**`DRUGDATABASE`**

Tabla de contenido

Tablas Drug (id, name, reviewed) y DrugSynset(id, drugId,synName,isBrand) 1

¿Cómo se cargaron los datos almacenados en estas dos tablas? 2

Tablas Effect(id,name,reviewed) y EffectSysnset(id, effectid,synName) 2

¿Cómo se cargaron los datos almacenados en estas dos tablas? 2

Tabla EffectTypes(id, effectTypeName) 3

Tabla DrugEffect(drugId,effectId,effectTypeId) 3

¿Cómo se cargaron los datos almacenados en esta tabla? 3

Tabla ATCCodes(idATC, description) 3

Tabla has\_ATC(idATC, drugId) 4

Tabla has\_externalIDEffect (effectId, idExternal,source) 4

Tabla has\_externalIDDrug (drugId, idExternal,source) 4

El objetivo principal de esta base de datos es almacenar las relaciones entre fármacos y sus posibles efectos: reacciones adversas e indicaciones. Esta es una importante contribución porque no existe ningún recurso que recoja esta información en formato estructurado. Sería deseable en un futuro que las relaciones se definan teniendo en cuenta la dosis del fármaco, ya que a dosis distintas se pueden obtener indicaciones y efectos adversos distintos. La base de datos también agrupar los fármacos y efectos con sus posibles sinónimos.

La clasificación de los fármacos es recogida mediante el código ATC de los fármacos.

Por último, el diseño también permite vincular el contenido de nuestra base de datos con recursos o bases de datos externas como UMLS, MedDRA, CIMA, DrugBank, SNOMED, entre otros.

**NOTA: ES IMPORTANTE RECORDAR QUE TODA LA INFORMACIÓN CARGADA EN LA BASE DE DATOS SE HA GENERADO DE FORMA COMPLETAMENTE AUTOMÁTICA SIN REALIZAR NINGÚN TIPO DE COMPROBACIÓN SOBRE LA INTEGRIDAD DE LA INFORMACIÓN (EL CÓDIGO JAVA SI SE HA PROBADO, PERO NO SE HA COMPROBADO POR EJEMPLO, SI LO QUE SE GUARDA ES REALMENTE UNA RELACIÓN O NO ENTRE UN FÁRMACO Y UN EFECTO). ADEMÁS, COMO MUCHA DE LA INFORMACIÓN TAMBIÉN SE HA DESCARGADO CON CRAWLERS (POR EJEMPLO, TODA LA CLASIFICACIÓN ATC), NO SERÍA RARO ENCONTRAR ALGUNOS ERRORES.**

**El código Java para cargar la base de datos se encuentra en el proyecto:**

[**https://svn.sintonia.inf.uc3m.es/trendminer/TrendMiner\_isegura**](https://svn.sintonia.inf.uc3m.es/trendminer/TrendMiner_isegura)

**A continuación, se describe cada una de las tablas.**

## Tablas Drug (id, name, reviewed) y DrugSynset(id, drugId,synName,isBrand)

La tabla Drug almacena los principios activos (fármacos genéricos).

Cada nombre en esta tabla es el término principal para designar todos los posibles nombres (sinónimos) de un fármaco (*ibuprofeno)*, es decir, tanto las posibles denominaciones del principio activo (*ibuprofeno o P-Isobutylhydratropic Acid*) como sus posibles marcas comerciales (*dalsy, espidifen, evasprin, junifen,* etc.). En general, en español un principio activo se corresponde con una única denominación. Sin embargo, en inglés es más común que un mismo principio activo se corresponda con varios nombres genéricos (muy similares entre ellos, por ejemplo, ibuprophen, ibuprofene)

La tabla DrugSynset almacenará todos los posibles sinónimos del fármaco.

OJO, el término principal (ibuprofeno) también es almacenado en la tabla DrugSynset. ES decir, ibruprofeno aparece en las dos tablas Drug y DrugSynset.

Los campos de la tabla Drug son:

* Id es un campo autonumérico y es la PK de la tabla.
* Name es el nombre del fármaco. En general, es un principio activo.
* Reviewed: Como la base de datos se carga siguiendo un proceso completamente automático a partir de distintas fuentes (CIMA pero también webs como Vademecum, MedLinePlus, Prospectos.NET y Propectos.ORG), podría ocurrir que en esta tabla estemos almacenando algunos fármacos que no sean principios activos sino marcas comerciales. Por tanto, este campo fue añadido por si en algún momento realizamos una revisión manual de los fármacos contenidos en esta tabla. Por el momento no lo estamos utilizando y se puede eliminar, si lo consideráis necesario.

Los campos de la tabla DrugSynset son:

* Id es un campo autonumérico y es la PK de la tabla.
* drugId es una clave ajena que se corresponde con el id de la tabla Drug. Este campo nos permite relacionar los distintos sinónimos con su término principal.
* synName es el nombre del fármaco.
* isBrand: es un campo booleano para indicar si el nombre se corresponde o no con una marca comercial. Por el momento, no lo estamos usando.

**Nota:** Conceptualmente hubiera preferido que el término principal se guarde únicamente en la tabla Drug, pero ahora no consigo recordar porque decidimos que el término principal fuera otro registro en la tabla DrugSynset!!!.

### ¿Cómo se cargaron los datos almacenados en estas dos tablas?

En primer lugar, la carga fue cargada a partir del diccionario de fármacos creado por Ricardo, que a su vez fue creado a partir del recurso CIMA (fichero trendminerplus-f.txt). Cada registro en este diccionario consta de un término principal (que es almacenado en la tabla Drug) y un conjunto de posibles sinónimos (que son almacenados en la tabla DrugSynset).

## Tablas Effect(id,name,reviewed) y EffectSysnset(id, effectid,synName)

Seguimos el mismo criterio que para fármacos. Los términos principales se almacenan en la tabla Effect, y sus posibles sinónimos (entre los que se incluye también el termino principal) son almacenados en la tabla EffectSynset.

### ¿Cómo se cargaron los datos almacenados en estas dos tablas?

En primer lugar, la carga fue cargada a partir del diccionario de fármacos creado por Ricardo, que a su vez fue creado a partir del recurso MedDRA, y que posteriormente fue modificado por Santiago para excluir los términos de la categoría de procedimientos y tests (fichero meddra-f.txt). Cada registro en este diccionario consta de un término principal (que es almacenado en la tabla Drug) y un conjunto de posibles sinónimos (que son almacenados en la tabla DrugSynset).

**Clase Java: labda.isb.bbdd.LoadDrugDB. En particular, el método** loadBD() contiene el código necesario para cargar los fármacos a partir del diccionario trendminerplus-f.txt y los efectos a partir del diccionario meddra-f.txt. Dicho método también se encarga de guardar las relaciones entre los fármacos y sus efectos (indicaciones o adversos).

## Tabla EffectTypes(id, effectTypeName)

Sólo tiene dos registros: 1: indication y 2: adverse effect. Nos planteamos usar una tabla por si en un futuro, necesitamos ampliar la clasificación de los posibles efectos de un fármaco.

## Tabla DrugEffect(drugId,effectId,effectTypeId)

Esta tabla se ocupa de almacenar las relaciones entre un fármaco y sus posibles efectos. Las relaciones se establecen a nivel de los términos principales de fármaco y efecto.

Así, los campos de la tabla DrugEffect son:

* drugId es clave ajena heredada de la tabla Drug (campo id).
* effectId clave ajena heredada de la tabla Effect (campo id).
* effectTypeId clave ajena heredada de la tabla EffectType (campo id).

La clave principal está formada por los 3 campos. Aunque no es frecuente un efecto podría ser tanto una indicación como un efecto adverso de un mismo fármaco (de hecho hay unos 1000 registros que cumplen está condición).

### ¿Cómo se cargaron los datos almacenados en esta tabla?

Se desarrollaron una serie de web crawlers (desarrollados con jsoup) que descargaron las secciones de indicaciones y efectos adversos de los fármacos de webs como MedLinePlus, Prospectos.NET y Prospectos.ORG.

Los crawlers se pueden encontrar en el paquete java: labda.isa.webcrawler.

Los textos descargados se pueden encontrar en el proyecto svn: <https://svn.sintonia.inf.uc3m.es/trendminer/BioCorpora>

A continuación, todos los efectos que aparecen en la sección “Indicaciones” de un fármaco, se grabaron como una indicación en la tabla DrugEffect. Del mismo modo, los efectos que aparecían descritos en la sección efectos adversos de un determinado fármaco, se almacenaron como un efecto adverso en la tabla DrugEffect.

**Nota:** Creo que he tenido un problema con el SVN, y falta parte del código que detecta las indicaciones y los efectos adversos en las secciones descargadas de los prospectos, los almacena en listas, que posteriormente se usan para grabar las relaciones en la tabla DrugEffect. No es importante, porque toda esta información ya está cargada en la base de datos, pero trataré de ver como puedo recuperarla (espero tenerla en el pc de la uni!).

## Tabla ATCCodes(idATC, description)

Esta tabla almacena todos los códigos ATC del sistema de clasificación de fármacos ATC. Este sistema de clasificación es propuesto y mantenido por la WHO y utilizado por la mayoría de los países europeos (en USA se usa el sistema de la FDA). El sistema consiste en una jerarquía de 5 niveles, siendo el quinto nivel donde se encuentran los principios activos. Los otros niveles serían las posibles grupos terapéuticos y farmacológicos que propone la clasificación.

Inicialmente esta tabla sólo contenía los códigos ATC de los principios activos (nivel 5), pero finalmente decidimos cargar también los códigos las familias de fármacos (del nivel 2 al 4). Los códigos ATC fueron descargados usando un web crawler (clase java CrawlerForATCWiki.java). Posteriormente, esta lista fue grabada en la tabla ATCCodes.

No hemos creido necesario almacenar las relaciones jerárquicas entre los grupos, porque el propio código ATC de un principio activo o grupo, permite inferir fácilmente cuales son sus padres.

## Tabla has\_ATC(idATC, drugId)

Los campos de la tabla son:

* idATC que es clave ajena de la tabla ATCCodes (campo idATC).
* drugId que es clave ajena de la tabla DrugSynset (campo id).

Ambos campos forman la clave principal. Un fármaco podría estar clasificado por varios códigos ATC.

Aunque inicialmente esta tabla fue diseñada para estar relacionada con la tabla Drug, y no con la tabla DrugSysnet, finalmente decidimos cambiar el diseño de la base de datos, porque curiosamente en CIMA el código ATC se relaciona a nivel de marca comercial (fichero prescripción.xml) y no de principio activo.

El código Java para cargar esta tabla se encuentra en el método **loadATCRelations** de la clase **LoadDrugDB**.

Los pasos seguidos para cargar esta tabla, fueron los siguientes:

* Recorrer el fichero prescription.xml y guardar en una lista de todos los fármacos contenidos y su código ATC. Para recuperar dicho código es necesario buscarlo en el fichero DICCIONARIO\_ATC.xml. El código encargado de recuperar esta información se encuentra en la clase **labda.isb.xml.CIMA.ReaderCIMA**.
* Recorrer todos los fármacos almacenados en la tabla DrugSynset y buscarlos en la lista recuperada en el paso anterior, para obtener su código ATC. En concreto, lo que hacemos es mirar si el nombre recuperado de la tabla DrugSynset está contenido en alguno de los nombres contenidos en la lista y que fueron recuperados del fichero prescription.xml. No hacemos una búsqueda exacta, porque los nombres de DrugSynset fueron “recortados” por Ricardo para eliminar la información de dosis, composición, fabricante, etc.
* Finalmente, la relación entre el fármaco y su código ATC es almacenada en la tabla has\_ATC.

## Tabla has\_externalIDEffect (effectId, idExternal,source)

El objetivo de la tabla es permitir vincular los efectos de nuestra base de datos con otras bases de datos externas como MedDRA, UMLS, etc. En este momento, la tabla sólo contiene los id de los efectos en MedDRA. Esto fue gracias a que nuestro diccionario fue generado conservando esta información de MEDRA. Desgraciadamente, no seguimos el mismo criterio para conservar los id de CIMA cuando creamos el diccionario de fármacos. Aunque se podría recuperar fácilmente.

Los campos son:

* effectId es clave ajena de la tabla EffectSynset (campo id). Es decir, la relación entre efectos y bases de datos externas hemos decidida guardarla a nivel de sinónimo. Es razonable pensar que unos sinónimos estarán almacenados en unas bases de datos, mientras que otros estarán registrados en otras bases de datos distintas.
* idExternal se corresponde con el código que tiene dicho efecto en la base de datos externa. Podría ser un código MedDRA, un código UMLS, un código SNOMED, etc… dependiendo de la base de datos.
* Source es un campo que permite indicar el nombre de la base de datos origen (MEDDRA, UMLS, etc.

La clave primaria está formada por los dos primeros campos, ya que queremos almacenar la relación entre un efecto y varias base de datos externas.

## Tabla has\_externalIDDrug (drugId, idExternal,source)

El objetivo de la tabla es permitir vincular los efectos de nuestra base de datos con otras bases de datos externas como MedDRA, UMLS, etc. En este momento, la tabla no contiene ningún registro. Desgraciadamente, cuando creamos nuestro diccionario a partir de CIMA, no conservamos sus códigos. Recuperarlos sería relativamente sencillo (voy a intentar de hacerlo yo antes en los próximos días).

Los campos son:

* drugId es clave ajena de la tabla DrugSynset (campo id). Es decir, la relación entre fármacos y bases de datos externas hemos decidida guardarla a nivel de sinónimo. Es razonable pensar que unos sinónimos estarán almacenados en unas bases de datos, mientras que otros estarán registrados en otras bases de datos distintas.
* idExternal se corresponde con el código que tiene dicho efecto en la base de datos externa. Podría ser un código CIMA, un código UMLS, un código SNOMED, un código en DrugBank, etc… dependiendo de la base de datos.
* Source es un campo que permite indicar el nombre de la base de datos origen (CIMA, DrugBank, UMLS, etc).

La clave primaria está formada por los dos primeros campos, ya que queremos almacenar la relación entre un fármaco y varias base de datos externas.