ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

ANALÍTICA DE DATOS APLICADA AL ENRIQUECIMIENTO DE RUTAS METABÓLICAS A TRAVÉS DE TARGET FISHING

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

CÁRDENAS AYALA VERÓNICA YAKELIN

veronica.cardenas@epn.edu.ec

YÁNEZ ESCARABAY KARLA VICTORIA

victoria.yanez@epn.edu.ec

DIRECTOR: CARRERA IZURIETA IVÁN MARCELO

ivan.carrera@epn.edu.ec

CODIRECTOR: TEJERA PUENTE EDUARDO

eduardo.tejera@udla.edu.ec

Quito, Febrero 2019

AVAL

Certificamos que el presente trabajo fue desarrollado por Verónica Cárdenas y Victoria Yánez, bajo nuestra supervisión.

IVÁN MARCELO CARRERA IZURIETA

DIRECTOR

EDUARDO TEJERA PUENTE

CODIRECTOR

DECLARACIÓN DE AUTORÍA

Nosotras, Verónica Cárdenas y Victoria Yánez declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

VERÓNICA CÁRDENAS	VICTORIA YÁNEZ

DEDICATORIA

Dedico el presente proyecto de investigación a mis padres, quienes con su sacrificio permitieron que pueda culminar esta etapa universitaria exitosamente. Además, a mis hermanas, quienes me apoyaron con su conocimiento y comprensión en momentos de dificultad.

Pero sobre todo dedico este trabajo a mi hermana Emilia, quien es la persona que siempre ha estado a mi lado y no solo como mi familia, sino como mi amiga y confidente.

Victoria.

Dedico el presente trabajo de investigación a mis padres quienes me han apoyado en cada decisión de mi vida, me han enseñado la importancia de la perseverancia y con su esfuerzo me permitieron lograr uno de mis sueños.

A mi abuelito Agustín Ayala que sé que aún me cuida desde el cielo y mi abuelita Amada Suárez, quienes con su cariño me han dado la fuerza de afrontar cada obstáculo que se ha presentado en mi carrera, brindando su amor y apoyo.

A mi hermana Nathaly Cárdenas quien es mi mejor amiga de vida, mi compañera y consejera.

Verónica.

AGRADECIMIENTO

Agradezco a Dios por ser mi guía en esta etapa universitaria y por permitirme culminarla exitosamente.

A mi madre Lupita y a mi padre Carlos, quienes me han enseñado que a pesar de todas las dificultades que se presenten en la vida, el amor por la familia es la fuerza que se necesita para continuar con los proyectos planteados. Además, les agradezco por ser padres amorosos que siempre se han preocupado por mi bienestar y me han apoyado tanto en momentos de tristeza y dolor, como en circunstancias de felicidad.

A mi hermana Karla por cuidarme, guiarme, aconsejarme y protegerme desde que era una niña. Porque a pesar de las peleas y conflictos que hemos tenido nunca se ha apartado de mi lado. A mi hermana Emilia quien ha sido mi compañera y amiga en todas las etapas de mi vida. Quien siempre ha confiado en lo que puedo lograr y lo que puedo llegar a ser. Quien se ha sentido orgullosa de mí y me ha apoyado cuando sentía que iba a desfallecer.

A Verónica mi amiga en esta etapa universitaria, a quien le agradezco por la paciencia y entrega para culminar el presente trabajo de investigación.

A Alexandra, Nathaly, Katherine, Daniela, José Miguel, Jose, Santiago, Jefferson, Galo y Cristian, con quienes he compartido los momentos más felices de mi vida universitaria. Les agradezco por estos años de amistad, por las experiencias buenas y malas que hemos compartido. A quienes les digo que son también mi familia.

A mis tutores del proyecto de investigación Iván y Eduardo, quienes con su conocimiento y entrega permitieron que este trabajo sea concluido satisfactoriamente.

Victoria.

Agradezco a Dios por guiarme en cada paso que doy y darme la fuerza para culminar esta etapa de mi formación profesional.

A mis padres Wilson Cárdenas y Martha Ayala por ser mis guías y estar conmigo en mis triunfos y fracasos, enseñándome que nunca debo rendirme ante la adversidad. Por ello sé que con esfuerzo podré lograr lo que me proponga en la vida. Gracias por su amor, consejos y constante apoyo.

A mi hermana Nathaly por sacarme una sonrisa siempre que me ve triste o enojada, por compartir mis alegrías, en fin, por ser mi punto de apoyo.

A mi abuelito Agustín que siempre estuvo pendiente de mi vida universitaria y a mi abuelita Amadita por siempre orar por todos sus nietos, para que tengamos la sabiduría de tomar buenas decisiones.

A mis amigos de la universidad con quienes compartí muchas alegrías y momentos de frustración. Momentos que con ellos se convirtieron en risas.

A Victoria por su dedicación y responsabilidad al realizar este proyecto y por ser mi compañera y amiga en toda circunstancia.

A mis amigas Diana, Daniela, Paola y Estefy, con quienes a pesar de la distancia hemos conservado esta hermosa amistad y me han acompañado en cada momento de mi vida universitaria. A Martín por ser mi confidente en todas mis tristezas y alegrías.

A mis tutores Iván y Eduardo por su tiempo y entrega para que podamos culminar con éxito este trabajo de investigación.

Verónica.

ÍNDICE DE CONTENIDO

AVAL I
DECLARACIÓN DE AUTORÍA II
DEDICATORIA III
AGRADECIMIENTO IV
ÍNDICE DE CONTENIDO VI
RESUMEN VIII
ABSTRACT IX
1. INTRODUCCIÓN 1
1.1 Pregunta de investigación 3
1.2 Objetivo General 3
1.3 Objetivos Específicos 3
1.4 Alcance 4
1.5 Marco Teórico 5
2. METODOLOGÍA 13
2.1 Identificación de datos 13
2.2. Adquisición, extracción y limpieza de datos 16
2.3. Agregación y representación de datos 21
2.4. Análisis de datos 24
2.5. Visualización de datos 28
2.6. Utilización de resultados del análisis 33
3. RESULTADOS Y DISCUSIÓN 34
3.1. Resultados 34
3.2. Discusión 37
4. CONCLUSIONES 39
5. REFERENCIAS BIBLIOGRÁFICAS 41
6. ANEXOS 46
ANEXO I 46
ANEXO II 47
ANEXO III 49
ANEXO IV 50
ANEXO V 51

ANEXO VI 52
ANEXO VII 53
ANEXO VIII 56
ANEXO IX 58
ANEXO X 60
ANEXO XI 64
ANEXO XII 65
ORDEN DE EMPASTADO 68

RESUMEN

El presente trabajo propone realizar Analítica de Datos utilizando la metodología del Ciclo de Vida de la Analítica de Big Data para enriquecer las rutas metabólicas de la base de datos *KEGG Pathway* aplicando la técnica de *Target Fishing*. El enriquecimiento de las rutas metabólicas consiste en proponer alternativas de compuestos químicos y blancos mediante el tratamiento de la información proveniente de cuatro fuentes internas y externas, que en este caso serán las bases de datos químico-biológicas: *KEGG Pathway*, UniProt, ChEMBL y ChEBI. Para ello, se aplicarán las etapas del Ciclo de Vida de la Analítica de Big Data. La primera fase abarca la identificación de las fuentes y el formato de la información; la segunda fase comprende un tratamiento conjunto de la adquisición, filtrado, extracción y limpieza de datos. La tercera fase consiste en la agregación y representación de datos, de la que se obtiene los subconjuntos de compuestos químicos y blancos. En la cuarta fase se realiza el análisis de datos utilizando la técnica de *Target Fishing.* Además, en la quinta fase, correspondiente a la visualización de los datos, se crea una interfaz web. Y finalmente, en la fase de utilización de resultados del análisis, se usa la distribución hipergeométrica de probabilidad para validar el enriquecimiento de la información. Al concluir con todas las etapas de la metodología se logró enriquecer las rutas metabólicas con alternativas de compuestos químicos y blancos.

PALABRAS CLAVE: Analítica de Datos, Blanco, Compuesto Químico, Rutas Metabólicas, Químico-Biológicas, Target Fishing

ABSTRACT

This work proposes to implement Data Analytics using the Big Data Analytics Lifecycle methodology to enrich the metabolic pathways of the KEGG Pathway database by applying the Target Fishing technique. Enrichment of the metabolic pathways consists of proposing alternatives of chemical compounds and targets through the treatment of information from four internal and external sources; which, in this case, will be the chemical-biological databases: KEGG Pathway, UniProt, ChEMBL and ChEBI. In order to achieve this, the stages of the Life Cycle of Big Data Analytics methodology will be applied. The first phase covers the identification of the sources and the format of the information; the second phase comprises a joint treatment of the acquisition, filtering, extraction and cleaning of data. The third phase consists in the aggregation and representation of data, from which the subsets of chemical compounds and targets are obtained. In the fourth phase, the data analysis is performed using the Target Fishing technique. In addition, in the fifth phase, corresponding to the visualization of the data, a web interface is created. And finally, in the phase of using the results of the analysis, the hypergeometric distribution of probability is used to validate the enrichment of the information. When concluding with all the stages of the methodology it was possible to enrich the metabolic routes with alternatives of chemical compounds and therapeutic targets.

KEYWORDS: Chemical-Biological, Chemical Compound, Data Analytics, Metabolic Pathways, Target Fishing

1. INTRODUCCIÓN

La analítica de datos es una disciplina que incluye la administración del ciclo de vida de los datos y engloba la recolección, limpieza, organización, almacenamiento, análisis y gobernanza de estos. La analítica de datos emplea el análisis predictivo en base a patrones, tendencias y relaciones que se encuentran en datos actuales o históricos [1]. El procesamiento de estos resultados puede conducir al descubrimiento de información y beneficios en distintas ramas de conocimiento, como en el caso de la Biología Computacional. Por ejemplo, el análisis de información de bases de datos químico-biológicas con la finalidad de obtener predicciones basadas en datos probados experimentalmente [2].

La base de datos *Kyoto Encyclopedia of Genes and Genomes (KEGG) Pathway* contiene una representación gráfica de los procesos celulares [3]. Los procesos celulares son sistemas básicos que involucran reacciones bioquímicas a nivel celular [4] como el transporte, el catabolismo, el metabolismo, el crecimiento y la muerte celular [5]. La información de *KEGG Pathway* se muestra a través del uso de grafos, en los que se encuentran representadas las interacciones moleculares entre genes, procesos y compuestos químicos [6].

En el presente proyecto, se plantea el enriquecimiento de la información contenida sobre las rutas metabólicas de *KEGG Pathway*. En este caso, el enriquecimiento de las rutas consiste en proponer alternativas de compuestos químicos y blancos mediante la integración de información de otras bases de datos como UniProt, ChEMBL y ChEBI, basados en datos experimentales, así como el uso la técnica de *Target Fishing*. Sin embargo, la localización de la información se ha convertido en un reto para los investigadores debido a que esta se encuentra almacenada en diferentes fuentes externas o internas, y también en distintos formatos, dificultando su integración [7].

En el presente trabajo se utilizará una cantidad de datos relativamente grande, con información en diferentes formatos, proveniente de diferentes fuentes. Por estos motivos, se elige utilizar la metodología del Ciclo de Vida de la Analítica de Big Data, propuesto por Thomas Erl [1]. Las etapas del ciclo se desarrollarán como se indica en la Figura 1.1.

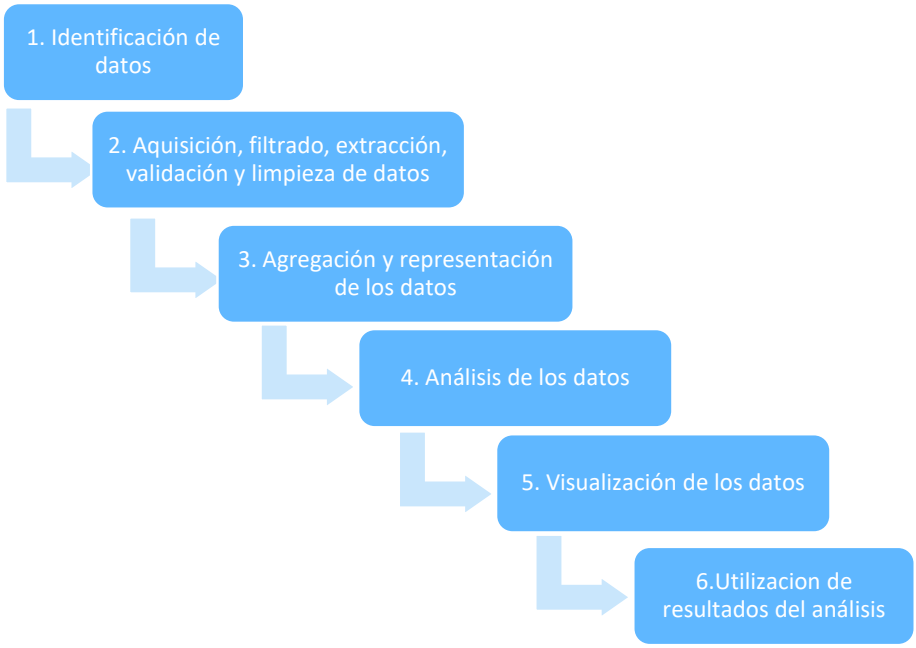

Figura 1.1. Ciclo de Vida de la Analítica de Big Data

Las etapas descritas en la Figura 1.1 comprenden lo siguiente:

1. Se identifican las fuentes de información y el formato de los datos.
2. Se adquiere, filtra, extrae, valida y limpia la información de *KEGG Pathway,* UniProt, ChEMBL y ChEBI.
3. Se obtienen los subconjuntos de compuestos químicos y blancos.
4. Se analiza los datos mediante *Target Fishing*.
5. Se crea una página web para presentar los resultados.
6. Se realiza la distribución hipergeométrica de probabilidad en base a los resultados del análisis.

Organización del presente documento

A continuación de la introducción se organiza el documento con la siguiente información: pregunta de investigación, el objetivo general, los objetivos específicos, el alcance y el marco teórico. Posteriormente, en el capítulo 2 se presenta la metodología, en donde se explica lo realizado en cada una de las fases del Ciclo de Vida de la Analítica de Big Data. En el capítulo 3, se muestran los resultados y discusión. Por último, en el capítulo 4, se presentan las conclusiones del presente proyecto de investigación.

1.1 Pregunta de investigación

Al inicio del presente trabajo, se realizó una investigación preliminar acerca de la información contenida en la base de datos *KEGG Pathway*; la cual incluye datos relevantes sobre los compuestos químicos que forman parte de la ruta metabólica, así como también, identificadores a otras bases de datos.

Dentro de la investigación preliminar, se pudo observar que sería interesante adicionar información relacionada a las interacciones entre los compuestos químicos y las proteínas para obtener un catálogo extendido de información química en la ruta metabólica. A esta adición de información se la referirá en el presente documento como 'enriquecimiento'.

Por tanto, para el presente proyecto, se define la siguiente pregunta de investigación:

> ¿Es posible enriquecer la información de las rutas metabólicas de la base de datos *KEGG Pathway* al considerar la relación entre compuestos químicos y blancos?

1.2 Objetivo General

Implementar el Ciclo de Vida de la Analítica de Big Data para enriquecer las rutas metabólicas de la base de datos *KEGG Pathway* a través de *Target Fishing*, utilizando la información de otras bases de datos químico-biológicas como UniProt, ChEMBL y ChEBI.

1.3 Objetivos Específicos

a) Usar el Ciclo de Vida de la Analítica de Big Data para la obtención y análisis de los datos.

b) Relacionar la información de las bases de datos *KEGG Pathway*, UniProt, ChEMBL y ChEBI.

c) Utilizar la técnica de *Target Fishing* para obtener la lista de compuestos químicos que generen una reacción en los blancos.

d) Elaborar una interfaz web para presentar las rutas metabólicas enriquecidas.

1.4 Alcance

El presente proyecto tiene la finalidad de enriquecer la información de las rutas metabólicas de seres humanos almacenadas en *KEGG Pathway* con la incorporación de datos acerca de blancos contenida en UniProt e información de compuestos químicos y datos experimentales contenida en ChEMBL y ChEBI a través de *Target Fishing*. El desarrollo del trabajo comprende el tratamiento de la información contenida en bases de datos químico-biológicas: *KEGG Pathway Database,* UniProt, ChEMBL y ChEBI. La relación de las bases de datos mencionadas y el análisis de los datos se muestran en la Figura 1.2.

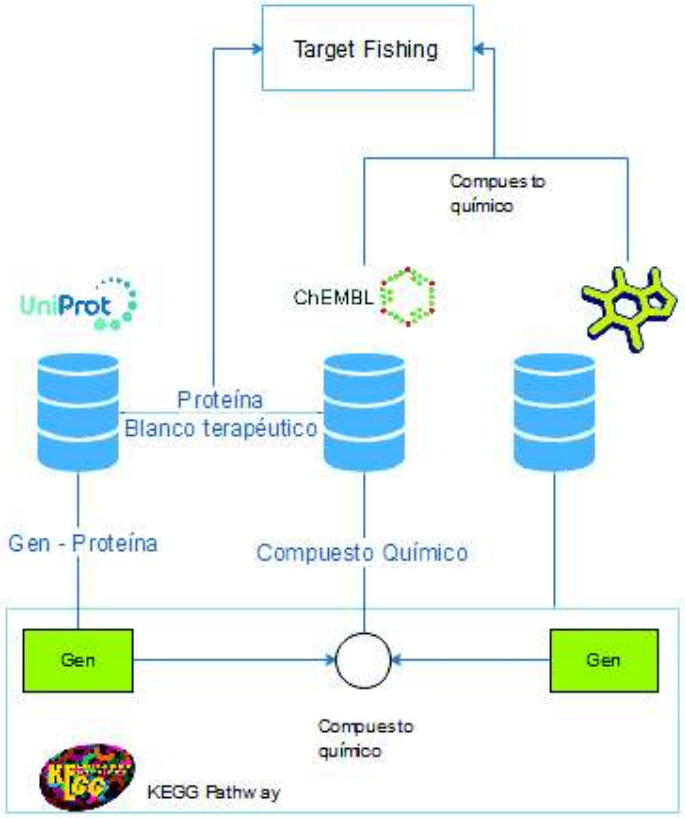

Figura 1.2. Relación de las bases de datos y análisis de los datos

En la Figura 1.2 se detalla lo siguiente:

- La base de datos *KEGG Pathway* contiene información sobre rutas metabólicas de seres humanos, la que incluye: genes, compuestos químicos e interacciones entre ellos.
- Se añade información revisada (Swiss-Prot) de genes y proteínas de la base de datos UniProt.

- Se incluye información de compuestos químicos, blancos y datos experimentales de la base de datos química ChEMBL 22 simplificada.
- Se agrega información sobre compuestos químicos de la base de datos ChEBI.
- Para realizar el análisis de los datos se utiliza la técnica de *Target Fishing*, la cual se emplea para descubrir nuevas relaciones entre compuestos químicos y sus blancos.

Para el presente trabajo, el tratamiento de los datos se realizará en base a la metodología del Ciclo de Vida de la Analítica de Big Data.

1.5 Marco Teórico

En los últimos años, debido a la alta disponibilidad de datos y los avances tecnológicos para la recolección de información, la analítica de datos ha surgido como una herramienta para realizar investigaciones descriptivas y predictivas sobre una gran cantidad de datos, con el objetivo de formular decisiones inteligentes e informadas [8]. La analítica de datos ha sido aplicada en investigaciones de biología computacional debido al incremento del volumen de datos, ya que no se limitan únicamente al almacenamiento de índices o información relevante, sino a toda la información relacionada a los datos experimentales [8]. Por ejemplo, el Instituto Europeo de Bioinformática (EBI), uno de los depósitos de datos de biología más grandes, tenía aproximadamente 40 petabytes de datos sobre genes, proteínas y moléculas pequeñas en 2014, en comparación con 18 petabytes en 2013 [8]. Esta comparación indica que en un año ha existido un incremento de más del 100%. De igual manera, para bases de datos como *KEGG*, la información ha incrementado mediante varios *release* desde 1995 hasta las últimas actualizaciones, realizadas el primero de octubre de 2018 [9]. En la Figura 1.3, Figura 1.4 y Figura 1.5, se detalla cómo ha aumentado la información desde octubre del 2011 a octubre del 2018, entre los cuales se puede observar que 121 rutas metabólicas (*pathways*), 2146 compuestos químicos, y 20232694 genes han sido agregados respectivamente. Cabe recalcar que *KEGG* almacena información complementaria de cada ruta metabólica, compuesto químico y gen.

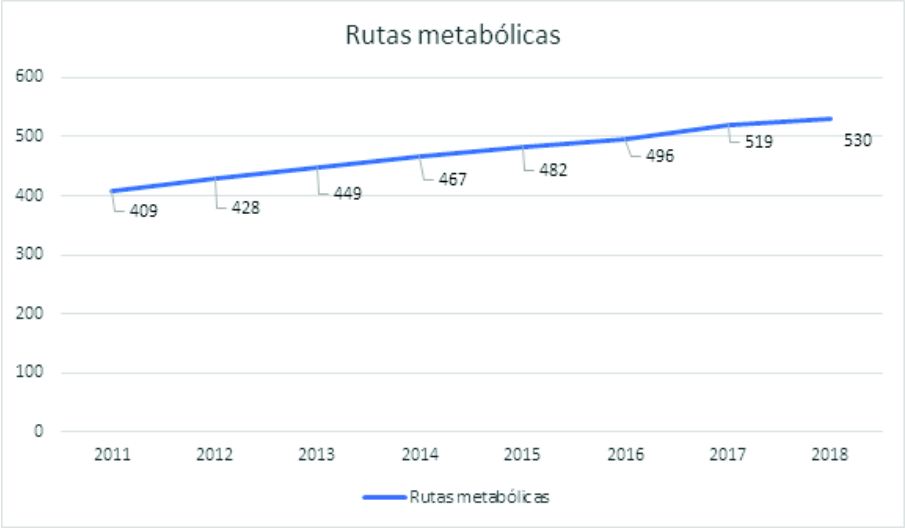

Figura 1.3. *Release history* de rutas metabólicas *[10]*

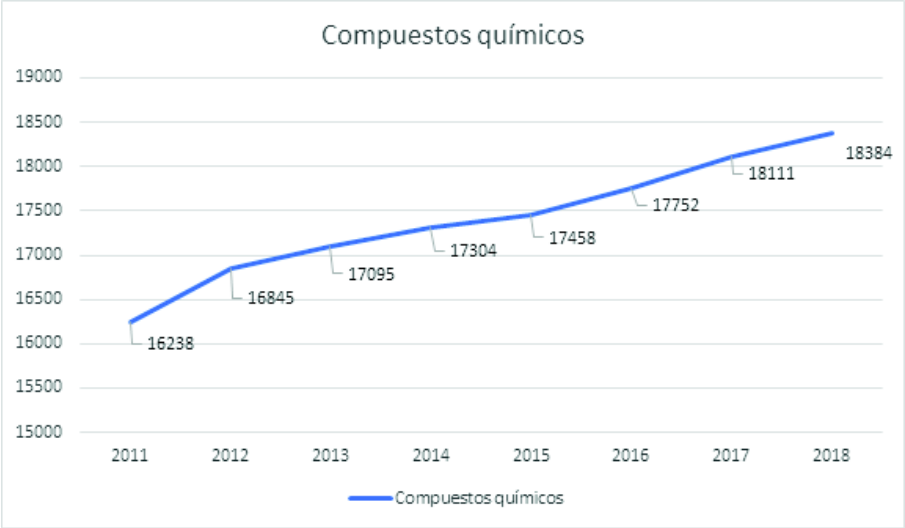

Figura 1.4. *Release history* de compuestos químicos *[10]*

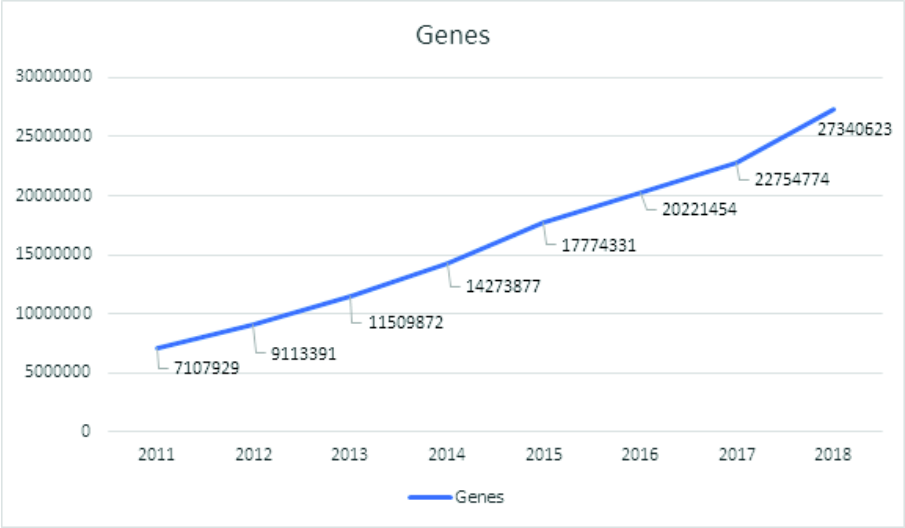

Figura 1.5. *Release history* de genes *[10]*

La analítica de datos a gran escala plantea una serie de desafíos, incluida la transferencia, el acceso, y la administración de los datos; así como la estandarización de los formatos de datos y la integración de múltiples tipos de datos para modelar con precisión la información de sistemas biológicos [11]. Por este motivo, se han implementado metodologías como el Ciclo de Vida de la Analítica de Big Data en experimentos de biología computacional [11]. Por ejemplo, en el proceso de descubrimiento de fármacos, el cual aplica la metodología desde la identificación del objetivo y el mecanismo de acción, hasta encontrar nuevas relaciones y posibles fármacos [12].

Para relacionar la información entre distintas fuentes de datos es necesario comprender los mecanismos biológicos fundamentales del desarrollo de medicamentos, y requiere la integración de información de diferentes niveles y fuentes para considerar más datos relacionados [13]. Estos niveles de información pueden provenir de fuentes externas o internas y comprenden grandes volúmenes de datos que deberán ser analizados mediante técnicas *in silico* como el *Target Fishing*. El término *in silico* se utiliza para referirse a la experimentación realizada por computadora. Esta experimentación se emplea para crear modelos computacionales o simulaciones que pueden usarse para realizar predicciones, sugerir hipótesis y, en última instancia, proporcionar descubrimientos o avances [14].

El *Target Fishing* ha surgido como un campo interdisciplinario con un gran potencial para avanzar en el diseño y descubrimiento de medicamentos y con ello, los problemas relacionados a toxicidad, la eficacia en seres humanos, el descubrimiento de las primeras etapas de los fármacos y encontrar posibles blancos aún no relacionados con ciertos compuestos químicos [12]. Esto es posible ya que combina técnicas de quimio-informática

y biología computacional para desarrollar enfoques que puedan priorizar los posibles blancos de un compuesto químico [12].

Biología computacional

La Biología Computacional se define como el desarrollo y aplicación de analítica de datos, métodos teóricos, modelos matemáticos y técnicas de simulación computacional para el estudio de sistemas biológicos, conductuales y sociales [15]. Además, es una herramienta que apoya el área de investigación biológica puesto que parte de una hipótesis que necesitará de conocimientos computacionales como: almacenamiento, simulación, desarrollo de algoritmos, entre otros; para validar o refutar dicha hipótesis. Actualmente, presenta nuevos desafíos en términos del manejo de datos, consultas y análisis, explícitamente relacionado a los grandes conjuntos de datos y a sus diversos formatos. Por tanto, esta área se ha complementado con soluciones de Big Data para obtener resultados que aporten con optimización operativa, inteligencia procesable, identificación de nuevos mercados, predicciones, detección de fallas, toma de decisiones mejorada, entre otros [16].

Ciclo de Vida de la Analítica de Big Data

El procedimiento utilizado para la obtención y análisis de los datos se enmarca en las etapas del Ciclo de Vida de la Analítica de Big Data propuestos en [1] y se detallan a continuación: Identificación de datos, Adquisición y filtrado de datos, Extracción de datos, Validación y limpieza de datos, Agregación y representación de datos, Análisis de datos, Visualización de datos y, por último, Utilización de resultados del análisis.

1. Identificación de datos. Identifica los conjuntos de datos y recursos que son requeridos para el análisis del proyecto, los cuales pueden provenir de fuentes externas o internas para ser recolectadas.
2. Adquisición y filtrado de datos. Recolecta la información establecida en el paso previo, la cual es filtrada dependiendo del tipo de fuente y proveedor, con el objetivo de eliminar datos innecesarios o corruptos que no tendrán valor en el propósito del análisis.
3. Extracción de datos. Extrae datos provenientes de formatos diferentes y los transforma a un formato comprensible para el procesamiento y análisis de datos.

4. Validación y limpieza de datos. Se establece las reglas de validación de los datos, elimina cualquier dato inválido y determina si la redundancia permite explorar un conjunto de datos interconectados para completar la información con datos válidos faltantes.
5. Agregación y representación de datos. Se dedica a integrar múltiples conjuntos de datos para llegar a una sola vista unificada. Para ello, debe considerar la estructura de los datos y la semántica.
6. Análisis de datos. Se realiza el análisis de la información utilizando el análisis exploratorio. El cual se caracteriza por un enfoque inductivo que facilita el descubrimiento de patrones en base a métodos o técnicas.
7. Visualización de datos. Se dedica a usar técnicas y herramientas de visualización de datos para comunicar gráficamente los resultados del análisis.
8. Utilización de resultados del análisis. Se determina cómo y dónde pueden emplearse los datos procesados del análisis en el caso de estudio.

Bases de datos:

El Ciclo de Vida de la Analítica de Big Data será aplicado en el presente proyecto de investigación sobre las bases de datos químico-biológicas mencionadas a continuación:

- *Kyoto Encyclopedia of Genes and Genomes* (KEGG)

Un recurso integrado de bases de datos para comprender funciones de alto nivel y la utilidad del sistema biológico como la célula, el organismo y el ecosistema, a partir de información a nivel molecular generada mediante la secuenciación del genoma y tecnologías experimentales de alto rendimiento [17].

- *Universal Protein Resources* (UniProt)

Una colección de base de datos que contiene información sobre la secuencia y anotaciones asociadas a las proteínas. Las anotaciones consisten en el análisis, la comparación y la fusión de todas las secuencias disponibles para una proteína determinada, así como una revisión crítica de los datos experimentales y pronósticos asociados [18]. Para ello, se extrae información biológica de la literatura y realizan numerosos análisis computacionales para proporcionar la información relevante conocida sobre una proteína en particular. En ella se describen, en un solo registro, los diferentes productos proteínicos derivados de un

determinado gen de una especie específica. UniProt se estructura en dos secciones: Swiss-Prot y TrEMBL. La primera se caracteriza por la revisión de los resultados experimentales, características computacionales y conclusiones científicas; por otro lado, TrEMBL comprende las anotaciones de proteínas, pero sin revisión de los datos [19].

- ChEMBL

Una base de datos abierta que contiene información acerca de compuestos bioactivos, es decir compuestos que tienen actividad biológica dentro del organismo. Además, la información de publicaciones científicas es ingresada manualmente. Dicha información se encuentra limpia y estandarizada, lo cual es un criterio que maximiza su calidad y utilidad en la resolución de problemas de investigación biológica y en el descubrimiento de fármacos. La información se encuentra publicada y se puede acceder a ella mediante una interfaz web [20].

- ChEBI

ChEBI es una base de datos de entidades químicas de interés biológico [21] que almacena información sobre productos de la naturaleza o productos sintéticos utilizados para intervenir en los procesos de los organismos vivos [22]. La base de datos ChEBI combina nomenclatura química, estructuras, sinónimos e información química relacionada a los compuestos químicos para proporcionar una amplia gama de datos relacionados como fórmulas, enlaces a otras bases de datos y una clasificación ontológica, por la cual se especifican las relaciones padre-hijo entre entidades moleculares [23].

Target Fishing

Target Fishing in silico es una técnica usada para la experimentación biológica computacional que permite la predicción de los blancos de compuestos químicos en base a su estructura química, usando información disponible en bases de datos químico-biológicas [14]. Esta técnica se utiliza para encontrar blancos no relacionados [12].

Target Fishing permite encontrar blancos mediante el cálculo de similaridad entre los identificadores *SMILES* de compuestos químicos. El cálculo de similaridad se realiza considerando el coeficiente de distancia Jaccard-Tanimoto o Similaridad de Tanimoto entre las representaciones en vectores binarios de las moléculas, conocidas también como el *fingerprint* de los compuestos químicos.

Simplified Molecular Input Line Entry System (*SMILES*) es una notación química que se utiliza para traducir la estructura tridimensional de un producto químico en una cadena de símbolos que el software de computadora entiende fácilmente [24]. Para ello, sigue cinco reglas que detallan la representación de los átomos, relaciones, cadenas simples, árboles de cadenas y estructura de anillos [25]. Además, cabe recalcar que existen dos tipos: *SMILES* genérico y *SMILES* canónico. El primero representa todas las posibilidades de una estructura, pero el segundo emplea un algoritmo de canonización que lo convierte en un *SMILES* único. Por ejemplo: un *SMILES* genérico representaría al etanol de las siguientes formas: OCC, [CH3][CH2][OH], C-C-O, C(O)C; sin embargo, al aplicar las reglas definidas en el algoritmo se obtendría para todos los casos, un *SMILES* igual a CCO [26].

Similaridad de Tanimoto es un método que calcula la similitud entre dos estructuras químicas mediante dos componentes principales de la búsqueda: representación de la estructura y el coeficiente de similitud. Para ello, realiza una representación de cadena de bits en el que cada entrada corresponde a alguna característica estructural; el valor de una entrada está representado por uno solo si la característica correspondiente está presente en la estructura, caso contrario se representa por un cero [27] [28]. El coeficiente de Tanimoto o coeficiente de Jaccard es ampliamente usado en sistemas de software para la gestión de información química [29]. La Similaridad de Tanimoto de dos moléculas se calcula mediante la Ecuación 1.1, donde A y B son moléculas, a y b el número de unos de su representación en bits, también denominados *fingerprint*, y c los bits igual a 1 que son comunes entre a y b.

$$S_{AB} = \frac{c}{a + b - c}$$

Ecuación 1.1. Similaridad de Tanimoto

Fingerprint es el término usado para referirse a la representación de moléculas en vectores binarios; y mediante esta representación es posible identificar la presencia o ausencia de características o subestructuras particulares. Así, dadas dos moléculas y sus *fingerprint* comprimidas a una longitud específica (número de bits), se puede obtener su similitud [30], con diferentes métodos, tales como: Similaridad de Tanimoto, Coeficiente Euclidiano, Coeficiente Russell-Rao, Coeficiente Forbes, entre otros [31].

RDKIT es un conjunto de herramientas usadas en quimio-informática. Sus módulos pueden ser importados en programas Python. Dichas herramientas permiten realizar operaciones moleculares 2D y 3D, entre ellas se encuentran la Similaridad de Tanimoto de *SMILES* genéricos y *SMILES* Canónicos [32] mediante funciones como *FingerprintSimilarity*.

La distribución hipergeométrica de probabilidad se usa para calcular la probabilidad de obtener un determinado número de éxitos sobre un espacio muestral N [33]. Para ello, se considera la Ecuación 1.2, la que está definida por cuatro variables:

$$P = \frac{\left(\frac{k}{m}\right)\left(\frac{N-k}{x-m}\right)}{\left(\frac{N}{x}\right)}$$

Ecuación 1.2. Distribución hipergeométrica de probabilidad

N = Población

x = Muestra

k = Número de éxitos en la población N

m = Número de éxitos de la muestra

2. METODOLOGÍA

El procedimiento utilizado para la obtención y análisis de los datos se enmarca en las etapas del Ciclo de Vida de la Analítica de Big Data. Tal como se explica en el apartado 1.5 del presente documento.

Las etapas de la metodología incluyen: Identificación de datos, Adquisición y filtrado de datos, Extracción de datos, Validación y limpieza de datos, Agregación y representación de datos, Análisis de datos, Visualización de datos, por último, Utilización de resultados del análisis.

2.1. Identificación de datos

La etapa inicial del Ciclo de Vida de la Analítica de Big Data comprende la identificación de las fuentes y formatos de información que serán utilizadas en la investigación. En este caso, las principales fuentes de información son bases de datos químico-biológicas como *KEGG Pathway*, UniProt, ChEMBL y ChEBI.

KEGG Pathway

El acceso a la base de datos *KEGG Pathway* se realiza mediante servicios Web publicados en una interfaz de programación de aplicaciones tipo *REST*[1] que contiene las peticiones para el acceso a la información en diferentes formatos como imágenes, *xml*, *json*, texto plano, y otros. La información de la base de datos *KEGG Pathway* utilizada para el presente trabajo corresponde al *release* publicado hasta el 29 de septiembre del 2017, donde la información fundamental sobre rutas metabólicas se encuentra en formato de *xml* y para compuestos químicos en texto plano; como se muestra en la Figura 2.1.

[1] https://www.kegg.jp/kegg/rest/keggapi.html

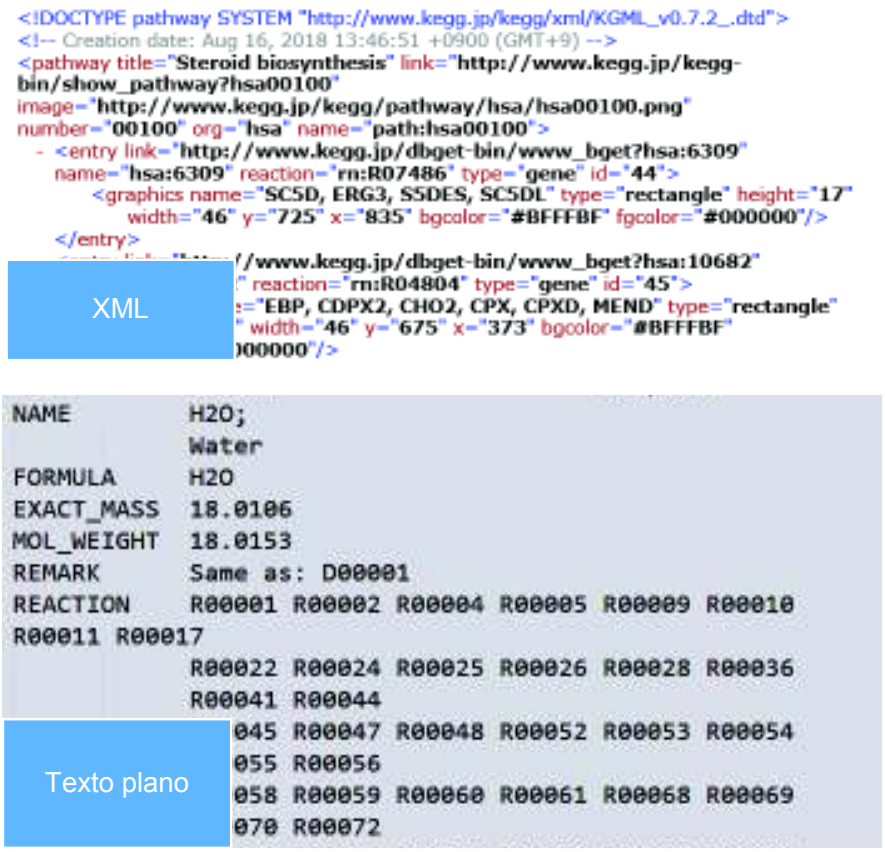

Figura 2.1. Formatos de información de la base de datos KEGG *Pathway*

UniProt

El acceso a la base de datos UniProt se realiza mediante la opción de descarga de la página web[2], donde se puede encontrar la información revisada de proteínas (Swiss-Prot) en texto plano, como se muestra en la Figura 2.2. La versión de la base de datos UniProt usada en el presente trabajo de investigación es la publicada el 27 de septiembre del 2017.

[2] https://www.uniprot.org/downloads

```
DR   PIRSF; PIRSF004030; Z_ArenaV; 1.
PE   3: Inferred from homology;
KW   Complete proteome; Host cell membrane; Host cytoplasm; Host membrane;
KW   Host-virus interaction;
KW   Inhibition of host innate immune response by virus;
KW   Inhibition of host RIG-I by virus;
KW   Inhibition of host RLR pathway by virus; Lipoprotein; Membrane;
KW   Metal-binding; Myristate; Viral budding;
KW   Viral budding via the host ESCRT complexes; Viral immunoevasion;
KW   Viral release from host cell; Virion; Zinc; Zinc-finger.
FT   INIT_MET      1      1       Removed; by host. {ECO:0000250}.
FT   CHAIN         2     95       RING finger protein Z.
FT                              /FTId=PRO_0000361042.
FT   ZN_FING      38     74       RING-type; atypical.
FT   MOTIF        88     91       PTAP/PSAP motif.
                   2      2       N-myristoyl glycine; by host.
                                  {ECO:0000250}.
 AA;  10916 MW;  D60C0C51C3619E42 CRC64;
YGDREPEG RIPITLNMPQ SLYGRYNCKS CWFANKGLLK CSNHYLCLKC
ICGEVLPK KLVFENSPSA PPYEA
```

Texto plano

Figura 2.2. Formato de información de base de datos UniProt

ChEMBL

Se utiliza la base de datos ChEMBL versión 22 simplificada, la cual se obtuvo del trabajo de titulación [34] y está en formato SQL. Éste contiene las tablas: *compound, compound_statistics, data_experimental, protein* y *protein_statistics*; como se muestra en la Figura 2.3. Además, se obtendrá información de compuestos químicos del servicio web de ChEMBL[3].

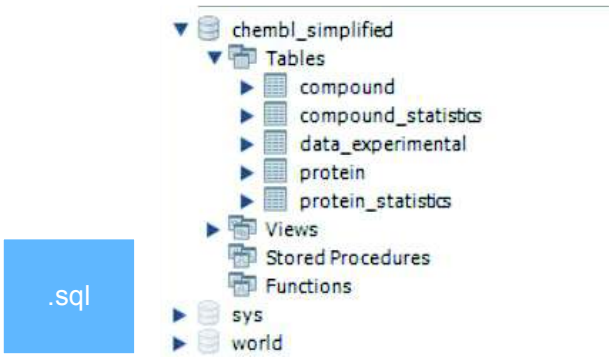

Figura 2.3. Tablas de base de datos ChEMBL

[3] https://www.ebi.ac.uk/chembl/ws

ChEBI

Se utiliza la base de datos ChEBI para obtener la información de compuestos químicos con formato *XML*, como se muestra en la Figura 2.4.


```
<TABLE>
  <ENTRY>
    <ID>CHEBI:65312</ID>
    <NAME>corticotropin-releasing hormone (human)</NAME>
    <DEFINITION>
      A corticotropin-releasing hormone from human/rat composed of Ser, Glu,
      Glu, Pro, Pro, Ile, Ser, Leu, Asp, Leu, Val, Phe, His, Leu, Leu, Arg,
      Glu, Val, Leu, Glu, Met, Ala, Arg, Ala, Glu, Gln, Leu, Ala, Gln, Gln,
      Ala, His, Ser, Asn, Arg, Lys, Leu, Met, Glu, Ile and Ile-NH2 residues
      joined in sequence.
    </DEFINITION>
    <STAR>3</STAR>
    <SYNONYM>
      Ser-Glu-Glu-Pro-Pro-Ile-Ser-Leu-Asp-Leu-Thr-Phe-His-Leu-Leu-Arg-Glu-Val-
      Leu-Glu-Met-Ala-Arg-Ala-Glu-Gln-Leu-Ala-Gln-Gln-Ala-His-Ser-Asn-Arg-Lys-
      Leu-Met-Glu-Ile-Ile-NH2
    </SYNONYM>
    <SYNONYM>SEEPPISLDLTFHLLREVLEMARAEQLAQQAHSNRKLMEII-NH2</SYNONYM>
    <SYNONYM>human/rat CRF</SYNONYM>
    <SYNONYM>CRH</SYNONYM>
    otropin-releasing factor (human/rat)</SYNONYM>
    otropin-releasing factor (human)</SYNONYM>
    otropin releasing hormone</SYNONYM>
    oliberin</SYNONYM>
```


XML

Figura 2.4. Formato de información de base de datos ChEBI

2.2. Adquisición, extracción y limpieza de datos

Esta etapa comprende las tres fases del Ciclo de Vida de la Analítica de Big Data: Adquisición y filtrado de datos, Extracción de datos, Validación y limpieza de datos. Cabe recalcar que la unificación de ellas se debe a la necesidad de relacionar los subconjuntos de datos y eliminar datos innecesarios o irrelevantes. Las bases de datos sobre las cuales se aplica esta etapa son las siguientes:

KEGG Pathway

KEGG Markup Language (KGML) es una representación en formato *xml* de los gráficos de las rutas metabólicas de KEGG [35]. Los KGML son grafos que especifican a los nodos como entradas, las cuales pueden ser genes, compuestos químicos y otras rutas metabólicas; y a los enlaces como relaciones y reacciones. Los elementos de relación y reacción indican los patrones de conexión entre productos genéticos y compuestos químicos.

La obtención de los KGML se efectúa mediante el código descrito en el ANEXO I, el cual utiliza las librerías *requests* y *csv* para ejecutar las peticiones hacia la API de KEGG, realizar la copia, la lectura y la escritura de archivos. Por tanto, a partir de la lista de rutas metabólicas del ser humano se procede a la descarga de archivos KGML identificados por el formato de nombre formado de la siguiente manera: <identificador del organismo><número de ruta metabólica>. Estos archivos permitirán extraer la información de genes y compuestos químicos.

A partir de los archivos de las rutas metabólicas se obtiene el listado de los genes y compuestos mediante el código en el ANEXO II. Para ello, se usa la librería *xml.etree.cElementTree* que permite recorrer jerárquicamente los archivos *xml* y adquirir los datos de las etiquetas. Para este caso, se necesita la información de la etiqueta *entry*.

La etiqueta *entry* contiene información acerca de un nodo en la ruta metabólica, como se muestra en la Tabla 2.1, donde se detallan los atributos y sus características [36]. Para este caso se obtendrán solo los atributos *id* y *name.*

Tabla 2.1. Datos de la etiqueta *entry*

Nombre del atributo	**Tipo de dato**	**Descripción**	**Requerido/ Implícito**
id	Id	El id del nodo en la ruta metabólica.	Requerido
name	Keggid	El KEGGID del nodo.	Requerido
type	Entry_type	El tipo del nodo.	Requerido
link	URL	La localización de los recursos e información del elemento.	Implícito
reaction	Keggid	El KEGGID correspondiente a la reacción.	Implícito

Además, es necesario tener más información acerca de los compuestos químicos que se encuentran en las rutas metabólicas. Para ello, se descarga y guarda la documentación contenida en la base de datos KEGG, en archivos de texto plano, usando la librería *requests* y *csv*; como se evidencia en el código del ANEXO III. Una vez descargados estos archivos, se procede a la lectura para obtener el nombre del compuesto químico y los identificadores de las bases de datos ChEBI y ChEMBL mediante el código en el ANEXO IV. En el cual, se usa las librerías csv y os, éste último permite la manipulación de la estructura de los directorios. Para este caso, se obtiene el listado de todos los compuestos y dentro de cada archivo se buscará el contenido que esté después de las palabras claves:

NAME que representa el nombre del compuesto, *ChEMBL* el id de ChEMBL y *ChEBI* el id de ChEBI. La búsqueda se realiza a través del recorrido de cada línea del documento, de la que se obtiene la lista de palabras de dicha línea a través del método *split().* Este método divide la cadena bajo la especificación de un separador, que para este caso será el tabulador. Por lo que, una vez que cumpla la condición de búsqueda, almacenará en las variables el elemento siguiente de la lista y en el caso de no encontrar se guardará un elemento en blanco.

ChEMBL

De la base de datos ChEBML 22 Simplificada se usa la información de las tablas: *protein*, *compound* y *data_experimental*, cuyo detalle se presenta en la Tabla 2.2.

Tabla 2.2. Información de tablas de ChEBML 22 Simplificada *[34]*

Tabla	Atributo	Descripción
protein	Tid	Identificador de la proteína.
	pref_name	Nombre preferido de la proteína.
	chembl_id	Identificador de ChEBML de la proteína.
	Accession	Número secuencial en la base de datos de origen a la que es referida.
data_experimental	data_experimental_id	Identificador del dato experimental.
	Molregno	Identificador del compuesto.
	Tid	Identificador de la proteína.
	assay_organism	Nombre del organismo de ensayo.
compound	Molregno	Identificador del compuesto.
	pref_name	Nombre preferido del compuesto.
	chembl_id	Identificador de ChEBML del compuesto.
	canonical smile	Representación de una molécula 2D.

Además, se procede a extraer los *SMILES* canónicos de los compuestos químicos de *KEGG Pathway* que tienen ChEMBLID a través de los servicios web de ChEMBL. Para ello, se utiliza el código indicado en el ANEXO V, en el cual se realiza la lectura del documento de compuestos químicos de la base de datos *KEGG Pathway* y se obtiene la lista de elementos mediante el método *split().* Para cada elemento, en el caso de existir información en la columna que referencia al identificador de ChEMBL, se procede a extraer el *SMILES* mediante el método *get()* de la librería *requests*. El método *get()* se conforma por tres componentes: la URL del servicio web, el parámetro para realizar la petición y el

formato de descarga de la información. Por ejemplo, la URL del servicio web de ChEBML es https://www.ebi.ac.uk/chembl/api/data/molecule/, el parámetro de petición corresponde al identificador del compuesto químico C00068 y el formato de descarga será *json,* como se muestra a continuación: *'https://www.ebi.ac.uk/chembl/api/data/molecule/" +C00068+'.json'.* Para este caso, se obtiene el texto en formato *json*, el que es decodificado, es decir se convierte en un objeto de tipo diccionario, a través del método *loads()*. De este objeto se busca los elementos con las etiquetas < *molecule_structures*> < *canonical_smiles*> y < *molecule_chembl_id* >, los que llenan el arreglo denominado *resultados* y posteriormente se almacenan en un archivo.

ChEBI

De la base de datos ChEBI se obtiene la información de compuestos químicos en formato *xml*. La información obtenida de cada *xml* se detalla en la Tabla 2.3.

Tabla 2.3. Información del formato *xml [22]*

Identificador	**Descripción**
ID	Identificador de ChEBI.
NAME	Nombre del compuesto químico.
DEFINITION	Determina la composición del compuesto químico.
SYNONYM	Especifica otros nombres del compuesto químico.
IUPAC_NAME	Nomenclatura del compuesto químico.
FORMULA	Fórmula molecular del compuesto químico.
MASS	Porcentaje de masa.
CHARGE	Carga molecular del compuesto químico.
SMILES	Representación gráfica de una estructura molecular.
INCHIKEY	Identificador internacional del compuesto químico.
REAXYS_REGISTRY_NUMBER	Secuencial de registros de compuestos químicos.
KEGG_COMPOUND_ACCESSION	Número secuencial en la base de datos KEGG.
WIKIPEDIA_ACCESSION	Nombre de identificación en la base de datos de Wikipedia.
PUBMED_CITATION	Identificador de la citación o del artículo de investigación de base de datos MEDLINE.

La información que se extrae de esta base de datos corresponde a los *SMILES* genéricos de los compuestos químicos de *KEGG Pathway* que no tienen un identificador ChEMBLID,

pero sí cuentan con el identificador de la base de datos ChEBI. La descarga se la realiza mediante el código en el ANEXO V el cual utiliza la librería *urllib.requests* y *urlretrive* para generar la petición y guardar el archivo en formato *xml*. La librería *urlretrive* recibe dos argumentos, el primero es la URL para la petición y el segundo, el directorio local en el que se guardarán los archivos.

Cabe mencionar que no todos los archivos descargados poseen una estructura correcta, porque no siguen una de las principales reglas de sintaxis de *xml*: *Todos los elementos del archivo xml deben tener una etiqueta de cierre* [37], por lo que es necesario tratarlos como texto plano y no como un documento *xml*. A continuación, para extraer los *SMILES* de la información de compuestos químicos de ChEBI, se usa el código del ANEXO VI, el cual importa la librería *os* y *csv*. Mediante este código se busca la etiqueta <*SMILES*> y se adquiere su contenido, para luego guardar esta información en un archivo que contendrá los identificadores de *KEGG Pathway* y ChEBI, además del respectivo *SMILES* genérico.

UniProt

La información de la base de datos UniProt se adquiere mediante la descarga del archivo *uniprot_sprot.dat*[4] de la opción *Reviewed (Swiss-Prot)* presente en la página web. El archivo se encuentra estructurado por un conjunto de líneas que empiezan con un identificador de dos caracteres, el cual es utilizado para distinguir el tipo de dato respecto a la proteína y a continuación el dato respectivo, como se muestra en la Tabla 2.4.

Tabla 2.4. Estructura de archivo *uniprot_sprot.dat*[5]

Identificador	**Descripción**
ID	Identificador de la proteína en la base de datos UniProt.
AC	Número de *accession*.
DT	Día de la última actualización.
DE	Descripción de la proteína.
GN	Identificador del gen.
GO	Especie del organismo.
OG	Organelo.
OC	Clasificación del organismo.
RN	Número de referencia.
RP	Posición de referencia.

[4] ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/idmapping/by_organism/
[5] http://athina.biol.uoa.gr/FT/format_SwissProt.html

RC	Comentario de referencia.
RX	Referencias cruzadas.
RA	Autores de referencia.
RL	Localización de referencia.
CC	Comentarios o notas.
DR	Referencia con otras bases de datos.
KW	Palabras clave.
SQ	Encabezado de la secuencia.
(Espacio en blanco)	Terminación de una línea.
FT	Tabla de características de los datos.
//	Separación entre proteínas.

A partir de este archivo se procede a extraer, de cada proteína, el contenido de las siguientes etiquetas: *ID, GN, AC* y GO. Para ello, se utiliza el código en el ANEXO VII, en el cual se implementa el módulo *MySQLdb*. Este módulo se usa para establecer la conexión con el sistema de gestión de bases de datos MySQL mediante el método *connect*(); el que necesita como parámetros: el *host,* la contraseña, el usuario y el nombre de la base de datos. Además, con este método, se crea la tabla EVOLUNIPROTDB con los campos UNIPROTID, GENEID, UNIPROTAC y GO a través de una sentencia MySQL. La sentencia de creación de tablas está almacenada en la variable *TABLES* y es ejecutada por el método *execute().* Una vez creada la tabla, se procede a la lectura de cada línea del documento y a la separación de los elementos de dicha línea, mediante el método *split()*. El primer elemento se comparará con las etiquetas mencionadas previamente, por lo que, si la condición es verdadera, se obtiene la información del siguiente elemento y se añade al arreglo *TEMP.* Una vez que se ha llenado el arreglo *TEMP* por cada proteína, se realizará la inserción en la tabla únicamente si el campo GO corresponde a *Homo sapiens (Human).* Cabe recalcar, que se distingue la información entre proteínas cuando se valida la presencia del identificador: //.

2.3. Agregación y representación de datos

La siguiente etapa consiste en identificar la información de los diferentes conjuntos de datos obtenidos previamente para relacionarlos mediante los campos comunes. Las relaciones encontradas fueron las siguientes: *KEGG Pathway* – UniProt, conjunto de datos de proteínas – ChEMBL, *KEGG Pathway* – ChEMBL / ChEBI.

KEGG Pathway – UniProt

La primera relación encontrada es entre los genes contenidos en la base de datos *KEGG Pathway* y los que genes se encuentran en UniProt, a través del campo común *GENEID,* como se indica en la Figura 2.5.

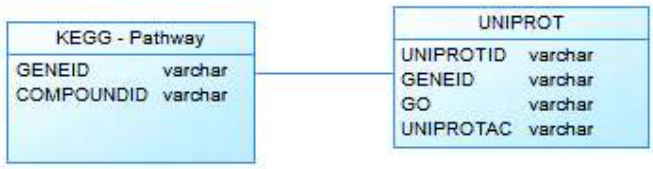

Figura 2.5. Relación entre base de datos KEGG *Pathway* y UniProt

A partir de esta relación, se procede a extraer el subconjunto de información de las proteínas de UniProt correspondientes a los genes de *KEGG Pathway,* como se muestra en la Figura 2.6.

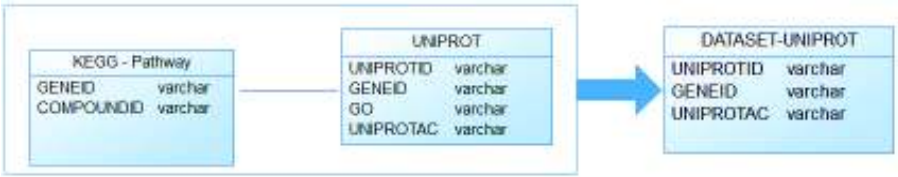

Figura 2.6. Conjunto de datos de proteínas

El conjunto de datos de proteínas, mostrado en la Figura 2.6, se obtiene mediante el código del Anexo ANEXO VIII. En el cual, se realiza la lectura y extracción de los GENEID presentes en la lista de genes de *KEGG Pathway*, cuyos elementos se almacenan en el arreglo *genesList*. A continuación, se establece la conexión con la base de datos EVOLUNIPROT y se define la sentencia de selección que obtiene el UNIPROTID, UNIPROTAC y GENEID para cada uno de los elementos que estén en el arreglo. Por último, mediante la librería *csv* y su método *write(),* se guarda la información mencionada previamente en un archivo.

UniProt – ChEMBL

La segunda relación identificada es entre las proteínas de ChEMBL Simplificada y el conjunto de datos de proteínas, mediante el campo *accession* y UNIPROTAC, respectivamente; como se muestra en la Figura 2.7

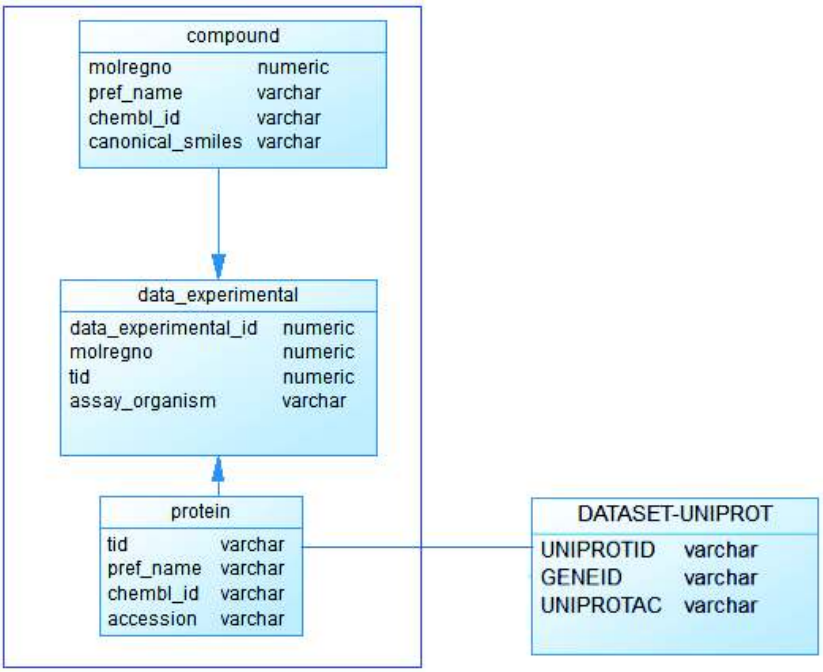

Figura 2.7. Relación entre base de datos ChEMBL Simplificada y *Dataset-UniProt*

Los blancos se definen como proteínas concretas sobre las cuales actúa un compuesto químico [38]. Debido a la relación existente entre proteína y compuesto químico presente en la tabla de datos experimentales en ChEMBL Simplificada, se considera a las proteínas como blancos.

A partir de la segunda relación, se extrae los *SMILES* canónicos correspondientes a los compuestos químicos que tienen interacción con un blanco en particular mediante el código en el ANEXO IX. En el cual, se realiza lectura y extracción del UNIPROTID presente en el *dataset* de proteínas, cuyos elementos se almacenan en el arreglo *proteinsIds.* A continuación, se establece la conexión con la base de datos *chembl_simplified* y se define la sentencia de selección que obtiene el TID, PREF_NAME, ChEMBLID y ACCESSION del blanco, para cada uno de los elementos de *proteinsIds*. Toda esta información se almacena en otro arreglo denominado *proteinsChembl.*

A continuación, para cada elemento del arreglo *proteinsChembl,* se crea un archivo nombrado con el identificador del blanco, en el cual almacena la siguiente información:

1. La primera línea contiene: el identificador del blanco, el identificador del gen, el tipo de blanco, el nombre y el número de *accession*.

2. La lista de *SMILES* canónicos de la tabla de compuestos donde el *molregno* se encuentra en la tabla de datos experimentales para el blanco.

KEGG Pathway - ChEMBL / ChEBI

La tercera relación reconocida es entre la base de datos *KEGG Pathway* y la tabla *compound* de la base de datos ChEMBL, a través del campo común CHEMBLID y chembl_id, respectivamente; y la base de datos *KEGG Pathway* y la base de datos de compuestos químicos de ChEBI, mediante el campo común CHEBIID; como se muestra en la Figura 2.8.

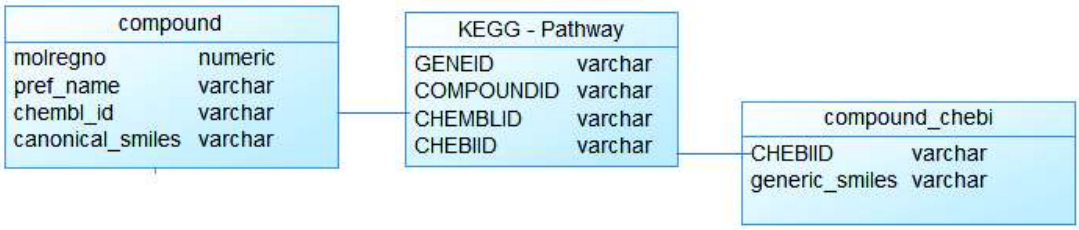

Figura 2.8. Relación entre base de datos KEGG *Pathway* y ChEMBL / ChEBI

2.4. Análisis de datos

La fase de análisis de datos consiste en identificar posibles blancos para los compuestos químicos de *KEGG Pathway*. Para ello, se usa la técnica de *Target Fishing* mencionada en el apartado 1.5, la cual tiene como datos de entrada y datos base la información relacionada en el apartado 2.3: el *dataset* de proteínas con los *SMILES* de los compuestos químicos; y el *SMILES* genérico o *SMILES* canónico del compuesto químico de *KEGG Pathway,* representados por la segunda y tercera relación; respectivamente; como se muestra en la Figura 2.9.

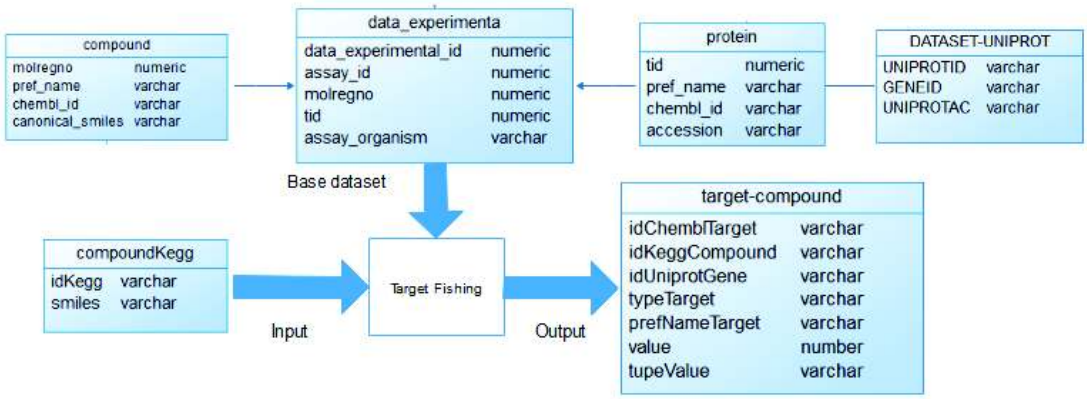

Figura 2.9. Entrada y salida de datos *Target Fishing*

El proceso empleado para el *Target Fishing* comprende los siguientes pasos, los cuales están ilustrados en la Figura 2.10:

1. Se obtiene el *fingerprint* de los *SMILES* de los datos de entrada y datos base.
2. Para cada elemento del *dataset* de entrada se realiza el cálculo de similaridad con todos los elementos del *dataset* base a través del método de Similaridad Tanimoto.
3. Se evalúa el resultado de similaridad en base a los valores *max* y *c3*. Estos valores representan el porcentaje de similitud entre las estructuras moleculares de los compuestos químicos. Ya que, moléculas con estructuras semejantes tienen una actividad similar, se propuso que los valores de corte sean 0.9 para el *max* y 0.55 para *c3* con la finalidad de eliminar los falsos positivos [39] [40].
4. Se realiza el almacenamiento de cada compuesto químico del *dataset* de entrada con sus blancos resultantes del *Target Fishing*.

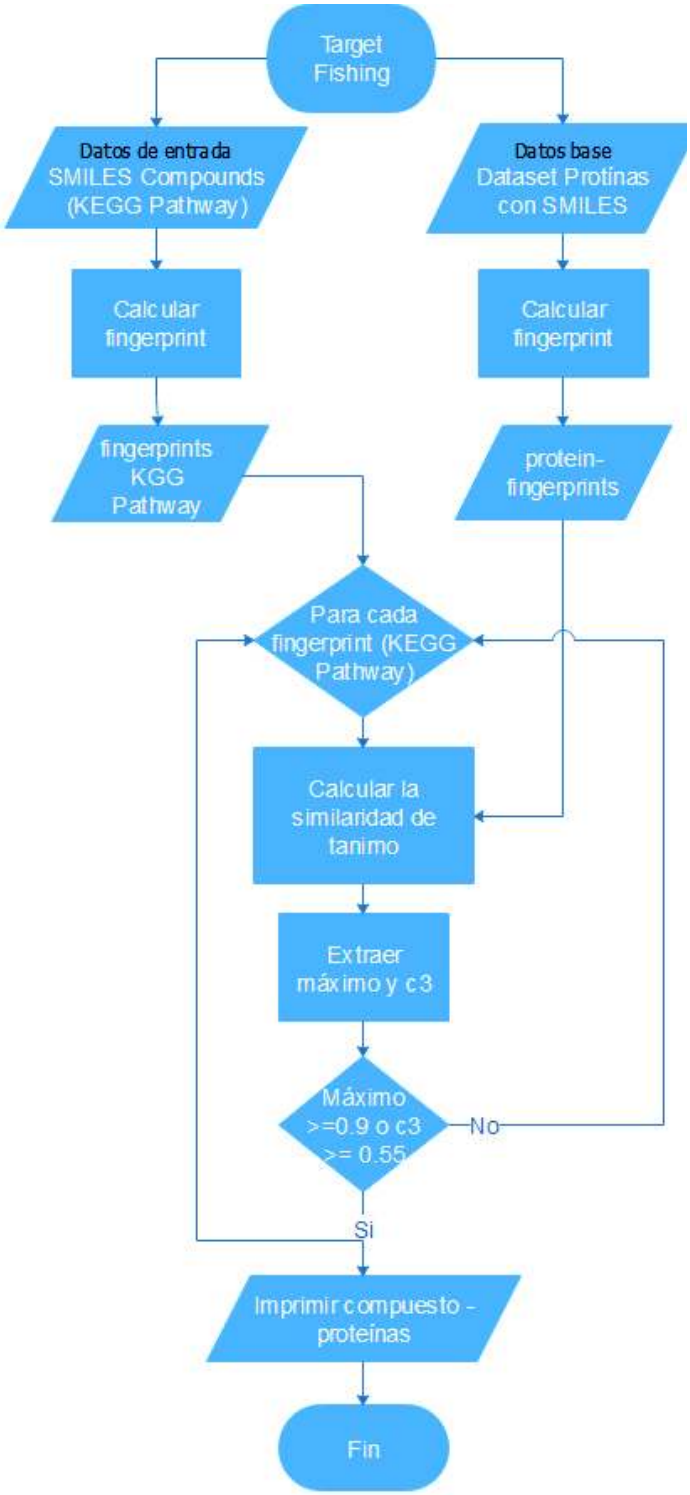

Figura 2.10. Proceso de *Target Fishing*

El código en el ANEXO X implementa el proceso del *Target Fishing*, en el cual se importan las librerías de RDKIT para la obtención del *fingerprint* y el cálculo de la Similaridad de Tanimoto. El tamaño del *fingerprint* tiene como valor mínimo 64 y como máximo 2048 bits. Para el presente proyecto se utilizará el valor por defecto que corresponde a 2048 bits. A continuación, se carga a memoria toda la información de las proteínas y compuestos, que serán representados por los objetos: Proteínas y Compuestos, respectivamente. A este último, también se le asigna el *fingerprint* como parte del objeto.

Después, se procede con el cálculo de la Similaridad de Tanimoto para cada *fingerprint* del compuesto químico de *KEGG Pathway* con todos los *fingerprint* de los compuestos químicos de cada blanco mediante el módulo de RDKIT *DataStructs*. Este módulo contiene una variedad de funcionalidades para estructuras de datos básicas, entre ellas la función *FingerprintSimilarity (a,b,metric=DataStructs.TanimotoSimilarity).* Esta función devuelve la similitud calculada entre dos *fingerprint* a partir de la métrica escogida para dicho proceso, en este caso *TanimotoSimilarity*.

Una vez finalizado el proceso, se carga esta información en una lista de valores y se la ordena de mayor a menor para así identificar el *max* y el *c3*. Los valores usados como *max* y *c3* son 0.9 y 0.55, respectivamente. El *max* corresponde al primer elemento de la lista ordenada y el *c3* representa el promedio entre los tres primeros elementos. Para ello, se utiliza la librería *nlargest*, la cual devuelve para este caso, los tres valores más grandes del conjunto de datos. A continuación, se guardará los datos de *c3* y max, si c3 es mayor o igual a 0.55 o si max es mayor o igual a 0.9; y si no cumple alguna de estas condiciones, se descarta el blanco. Finalmente, se guarda la información obtenida, la que comprende el compuesto de *KEGG Pathway* con los blancos que resultan del análisis.

Para la ejecución del código del ANEXO X, mencionado anteriormente, se utiliza multiprocesamiento con 4 núcleos. Los procesos son asignados mediante una función que reparte el procesamiento de la información correspondiente a los compuestos en un número de procesos asignados para el número de núcleos de procesamiento disponibles. En ella, se asigna equitativamente el conjunto de datos a ser procesados y en el caso de no obtener una división par, se asigna la diferencia al último núcleo.

Por último, mediante el código en el ANEXO XI se reordena la información obtenida para almacenar el blanco con los respectivos compuestos de *KEGG Pathway.*

2.5. Visualización de datos

La visualización de la información obtenida se realiza a través de una página web creada en base al patrón Modelo-Vista-Controlador. El patrón se caracteriza por tratar los componentes por separado, es decir, que se construyen independientemente y unifican su funcionalidad en tiempo de ejecución [41]. Para implementar el patrón se realiza la aplicación usando los marcos de trabajo para la creación de aplicaciones Web: *Flask* y *Angular*.

- *Flask*

Flask es un framework para el desarrollo de aplicaciones Web, escrito en Python. *Flask* pretende mantener un núcleo simple pero extensible, de esta manera no tiene funcionalidades que no se usen y todo depende del programador, ya que se pueden incluir las diversas librerías que existen según la necesidad [42].

- *Angular*

Es una plataforma y un marco de trabajo para crear aplicaciones en *HTML* y *TypeScript*. Angular implementa plantillas declarativas, dependencia, sincronización automática entre el modelo y los componentes de la vista para reducir la cantidad de código. Además, crea aplicaciones de una sola página para que el usuario tenga una experiencia más fluida [43].

Metabolic Pathway Enrichment

La aplicación consta de dos módulos:

1. *Chemical Compounds*

La pantalla mostrada en la Figura 2.11 presenta la lista de compuestos químicos de *KEGG Pathway*, en la cual se detalla las siguientes columnas:

- *Has prediction:* El compuesto químico con información agregada se muestra con un visto en la parte gráfica y se filtra por la palabra *true*, caso contrario se observa el campo en blanco y se filtra por la palabra *false.*
- *KEGG Name:* Nombre conocido por la base de datos *KEGG Pathway.*
- *ChEMBL/ChEBI ID:* Identificador del compuesto químico en las bases de datos ChEMBL y ChEBI.
- *Name:* Alternativa del nombre del compuesto químico.

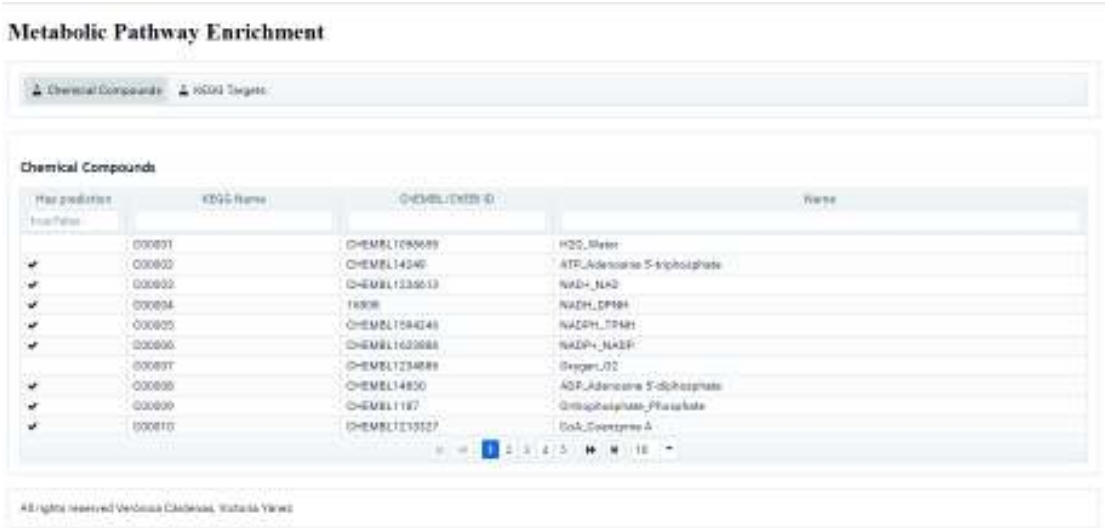

Figura 2.11. Lista de compuestos químicos de *KEGG Pathway*

Al seleccionar un registro se mostrará la lista de *KEGG targets* y las rutas metabólicas enriquecidas, como se muestra en la Figura 2.12 y Figura 2.13, respectivamente. Las figuras representan información para el compuesto C00002.

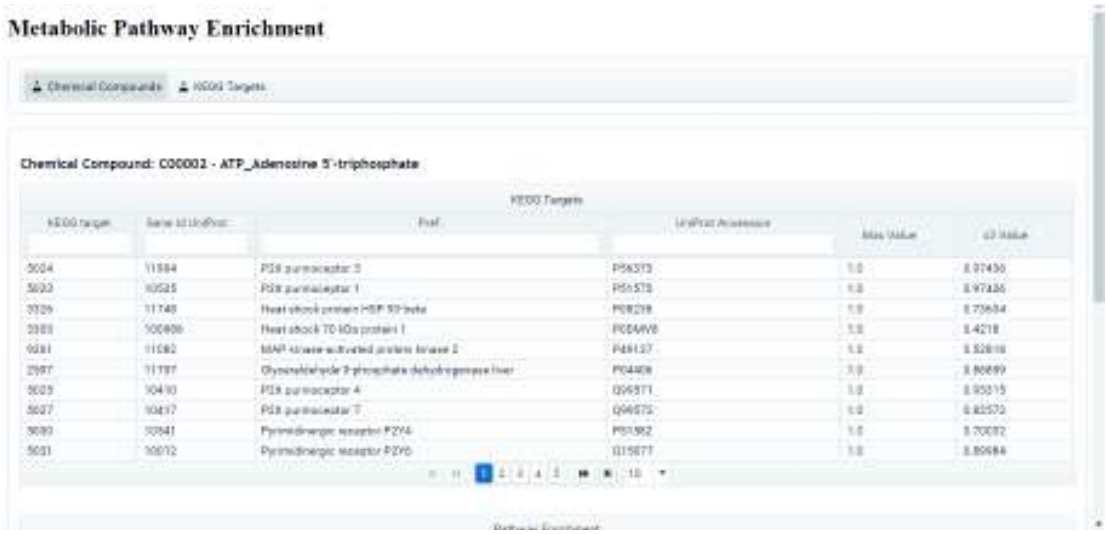

Figura 2.12. Blancos terapéuticos para el compuesto químico C00002

En la Figura 2.12 se detalla la tabla *KEGG targets* con las siguientes columnas:

- *KEGG Target:* Identificador de la base de datos ChEMBL.
- *Gene Id UniProt:* Identificador de la base de datos *KEGG Pathway* y UniProt.
- *Pref:* Nombre del *KEGG targets.*
- *UniProt Accession:* Número de *accession*.
- *Max Value:* Valor máximo obtenido del T*arget Fishing*.

- *c3 Value:* Valor de c3 obtenido del T*arget Fishing*.

Figura 2.13. Rutas metabólicas enriquecidas para el compuesto químico C00002

En la Figura 2.13 se detalla la tabla *Pathways Enrichment* con las siguientes columnas:

- *Pathway Id:* Identificador de la ruta metabólica en la base de datos *KEGG Pathway.*
- *Pathway name:* Nombre de la ruta metabólica en la base de datos *KEGG Pathway.*
- *pValue:* Valor de probabilidad obtenido por la distribución hipergeométrica de probabilidad.
- *Image:* Botón que redirecciona a la imagen de la ruta metabólica como se muestra en la Figura 2.14.
- *Link:* Botón que redirecciona a la página oficial de la ruta metabólica de *KEGG Pathway* como se muestra en la Figura 2.15.

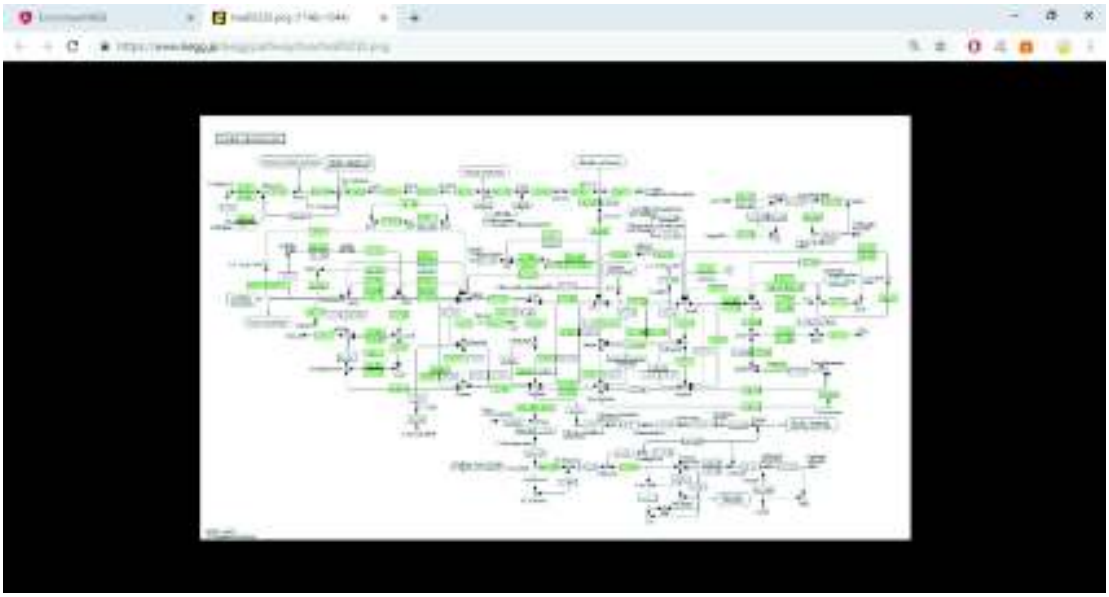

Figura 2.14. Imagen de la ruta metabólica hsa00230

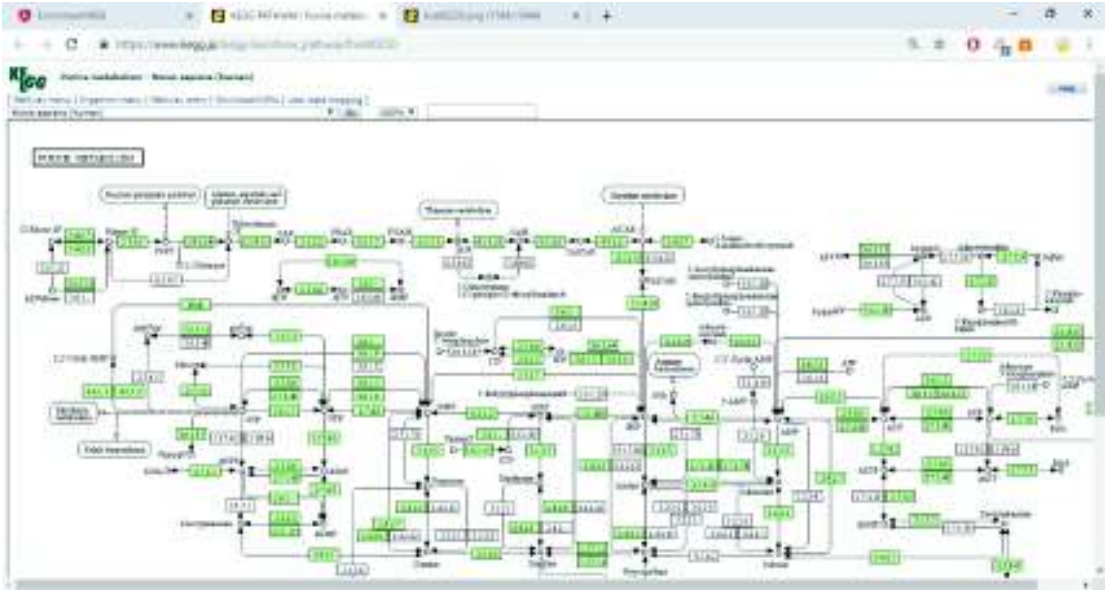

Figura 2.15. Página oficial de la ruta metabólica hsa00230

2. *KEGG targets*

La pantalla mostrada en la Figura 2.16 presenta la lista de *KEGG targets* de la base de datos *KEGG Pathway*. En la cual, se detalla las siguientes columnas:

- *Has prediction:* El compuesto químico con información agregada se representa por un visto en la parte gráfica y se filtra por la palabra *true*, caso contrario se muestra el campo en blanco y se filtra por la palabra *false.*
- *Protein ID UniProt:* Identificador de la base de datos UniProt.
- *Gene ID UniProt/ KEGG:* Identificador del *KEGG target* que se encuentra en las bases de datos UniProt y *KEGG Pathway*.
- *Name:* Nombre del *KEGG target.*
- *Accession UniProt:* Número de *accession.*

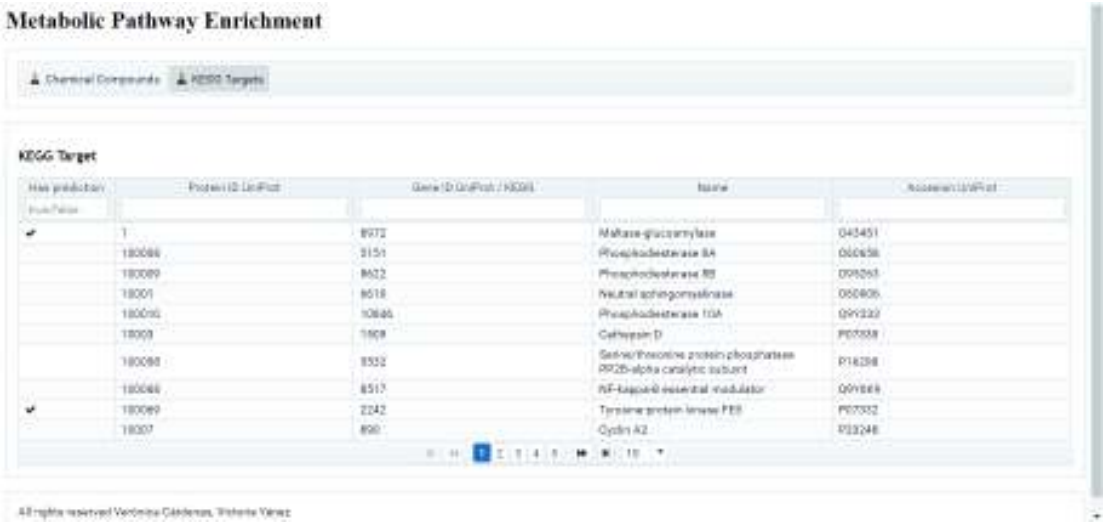

Figura 2.16. Lista de *KEGG targets*

Al seleccionar un registro se mostrará la lista de compuestos químicos como se muestra en la Figura 2.17. Las figuras representan información para el *KEGG target* 134.

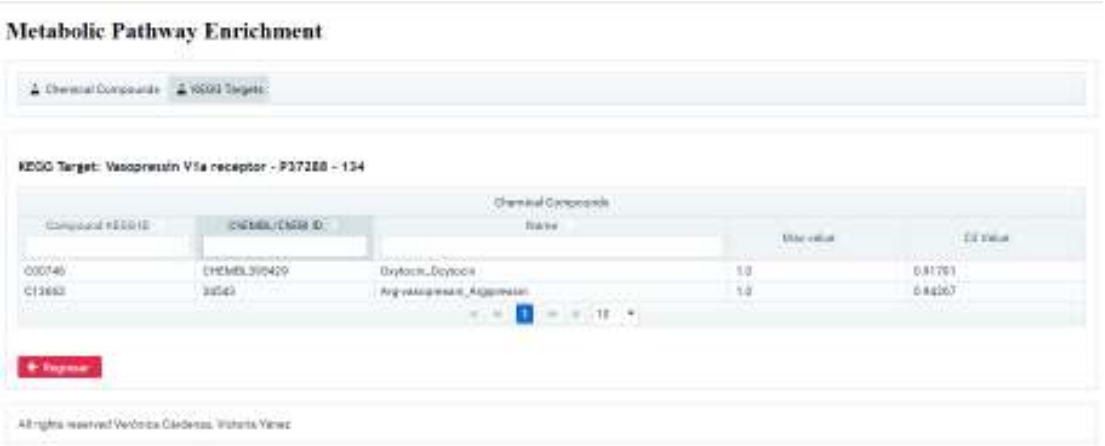

Figura 2.17. Información para el *KEGG target* 134

En la Figura 2.17 se detalla la tabla *Chemical Compounds*, la que describe la siguiente información:

- *Compound KEGG ID:* Identificador del compuesto químico en la base de datos *KEGG Pathway.*
- *ChEMBL/ChEBI ID:* Identificador del compuesto químico en las bases de datos ChEMBL y ChEBI.
- *Name:* Nombre del compuesto químico.
- *Max Value:* Valor máximo obtenido del *Target Fishing*.
- *c3 Value:* Valor de c3 obtenido del *Target Fishing*.

2.6. Utilización de resultados del análisis

La fase de utilización de resultados consiste en determinar el enriquecimiento de los compuestos químicos y los blancos de las rutas metabólicas mediante el cálculo de la distribución hipergeométrica de probabilidad descrita en el apartado 1.5. El cálculo del valor de probabilidad se define como *pvalue* y refiere a la probabilidad de encontrar resultados observables para validar que una hipótesis es verdadera [44]. Para ello, se considera que el *pvalue* debe ser igual o inferior a 0.05. Este valor es definido por varios autores como un dato estadísticamente significativo para validar resultados en un trabajo de investigación [45]. Por ello, en la interfaz web de compuestos y blancos se identifican únicamente las rutas que han cumplido con esta condición. Cabe mencionar que el valor se calcula en tiempo de ejecución e implementa la librería s*cipy.stats.* Esta librería permite realizar operaciones sobre distribución de probabilidad y para este caso, se emplea la siguiente función: *hypergeom.sf(m-1,N,x,k)*, la cual utiliza cuatro parámetros. Cada parámetro se detallará a continuación para los compuestos químicos.

Compuestos químicos

El enriquecimiento de los compuestos químicos considera las siguientes variables para el cálculo de la distribución hipergeométrica de probabilidad:

- *N* es el número de blancos del *dataset* de proteínas utilizados en el apartado 2.4, el cual equivale a un total de 1736 y se los denomina como *KEGG targets.* Los identificadores de *KEGG targets* se encuentran listados en el ANEXO XII.
- *k* es el número de *KEGG targets* que se obtuvieron del *Target Fishing* para el compuesto químico escogido.
- *x* es el número de *KEGG targets* que están en la ruta metabólica.
- *m* es el número de *KEGG targets* que se obtienen entre la intersección de *k* y *x* para cada ruta metabólica del compuesto químico escogido.

La visualización de esta información está presente en el apartado 2.5.

3. RESULTADOS Y DISCUSIÓN

3.1. Resultados

El enriquecimiento de la información de rutas metabólicas es posible gracias a la relación existente entre compuestos químicos y blancos. De acuerdo con los principios de la biología de sistemas, un compuesto químico simple que perturba una red de blancos dentro de una ruta metabólica puede desencadenar reacciones complejas y de esta manera es posible conectar datos experimentales *in vitro* con datos obtenidos por análisis *in sílico*. Estas conexiones se fundamentan en la técnica de *Target Fishing,* dado que esta permite predecir los blancos de un compuesto químico basado en el cálculo de similaridad de la estructura química y utilizando información en bases de datos químico-biológicas.

Para validar los resultados de las relaciones compuesto – ruta metabólica se establece como método a la distribución hipergeométrica de probabilidad con *pvalue* menor o igual a 0.05. Esto se debe a que el valor de probabilidad obtenido quiere validar que los resultados no sean al azar, sino que exista una relación estadísticamente significativa entre los datos empleados en el análisis [46]. Es por esto por lo que se considera calcular el *pvalue* para mostrar que rutas metabólicas fueron enriquecidas por los compuestos químicos y sus *KEGG targets.*

Compuestos químicos

Al seleccionar un compuesto químico se despliega información de las alternativas de *KEGG* targets y las rutas metabólicas enriquecidas que cumplieron con el valor establecido para el *pvalue,* como se muestran en la Figura 3.1 y Figura 3.2, respectivamente; para el compuesto químico C00002 *(Adenosine 5'-triphosphate).*

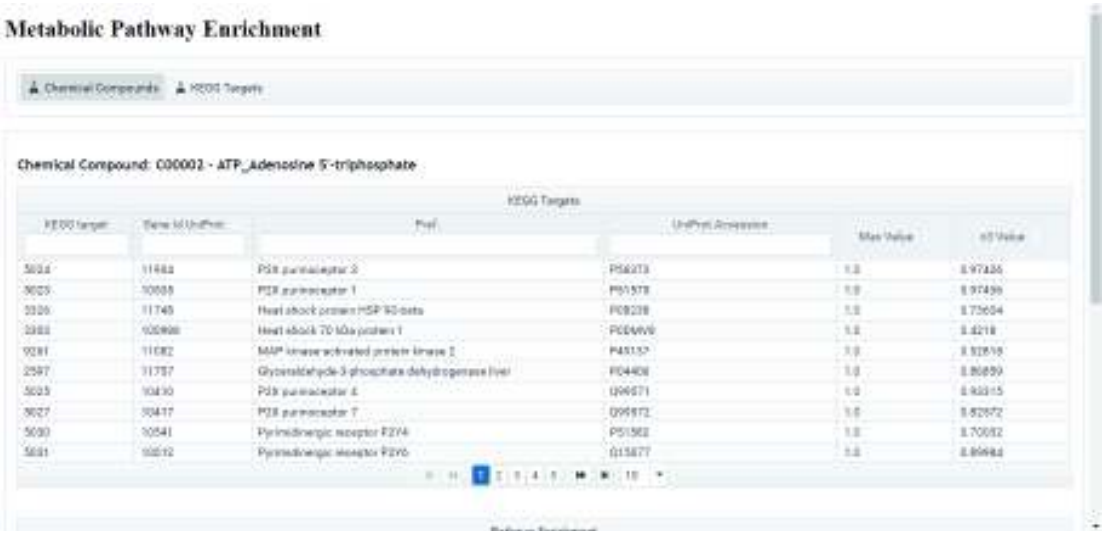

Figura 3.1. *KEGG targets* para el compuesto químico C00002

Figura 3.2. Rutas metabólicas enriquecidas del compuesto químico C00002

La Figura 3.2 detalla las rutas metabólicas que fueron enriquecidas al utilizar como método de validación la distribución hipergeométrica de probabilidad. Por ejemplo, para la ruta metabólica hsa04080 (*Neuroactive ligand-receptor interaction*), se define la siguiente información para los parámetros mencionados en el apartado 2.6:

- *N* = 1736
- *k* es el número de *KEGG targets* que se obtuvieron del *Target Fishing* para el compuesto químico C00002, cuyo valor equivale a 50 y se puede observar los identificadores en la Tabla 3.1.

Tabla 3.1. Identificadores de *KEGG targets* para *k*

124	140	1103	2272	3309	3615	5027	5032	6041	22953
132	191	1312	2527	3312	3939	5028	5167	6714	60482
134	204	1841	2597	3320	5023	5029	5331	7226	64805
135	262	2194	2710	3326	5024	5030	5335	9261	65220
136	440	2203	3303	3614	5025	5031	5631	10690	84444

- *x* es el número total de *KEGG targets* para la ruta metabólica hsa04080, el cual equivale a 219 y se señalan los identificadores en la Tabla 3.2.

Tabla 3.2. Identificadores de *KEGG targets* para *x*

117	886	1909	2831	2918	4886	5732	9002	134864
134	887	1910	2837	2925	4887	5733	9127	339403
135	1128	2147	2846	3061	4889	5734	9170	100996758
136	1129	2149	2847	3062	4923	5737	9294	
140	1131	2150	2859	3269	4985	5739	9340	
146	1132	2357	2862	3274	4986	5745	9934	
147	1133	2358	2890	3350	4987	6344	10161	
148	1134	2492	2891	3351	4988	6751	10203	
150	1136	2550	2892	3352	5021	6752	10316	
151	1137	2554	2893	3354	5023	6753	10800	
152	1139	2555	2897	3355	5024	6754	10886	
153	1141	2556	2898	3356	5025	6755	11255	
154	1144	2557	2899	3357	5026	6865	22953	
155	1145	2558	2901	3358	5027	6869	23566	
185	1241	2559	2902	3360	5028	6870	23620	
186	1268	2562	2903	3361	5029	6915	27334	
187	1269	2569	2904	3362	5030	7067	51289	
552	1395	2570	2905	3363	5031	7068	53637	
553	1511	2587	2906	3973	5032	7201	55584	
554	1812	2642	2908	4157	5340	7253	56413	
623	1813	2692	2911	4158	5618	7433	56923	
624	1814	2693	2912	4159	5644	7434	57105	
680	1815	2696	2913	4160	5645	7442	59340	
706	1816	2740	2914	4161	5646	8001	59350	
719	1901	2741	2915	4543	5724	8484	64106	
728	1902	2742	2916	4544	5729	8698	84539	
799	1903	2798	2917	4829	5731	8811	84634	

- *m* es el número de *KEGG targets* que se encuentran tanto en *k* y en *x,* el que equivale a 14 para el compuesto C00002 y la ruta metabólica hsa04080, como se muestra en la Tabla 3.3.

Tabla 3.3. identificadores de *KEGG targets* de *m*

117	886	1909	2831	2918	4886	5732	9002	134864
134	887	1910	2837	2925	4887	5733	9127	339403
135	1128	2147	2846	3061	4889	5734	9170	100996758
136	1129	2149	2847	3062	4923	5737	9294	
140	1131	2150	2859	3269	4985	5739	9340	
146	1132	2357	2862	3274	4986	5745	9934	
147	1133	2358	2890	3350	4987	6344	10161	
148	1134	2492	2891	3351	4988	6751	10203	
150	1136	2550	2892	3352	5021	6752	10316	
151	1137	2554	2893	3354	5023	6753	10800	
152	1139	2555	2897	3355	5024	6754	10886	
153	1141	2556	2898	3356	5025	6755	11255	
154	1144	2557	2899	3357	5026	6865	22953	
155	1145	2558	2901	3358	5027	6869	23566	
185	1241	2559	2902	3360	5028	6870	23620	
186	1268	2562	2903	3361	5029	6915	27334	
187	1269	2569	2904	3362	5030	7067	51289	
552	1395	2570	2905	3363	5031	7068	53637	
553	1511	2587	2906	3973	5032	7201	55584	
554	1812	2642	2908	4157	5340	7253	56413	
623	1813	2692	2911	4158	5618	7433	56923	
624	1814	2693	2912	4159	5644	7434	57105	
680	1815	2696	2913	4160	5645	7442	59340	
706	1816	2740	2914	4161	5646	8001	59350	
719	1901	2741	2915	4543	5724	8484	64106	
728	1902	2742	2916	4544	5729	8698	84539	
799	1903	2798	2917	4829	5731	8811	84634	

Al realizar el reemplazo de la función: *hypergeom.sf(13,1736,50,219)* se obtiene que el *pvalue* es igual a 0.00243, siendo este menor al valor establecido de 0.05. Por lo que, se puede afirmar que existe un enriquecimiento en la ruta metabólica hsa04080 por el compuesto C00002 y sus respectivos *KEGG targets*.

3.2. Discusión

Si bien dos de los objetivos descritos en el apartado 1.3 eran agregar la información de compuestos químicos sobre los archivos de las rutas metabólicas y elaborar una interfaz web para presentar las rutas metabólicas enriquecidas, se evidenció que no se puede afirmar que un compuesto químico tiene actividad directa con una proteína de la ruta metabólica. Esto se debe a que existen ciertas limitantes biológicas, como que la

membrana celular puede impedir que el compuesto químico ingrese, que restringirían que cualquier compuesto químico forme parte de una ruta metabólica, aun cuando exista el potencial de la interacción entre compuesto y su blanco. Por ello, se consideró mostrar en la página web los compuestos químicos que tienen activad sobre las rutas metabólicas en base al cálculo de la distribución hipergeométrica de probabilidad con *pvalue* menor o igual a 0.05, considerando la relación compuesto químico y ruta metabólica.

En el presente trabajo se manejó el almacenamiento de la información a través de ficheros en lugar de la utilización de un gestor de base de datos. Esto se debe a que los ficheros no requieren de un espacio amplio de memoria para guardar la información, al contrario de las bases de datos que al ser programas complejos necesitan de espacio en disco y memoria para trabajar eficazmente [47]. Además, el rendimiento es mejor a través de ficheros puesto que el acceso a la información es simple y ligera, a diferencia de las bases de datos, en las que, si no se han definido índices durante la creación de la tabla, podría ralentizar el tiempo de la consulta [48]. Adicionalmente, la migración de los datos es un proceso fácil, ya que se puede copiar y pegar los ficheros a una carpeta de destino y proporcionar los permisos de escritura o lectura; por otro lado, en el caso de una base de datos, se debe exportar las tablas en los formatos disponibles, crear la base de datos y la tabla; para luego realizar la importación en la otra base de datos; lo que significa un proceso demoroso para el usuario [47]. Otra razón por la que se usó ficheros es debido a la facilidad de lectura para los usuarios finales, ya que algunos de ellos no son técnicos informáticos y no es necesario tener personal especializado para el manejo de la información [49]. Sin embargo, es importante mencionar que, en el caso de haber utilizado un gestor de base de datos para el almacenamiento de la información, se habría optimizado el tiempo de programación. Esto se debe a que se accedería a la información directamente por consultas específicas en las que se definirían las condiciones de relación entre las tablas y no sería necesario la navegación entre directorios y ficheros para la obtención de datos.

La lectura y escritura de datos durante la ejecución de los códigos se realizó por medio de memoria RAM y no a través de memoria en disco, esto se debe a que el acceso a la información es instantánea y concurrente, por tanto, no debe ubicarse los datos, leerlos y enviarlos antes de procesarlos, como en el caso de la memoria en disco [50] [51]. Además, el rendimiento para la lectura mediante memoria RAM es mejor, puesto que se accede a la información en el orden de los nanosegundos a diferencia de memoria en disco, el cual se tarda en la lectura en el orden de los milisegundos [52]. Sin embargo, pueden existir inconvenientes como la pérdida de información en el caso de un bloqueo en el computador

o fallos de alimentación [50], así como también, problemas de almacenamiento ya que no se utiliza para guardar información en memoria a largo plazo [52].

4. CONCLUSIONES

A partir de este trabajo se determina que es posible enriquecer la información de rutas metabólicas considerando la relación entre compuestos químicos y blancos mediante la gestión de datos dispersos en diferentes bases de datos químico-biológicas al tomar como referencia el Ciclo de Vida de la Analítica de Big Data para su gestión.

Este trabajo de investigación utilizó la metodología del Ciclo de Vida de la Analítica de Big Data organizar actividades y tareas relacionadas con la adquisición, procesamiento y análisis de datos, debido a que se caracteriza por ser grandes conjuntos de datos, estar almacenados en diferentes fuentes y diversos formatos. Por ello, se estructuró el tratamiento de los datos en base a las etapas propuestas por dicho método. Además, fue necesario unificar las siguientes etapas del Ciclo de Vida de la Analítica de Big: Adquisición y filtrado de datos, Extracción de datos, Validación y limpieza de datos, en una sola etapa. Esto se debe a que algunos conjuntos de datos sirven para la extracción y limpieza de otros conjuntos. Por ejemplo: la información de *KEGG Pathway* luego de ser extraída, filtrada y verificada se usa como datos base para la adquisición, filtrado, extracción y validación de los datos de UniProt, ChEMBL y ChEBI.

El *Target Fishing in silico* permitió encontrar posibles blancos no relacionados con ciertos compuestos químicos mediante la relación de la información entre bases de datos químico-biológicas y la implementación de las herramientas computacionales como RDKIT. Este último, facilitó las operaciones moleculares 2D para la transformación de los *SMILES* a fingerprint y el cálculo del valor de la similaridad en base al método de Similaridad de Tanimoto.

La implementación de multiprocesamiento en la ejecución del código en la fase de análisis de datos del Ciclo de Vida de la Analítica de Big Data permitió optimizar el tiempo de ejecución del *Target Fishing*, ya que dividió el procesamiento de acuerdo con el número de núcleos disponibles y la cantidad de compuestos químicos de entrada.

La utilización de la distribución hipergeométrica de probabilidad y el valor de probabilidad equivalente a 0.05 permitieron distinguir qué rutas metabólicas fueron enriquecidas por los compuestos químicos y sus *KEGG targets.* Esto se realizó con el propósito de eliminar resultados que podían obtenerse al azar y determinar qué compuestos tienen actividad sobre la ruta metabólica.

Trabajos futuros

El presente trabajo tomó como base para el *Target Fishing* el conjunto de compuestos contenidos en *KEGG Pathway.* Un trabajo futuro podría ser: agregar más información a las rutas metabólicas realizando *Compund Fishing*. Para ello, en la fase de Análisis de datos del Ciclo de la Vida de la Analítica de Datos, se debería tomar como datos base los compuestos químicos de la base de datos ChEMBL. Esto tendrá como resultado más alternativas de compuestos químicos para la ruta metabólica.

Además, el presente trabajo implementó el *Target Fishing* usando un compuesto químico de la base de datos *KEGG Pathway*. Otro proyecto a futuro consistiría en realizar el *Target Fishing* sobre un conjunto de compuestos para emplearlo en alternativas de mezclas como productos naturales o en reposicionamiento de fármacos. El reposicionamiento de fármacos consiste en el proceso de identificación de nuevas aplicaciones para medicamentos existentes, descontinuados o archivados que actualmente están en desarrollo para otras condiciones médicas, en la que puede incluso variar la dosis empleada [53] [54].

Además, se podría agregar información toxicológica sobre los efectos secundarios disponibles en la base de datos *Side Effects Resource* para los compuestos químicos de tipo *DRUG* mediante el nombre almacenado en la base de datos *KEGG DATABASE. Side Effects Resource* contiene información sobre los efectos secundarios, la frecuencia de los efectos, información sobre el medicamento a nivel comercial e identificadores a otras bases de datos [55].

5. REFERENCIAS BIBLIOGRÁFICAS

[1] T. Erl, W. Khattak y P. Buhler, Big Data Fundamentals: Concepts, Drivers & Techniques, Indiana: Prentice Hall, 2016, p. 19.

[2] R. González, M. Sánchez y M. Galindo, «Software para predecir la interacción, actividad y función de fármacos,» Mayo 2016. [En línea]. Available: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/53882/7/rgonzalezbalTFM0516memoria.pdf. [Último acceso: 1 Octubre 2018].

[3] Kanehisa Laboratories, «KEGG: Kyoto Encyclopedia of Genes and Genome,» 2018. [En línea]. Available: https://www.genome.jp/kegg/. [Último acceso: 25 07 2018].

[4] TOCRIS, «Cell Biology,» 2018. [En línea]. Available: https://www.tocris.com/cell-biology. [Último acceso: 16 Octubre 2018].

[5] Kyoto Encyclopedia of Genes and Genomes, «KEGG PATHWAY Database,» 21 Agosto 2018. [En línea]. Available: https://www.genome.jp/kegg/pathway.html. [Último acceso: 16 Octubre 2018].

[6] K. Minoru, G. Susumu, S. Yoko, K. Masayuki, F. Miho y T. Mao, «Data, information, knowledge and principle: back to metabolism in KEGG,» *Nucleic Acids Research,* vol. 42, nº 1, p. 199–205, 2014.

[7] D. Kumari y R. Kumar, «Impact of Biological Big Data in Bioinformatics,» *International Journal of Computer Applications,* vol. 101, nº 11, pp. 22-24, 2014.

[8] H. Kashyap, H. Ahmed, N. Hoque, S. Roy y D. Bhattacharyya, «Big Data Analytics in Bioinformatics: A Machine Learning Perspective,» *Journal of Latex Class Files,* vol. 13, nº 9, 2014.

[9] Kyoto Encyclopedia of Genes and Genomes, «KEGG Release Notes,» [En línea]. Available: https://www.kegg.jp/kegg/docs/relnote.html. [Último acceso: 10 Octubre 2018].

[10] Kyoto Encyclopedia of Genes and Genomes, «KEGG release history,» 2018. [En línea]. Available: https://www.genome.jp/kegg/docs/upd_all.html. [Último acceso: 17 Octubre 2018].

[11] M. Linderman, J. Sorenson, L. Lee y G. Nolan, «Computational solutions to large-scale data management and analysis,» *Nature Reviews Genetics ,* vol. 11, pp. 647-657, 2010.

[12] L. Wang y X. Qung Xie, «Computational target fishing: what should chemogenomics researchers expect for the future of in silico drug design and discovery?,» *Future Med Chem,* vol. 6, nº 3, pp. 247-249, 2014.

[13] T. Katsila, G. Spyroulias, G. Patrinos y M. Matsoukas, «Computational approaches in target identification and drug discovery,» *Computational and Structural Biotechnology Journal,* vol. 14, pp. 177-184, 2016.

[14] S. Ekins, J. Mestres y B. Testa, «In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling,» *Br J Pharmacol,* vol. 152, nº 1, p. 9–20, 2007.

[15] R. Lyngsø, Computational Biology, Dinamarca: BRICS Dissertation Series, 2000.

[16] X. Su, «Introduction to Big Data,» 29 Agosto 2017. [En línea]. Available: https://www.ntnu.no/iie/fag/big/lessons/lesson2.pdf. [Último acceso: 16 Enero 2018].

[17] K. Minoru y G. Susumu, «KEGG: Kyoto Encyclopedia of Genes and Genomes,» *Nucleic Acids Research,* vol. 28, nº 1, pp. 27-30, 2000.

[18] The UniProt Consortium, «UniProt: the universal protein knowledgebase,» *Nucleic Acids Research,* vol. 45, nº 5, p. 2699, 2018.

[19] The UniProt Consortium, «UniProt: the Universal Protein,» [En línea]. Available: https://www.uniprot.org/docs/uniprot_flyer.pdf. [Último acceso: 29 Julio 2018].

[20] A. Gaulton, L. Bellis, P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani y J. Overington, «ChEMBL: a large-scale bioactivity database for drug discovery,» *Nucleic Acids Research,* vol. 40, nº 1, pp. 1100-1107, 2012.

[21] J. Hastings, P. de Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, V. Muthukrishnan, G. Owen, S. Turner, M. Williams y C. Steinbeck, «The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013,» *Nucleic Acids Research,* vol. 41, nº 1, pp. 456-463, 2013.

[22] ChEBI, «About ChEBI,» 2018. [En línea]. Available: https://www.ebi.ac.uk/chebi/aboutChebiForward.do. [Último acceso: 23 Octubre 2018].

[23] European Bioinformatics Institute, «What is ChEBI?,» 2018. [En línea]. Available: https://www.ebi.ac.uk/training/online/course/chebi-quick-tour/what-chebi. [Último acceso: 5 Noviembre 2018].

[24] United States Environmental Protection Agency, *Appendix F. SMILES Notation Tutorial,* Washington D.C., 2017.

[25] United States Environmental Protection Agency, «SMILES Tutorial,» 21 Febrero 2016. [En línea]. Available: https://archive.epa.gov/med/med_archive_03/web/html/smiles.html. [Último acceso: 26 Julio 2018].

[26] Daylight Chemical Information Systems, «4. SMARTS - A Language for Describing Molecular Patterns,» 2008. [En línea]. Available: http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html. [Último acceso: 26 Julio 2018].

[27] A. Lipkus, «A proof of the triangle inequality for the Tanimoto distance,» *Journal of Mathematical Chemistry,* vol. 26, nº 1-3, p. 263–265, 1999.

[28] S. Bero, A. Muda, Y.-H. Choo, N. Muda y S. Pratama, «Weighted Tanimoto Coefficient for 3D Molecule,» 10 Junio 2018. [En línea]. Available: https://arxiv.org/abs/1806.05237v1. [Último acceso: 10 Octubre 2018].

[29] P. Willett, «Similarity-based virtual screening using 2D fingerprints,» *Drug Discovery Today,* vol. 11, nº 23/24, pp. 1046-1053, 2006.

[30] J. Swamidass† y P. Baldi, «Mathematical Correction for Fingerprint Similarity Measures to Improve Chemical Retrieval,» *Journal of Chemical Information and Modeling,* vol. 47, nº 1, pp. 952-964, 2006.

[31] S. Arif, J. Holliday y P. Willett, «Comparison of chemical similarity measures using different numbers of query structures,» *Journal of Information Science,* vol. 39, nº 1, pp. 1-8, 2013.

[32] G. Landrum, «RDKit Documentation,» 01 Marzo 2018. [En línea]. Available: https://www.rdkit.org/RDKit_Docs.current.pdf. [Último acceso: 10 Septiembre 2018].

[33] L. Sánchez, «Distribución hipergeométrica de probabilidad,» 29 Octubre 2014. [En línea]. Available: https://estadisticayadministracion.wordpress.com/2014/10/29/distribucion-hipergeometrica-de-probabilidad-cero-complicada/. [Último acceso: 16 Noviembre 2018].

[34] S. P. Molina Sierra, «Evaluación Comparativa del desempeño computacional de la ejecución de consultas usando la Base de Datos ChEMBL,» Quito, 2018.

[35] KEGG, «KEGG Markup Language,» [En línea]. Available: https://www.kegg.jp/kegg/xml/docs/. [Último acceso: 1 Agosto 2018].

[36] Kyoto Encyclopedia of Genes and Genomes, «KEGG Markup Language,» [En línea]. Available: https://www.kegg.jp/kegg/xml/docs/. [Último acceso: 7 Agosto 2018].

[37] W3Schools, «XML Syntax Rules,» 2018. [En línea]. Available: https://www.w3schools.com/xml/xml_syntax.asp. [Último acceso: 17 Octubre 2018].

[38] INDACEA , «¿Qué es una diana terapéutica?,» 22 Abril 2018. [En línea]. Available: https://indacea.org/que-es-una-diana-terapeutica/. [Último acceso: 23 Octubre 2018].

[39] D. Stumpfe y J. Bajorath, «Exploring Activity Cliffs in Medicinal Chemistry,» *Journal of Medical Chemistry,* vol. 55, nº 7, p. 2932−2942, 2012.

[40] E. D. Valencia Cristancho, «Diseño Racional de Compuestos Espirotizolidínicos Pirazolotiazolidónicos y Pirazolo ß-Lactámicos con Potencial Actividad Antimicrobiana,» Bogotá, 2014.

[41] J. G. Camarena Sagredo, A. Trueba Espinosa, M. Martínez Reyes y M. d. L. López García, «Automatización de la codificación del patrón modelo vista controlador (MVC) en proyectos orientados a la Web,» *Ciencia Ergo Sum,* vol. 19, nº 3, pp. 239-250, 2012.

[42] Pocoo, «Flask web development, on drop at time,» 2010. [En línea]. Available: http://flask.pocoo.org/docs/1.0/. [Último acceso: 05 Septiembre 2018].

[43] ANGULAR, «What is Angular?,» 2018. [En línea]. Available: https://angular.io/docs. [Último acceso: 25 Octubre 2018].

[44] StatsDirect Limited, «P Values,» 2018. [En línea]. Available: https://www.statsdirect.com/help/basics/p_values.htm. [Último acceso: 11 Noviembre 2018].

[45] R. Fisher, «RA The Arrangement of Field Experiments,» *Journal of the Ministry of Agriculture,* vol. 33, pp. 503-513, 1933.

[46] Giordano, «EL FAMOSO VALOR "P",» 7 Junio 2010. [En línea]. Available: http://www.sinestetoscopio.com/el-famoso-valor-p/. [Último acceso: 20 Noviembre 2018].

[47] J. D. J. Durán, F. Astier y S. Banov, «Bases de Datos vs Sistemas de Archivos,» 22 Enero 2014. [En línea]. Available: https://prezi.com/jgrydc9ncude/bases-de-datos-vs-sistema-de-archivos/. [Último acceso: 12 Noviembre 2018].

[48] A. Sulaiman, «File System vs. Database,» 27 Abril 2017. [En línea]. Available: https://dzone.com/articles/which-is-better-saving-files-in-database-or-in-fil. [Último acceso: 12 Noviembre 2018].

[49] Fundamentos de Bases de Datos, «1.4 Sistemas de bases de datos frente a los sistemas de archivos,» Mayo 2010. [En línea]. Available: https://fundamentosdebasededatos.files.wordpress.com/2010/05/equipo2.pdf. [Último acceso: 12 Noviembre 2018].

[50] International Multimedia Resource Center, «RAM vs. Hard Drive Memory,» 2018. [En línea]. Available: https://www.lehigh.edu/~inimr/computer-basics-tutorial/ramvsdiskspacehtm.htm. [Último acceso: 13 Noviembre 2018].

[51] Indiana University, «ARCHIVED: What is the difference between memory and disk storage?,» 18 Enero 2018. [En línea]. Available: https://kb.iu.edu/d/ahtx. [Último acceso: 13 Noviembre 2018].

[52] A. Kumar, «Difference between Memory and Storage,» 7 Julio 2018. [En línea]. Available: http://www.differencebetween.net/technology/difference-between-memory-and-storage/. [Último acceso: 13 Noviembre 2018].

[53] M. Barratt y D. Frail, Drug Reposition, Nueva Yersey: Wiley, 2012.

[54] F. Saldivar, F. Prieto y J. Medina, «Descubrimiento y desarrollo de fármacos: un enfoque computacional,» *Educación Química,* vol. 28, nº 1, pp. 51-58, 2017.

[55] SIDER, «SIDER 4.1 : Side Effect Resource,» 2018. [En línea]. Available: http://sideeffects.embl.de/. [Último acceso: 25 Octubre 2018].

[56] L. Carvajal, Metodología de la Investgación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.

[57] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono y M. Kanehisa, «KEGG: Kyoto Encyclopedia of Genes and Genomes,» *Nucleic Acids Research,* vol. 27, nº 1, pp. 29-34, 1999.

[58] R. B. Lyngsø, Computational Biology, Dinamarca: BRICS Dissertation Series publications, 2000.

[59] PubChem, «About PubChem,» [En línea]. Available: http://pubchemdocs.ncbi.nlm.nih.gov/about. [Último acceso: 26 Julio 2018].

[60] F. Haseltine, M. Huerta, Y. Liu, G. Downing y B. Seto, «NIH Working Definition of Bioinformatics and Computational Biology,» 17 Julio 2000. [En línea]. Available: http://www.bisti.nih.gov/docs/CompuBioDef.pdf. [Último acceso: 6 Agosto 2018].

[61] M. Cruz Monteagudo, E. Tejera, Y. Pérez, J. Medina Fronco, A. Sánchez Rodríguez y F. Borges, «Systemic QSAR and phenotypic virtual screening: chasing butterflies in drug discovery,» *Drug Discovery Today,* vol. 22, nº 7, pp. 994-1007, 2017.

[62] N. Wale y G. Karypis, «Target Fishing for Chemical Compounds Using Target-Ligand Activity Data and Ranking Based Methods,» *Journal of Chemical Information and Modeling,* vol. 49, nº 10, p. 2190–2201, 2009.

[63] El Pasante, «Ventajas y desventajas de las bases de datos,» 17 Junio 2015. [En línea]. Available: https://educacion.elpensante.com/ventajas-y-desventajas-de-las-bases-de-datos/. [Último acceso: 12 Noviembre 2018].

[64] Probability Formula, «Hypergeometric Distribution,» [En línea]. Available: http://www.probabilityformula.org/hypergeometric-distribution.html. [Último acceso: 16 Noviembre 2018].

6. ANEXOS

ANEXO I

```
# Código para realizar la copia, la lectura y la escritura de archivos
KGML obtenidos del api de KEGG

import requests
import os
import config
import directoryStructure

organism = config.organism
urlList = 'http://rest.kegg.jp/list/pathway/'
urlKgml = 'http://rest.kegg.jp/get/'
directory = directoryStructure.keggpathway
dirkgml = directoryStructure.keggpKgml

pathways = requests.get(urlList + organism)
for line in pathways.text.split('\n'):
    pathwayid = line.split('\t')[0].replace('path:','')
    print(pathwayid)
    kgml = requests.get(urlKgml + pathwayid + '/kgml')
    f = open(dirkgml + '/' + pathwayid + '.xml', 'w')
    f.write(kgml.content.decode("utf-8"))
    f.close
```

ANEXO II

```
# Código para obtener el listado de genes y compuestos de los archivos
KGML

import csv, os, xml.etree.cElementTree as ET
import config
import directoryStructure

dirPathways = directoryStructure.keggpKgml
cleanInfoKegg = directoryStructure.cleanInfoKegg

files = os.listdir(dirPathways)
fileGenes = open(cleanInfoKegg + '/listGenesPathway.csv', 'w')
fileCompounds = open( cleanInfoKegg + '/listCompoundsPathway.csv', 'w')

fileGeneID = open(cleanInfoKegg + '/listGenesID.csv', 'w')
fileCompoundID = open(cleanInfoKegg + '/listCompoundsID.csv', 'w')


resultados = []
resultadosUnicos = []
resultadosUnicosC = []
setGenes = set()
setCompounds = set()

def writeFiles (list, type):
    if type == 'gene':
        for listItem in list:
            fileGeneID.write(listItem+'\n')
    else:
        for listItem in list:
            fileCompoundID.write(listItem+'\n')

for file in files:

    tree = ET.ElementTree(file=dirPathways +'/'+file)
    root = tree.getroot()

    for entry in root.findall('entry'):
        type = entry.get("type")
        nameEntry = entry.get("name")
        if (nameEntry == None):
            nameEntry=''

        if type == "compound" or type == "gene":
            for graphics in entry.findall("graphics"):
                name = graphics.get("name")
                nameCorrect =name.find('...')
                nameTotal = nameEntry
                namesEntry = nameEntry.split(" ")

                if nameCorrect != -1:
                    name = name[:nameCorrect]
                if type == "gene":
                    reaction = ''
                    if entry.get("reaction") != None:
```

```
                        reaction = entry.get("reaction")
                    genes = nameTotal +';'+name + ';' +file[:-4]+'\n'
                    fileGenes.write(genes)
                else:
                    compounds = name + ';' +file[:-4]+'\n'
                    fileCompounds.write(compounds)

                for nameEntry in namesEntry:
                    resultados = [nameEntry]
                    for resultados in resultados:

                        if type == 'gene':
                            setGenes.add(resultados)

                        else:
                            setCompounds.add(name)
writeFiles(setGenes,'gene')
writeFiles(setCompounds,'compound')

fileGenes.close()
fileCompounds.close()
fileGeneID.close()
fileCompoundID.close()
```

ANEXO III

```
# Código para obtener información complementaria sobre los compuestos
químicos desde el api de KEGG

import csv, requests
import directoryStructure

dirCompounds = directoryStructure.keggCompounds
listCompoundID = directoryStructure.cleanInfoKegg +
'/listCompoundsID.csv'

with open( listCompoundID, 'r') as csvfile:
  spamreader = csv.reader(csvfile, delimiter='\n')
  for row in spamreader:
        compoundID = row[0][0:6]
        txt = requests.get('http://rest.kegg.jp/get/' + compoundID)
        f = open(dirCompounds + '/' + compoundID[0:6] + '.txt', 'w')
        f.write(txt.content.decode("utf-8"))
        f.close
```

ANEXO IV

```
# Código para obtener el nombre del compuesto químico y los identificadores
de las bases de datos ChEBI y ChEMBL

import csv, os
import directoryStructure

dirCompounds = directoryStructure.keggCompounds
filesCompound = os.listdir(dirCompounds)
fileInformation = open(directoryStructure.cleanInfoKegg +
'/compoundIDdataBases.csv', 'w')

for file in filesCompound:
    name = ""
    chembl = ""
    chebi = ""
    comment = ""
    with open(dirCompounds + '/'+file, 'r') as csvfile:
        lines = csvfile.readlines()
        for linea in lines:
            for lineaS in linea.split():
                if 'NAME' == lineaS:
                    nameLine = linea.split()
                    name = nameLine[1]
                if 'ChEMBL:' == lineaS:
                    chemblLine = linea.split()
                    chembl = chemblLine[1]
                if 'ChEBI:' == lineaS:
                    chebiLine = linea.split()
                    chebi = chebiLine[1]
                if 'COMMENT' == lineaS:
                    commentLine = linea.split()
                    comment = 'comment: ' + commentLine[1]

        resultados = file[:6]+";"+ name +";"+ chembl  + ";"+ chebi+ ";"+
comment
        print (1,resultados)
        fileInformation.write(resultados+'\n')

fileInformation.close()
```

ANEXO V

Código para obtener los SMILES canónicos de los compuestos químicos con identificador de ChEMBL. Además, se utiliza para la descarga de archivos con identificador de ChEBI.

```
import json
import csv
import requests
import directoryStructure
from urllib.request import  Request, urlopen, urlretrieve

dirInfoCompounds = directoryStructure.cleanInfoKegg +
'/compoundIDdataBases.csv'
dircanSmiles = directoryStructure.chembl + '/canonicalSmiles.csv'
chebiDir = directoryStructure.genericSmilesChebi

file =  open(dirInfoCompounds, 'r')
spamreader = file.read().split('\n')
compoundSmileIdchembl = open(dircanSmiles, "w")
chemblId = ''
cheid = ''
name = ''
smile= ''
for row in spamreader:
    rowArray = row.split(';')
    resultado=[]
    if rowArray[2] != '':
        conection =
requests.get('https://www.ebi.ac.uk/chembl/api/data/molecule/' +
rowArray[2]+ '.json', timeout=None)
        texto = conection.content.decode("utf-8")
        try:
            compound = json.loads(texto)
            if compound['molecule_structures'] != None:
                smile =
compound['molecule_structures']['canonical_smiles']
                chemblId = compound['molecule_chembl_id']
                resultado = rowArray[0]+";"+chemblId+';'+ smile
+';'+rowArray[1]
                print (resultado)
                compoundSmileIdchembl.write(resultado + '\n')
        except:
            continue

    else:
        if rowArray[3] != '':
            url =
"http://www.ebi.ac.uk/chebi/saveStructure.do?xml=true&chebiId="+
rowArray[3]+"&imageId=0"
            savein = chebiDir + '/' +rowArray[0]+"-"+rowArray[3] + ".xml"
            urlretrieve(url, savein)

compoundSmileIdchembl.close()
```

ANEXO VI

```
# Código para obtener el SMILES genérico de los archivos descargados en el
ANEXO V

import csv, os, xml.etree.cElementTree as ET
import directoryStructure

dirChebismilies = directoryStructure.genericSmilesChebi
archivos = os.listdir(dirChebismilies)
resultados = []
resultadosUnicos = []

fileGenericSmiles = directoryStructure.chebi +
'/compoundGenericSmiles.csv'
with open( fileGenericSmiles, "w",  newline='') as csv_file:
    writer = csv.writer(csv_file)
    for archivo in archivos:
        with open(dirChebismilies + '/' +archivo, 'r' , encoding='utf-8')
as compoundChebi:
            if archivo != 'readChebiGetSmile.py':
                lines = compoundChebi.readlines()
                smilestr = []
                for linea in lines:
                    for lineaS in linea.split(">"):
                        if lineaS == '<SMILES':
                            smile = linea.split("<")
                            smilestr = smile[1].split(">")

                if len (smilestr) !=0:
                    resultado = [archivo[:6]+ ";"+archivo[7:-4]+
";"+smilestr[1]]
                    writer.writerow(resultado)
```

ANEXO VII

```
# Código para extraer la información del archivo uniprot_sprot.data para
almacenarla en la tabla EVOLUNIPROTDB

import MySQLdb
import config

#   PROCESSING UNIPROT DATA FILES AND CREATING DATABASE

def ExtractAllWords(Line):
    TempWords = str(Line).split(' ')
    WordAll = []
    for word in TempWords:
        if word != '': WordAll.append(word)
    return (WordAll)
def Extract*SEQ(ListLines):
    Out = []
    for Lines in ListLines:
        Temp = ExtractAllWords(Lines)
        Temp = [item.split('\n')[0] for item in Temp]
        Out = Out+ Temp
    Out = ''.join(Out)
    return (Out)

def filterData(cur, temp0, temp1,
temp2,temp3,temp4,temp5,temp6,temp7,temp8):
    add_register = ("INSERT INTO evoluniprotv1.UNIPROTDB "
               "(UNIPROTID, UNIPROTAC, GENEID, GO) "
               "VALUES (%s, %s, %s, %s)")

    if temp6 == 'Homo sapiens (Human).\n':
        temp2 = temp2.split(';')[0]
        data_register = (temp0, temp2, temp3,temp8)
        cur.execute(add_register, data_register)


def UniprotDataBaseHS_Create():
    global DatabaseName


    TABLES= (
        "CREATE TABLE `UNIPROTDB` ("
        "  `UNIPROTID` varchar(50),"
        "  `UNIPROTAC` TEXT,"
        "  `GENEID` TEXT,"
        "  `GO` TEXT"
        ")")
    db=
MySQLdb.connect(host=config.host,passwd=config.password,user=config.user,
db=DatabaseName)
    cur = db.cursor()
    print("Creating table : UNIPROTDB_FILTERED")
    cur.execute(TABLES)
    print ('UNIPROTDB_FILTERED was created')
```

```
def UniprotDataBaseHS_ADD(UFile):

    global DatabaseName


    db=
MySQLdb.connect(host=config.host,passwd=config.password,user=config.user,
db="")
    print("Adding records to table : UNIPROTDB_ALL")
    f = open(UFile,'r')

    Continue = False
    for Line in f:

        Line = str(Line)
        TempList = ExtractAllWords(Line)

        if TempList[0]=='ID':
            Continue = False
            Record_ID = []
            Record_AC = []
            Record_ST = []
            Record_GID = []
            Record_GO = []
            Record_OX = []
            Record_OC = []
            Record_OS = []
            Record_SEQ = []
            Record_ID.append(TempList[1])
            T = str(TempList[2]).split(';')[0]
            Record_ST.append(T)
        if TempList[0]=='AC':
            T = TempList[1:len(TempList)]
            T = ''.join(T)
            T = str(T).split('\n')[0]
            if T[len(T)-1] == ';': T = T[0:len(T)-1]
            Record_AC.append(T)
        if TempList[0]=='GeneID':
            T = str(str(TempList[1]).split(';')[0]).split('=')[1]
            Record_OX.append(T)
        if TempList[0]=='OS':

            T = TempList[1:len(TempList)]
            T = ' '.join(T)
            Record_OS.append(T)
        if TempList[0]=='OC':

            T = TempList[1:len(TempList)]
            T = ' '.join(T)
            Record_OC.append(T)
        if TempList[0]=='DR' and TempList[1]=='GeneID;':
            T= str(TempList[2]).split(';')[0]
            Record_GID.append(T)
        if TempList[0]=='SQ': Continue = True
        if Continue == True and TempList[0]!='SQ' and
(Line[0]+Line[1])!='//': Record_SEQ.append(Line)
        if TempList[0]=='DR' and TempList[1]=='GO;':
            T= str(TempList[2]).split(';')[0]
            Record_GO.append(T)
        if (Line[0]+Line[1])=='//':
```

```
            Continue = False
            Temp = []
            Temp.append(Record_ID[0])
            Temp.append(Record_ST[0])
            if len(Record_AC)>0:
                Temp.append(Record_AC[0])
            else:
                Temp.append('')
            if len(Record_GID)>0:
                Temp.append(Record_GID[0])
            else:
                Temp.append('')
            if len(Record_OX)>0:
                Temp.append(Record_OX[0])
            else:
                Temp.append('')
            if len(Record_OC)>0:
                Temp.append(''.join(Record_OC))
            else:
                Temp.append('')
            if len(Record_OS)>0:
                Temp.append(''.join(Record_OS))
            else:
                Temp.append('')
            if len(Record_SEQ)>0:
                T = ExtractSEQ(Record_SEQ)
                Temp.append(T)
            else:
                Temp.append('')
            if len(Record_GO)>0:
                Temp.append(';'.join(Record_GO))
            else:
                Temp.append('')
            cur = db.cursor()
            filterData(cur, Temp[0], Temp[1],
Temp[2],Temp[3],Temp[4],Temp[5],Temp[6],Temp[7],Temp[8])
            db.commit()

    f.close
    print ('File Processed')

#   Main program UniprotDB_ALL.py

if __name__ == '__main__':

    dirSprotDat = config.basedir + 'resources/uniprot_sprot.dat'
    DatabaseName = 'evoluniprot'
    SprotFile = dirSprotDat
    UniprotDataBaseHS_Create()
    UniprotDataBaseHS_ADD (SprotFile)
    print ('SWI Processed')
```

ANEXO VIII

```
# Código para extraer el subconjunto de información de las proteínas de
UniProt correspondientes a los genes de KEGG Pathway

import MySQLdb
import time, csv
import config
import directoryStructure


def connection ():
    db = MySQLdb.connect( host= config.host,
                          user= config.user,
                          passwd = config.password,
                          db=config.databaseChembl)
    cur = db.cursor()
    return cur;

def getProteins (geneL, cur):
    geneL = tuple(geneL)
    print (len(geneL))
    params = {'geneList': geneL}
    s= ('SELECT * FROM evoluniprot.uniprotdb where GENEID in
%(geneList)s')

    cur.execute(s, params)
    resultado = cur.fetchall()
    return resultado

if __name__ == '__main__':
    start_time = time.time()
    fileProtsGenes = directoryStructure.uniprot
    fileGenesId = directoryStructure.cleanInfoKegg+'/listGenesID.csv'
    fileToWriteG = open(fileProtsGenes+'/proteinas-genesKegg.csv', 'w')
    genesList = []
    with open (fileGenesId, 'r') as genes:
        spamreader = genes.read().split('\n')
        for gene in spamreader:
            if gene != '':
                gen = gene.split (':')[1]
                genesList.append(gen)

    cur = connection()
    proteinas = getProteins(genesList, cur)
    resultado = []
    for protein in proteinas:
        fileToWriteG.write(protein[0]+ ';'+protein[1]+ ';'+ protein[2]+
'\n')

    fileToWriteG.close()
    print (len(proteinas))
```

ANEXO IX

Código para extraer los SMILES canónicos correspondientes a los compuestos químicos que tienen interacción con un blanco en particular

```
import MySQLdb
import time
import csv
import config
import directoryStructure

def connection ():
    db = MySQLdb.connect(host=config.host,
                         user=config.user,
                         passwd=config.password,
                         db=config.databaseChembl)
    cur = db.cursor()
    return cur;

def getProteins(proteinsA, cur):
    proteinsA = tuple(proteinsA)
    params = {'proteinsList':proteinsA}

    s= ('Select * from chembl_simplified.protein where accession in
%(proteinsList)s')

    cur.execute(s,params)
    resultado = cur.fetchall()
    return resultado

def getCompounds(idProtein, cur):

    s= ('select canonical_smiles from compound where molregno in ('
        'select molregno from data_experimental where tid = %(tid)s)')

    cur.execute(s, { 'tid': idProtein })
    resultado = cur.fetchall()
        #print (resultado)
    return resultado

if __name__ == '__main__':

    start_time = time.time()
    proteinsIds = []
    proteinsInfo = []
    fileProteinsGenes = directoryStructure.uniprot
    dirProteinsSmiles = directoryStructure.proteinsSmiles
    with open (fileProteinsGenes + '/proteinas-genesKegg.csv', 'r') as
proteins:
        spamreader = proteins.read().split('\n')
        for protein in spamreader:
            if protein != '':
                prot = protein.split (';')
                proteinsInfo.append(prot)
                proteinsIds.append(prot[1])
```

```
    cur = connection()
    proteinsChembl = getProteins(proteinsIds, cur)

    print (len(proteinsChembl))
    total= len(proteinsChembl)
    i = 0
    for protein in proteinsChembl:

        tid = protein[0]
        targetType = protein [1]
        prefName= protein[2]
        accesion = protein[9]
        gene = list (filter(lambda gene: gene[1] == accesion,
proteinsInfo))

        gene =gene[0][2]
        nameFile = str(tid)
        firstLine = str(tid)+'_'+gene+'_'+targetType+'_'+
prefName+'_'+accesion
        name= dirProteinsSmiles +'/'+ nameFile+ '.csv'
        compounds = getCompounds(tid, cur)

        if len(compounds) != 0:
            f = open(name, 'w')
            f.write(firstLine+'\n')
            for compound in compounds:
                #print(compound[0])
                f.write(compound[0]+'\n')
        i = i+1
        print ('Avance: '+ str(i) + '/' +str(total))
    print (time.time() - start_time)
```

ANEXO X

```
# Código que implementa el proceso de Target Fishing

import time, csv, os
from Protein import Protein
from Compound import Compound
from getNumberCompounds import calcularAreas
from rdkit import Chem
from rdkit import DataStructs
from rdkit.Chem.Fingerprints import FingerprintMols
from rdkit.DataStructs import FingerprintSimilarity
from targetFishingRdkitFiles import writeProteins
from multiprocessing import Pool
import multiprocessing
from rdkit.Chem import AllChem
import directoryStructure

def printS(compounds, proteins, rangoi, rangof):
    for i in range(rangoi, rangof):
        fingerPrint = compounds[i]
      if fingerPrint != None:
            compoundName = compounds[i].nameCompoundKegg
            writeProteins(compounds[i], proteins)


def cargarCompounds(line):
    print(line)
    comp = line.split(";")
    fingerPrint = getFingerPrint(comp[2])
    compound = Compound(comp[0], comp[1], comp[2], fingerPrint, comp[3])
    return compound

def getFingerPrint (smile):
    sanit = Chem.MolFromSmiles(smile,False)
    a = None
    try:
        Chem.SanitizeMol(sanit)
        m = Chem.MolFromSmiles(smile)
        a = AllChem.GetMorganFingerprintAsBitVect(m,2)
    except ValueError:
        a= None
    return a

def getFingerPrints (smiles):
    fingerPrints = []

    for i in range(1, len(smiles)) :
        fingerPrint = getFingerPrint(smiles[i])
        fingerPrints.append(fingerPrint)
    return fingerPrints

if __name__ == '__main__':
    start_time = time.time()
    dirProteinSmiles = directoryStructure.proteinsSmiles
    archivos = os.listdir(dirProteinSmiles)
    fileCanonicalSmiles = open (directoryStructure.chembl +
'/canonicalSmiles.csv', 'r')
```

```
    canonicalSmiles = fileCanonicalSmiles.read()
    fileGenericSmiles = open (directoryStructure.chebi +
'/compoundGenericSmiles.csv', 'r')
    genericSmiles = fileGenericSmiles.read()[:-1]
    smiles = canonicalSmiles + genericSmiles
    smiles = smiles.split('\n')
    proteins =[]
    i = 0

    pool = Pool(processes=4)
    compounds = pool.map(cargarCompounds, smiles, 4)
    print ("compoundsTotales")
    print (len(compounds))
    for archivo in archivos:
        compoundsOfSmile = open(dirProteinSmiles+'/'+ archivo, 'r')
        readCompoundsOfSmile = compoundsOfSmile.read()
      smiles = readCompoundsOfSmile.split ('\n')
      proteinInfo = smiles[0].split('_')
        tid = proteinInfo[0]
      gene = proteinInfo [1]
      targetType = proteinInfo[2]
      prefName = proteinInfo[3]
      accession = proteinInfo[4][:-1]
      fingerPrints = getFingerPrints (smiles)
      protein = Protein(tid, gene, targetType, prefName, accession,
fingerPrints)
      proteins.append(protein)

    print ("Carga completa de proteinas y fingerPrints")
    print ("Empieza calculo similitud tanimoto max = 0.9, c3 = 0.55")

    cores = 4;
    totalArchivos = len(compounds)
    print (totalArchivos)
    limitesFinal = calcularAreas(totalArchivos,cores)
    print(limitesFinal)
    limiteInicial = limitesFinal

    for core in range(1,cores+1):
        if core == 1:
            process = multiprocessing.Process ( target=printS, args=(
compounds, proteins,int(0),int(limitesFinal[core-1]),))
        else:
            process = multiprocessing.Process ( target=printS, args=(
compounds, proteins, int(limiteInicial[core-2]) ,int(limitesFinal[core-
1]),))
        process.start()

    print (time.time() -start_time)
```

```
from Protein import Protein
from heapq import nlargest
import csv, os
import time
from rdkit import Chem
from rdkit import DataStructs
from rdkit.Chem.Fingerprints import FingerprintMols
from rdkit.DataStructs import FingerprintSimilarity
import directoryStructure

def getValuesSimilitudes (fingerPrint , fingerPrints):
    similitudes = []

    for fp in fingerPrints:
        similitude = getSimilitude (fingerPrint, fp)
        similitudes.append(similitude)

    return similitudes


def getSimilitude (a, b):
      resultado = 0
      if a != None and b !=None:
            resultado =
FingerprintSimilarity(a,b,metric=DataStructs.TanimotoSimilarity)
      return resultado


def writeProteins(compound, proteins):
    values = []
    dirResultadosTF = directoryStructure.resultadosTargetfishing
    fileP = open(dirResultadosTF +'/'+compound.nameCompoundKegg+'.csv',
'w')
    maxValue = 0.9
    c3Value= 0.55
    firstLine = compound.nameCompoundKegg + ';' + compound.chemblid +
';'+compound.smile + ';'+compound.name
    fileP.write(firstLine)
    for protein in proteins:
        values = getValuesSimilitudes(compound.fingerPrint,
protein.fingerPrints)
      tid = protein.tid
      gene = protein.gene
      targetType = protein.targetType
      prefName = protein.prefName
      accession = protein.accession
      infoProtein = str(tid) +
';'+str(gene)+';'+targetType+';'+prefName+';'+accession
        if len(values) >0:
            maxV = max(values)
            if len(values) >= 3:
                c3List= nlargest(3, values)
                c3 =  sum(c3List)/3
                if c3 >= c3Value or  maxV >=  maxValue:

fileP.write(infoProtein+";"+str(maxV)+";"+str(c3)+"\n")
```

```
import time
import math


def calculateLines(lines,partialArea,x,nucleo,total):

    x1= partialArea
    x1 = math.ceil(x1)

    if int(x) == int(nucleo):
        x1 = total
    return x1


def calcularAreas(compoundLen, nucleos):

    resultados = []
    resultadosC = []
    compoundLen = int(compoundLen)

    partialArea =int(compoundLen/nucleos)
    firstLines = math.ceil(partialArea)
    #print(math.ceil(firstLines))
    resultados.append(math.ceil(firstLines))
    lines = int(firstLines)
    y= 0
    resultado = 0


    for x in range (2, nucleos+1):
        nucleo = nucleos
        y= resultado+y
        resultado =
calculateLines(lines+y,partialArea,x,nucleo,compoundLen)
        #print("Archivo len : "+ str(x))
        x = x+1
        resultados.append(resultado)
        resultados = sorted(resultados)

    inicial = 0
    for i in range(0,len(resultados)):
        inicial = resultados[i] + inicial
        if inicial > compoundLen:
            inicial = compoundLen
        resultadosC.append(inicial)
    return resultadosC
```

ANEXO XI

```
# Código que reordena la información obtenida para almacenar el blanco
con los respectivos compuestos de KEGG Pathway

import time, csv, os
from multiprocessing import Pool
import directoryStructure
import multiprocessing

dirResultadosTF = directoryStructure.resultadosTargetfishing
dirResultadosTFproteins = directoryStructure.resultadosTFProteins
dirProtein = directoryStructure.proteinsSmiles

if __name__ == '__main__':
    start_time = time.time()
    archivos = os.listdir(dirResultadosTF)
    archivosProtein = os.listdir(dirProtein)
    campounds = []
    proteinas = []
    proteinids = []
    compoundsOfProtein = []
    total= len(archivosProtein)
    i = 0;
    for archivoProtein in archivosProtein:
        proteins = open(dirProtein+'/'+archivoProtein, 'r')
        proteins = proteins.read()
        proteinid = archivoProtein.split(".")[0]
        infoProtein = proteins.split('\n')[0]
        proteinids.append(proteinid)
        file = open( dirResultadosTFproteins +'/'+ proteinid+'.csv', 'w')
        file.write(infoProtein+'\n')
        for archivo in archivos:
            compoundsTargeF = open(dirResultadosTF+'/'+ archivo, 'r')
            nameFile = archivo.split(".")[0]
            compound = nameFile
            campounds.append(compound)
            readcompoundsTargeF = compoundsTargeF.read()
            proteins = readcompoundsTargeF.split("\n")
            infoCompound = proteins[0].split(';')
            for proteinStr in proteins:
                proteinDatos = proteinStr.split(";")
                if proteinDatos[0] == proteinid and len(proteinDatos) >1:
                    resultado = compound+";"+infoCompound[1] +
";"+infoCompound[3]  +";"+proteinDatos[5]+";"+proteinDatos[6]+"\n"
                    file.write(resultado)
        i = i+1
        print("Avance " + str(i)+'/'+str(total))
```

ANEXO XII

9	154	317	599	790	978	1235	1522	1719	2023	2161	2348	2683	2893	3074
16	155	328	610	799	983	1236	1524	1723	2028	2162	2350	2692	2897	3075
18	156	329	613	814	993	1237	1536	1728	2030	2166	2356	2693	2898	3082
25	157	330	623	815	994	1241	1543	1742	2033	2167	2357	2695	2899	3091
26	159	332	624	816	995	1244	1544	1759	2034	2168	2358	2696	2901	3098
27	177	338	629	817	998	1263	1545	1777	2041	2169	2475	2710	2902	3099
28	178	351	635	818	1000	1268	1548	1785	2042	2170	2492	2717	2903	3105
31	185	354	641	821	1017	1269	1553	1786	2043	2171	2494	2720	2904	3125
32	186	367	643	823	1019	1312	1555	1788	2044	2185	2517	2729	2905	3141
41	187	369	650	824	1020	1326	1557	1789	2045	2188	2519	2735	2906	3146
43	191	374	652	834	1021	1357	1558	1791	2046	2194	2526	2736	2908	3156
47	196	375	654	835	1022	1359	1559	1803	2047	2203	2527	2739	2911	3162
52	204	383	657	836	1025	1360	1562	1806	2048	2205	2528	2740	2912	3164
90	205	384	658	837	1026	1361	1565	1812	2049	2208	2529	2741	2913	3172
91	207	387	659	838	1066	1374	1571	1813	2050	2209	2530	2742	2914	3176
92	208	396	672	839	1075	1375	1573	1814	2051	2212	2534	2744	2915	3178
93	210	408	673	840	1080	1376	1576	1815	2052	2215	2538	2752	2916	3242
100	213	412	680	841	1103	1384	1577	1816	2053	2222	2539	2770	2917	3248
102	216	427	695	842	1109	1387	1583	1839	2057	2224	2542	2773	2918	3249
107	217	432	699	843	1111	1395	1584	1841	2058	2237	2548	2775	2920	3251
111	218	440	706	846	1119	1410	1585	1843	2064	2241	2550	2798	2925	3265
116	219	462	715	847	1120	1432	1586	1844	2065	2242	2554	2810	2931	3269
117	220	466	716	873	1128	1436	1588	1845	2066	2244	2555	2826	2932	3274
124	231	471	718	883	1129	1441	1589	1846	2073	2245	2556	2831	2936	3276
125	238	472	719	886	1131	1445	1591	1848	2078	2246	2557	2833	2938	3283
126	239	476	728	887	1132	1452	1592	1854	2081	2247	2558	2837	2939	3284
127	240	479	747	890	1133	1453	1593	1855	2099	2249	2559	2846	2944	3290
128	241	487	759	891	1134	1455	1594	1856	2100	2252	2562	2847	2946	3291
131	246	525	760	896	1136	1456	1595	1857	2103	2260	2569	2852	2950	3292
132	247	526	761	898	1137	1457	1610	1890	2115	2261	2570	2859	2957	3293
134	248	537	762	904	1139	1459	1612	1901	2120	2263	2582	2862	2990	3294
135	249	545	763	914	1141	1499	1613	1902	2145	2264	2584	2864	2993	3297
136	250	552	765	920	1144	1504	1621	1903	2146	2268	2587	2867	2997	3301
140	251	553	766	933	1145	1506	1633	1909	2147	2272	2595	2868	3002	3303
142	262	554	768	949	1147	1508	1636	1910	2149	2288	2597	2869	3029	3309
143	270	558	771	952	1158	1509	1644	1915	2150	2289	2618	2870	3035	3312
146	271	572	773	953	1195	1510	1645	1917	2152	2308	2629	2872	3055	3315
147	272	581	774	954	1213	1511	1646	1956	2153	2309	2641	2876	3061	3320
148	276	586	775	956	1215	1512	1650	1969	2155	2321	2642	2885	3062	3326
150	279	587	776	958	1230	1513	1654	1973	2157	2322	2645	2887	3064	3350
151	290	596	777	960	1232	1514	1675	1977	2158	2324	2648	2890	3065	3351
152	314	597	779	972	1233	1515	1716	1991	2159	2335	2673	2891	3066	3352
153	316	598	781	975	1234	1520	1717	1994	2160	2346	2678	2892	3073	3354

3355	3678	3939	4199	4690	5024	5170	5338	5579	5693	5917	6403	6595	6872	7294
3356	3683	3945	4214	4697	5025	5173	5340	5580	5695	5925	6404	6597	6885	7295
3357	3685	3949	4215	4734	5026	5196	5346	5581	5696	5970	6416	6608	6897	7296
3358	3688	3973	4216	4758	5027	5207	5347	5582	5698	5972	6419	6609	6915	7297
3359	3695	3978	4217	4759	5028	5208	5360	5583	5699	5979	6422	6610	6916	7298
3360	3702	3984	4221	4780	5029	5209	5373	5584	5724	6009	6446	6611	6945	7299
3361	3708	3985	4233	4790	5030	5213	5406	5585	5729	6011	6464	6613	6999	7317
3362	3710	3988	4245	4792	5031	5218	5409	5586	5731	6041	6469	6622	7006	7318
3363	3716	3990	4247	4803	5032	5222	5422	5587	5732	6093	6470	6646	7010	7329
3373	3717	3991	4248	4804	5033	5223	5423	5588	5733	6095	6476	6647	7015	7334
3376	3718	3992	4282	4829	5045	5226	5424	5590	5734	6097	6480	6652	7040	7341
3383	3725	4005	4296	4837	5053	5228	5428	5591	5737	6117	6482	6654	7046	7345
3416	3735	4012	4297	4842	5054	5241	5429	5592	5739	6195	6487	6713	7048	7357
3417	3753	4023	4306	4843	5058	5243	5430	5593	5740	6196	6505	6714	7052	7364
3454	3758	4025	4311	4846	5062	5260	5465	5594	5742	6197	6506	6715	7054	7367
3458	3759	4047	4312	4851	5063	5261	5467	5595	5743	6198	6507	6716	7067	7372
3480	3761	4048	4313	4853	5073	5286	5468	5596	5745	6199	6510	6723	7068	7374
3482	3766	4049	4314	4854	5105	5287	5476	5597	5747	6232	6513	6732	7083	7376
3486	3767	4056	4316	4855	5111	5288	5478	5598	5770	6240	6514	6751	7084	7378
3551	3769	4067	4318	4860	5136	5289	5481	5599	5771	6241	6518	6752	7086	7381
3553	3776	4068	4322	4864	5137	5290	5494	5600	5777	6256	6523	6753	7097	7398
3558	3777	4074	4323	4878	5138	5291	5495	5601	5778	6257	6529	6754	7099	7403
3559	3778	4088	4325	4881	5139	5292	5498	5602	5781	6258	6531	6755	7113	7407
3567	3780	4099	4350	4886	5140	5293	5499	5603	5783	6261	6532	6768	7124	7409
3569	3781	4124	4351	4887	5141	5294	5501	5604	5787	6272	6538	6772	7132	7412
3572	3782	4125	4352	4889	5142	5295	5515	5605	5788	6300	6540	6774	7153	7415
3576	3783	4126	4353	4891	5143	5297	5520	5606	5792	6303	6546	6776	7155	7417
3577	3784	4128	4360	4893	5144	5298	5524	5607	5795	6319	6548	6777	7157	7421
3579	3785	4129	4361	4907	5145	5300	5525	5608	5797	6323	6550	6778	7158	7422
3611	3786	4137	4363	4914	5146	5313	5530	5609	5801	6326	6554	6783	7166	7428
3614	3791	4141	4481	4915	5147	5315	5532	5610	5810	6328	6555	6788	7167	7430
3615	3795	4145	4507	4916	5150	5319	5536	5618	5834	6331	6556	6789	7172	7433
3620	3798	4149	4512	4923	5151	5320	5538	5621	5836	6332	6558	6790	7173	7434
3632	3799	4157	4513	4929	5152	5321	5547	5624	5837	6335	6564	6793	7184	7442
3636	3800	4158	4522	4942	5153	5322	5550	5631	5859	6337	6570	6794	7189	7465
3643	3815	4159	4538	4953	5154	5327	5551	5641	5860	6342	6571	6817	7201	7532
3645	3816	4160	4543	4985	5155	5328	5562	5644	5871	6344	6572	6833	7222	7786
3654	3817	4161	4544	4986	5156	5329	5563	5645	5879	6347	6573	6839	7225	7852
3672	3818	4170	4547	4987	5159	5331	5564	5646	5888	6352	6579	6850	7226	7867
3673	3827	4190	4548	4988	5163	5333	5566	5663	5894	6356	6580	6865	7251	7874
3674	3845	4191	4597	5009	5167	5335	5567	5664	5906	6387	6581	6868	7253	7915
3675	3932	4193	4609	5021	5168	5336	5571	5689	5914	6401	6582	6869	7272	7941
3676	3938	4194	4638	5023	5169	5337	5578	5690	5915	6402	6584	6870	7276	8001

8085	8737	9311	10135	11069	25865	51166	56267	79646	147746
8174	8743	9340	10161	11073	25885	51176	56413	79723	158835
8202	8767	9344	10170	11127	26279	51179	56479	79813	162466
8313	8797	9373	10203	11181	26524	51268	56624	79837	162514
8317	8811	9376	10213	11184	26986	51284	56649	79885	166929
8318	8824	9388	10257	11185	27005	51289	56670	79966	169355
8394	8829	9415	10298	11200	27035	51305	56848	80025	196743
8396	8836	9429	10316	11201	27102	51311	56923	80142	204851
8398	8841	9446	10327	11213	27115	51347	57007	80168	206358
8399	8843	9448	10331	11221	27159	51426	57016	80201	221955
8408	8850	9451	10381	11238	27198	51478	57105	80854	225689
8424	8851	9453	10383	11253	27250	51520	57118	81027	246243
8431	8854	9475	10392	11255	27306	51548	57121	83401	259230
8435	8862	9493	10411	11313	27330	51564	57144	83732	259287
8438	8877	9517	10419	11343	27334	51701	57172	83933	259290
8443	8879	9536	10434	22861	27343	51738	57406	84254	260293
8473	8883	9572	10454	22943	27434	51776	57468	84335	260425
8484	8911	9615	10498	22953	28234	53354	57551	84444	285220
8491	8912	9641	10524	23035	28996	53637	57552	84539	338398
8493	8913	9706	10533	23049	29072	53944	57704	84634	338442
8517	8942	9734	10555	23192	29101	54106	57733	84649	339403
8528	8972	9748	10558	23193	29110	54363	58497	84787	340156
8529	8986	9757	10560	23235	29117	54575	58508	84867	340485
8536	8989	9759	10599	23236	29126	54576	59272	84876	353497
8550	9002	9817	10645	23239	29850	54577	59340	85366	373156
8556	9020	9869	10667	23324	29881	54578	59341	85464	376497
8560	9064	9934	10682	23396	29883	54583	59350	91807	377677
8564	9088	9971	10690	23410	29924	54600	60481	112398	377841
8565	9101	9986	10724	23411	30001	54657	60482	112399	387893
8566	9113	10000	10733	23417	30061	54658	63826	112401	440275
8569	9123	10013	10746	23450	30814	54659	64106	112817	494551
8605	9127	10014	10769	23463	30835	54677	64127	113235	729230
8607	9128	10018	10797	23498	30849	54898	64600	114548	100137049
8622	9132	10021	10800	23566	50507	55163	64772	120892	100533105
8639	9170	10038	10803	23604	50807	55193	64781	121278	100861540
8644	9175	10039	10825	23620	50832	55236	64805	121504	100996758
8647	9197	10055	10846	23621	50833	55361	64816	122011	101930123
8648	9215	10057	10855	23632	50835	55503	64834	129807	102724428
8654	9227	10058	10864	23659	50840	55584	64840	131890	102724560
8694	9252	10060	10886	23683	50940	55589	65220	134864	102800317
8698	9254	10062	10892	23743	51005	55781	79001	138474	106821730
8708	9261	10105	10919	25759	51056	55790	79054	139760	
8714	9265	10110	10999	25780	51094	55811	79071	140609	
8720	9266	10114	11035	25825	51111	55851	79444	140803	
8722	9294	10125	11040	25828	51135	55869	79602	140838	

ORDEN DE EMPASTADO