "ConsultaEspecializada" (
    "Centro" text,
    "Especialidad" text,
    "Origen" text,
    "CIAS" text,
    "PrimeraConsulta" character(1)
)
INHERITS ("Evento");
ALTER TABLE "ConsultaEspecializada" OWNER TO repoadmin;
```

```
-- Name: Farmacia; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Farmacia" (
  "GrupoTerapeutico" text,
  "CodigoATC" character varying(7) NOT NULL,
  "NombreComercial" text,
  "Empresa" text,
  "Administracion" text,
  "Unidades" integer NOT NULL,
  "Precio" numeric(10,2) DEFAULT 0 NOT NULL,
  "Aportacion" numeric(10,2) DEFAULT 0 NOT NULL,
  "Visado" character(1) DEFAULT 'N'::bpchar NOT NULL,
  "Cofinanciacion" integer NOT NULL,
  CONSTRAINT "CHK_FARMACIA_VISADO" CHECK (("Visado" = ANY (ARRAY['S'::bpchar,
'N'::bpchar])))
)
INHERITS ("Evento");
ALTER TABLE "Farmacia" OWNER TO repoadmin;

-- Name: Pacientes; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Pacientes" (
  "CIA" text NOT NULL,
  "Sexo" character(1),
  "FechaNacimiento" date,
  "Nacionalidad" integer,
  "PaisNacimiento" integer,
  "Alta" date,
  "ZonaSalud" text,
  "CentroSalud" text,
  "Activo" integer,
  "FechaBaja" date,
  "FechaFallecimiento" date,
  "TipoUsuario" character varying(6),
  "Situacion" integer,
  "Poblacion" character varying(11),
  "CP" integer,
  "Sector" text
);
ALTER TABLE "Pacientes" OWNER TO repoadmin;

-- Name: Poblaciones; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Poblaciones" (
  "Poblacion" character varying(11) NOT NULL,
  "Provincia" character(2) NOT NULL,
  "NombrePoblacion" text NOT NULL
);
ALTER TABLE "Poblaciones" OWNER TO repoadmin;

-- Name: Sectores; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Sectores" (
  "Sector" text NOT NULL,
  "NombreSector" text NOT NULL
);
ALTER TABLE "Sectores" OWNER TO repoadmin;

-- Name: Zonas; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Zonas" (
  "Zona" text NOT NULL,
  "NombreZona" text NOT NULL
);
ALTER TABLE "Zonas" OWNER TO repoadmin;
```

```
-- Name: ConsultaFarmacia; Type: MATERIALIZED VIEW; Schema: public; Owner: repoadmin
CREATE MATERIALIZED VIEW "ConsultaFarmacia" AS
SELECT "Farmacia"."IdEvento",
       "Farmacia"."CIA",
       "Farmacia"."Fecha",
       "Farmacia"."GrupoTerapeutico",
       "Farmacia"."CodigoATC",
       "Farmacia"."NombreComercial",
       "Farmacia"."Empresa",
       "Farmacia"."Administracion",
       "Farmacia"."Unidades",
       "Farmacia"."Precio",
       "Farmacia"."Aportacion",
       "Farmacia"."Visado",
       "Farmacia"."Cofinanciacion",
       "Pacientes"."Sexo",
       "Pacientes"."FechaNacimiento",
       "Pacientes"."Nacionalidad",
       "Pacientes"."PaisNacimiento",
       "Pacientes"."ZonaSalud",
       "Pacientes"."CentroSalud",
       "Centros"."NombreCentro",
       "Zonas"."NombreZona",
       "Pacientes"."Poblacion",
       "Pacientes"."Sector",
       "Poblaciones"."NombrePoblacion",
       "Sectores"."NombreSector"
FROM "Farmacia",
     "Pacientes",
     "Centros",
     "Zonas",
     "Poblaciones",
     "Sectores"
WHERE (((("Pacientes"."CIA" = "Farmacia"."CIA") AND ("Centros"."Centro" =
"Pacientes"."CentroSalud"))) AND ("Zonas"."Zona" = "Pacientes"."ZonaSalud")) AND
(("Poblaciones"."Poblacion")::text = ("Pacientes"."Poblacion")::text)) AND
("Sectores"."Sector" = "Pacientes"."Sector"))
WITH NO DATA;
ALTER TABLE "ConsultaFarmacia" OWNER TO repoadmin;

-- Name: IngresoHospitalario; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "IngresoHospitalario" (
  "Centro" text,
  "TipoAsistencia" text,
  "FechaIngreso" date,
  "TipoIngreso" text,
  "ProvinciaResidencia" integer,
  "MunicipioResidencia" text,
  "CP" integer,
  "ServicioAlta" text,
  "SeccionAlta" text,
  "MedicoAlta" text,
  "TipoAlta" integer,
  "Traslado" text,
  "VersionCIE" integer,
  "Diagnostico1" text,
  "Diagnostico2" text,
  "Diagnostico3" text,
  "Diagnostico4" text,
  "Diagnostico5" text,
  "Diagnostico6" text,
  "Diagnostico7" text,
```



```

    "Diagnostico8" text,
    "Diagnostico9" text,
    "Diagnostico10" text,
    "Diagnostico11" text,
    "Diagnostico12" text,
    "Diagnostico13" text,
    "Diagnostico14" text,
    "Diagnostico15" text,
    "Procedimiento1" text,
    "Procedimiento2" text,
    "Procedimiento3" text,
    "Procedimiento4" text,
    "Procedimiento5" text,
    "Procedimiento6" text,
    "Procedimiento7" text,
    "Procedimiento8" text,
    "Procedimiento9" text,
    "Procedimiento10" text,
    "Procedimiento11" text,
    "Procedimiento12" text,
    "Procedimiento13" text,
    "Procedimiento14" text,
    "Procedimiento15" text,
    "Terapia1" text,
    "Terapia2" text,
    "Terapia3" text,
    "Terapia4" text,
    "Terapia5" text,
    "Terapia6" text,
    "Terapia7" text,
    "Terapia8" text,
    "Terapia9" text,
    "Terapia10" text,
    "Reingreso" character(1),
    "Exitus" character(1),
    "VersionGRD" integer,
    "CodigoGRD" text,
    "Severidad" integer,
    "Mortalidad" integer,
    "Zona" text,
    "Sector" text
)
INHERITS ("Evento");
ALTER TABLE "IngresoHospitalario" OWNER TO repoadmin;

-- Name: TipoAlta; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "TipoAlta" (
    "TipoAlta" integer NOT NULL,
    "DescriptorAlta" text
);
ALTER TABLE "TipoAlta" OWNER TO repoadmin;

-- Name: ConsultaIngresos; Type: MATERIALIZED VIEW; Schema: public; Owner: repoadmin
CREATE MATERIALIZED VIEW "ConsultaIngresos" AS
SELECT "IngresoHospitalario"."CIA",
    "IngresoHospitalario"."Fecha",
    "IngresoHospitalario"."Centro",
    "IngresoHospitalario"."TipoAsistencia",
    "IngresoHospitalario"."FechaIngreso",
    "IngresoHospitalario"."TipoIngreso",
    "IngresoHospitalario"."ProvinciaResidencia",
    "IngresoHospitalario"."MunicipioResidencia",

```

```

    "IngresoHospitalario"."CP",
    "IngresoHospitalario"."ServicioAlta",
    "IngresoHospitalario"."SeccionAlta",
    "IngresoHospitalario"."MedicoAlta",
    "IngresoHospitalario"."TipoAlta",
    "TipoAlta"."DescriptorAlta",
    "IngresoHospitalario"."Traslado",
    d1."DiagnosticoCIE" AS diagnostico1,
    d1."DescriptorCIE" AS descriptordiagnostico1,
    d2."DiagnosticoCIE" AS diagnostico2,
    d2."DescriptorCIE" AS descriptordiagnostico2,
    d3."DiagnosticoCIE" AS diagnostico3,
    d3."DescriptorCIE" AS descriptordiagnostico3,
    d4."DiagnosticoCIE" AS diagnostico4,
    d4."DescriptorCIE" AS descriptordiagnostico4,
    d5."DiagnosticoCIE" AS diagnostico5,
    d5."DescriptorCIE" AS descriptordiagnostico5,
    d6."DiagnosticoCIE" AS diagnostico6,
    d6."DescriptorCIE" AS descriptordiagnostico6,
    d7."DiagnosticoCIE" AS diagnostico7,
    d7."DescriptorCIE" AS descriptordiagnostico7,
    d8."DiagnosticoCIE" AS diagnostico8,
    d8."DescriptorCIE" AS descriptordiagnostico8,
    d9."DiagnosticoCIE" AS diagnostico9,
    d9."DescriptorCIE" AS descriptordiagnostico9,
    d10."DiagnosticoCIE" AS diagnostico10,
    d10."DescriptorCIE" AS descriptordiagnostico10,
    d11."DiagnosticoCIE" AS diagnostico11,
    d11."DescriptorCIE" AS descriptordiagnostico11,
    d12."DiagnosticoCIE" AS diagnostico12,
    d12."DescriptorCIE" AS descriptordiagnostico12,
    d13."DiagnosticoCIE" AS diagnostico13,
    d13."DescriptorCIE" AS descriptordiagnostico13,
    d14."DiagnosticoCIE" AS diagnostico14,
    d14."DescriptorCIE" AS descriptordiagnostico14,
    d15."DiagnosticoCIE" AS diagnostico15,
    d15."DescriptorCIE" AS descriptordiagnostico15,
    "IngresoHospitalario"."IdEvento",
    "Centros"."NombreCentro"
FROM (((((((((((((((("IngresoHospitalario"
    LEFT JOIN "TipoAlta" ON (("IngresoHospitalario"."TipoAlta" =
"TipoAlta"."TipoAlta"))))
    LEFT JOIN "Centros" ON (("IngresoHospitalario"."Centro" = "Centros"."Centro"))))
    LEFT JOIN "CIE" d1 ON (((("IngresoHospitalario"."VersionCIE" = d1."VersionCIE") AND
("IngresoHospitalario"."Diagnostico1" = d1."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d2 ON (((("IngresoHospitalario"."VersionCIE" = d2."VersionCIE") AND
("IngresoHospitalario"."Diagnostico2" = d2."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d3 ON (((("IngresoHospitalario"."VersionCIE" = d3."VersionCIE") AND
("IngresoHospitalario"."Diagnostico3" = d3."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d4 ON (((("IngresoHospitalario"."VersionCIE" = d4."VersionCIE") AND
("IngresoHospitalario"."Diagnostico4" = d4."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d5 ON (((("IngresoHospitalario"."VersionCIE" = d5."VersionCIE") AND
("IngresoHospitalario"."Diagnostico5" = d5."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d6 ON (((("IngresoHospitalario"."VersionCIE" = d6."VersionCIE") AND
("IngresoHospitalario"."Diagnostico6" = d6."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d7 ON (((("IngresoHospitalario"."VersionCIE" = d7."VersionCIE") AND
("IngresoHospitalario"."Diagnostico7" = d7."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d8 ON (((("IngresoHospitalario"."VersionCIE" = d8."VersionCIE") AND
("IngresoHospitalario"."Diagnostico8" = d8."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d9 ON (((("IngresoHospitalario"."VersionCIE" = d9."VersionCIE") AND
("IngresoHospitalario"."Diagnostico9" = d9."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d10 ON (((("IngresoHospitalario"."VersionCIE" = d10."VersionCIE") AND

```

```

("IngresoHospitalario"."Diagnostico10" = d10."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d11 ON (((("IngresoHospitalario"."VersionCIE" = d11."VersionCIE") AND
("IngresoHospitalario"."Diagnostico11" = d11."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d12 ON (((("IngresoHospitalario"."VersionCIE" = d12."VersionCIE") AND
("IngresoHospitalario"."Diagnostico12" = d12."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d13 ON (((("IngresoHospitalario"."VersionCIE" = d13."VersionCIE") AND
("IngresoHospitalario"."Diagnostico13" = d13."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d14 ON (((("IngresoHospitalario"."VersionCIE" = d14."VersionCIE") AND
("IngresoHospitalario"."Diagnostico14" = d14."DiagnosticoCIE"))))
    LEFT JOIN "CIE" d15 ON (((("IngresoHospitalario"."VersionCIE" = d15."VersionCIE") AND
("IngresoHospitalario"."Diagnostico15" = d15."DiagnosticoCIE"))))
    WITH NO DATA;
ALTER TABLE "ConsultaIngresos" OWNER TO repoadmin;

-- Name: Urgencias; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Urgencias" (
    "TipoConsulta" text,
    "VersionCIE" integer DEFAULT 9 NOT NULL,
    "Diagnostico1" text,
    "Diagnostico2" text,
    "Diagnostico3" text,
    "TipoAlta" integer NOT NULL
)
INHERITS ("Evento");
ALTER TABLE "Urgencias" OWNER TO repoadmin;

-- Name: ConsultaUrgencias; Type: MATERIALIZED VIEW; Schema: public; Owner: repoadmin
CREATE MATERIALIZED VIEW "ConsultaUrgencias" AS
SELECT "Pacientes"."Sexo",
    "Pacientes"."FechaNacimiento",
    "Pacientes"."Nacionalidad",
    "Pacientes"."PaisNacimiento",
    "Pacientes"."ZonaSalud",
    "Pacientes"."CentroSalud",
    "Centros"."NombreCentro",
    "Zonas"."NombreZona",
    "Pacientes"."Poblacion",
    "Pacientes"."Sector",
    "Poblaciones"."NombrePoblacion",
    "Sectores"."NombreSector",
    "Urgencias"."IdEvento",
    "Urgencias"."CIA",
    "Urgencias"."Fecha",
    "Urgencias"."TipoConsulta",
    "Urgencias"."VersionCIE",
    "Urgencias"."Diagnostico1",
    "Urgencias"."Diagnostico2",
    "Urgencias"."Diagnostico3",
    "Urgencias"."TipoAlta",
    cie1."DescriptorCIE" AS descriptorciel,
    cie2."DescriptorCIE" AS descriptorcie2,
    cie3."DescriptorCIE" AS descriptorcie3,
    "TipoAlta"."DescriptorAlta"
FROM "Pacientes",
    "Centros",
    "Zonas",
    "Poblaciones",
    "Sectores",
    (("Urgencias"
    LEFT JOIN "CIE" cie1 ON (((("Urgencias"."VersionCIE" = cie1."VersionCIE") AND
("Urgencias"."Diagnostico1" = cie1."DiagnosticoCIE"))))
    LEFT JOIN "CIE" cie2 ON (((("Urgencias"."VersionCIE" = cie2."VersionCIE") AND

```

```

("Urgencias"."Diagnostico1" = cie2."DiagnosticoCIE"))))
  LEFT JOIN "CIE" cie3 ON (("Urgencias"."VersionCIE" = cie3."VersionCIE") AND
("Urgencias"."Diagnostico1" = cie3."DiagnosticoCIE"))),
  "TipoAlta"
  WHERE (((("Centros"."Centro" = "Pacientes"."CentroSalud") AND ("Zonas"."Zona" =
"Pacientes"."ZonaSalud")) AND ("Poblaciones"."Poblacion")::text =
("Pacientes"."Poblacion")::text)) AND ("Sector"."Sector" = "Pacientes"."Sector")) AND
("Urgencias"."CIA" = "Pacientes"."CIA")) AND ("Urgencias"."TipoAlta" =
"TipoAlta"."TipoAlta"))
  WITH NO DATA;
ALTER TABLE "ConsultaUrgencias" OWNER TO repoadmin;

```

```
-- Name: ConsultaVisitaAP; Type: MATERIALIZED VIEW; Schema: public; Owner: repoadmin
```

```
CREATE MATERIALIZED VIEW "ConsultaVisitaAP" AS
```

```

SELECT "Pacientes"."Sexo",
  "Pacientes"."FechaNacimiento",
  "Pacientes"."Nacionalidad",
  "Pacientes"."PaisNacimiento",
  "Pacientes"."ZonaSalud",
  "Pacientes"."CentroSalud",
  "Centros"."NombreCentro",
  "Zonas"."NombreZona",
  "Pacientes"."Poblacion",
  "Pacientes"."Sector",
  "Poblaciones"."NombrePoblacion",
  "Sector"."NombreSector",
  "ConsultaAP"."IdEvento",
  "ConsultaAP"."CIA",
  "ConsultaAP"."Fecha",
  "ConsultaAP"."DiagnosticoCIAP",
  "ConsultaAP"."FechaDiagnostico",
  "ConsultaAP"."CierreDiagnostico",
  "ConsultaAP"."Acto",
  "CIAP"."DescriptorCIAP"
FROM "Pacientes",
  "Centros",
  "Zonas",
  "Poblaciones",
  "Sector",
  "ConsultaAP",
  "CIAP"
  WHERE (((("Pacientes"."CIA" = "ConsultaAP"."CIA") AND ("Centros"."Centro" =
"Pacientes"."CentroSalud")) AND ("Zonas"."Zona" = "Pacientes"."ZonaSalud")) AND
(("Poblaciones"."Poblacion")::text = ("Pacientes"."Poblacion")::text)) AND
("Sector"."Sector" = "Pacientes"."Sector")) AND ("ConsultaAP"."DiagnosticoCIAP" =
"CIAP"."DiagnosticoCIAP"))
  WITH NO DATA;
ALTER TABLE "ConsultaVisitaAP" OWNER TO repoadmin;

```

```
-- Name: ConsultaVisitaEspecializada; Type: MATERIALIZED VIEW; Schema: public; Owner:
repoadmin
```

```
CREATE MATERIALIZED VIEW "ConsultaVisitaEspecializada" AS
```

```

SELECT "Pacientes"."Sexo",
  "Pacientes"."FechaNacimiento",
  "Pacientes"."Nacionalidad",
  "Pacientes"."PaisNacimiento",
  "Pacientes"."ZonaSalud",
  "Pacientes"."CentroSalud",
  "Centros"."NombreCentro",
  "Zonas"."NombreZona",
  "Pacientes"."Poblacion",
  "Pacientes"."Sector",

```

```

    "Poblaciones"."NombrePoblacion",
    "Sectores"."NombreSector",
    "ConsultaEspecializada"."IdEvento",
    "ConsultaEspecializada"."CIA",
    "ConsultaEspecializada"."Fecha",
    "ConsultaEspecializada"."Centro",
    "ConsultaEspecializada"."Especialidad",
    "ConsultaEspecializada"."Origen",
    "ConsultaEspecializada"."PrimeraConsulta",
    centrospecializada."NombreCentro" AS centroconsulta
FROM "Pacientes",
    "Centros",
    "Zonas",
    "Poblaciones",
    "Sectores",
    "ConsultaEspecializada",
    "Centros" centrospecializada
WHERE (((("Pacientes"."CIA" = "ConsultaEspecializada"."CIA") AND ("Centros"."Centro"
= "Pacientes"."CentroSalud")) AND ("Zonas"."Zona" = "Pacientes"."ZonaSalud")) AND
(("Poblaciones"."Poblacion")::text = ("Pacientes"."Poblacion")::text)) AND
("Sectores"."Sector" = "Pacientes"."Sector") AND ("ConsultaEspecializada"."Centro" =
centrospecializada."Centro"))
WITH NO DATA;
ALTER TABLE "ConsultaVisitaEspecializada" OWNER TO repoadmin;

-- Name: Especialidad; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Especialidad" (
    "Especialidad" text NOT NULL,
    "NombreEspecialidad" text NOT NULL
);
ALTER TABLE "Especialidad" OWNER TO repoadmin;

-- Name: Evento_IdEvento_seq; Type: SEQUENCE; Schema: public; Owner: repoadmin
CREATE SEQUENCE "Evento_IdEvento_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE "Evento_IdEvento_seq" OWNER TO repoadmin;

-- Name: Evento_IdEvento_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: repoadmin
ALTER SEQUENCE "Evento_IdEvento_seq" OWNED BY "Evento"."IdEvento";

-- Name: PacientesEventos; Type: MATERIALIZED VIEW; Schema: public; Owner: repoadmin
CREATE MATERIALIZED VIEW "PacientesEventos" AS
SELECT "Pacientes"."CIA",
    "Pacientes"."Sexo",
    "Pacientes"."FechaNacimiento",
    "Pacientes"."Nacionalidad",
    "Pacientes"."PaisNacimiento",
    "Pacientes"."Alta",
    "Pacientes"."ZonaSalud",
    "Pacientes"."CentroSalud",
    "Pacientes"."Activo",
    "Pacientes"."FechaBaja",
    "Pacientes"."FechaFallecimiento",
    "Pacientes"."TipoUsuario",
    "Pacientes"."Situacion",
    "Pacientes"."Poblacion",
    "Pacientes"."CP",
    "Pacientes"."Sector",

```

```
    pg_class.relname,
    "Evento"."IdEvento",
    "Evento"."Fecha"
FROM "Pacientes",
    "Evento",
    pg_class
WHERE (("Pacientes"."CIA" = "Evento"."CIA") AND ("Evento".tableoid = pg_class.oid))
WITH NO DATA;
ALTER TABLE "PacientesEventos" OWNER TO repoadmin;

-- Name: Países; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Países" (
    "Pais" integer NOT NULL,
    "NombrePais" text
);
ALTER TABLE "Países" OWNER TO repoadmin;

-- Name: Provincias; Type: TABLE; Schema: public; Owner: repoadmin
CREATE TABLE "Provincias" (
    idprovincia integer NOT NULL,
    provincia character varying NOT NULL
);
ALTER TABLE "Provincias" OWNER TO repoadmin;

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaAP" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: FechaCarga; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaAP" ALTER COLUMN "FechaCarga" SET DEFAULT now();

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaEspecializada" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: FechaCarga; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaEspecializada" ALTER COLUMN "FechaCarga" SET DEFAULT now();

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Evento" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Farmacia" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: FechaCarga; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Farmacia" ALTER COLUMN "FechaCarga" SET DEFAULT now();

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "IngresoHospitalario" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: FechaCarga; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "IngresoHospitalario" ALTER COLUMN "FechaCarga" SET DEFAULT now();

-- Name: IdEvento; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Urgencias" ALTER COLUMN "IdEvento" SET DEFAULT
nextval('"Evento_IdEvento_seq" '::regclass);

-- Name: FechaCarga; Type: DEFAULT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Urgencias" ALTER COLUMN "FechaCarga" SET DEFAULT now();
```

```
-- Name: PK_ATC; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ATC"
  ADD CONSTRAINT "PK_ATC" PRIMARY KEY ("ATC");

-- Name: PK_CENTROS; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Centros"
  ADD CONSTRAINT "PK_CENTROS" PRIMARY KEY ("Centro");

-- Name: PK_CIAP; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "CIAP"
  ADD CONSTRAINT "PK_CIAP" PRIMARY KEY ("DiagnosticoCIAP");

-- Name: PK_CIE; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "CIE"
  ADD CONSTRAINT "PK_CIE" PRIMARY KEY ("VersionCIE", "DiagnosticoCIE");

-- Name: PK_CONSULTAAP; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaAP"
  ADD CONSTRAINT "PK_CONSULTAAP" PRIMARY KEY ("IdEvento");

-- Name: PK_ConsultaEspecializada; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "ConsultaEspecializada"
  ADD CONSTRAINT "PK_ConsultaEspecializada" PRIMARY KEY ("IdEvento");

-- Name: PK_ESPECIALIDAD; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Especialidad"
  ADD CONSTRAINT "PK_ESPECIALIDAD" PRIMARY KEY ("Especialidad");

-- Name: PK_EVENTO; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Evento"
  ADD CONSTRAINT "PK_EVENTO" PRIMARY KEY ("IdEvento");

-- Name: PK_FARMACIA; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Farmacia"
  ADD CONSTRAINT "PK_FARMACIA" PRIMARY KEY ("IdEvento");

-- Name: PK_INGRESOHOSPITALARIO; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "IngresoHospitalario"
  ADD CONSTRAINT "PK_INGRESOHOSPITALARIO" PRIMARY KEY ("IdEvento");

-- Name: PK_PACIENTE; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Pacientes"
  ADD CONSTRAINT "PK_PACIENTE" PRIMARY KEY ("CIA");

-- Name: PK_PAIS; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Paises"
  ADD CONSTRAINT "PK_PAIS" PRIMARY KEY ("Pais");

-- Name: PK_POBLACIONES; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Poblaciones"
  ADD CONSTRAINT "PK_POBLACIONES" PRIMARY KEY ("Poblacion");

-- Name: PK_PROVINCIAS; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Provincias"
  ADD CONSTRAINT "PK_PROVINCIAS" PRIMARY KEY (idprovincia);

-- Name: PK_SECTOR; Type: CONSTRAINT; Schema: public; Owner: repoadmin
ALTER TABLE ONLY "Sectores"
  ADD CONSTRAINT "PK_SECTOR" PRIMARY KEY ("Sector");
```

```
-- Name: PK_TIPOALTA; Type: CONSTRAINT; Schema: public; Owner: repoadmin  
ALTER TABLE ONLY "TipoAlta"  
  ADD CONSTRAINT "PK_TIPOALTA" PRIMARY KEY ("TipoAlta");
```

```
-- Name: PK_URGENCIAS; Type: CONSTRAINT; Schema: public; Owner: repoadmin  
ALTER TABLE ONLY "Urgencias"  
  ADD CONSTRAINT "PK_URGENCIAS" PRIMARY KEY ("IdEvento");
```

```
-- Name: PK_ZONAS; Type: CONSTRAINT; Schema: public; Owner: repoadmin  
ALTER TABLE ONLY "Zonas"  
  ADD CONSTRAINT "PK_ZONAS" PRIMARY KEY ("Zona");
```


7.3. Anexo 3: Licencia

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

- a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
- c. **BY-NC-SA Compatible License** means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.
- d. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section [2\(b\)\(1\)-\(2\)](#) are not Copyright and Similar Rights.
- e. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- f. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- g. **License Elements** means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution, NonCommercial, and ShareAlike.
- h. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
- i. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- j. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.
- k. **NonCommercial** means not primarily intended for or directed towards commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange.
- l. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- m. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

- n. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
 - A. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and
 - B. produce, reproduce, and Share Adapted Material for NonCommercial purposes only.
2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
3. Term. The term of this Public License is specified in Section [6\(a\)](#).
4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section [2\(a\)\(4\)](#) never produces Adapted Material.
5. Downstream recipients.
 - A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
 - B. Additional offer from the Licensor – Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.
 - C. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section [3\(a\)\(1\)\(A\)\(i\)](#).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
2. Patent and trademark rights are not licensed under this Public License.
3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:
 - A. retain the following if it is supplied by the Licensor with the Licensed Material:
 - i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
 - ii. a copyright notice;

- iii. a notice that refers to this Public License;
 - iv. a notice that refers to the disclaimer of warranties;
 - v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
- B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
 - C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
2. You may satisfy the conditions in Section [3\(a\)\(1\)](#) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
 3. If requested by the Licensor, You must remove any of the information required by Section [3\(a\)\(1\)\(A\)](#) to the extent reasonably practicable.
- b. ShareAlike.**
- In addition to the conditions in Section [3\(a\)](#), if You Share Adapted Material You produce, the following conditions also apply.
1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-NC-SA Compatible License.
 2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.
 3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section [2\(a\)\(1\)](#) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section [3\(b\)](#); and
- c. You must comply with the conditions in Section [3\(a\)](#) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section [4](#) supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

- a. **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**
- b. **To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.**
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate

automatically.

- b. Where Your right to use the Licensed Material has terminated under Section [6\(a\)](#), it reinstates:
 1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
 2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section [6\(b\)](#) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections [1](#), [5](#), [6](#), [7](#), and [8](#) survive termination of this Public License.

Section 7 – Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.
- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.