



TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Detección de errores en bases de datos químicas

Autor

Jesús Navarro Merino

Directora

Rocío Celeste Romero Zaliz



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, julio de 2023

Detección de errores en bases de datos químicas

Jesús Navarro Merino

Palabras clave: palabra_clave1, palabra_clave2, palabra_clave3,

Resumen

Poner aquí el resumen.

Project Title: Project Subtitle

First name, Family name (student)

Keywords: Keyword1, Keyword2, Keyword3,

Abstract

Write here the abstract in English.

Yo, **Jesús Navarro Merino**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 15429457E, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Jesús Navarro Merino

Granada a X de MES de 201 .

Dña. **Rocío Celeste Romero Zaliz**, Profesora del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Detección de errores en bases de datos químicas*, ha sido realizado bajo su supervisión por **Jesús Navarro Merino**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 201 .

La directora:

Rocío Celeste Romero Zaliz

Agradecimientos

Poner aquí agradecimientos...

Índice general

1. Introducción	1
1.1. Motivación y objetivos	2
1.2. Objetivos	7
1.3. Estructura de la memoria	7
2. Estado del arte y fundamentos teóricos	9
3. Gestión y Planificación del proyecto	11
3.1. Metodología	11
3.1.1. Aplicación de la 'metodología' al proyecto	11
3.2. Gestión de la configuración	11
3.2.1. Gestión del código	11
3.2.2. Gestión de la documentación	12
3.3. Gestión de recursos	12
3.3.1. Recursos humanos	12
3.3.2. Recursos materiales	12
3.3.3. Recursos software	13
3.4. Gestión de costes	14
3.4.1. Coste de recursos humanos	15
3.4.2. Costes de recursos materiales	15
3.4.3. Costes software	16
3.4.4. Otros costes	16
3.4.5. Presupuesto final	17
3.5. Planificación	18
3.6. Análisis de riesgos	18
3.6.1. Riesgos materializados	18
4. Diseño e Implementación	21
5. Experimentación	23
A. Tabla comparativa del set completo de moléculas	27

Índice de figuras

1.1.	Distintas cadenas SMILES válidas para el 1-methyl-3-bromo-ciclohexeno. (a) Considera el ciclo como la rama principal y el bromo como ramificación. (b) Hace el recorrido que marca la flecha, dejando parte del ciclo como una ramificación. Imagen extraída de [3]	3
3.1.	Riesgos del proyecto, causas, y planes de actuación	19
3.2.	Matriz de probabilidad-impacto de riesgos	19

Índice de tablas

1.1. Códigos SMILES y sus representaciones visuales según Sigma-Aldrich	5
1.2. Códigos SMILES y sus representaciones visuales según SciFinder	6
3.1. Tabla de los costes materiales	16
3.2. Tabla de costes adicionales	17
3.3. Presupuesto total del proyecto	17
A.1. Tabla de índices con los nombres de las moléculas de la Tabla A.2	27
A.2. Tabla extendida para el set de datos de 30 moléculas. Contiene la cadena SMILES extraída de Sigma-Aldrich (SA), la cadena SMILES extraída de SciFinder (SF), y las imágenes de las respectivas bases de datos (SA y SF)	29

Capítulo 1

Introducción

La Química estudia la composición y estructura de la materia, sus propiedades y transformaciones. Estudia las sustancias, la energía y sus cambios durante las reacciones. Desde que se tienen registros, la química ha sido fundamental para el desarrollo de la humanidad, ya que ha permitido la producción de materiales, alimentos, medicamentos y energía, entre otros. Esto ha sido un proceso lento y exhaustivo a través de la experimentación. Por ejemplo, en 1881, Beilstein publica su Manual de Química Orgánica, que recogía 15000 compuestos orgánicos con sus propiedades [2]. Conforme la química se iba expandiendo, también lo hacía el volumen de datos que se generaban, siendo cada vez mas frecuentes preguntas como "¿alguien habrá sintetizado ya este compuesto?" [9]

Eventualmente, hace unas cuantas décadas, se pensó que la cantidad de información que cada químico por su cuenta había acumulado, se podía compartir y hacer accesible a la comunidad científica a través de su almacenamiento en bases de datos [4]. Con el desarrollo de técnicas de manipulación y tratamiento de esos datos surgió el término *chemoinformatics*.

Las *chemoinformatics* han cobrado gran importancia en los últimos años debido al aumento exponencial de datos experimentales generados en la investigación biomédica y química, y a la necesidad de manejar y analizar esta información de manera eficiente. Esta disciplina ha sido influenciada por diversas áreas, como la química, matemáticas, estadística, biología y ciencias de la computación entre otras. Al parecer, su origen se remonta a la década de 1940, habiendo ya algunas investigaciones en el área, pero el término 'chemoinformatics' como tal se lleva utilizado más bien poco (1998) [5]. Como tal, aun no hay un acuerdo en cuanto a su definición, seguramente por su carácter interdisciplinar, ni si quiera en cómo deletrearlo, pudiendo aparecer también como *cheminformatics*, *chemical informatics*, *chemi-informatics*, y *molecular informatics* entre otras [5, 7]. En la literatura se discuten varias interpretaciones sobre su definición, unas más precisas y

otras más generales: [5, 6, 7, 4]

La mezcla de recursos de información para transformar datos en información, y la información en conocimiento, con el fin de tomar decisiones más rápidas y efectivas en la identificación y optimización de fármacos [Brown 1998]

Chem(o)informatics es un término genérico que abarca el diseño, la creación, la organización, la gestión, la recuperación, el análisis, la difusión, la visualización y el uso de la información química. [G. Paris 1999]

La aplicación de métodos informáticos para resolver problemas de química [J. Gasteiger and T. Engel 2006]

A pesar de ello, son a día de hoy un componente esencial en el descubrimiento de sustancias químicas; sin duda es un área en constante evolución y su importancia solo aumentará en los próximos años, tanto en el descubrimiento de fármacos —que es como originariamente surgió y donde más impacto tiene en la sociedad— como en otros campos de la química.

Una herramienta también de vital importancia en este ámbito son los sistemas de representación lineal. Surgieron a medida que la química y la tecnología computacional avanzaban, y nos permiten codificar moléculas para su análisis y almacenamiento en bases de datos. En el siglo XIX, se desarrollaron varias formas de representación visual de moléculas, como las fórmulas estructurales que permitieron a los químicos dibujar y visualizar moléculas de manera más efectiva. Sin embargo, estas formas de representación no son adecuadas para su uso en la computación, ya que no son fácilmente legibles para los programas informáticos. Nosotros, los humanos, cuando vemos una estructura molecular dibujada la entendemos directamente, obtenemos una visión global de los símbolos que representan los enlaces y la distribución espacial de los átomos que la componen, pero los computadores no tienen esa facilidad. Por ello, se desarrollaron sistemas de notación lineal que permitían describir de manera más precisa y eficiente la estructura molecular, trabajando con tipos de datos sencillos, cadenas de caracteres.

1.1. Motivación y objetivos

Los formatos de notación lineal llevan siendo un tema de interés e investigación para los científicos desde mediados del siglo 19, evolucionando poco a poco y desarrollándose nuevas notaciones en función de las necesidades —principalmente computacionales— del momento y las limitaciones que se iban descubriendo [1]. En la actualidad, existen varias representaciones lineales, siendo las más usadas SMILES, InChI, y SELFIES [12]. Como

comenté antes, una forma muy potente de representar moléculas y compuestos químicos es mediante cadenas strings, y de esto justamente se encargan las representaciones lineales: traducir una molécula, con sus átomos, enlaces entre ellos, ciclos y otras propiedades características, en una cadena string que la represente, y que la máquina y los propios químicos puedan entender. Sin embargo, hay diferencias notables entre las representaciones, tanto en la sintaxis de las cadenas que se generan como en las aplicaciones que se le puede dar a cada una de ellas.

SMILES, ideada por David Weininger, sale a la luz en 1988 satisfaciendo con creces las necesidades de procesamiento de información química que había, desbancando a la representación estandarizada del momento, Wiswesser Line Notation (WLN). Desde ese entonces SMILES se convirtió —y sigue siendo a día de hoy— en el estándar de representación lineal, ya que permite describir estructuras moleculares de una forma sencilla en un formato fácil de leer, lo que ha hecho que sea una herramienta popular en la química computacional, siendo la más usada entre investigadores y químicos. Pese a esto, SMILES tiene dos grandes inconvenientes: una misma molécula puede escribirse con varias cadenas SMILES distintas válidas, es decir, tiene sinónimos (Figura 1.1); y no es robusto ni sintáctica ni semánticamente. En este sentido se podría generar un string que no represente una molécula válida, como lo es CC(CCCC, el cual tiene un paréntesis sin cerrar (lo que implica que no se delimita cuándo acaba la rama). O generar una molécula que no sea químicamente viable como CO=CC, que muestra un átomo de oxígeno neutro formando tres enlaces (superando el límite de enlaces covalentes que un oxígeno neutro puede tener) [12].

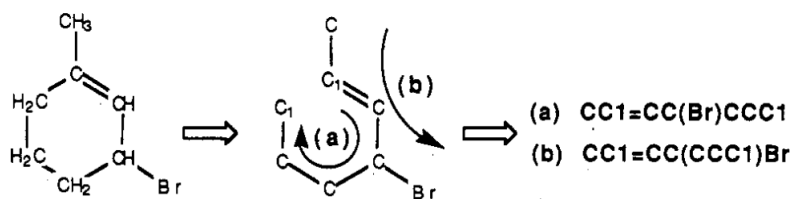


Figura 1.1: Distintas cadenas SMILES válidas para el 1-methyl-3-bromociclohexeno. (a) Considera el ciclo como la rama principal y el bromo como ramificación. (b) Hace el recorrido que marca la flecha, dejando parte del ciclo como una ramificación. Imagen extraída de [3]

Esto tiene especial relevancia en el ámbito del Machine Learning (ML). Aunque se sale del alcance de este trabajo, uno de los grandes objetivos de la química computacional es la creación o diseño de nuevas moléculas. Se podrían crear modelos de ML o redes neuronales capaces de generar moléculas ficticias válidas, para posteriormente ver sus propiedades, valorarlas energéticamente para ver cuán estables son, y estudiar su viabilidad en

distintas aplicaciones, entre otras cosas. SMILES dificulta esta tarea, y por ello, aparece en 2020, SELFIES (SELF-referencIng Embedded Strings), una nueva representación lineal 100 % robusta, muy usada actualmente para modelos generativos. Ver [12, 10] para más detalles de cómo soluciona los problemas de robustez y otras características de la representación. SELFIES es relativamente reciente y continuamente está ampliando sus funcionalidades, mejorando su simplicidad y facilidad de uso para el usuario [13]. Aun así, no se termina de instaurar entre la comunidad investigadora. Por último, InChI es creado en 2013 por la IUPAC (International Union of Pure and Applied Chemistry) como un proyecto para estandarizar el proceso de búsqueda de estructuras moleculares entre distintas bases de datos. Esto es porque InChI (International Chemical Identifier) genera una cadena canónica única para cada molécula, de manera que cada molécula tiene una sola representación, y dicha representación solamente hace referencia a esa molécula. La principal desventaja radica en su sintaxis y su estructura jerárquica, haciéndola complicada de leer y utilizar por los humanos. Por esto mismo también, no es la mejor opción para usar en modelos generativos, pues tiene una serie de reglas y normas gramaticales y aritméticas que son complejas de aplicar al generar moléculas a través de modelos de ML.[8]

Por todo lo anterior, me centraré en la notación SMILES durante el desarrollo de este trabajo. Dicho esto, existen diversas bases de datos en química donde se recoge gran cantidad de información acerca de los compuestos. Entiéndase esto como una colección estructurada y organizada que contiene datos sobre compuestos químicos, sus propiedades y relaciones con otros compuestos. Se utilizan para almacenar y recuperar información sobre moléculas, sustancias, reacciones, propiedades fisicoquímicas, e incluso literatura científica relacionada. Mencionaré ahora las más importantes y las que serán objeto de interés. *PubChem*, una base de datos abierta que sirve información a millones de usuarios en todo el mundo, desde investigadores y estudiantes hasta el público general. Recogen para cada compuesto, información sobre su estructura, representaciones 2D y 3D, identificadores, propiedades químicas y físicas, patentes, avisos de toxicidad, etc. [15]

SciFinder, una herramienta de investigación muy potente que permite explorar las bases de datos de CAS (American Chemical Society) las cuales contienen literatura sobre Química y otras disciplinas afines como Física, Biomedicina, Geología, Ingeniería Química, etc. Incluye referencias bibliográficas y resúmenes de artículos, informes, y libros entre otras cosas. Permite realizar búsquedas por estructura, nombres de sustancias o identificadores, reacciones en la que participa dicha sustancia, artículos y publicaciones que nombren el compuesto en cuestión, e incluso proveedores de compra [16]. Para el uso de esta herramienta es necesario acceder mediante la red de una institución autorizada (en este caso trabajo mediante VPN de

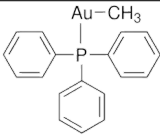
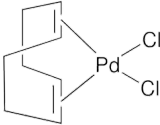
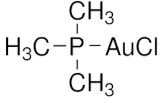
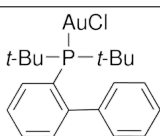
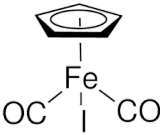
Código SMILES	Representación 2D
<chem>C[Au].c1ccc(cc1)P(c2ccccc2)c3ccccc3</chem>	
<chem>Cl[Pd]Cl.C1CC=CCCC=C1</chem>	
<chem>Cl[Au].CP(C)C</chem>	
<chem>Cl[Au].CC(C)(C)P(c1ccccc1-c2ccccc2)C(C)(C)C</chem>	
<chem>[Fe]I.[C-]#[O+].[C-]#[O+].[CH]1[CH][CH][CH][CH]1</chem>	

Tabla 1.1: Códigos SMILES y sus representaciones visuales según Sigma-Aldrich

la UGR) y seguir los pasos para registrarte ¹. Y *Sigma-Aldrich*, una compañía de ciencia, química y biotecnología que se dedica a la producción y venta de productos químicos, reactivos, equipos y materiales de laboratorio. Ofrece herramientas, servicios, artículos y una gran variedad de productos químicos que se utilizan en investigación, biofarmacéutica, e industria entre otros ámbitos [17]. A través de su página web se enfocan al comercio electrónico pudiendo buscar y comprar productos, compuestos orgánicos e inorgánicos, agentes reactivos, isótopos para síntesis químicas, proteínas, enzimas, etc. De cada producto muestra información relevante como la ficha de datos de seguridad, detalles de las propiedades físicas y químicas así como algunas representaciones lineales del compuesto y la representación del grafo molecular, que es lo interesante en este caso realmente.

Desde la Universidad de Granada, la tutora de este TFG colabora con el grupo de investigación de químicos del ICIQ (Instituto Catalán de Investigación Química) liderado por la profesora Mónica H. Pérez-Temprano. Su foco de investigación gira en torno al entendimiento de transformaciones catalíticas en las que participan compuestos organometálicos, descubriendo y

¹Pasos para el registro en SciFinder https://bibliotecaugr.libguides.com/scifinder_scholar

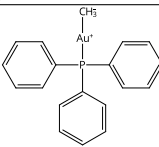
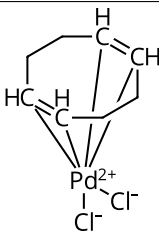
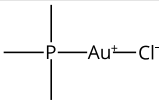
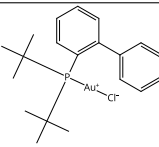
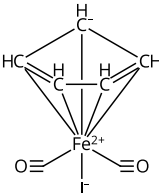
Código SMILES	Representación 2D
<chem>[Au+](CH3-)[P](C=1C=CC=CC1)(C=2C=CC=CC2)C=3C=CC=CC3</chem>	
<chem>[Cl-][Pd+2]123([Cl-])[CH]=4CC[CH]3=[CH]2CC[CH]41</chem>	
<chem>[Cl-][Au+][P](C)(C)C</chem>	
<chem>[Cl-][Au+][P](C=1C=CC=CC1C=2C=CC=CC2)(C(C)(C)C)C(C)(C)C</chem>	
<chem>O#C[Fe+2]1234([I-])(C#O)[CH]=5[CH]4=[CH]3[CH-]2[CH]51</chem>	

Tabla 1.2: Códigos SMILES y sus representaciones visuales según SciFinder

diseñando reacciones más eficientes basadas en catalizadores metálicos. Para más detalle sobre el grupo de investigación y sus ámbitos de trabajo, ver su sitio web [14]. En resumen, intentan desarrollar enfoques más sostenibles para la síntesis de moléculas orgánicas usando la química organometálica. Como tal, necesitan codificar correctamente una molécula de organometálica en pos de trabajar con ella adecuadamente y utilizar todas las herramientas, para, entre otras cosas, poder dibujarla y entenderla mejor.

Uno de los principales problemas que se detectan en este ámbito es la heterogeneidad en las distintas bases de datos para un mismo compuesto o molécula. Para ilustrar esto, presento las tablas 1.1 y 1.2. Ambas tablas comparan las mismas moléculas, mostrando el código SMILES y la representación visual que ofrecen las bases de datos Sigma-Aldrich y SciFinder respectivamente. Vemos diferencias claras en el tratamiento de los ciclos aromáticos, la especificación de las cargas de los átomos y la posición de algunas ramificaciones. Utilizo un subset de 5 moléculas pertenecientes a

la organometálica, seleccionadas desde un conjunto de datos de 30 moléculas considerados de interés por los químicos con los que colabora la tutora (disponible para su consulta en mi GitHub). En el Apéndice A, se puede consultar una tabla comparativa con el set de moléculas al completo.

1.2. Objetivos

Por tanto, el objetivo principal de este Trabajo Fin de Grado sería modificar el paquete open-source OpenBabel creando un método para canonizar códigos SMILES, orientado específicamente para compuestos organometálicos. Para ello, se establecen los siguientes subobjetivos:

- Analizar y comparar las cadenas SMILES de distintas bases de datos (p.ej. Sigma-Aldrich, SciFinder) viendo los posibles sinónimos para una misma molécula.
- Determinar un sistema que genere, a partir de cualquier sinónimo SMILES de la misma molécula, un único SMILES canónico.
- Definir otro algoritmo o conjunto de reglas que mejore, aunque sea mínimamente, el sistema de dibujado de las moléculas.

1.3. Estructura de la memoria

esperar a tenerla mas avanzada para completar esto

Capítulo 2

Estado del arte y fundamentos teóricos

Quizas sea mejor mover esta seccion justo despues de la introduccion para seguir con la tematica de la motivacion, y ya luego me meto con la gestion y planificacion

Puedo hacer una revision de la literatura existente hasta dia de hoy sobre el tema Usar SCOPUS para esto, con terminos tipo: "SMILESmolecule organometalic" (juntarlos o separarlos segun vea)

Hablar por aqui de la organometalica, representacion de moléculas, Hablar mas extendido de SMILES, SELFIES, e INCHI; cosas de dibujado de moleculas (los paquetes que hay),) No se si meterlo aqui o en otro apartado, el diagrama de clases

la historia de la humanidad está marcada por la búsqueda de materiales que mejoren su calidad de vida, y los metales han sido parte crucial de esos cambios

La materia que forma los seres vivos tiene en su composición sustancias cuya base principal es el carbono. El estudio de estos compuestos constituye una rama de la química llamada química orgánica. La abundancia del carbono en el planeta es relativamente pequeña: aproximadamente un 0,03 %; sin embargo, da lugar a millones de sustancias diferentes, mientras que los compuestos inorgánicos son solo unos pocos miles. ¿Qué hace a este elemento tan especial? Su estructura singular, que le permite formar largas cadenas en las cuales una pequeña variación da lugar a un compuesto distinto al anterior.

(del archivo de informe Reuniones puedo ir sacando cosas para meter en el estado del arte) Openbabel como tal no soporta el dibujado en 3D de las moléculas, por lo que los únicos dibujos que puede hacer son en 2D.

Openbabel tampoco soporta el uso de wedge ni hash bonds para el dibujo de moléculas en 2D con perspectiva (es curioso porque en el código sí que hay funciones dedicadas a esto, pero luego cuando le metes símbolos SMILES de @@ y demás, los medio ignora. Sigo ejemplos del tutorial de Daylight, pero no salen los mismos dibujos. Otros símbolos más dedicados a geometría que viene en Daylight o en OpenSMILES, tipo @SP, @TB, @OH, los ignora por completo). Pero sí es capaz de generar archivos .sdf con información 3D, que se pueden usar en otros softwares de dibujo específicos como Avogadro (y no están mal, mucho mejor que los 2D desde luego)

Capítulo 3

Gestión y Planificación del proyecto

3.1. Metodología

3.1.1. Aplicación de la 'metodología' al proyecto

3.2. Gestión de la configuración

En este apartado se describirá cómo se ha llevado a cabo la gestión de los activos de este proyecto, es decir, el código desarrollado y la documentación generada. Dado que son partes fundamentales del trabajo, atendiendo al análisis de riesgos descrito en la sección 3.6 y el plan de actuación frente a la pérdida de información, se detalla a continuación la gestión de ambas.

3.2.1. Gestión del código

Para la gestión del código se ha usado un control de versiones a través de Git y Github. Ambas herramientas en conjunto permiten ir creando versiones intermedias del código conforme se va desarrollando y hacer copias de seguridad en la nube. También son muy útiles para proyectos colaborativos, donde varias personas del equipo pueden combinar fácilmente su parte del código desarrollado y revisar el progreso subido hasta el momento. El procedimiento a seguir en la mayoría de casos es bastante similar. Se creará un repositorio remoto en la plataforma de GitHub con el nombre de *TFG* —o el que uno prefiera— donde se irán almacenando los cambios¹. En la carpeta de trabajo local de nuestra computadora, carpeta que contendrá en

¹<https://github.com/Jesnm01/TFG>

mi caso todos los elementos de los que quiera llevar un control, se creará un repositorio local usando Git, que habrá que vincular con el remoto para poder ir sincronizando los cambios.

3.2.2. Gestión de la documentación

En lo relativo a la documentación, el proceso de gestión será similar ya que también se ha usado GitHub para su control de versiones. Esta se ha redactado en LaTeX usando el servicio online Overleaf. Overleaf cuenta con una opción para sincronizar el proyecto con un repositorio de GitHub, pero es una opción de pago. En su lugar, descargo el proyecto en el repositorio local y desde ahí ya realizo dicha sincronización para subir los cambios en el remoto.

Además, dada la naturaleza del proyecto y de la metodología de desarrollo utilizada, se ha llevado un registro de las reuniones con la tutora que también se ha ido actualizando periódicamente. Este documento pretende recoger los contenidos más relevantes de las reuniones con Rocío: preguntas, comentarios, anotaciones, tareas que hacer, cosas pendientes de una reunión a otra, revisiones, y puntos a mejorar, entre otras cosas.

3.3. Gestión de recursos

3.3.1. Recursos humanos

- **Dña. Rocío Celeste Romero Zaliz**, profesora del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada en calidad de tutora del proyecto. A cargo de la supervisión y guía del alumno durante su desarrollo del trabajo.
- **Jesús Navarro Merino**, estudiante del grado en Ingeniería Informática en la Escuela Técnica Superior de Ingenierías Informática y Telecomunicación.

3.3.2. Recursos materiales

Para este proyecto no se han necesitado recursos adicionales, habiéndose usado únicamente los siguientes recursos materiales, ya poseídos por el alumno:

- **Portátil personal**: Portátil ACER Aspire A515-51G-8907 con un procesador Intel Core i7 8550U 1.8GHz, 20GB de memoria RAM y una

arquitectura de 64 bits. Se ha usado durante todo el proyecto, para labores de programación y redacción de la memoria.

- **Pantalla.** Monitor utilizado de apoyo a la pantalla propia del portátil. De la marca AOC, de 24 pulgadas con una resolución de 1920x1080.

3.3.3. Recursos software

En esta sección describiré todas aquellas herramientas software empleadas durante la realización del proyecto. Todas y cada una de ellas son herramientas de software libre, gratuitas o disponibles a través de licencias de estudiantado. A continuación se lista el software usado:

- **Sistema operativo:** Windows 10 Home. Aunque por lo general Windows no es gratuito, al estar utilizando la típica licencia OEM que trae preinstalada el ordenador al comprarlo, la considero como tal.
- **Visual Studio C++:** es un IDE muy potente de Microsoft orientado a crear aplicaciones .NET y C++ para Windows. Se ha usado su versión gratuita Visual Studio Community 2022. Permite editar, depurar, realizar pruebas de testing, además de tener control de versiones integrado, entre otras cosas. Aquí se ha llevado a cabo todo el desarrollo del código.
- **OpenBabel:** Open Babel es una biblioteca de código abierto multi-plataforma utilizada en química computacional y ciencias relacionadas para la conversión y manipulación de estructuras químicas en varios formatos. He trabajado con el release 3.1.1 disponible en su repositorio de GitHub oficial².
- **CMake:** es una herramienta de generación de archivos de compilación que simplifica el proceso de compilación y construcción de proyectos, permitiendo una configuración flexible e independiente de la plataforma. CMake utiliza archivos de configuración llamados CMakeLists.txt para describir la estructura del proyecto y las dependencias necesarias. Se ha usado en su versión 3.25.2 para la compilación y creación de soluciones de OpenBabel.
- **Git:** software de código abierto para el control de versiones de un proyecto.
- **Github:** es una plataforma donde se alojará el código y la documentación del proyecto. Utiliza Git por debajo y es una de las plataformas gratuitas para alojamiento de código mas empleadas a nivel mundial.

²<https://github.com/openbabel/openbabel>

- **Google Colab:** es una plataforma en línea gratuita ofrecida por Google que permite a cualquier usuario escribir y ejecutar código Python en el navegador. Es una herramienta basada en la nube que proporciona un entorno de ejecución como si fueran notebooks de Jupyter, es decir, se puede escribir, editar y ejecutar código en bloques/celdas interactivos. Se ha usado durante las etapas iniciales del proyecto para la experimentación con diversas moléculas.
- **Zotero:** software de gestión de referencias bibliográficas que permite recopilar, organizar, citar y generar fácilmente una bibliografía en varios estilos de formato estándares según los documentos, páginas webs, artículos o archivos PDF guardados. Facilita la creación de referencias y citas en documentos académicos, ahorrando tiempo y asegurando un uso correcto de las fuentes consultadas.
- **Google Meet:** servicio de videoconferencias de Google. Plataforma utilizada para las reuniones con la tutora.
- **Google Drive Sync:** ahora llamada Google Drive para PC, es una aplicación de Google que permite sincronizar los archivos y carpetas de la computadora con la cuenta de Drive. Usado para realizar copias de seguridad —adicionales a lo almacenado en GitHub— de algunos archivos importantes.
- **Overleaf:** herramienta online para la redacción de documentos en LaTeX usada para la documentación de este proyecto.
- **Correo UGR:** servicio de correo electrónico institucional de la UGR.
- **Microsoft Word:** procesador de textos de Microsoft usado para apuntes personales y documentación en sucio. Disponible a través de la cuenta de Microsoft Office 365 que ofrece la universidad.
- **Microsoft Excel:** utilizado para la creación de algunas tablas y gráficos incluidos en la memoria. Disponible a través de la cuenta de Microsoft Office 365 que ofrece la universidad.
- **Visual Studio Code y LaTeX:** VSCode es un editor de texto, que a través de algunas extensiones, permite editar, compilar y visualizar ficheros LaTeX. Es la alternativa a Overleaf según lo descrito en la sección 3.6.

3.4. Gestión de costes

TERMINAR esto cuando vaya acabando el trabajo

En esta sección se realizará una estimación de los costes asociados al proyecto, atendiendo a los recursos descritos en la Sección 3.3. La elaboración de un presupuesto preciso en muchos casos puede suponer un desafío, ya que los proyectos software a menudo están sujetos a cambios y variables imprevistas, pero es una tarea importante para estudiar su viabilidad.

3.4.1. Coste de recursos humanos

Si actuáramos como una empresa, en un proyecto software al uso existen muchos componentes que tener en cuenta para montar esta sección del presupuesto. Aspectos como el salario de los trabajadores y compensaciones varias, el coste de los procesos de selección y contratación, formación y desarrollo del personal, costes laborales adicionales (seguros, vacaciones, etc), o costes de personal externo (consultores o subcontratistas) entre otras cosas. Dada la naturaleza de este proyecto, en relación a los gastos asociados a recursos humanos contamos con un equipo de desarrollo formado por una persona que tendrá el papel de Ingeniero Informático. Para calcular el costo de su trabajo, se indica el número total de horas dedicadas y su correspondiente coste.

HACER una estimación según el horario normal de un trabajador durante todos los días de X meses. Contrastarlo con las horas reales dedicadas según clockify.

3.4.2. Costes de recursos materiales

Dado que no se han adquirido expresamente para este proyecto, ya que se poseían con anterioridad, no se valora su precio de compra como tal sino su valor de depreciación. Los productos electrónicos experimentan un proceso llamado depreciación, que conlleva una devaluación gradual a lo largo de su vida útil. Es importante tener esto en cuenta para estimar su valor actual y el coste del recurso. Esta estimación refleja el valor de un activo desde un punto de vista contable.

Al referirnos a la depreciación de un activo a lo largo de su vida útil, no se incluyen situaciones en las que sufra daños debido a accidentes, desastres naturales u otros eventos similares. En cambio, estamos hablando del desgaste del uso cotidiano, así como los impactos derivados de las innovaciones tecnológicas que surgen durante ese período que puedan dejar obsoleto el dispositivo.

Para calcular el valor actual de los activos utilizaré el método de depreciación lineal, que considera un desgaste uniforme durante su uso, mostrando el resultado del gasto anual de depreciación[11]. Necesitamos lo primero, hallar el valor residual del activo, es decir, el valor que se estima que tendrá

cuando llegue al final de su vida útil. Usaré para los dispositivos electrónicos una vida útil de 8 años. Haré un ejemplo con el coste del portátil.

$$\text{Valor residual} = \frac{\text{Coste inicial}}{\text{Vida útil (años)}} = \frac{689\text{€}}{8 \text{ años}} = 86,125\text{€}$$

Con el valor residual estimado, se puede calcular la depreciación lineal anual con la siguiente fórmula.

$$\text{Depreciación} = \frac{\text{Coste inicial} - \text{Valor residual}}{\text{Vida útil (años)}} = \frac{689\text{€} - 86,125\text{€}}{8 \text{ años}} = 75,36\text{€}$$

Teniendo estos 2 datos, se puede calcular el valor actual del activo teniendo en cuenta sus años de antigüedad. Se muestran todos los datos relativos a los costes materiales en la Tabla 3.1.

Recurso	Valor inicial (€)	Depreciación anual (€)	Antigüedad (años)	Valor actual (€)
Portátil personal	689	75,36	5	312,2
Pantalla AOC	230	25,15	2	179,69
Total:				491,89

Tabla 3.1: Tabla de los costes materiales

3.4.3. Costes software

Los costes relacionados con los recursos software son nulos. Como indico en el listado de recursos de la Sección 3.3.3, son herramientas de software libre o utilizadas mediante licencias gratuitas, por lo que no suponen coste alguno en el desarrollo de este proyecto.

3.4.4. Otros costes

Aquí se incluyen todos los demás gastos que han sido necesarios para el desarrollo del proyecto y que no pertenecen a los apartados anteriores. Principalmente son los gastos vinculados a las facturas de la luz e Internet. El coste de la tarifa de Internet contratada es de 40€/mes, que a lo largo de los 5 meses el total asciende a 200€. Para la luz, una estimación posible serían 19,58€, teniendo en cuenta el gasto que suponen los recursos materiales en base a una factura trimestral de 11,80€.

Recursos	Importe (€)
Luz	19,58
Internet	200

Tabla 3.2: Tabla de costes adicionales

3.4.5. Presupuesto final

Se presenta por tanto el presupuesto completo asociado al proyecto, dividido en cada una de las secciones tratadas anteriormente. Se ve el desglose en la Tabla 3.3.

Detalle	Importe
Costes de recursos humanos	X €
Trabajo autónomo	X €
Costes de recursos materiales	X €
Portátil personal:	312,2 €
Pantalla de apoyo	179,69 €
Costes de recursos software	0,00 €
Windows 10	0,00 €
Visual Studio C++	0,00 €
OpenBabel	0,00 €
CMake	0,00 €
Git	0,00 €
GitHub	0,00 €
Google Colab	0,00 €
Zotero	0,00 €
Google Meet:	0,00 €
Google Drive Sync:	0,00 €
Overleaf	0,00 €
Correo UGR	0,00 €
Microsoft Word	0,00 €
Microsoft Excel	0,00 €
Visual Studio Code y LaTeX	0,00 €
Costes adicionales	X €
Internet	40€ x 5 meses = 200€
Factura de la luz	19,58 €
Total:	X €

Tabla 3.3: Presupuesto total del proyecto

3.5. Planificación

Esto quizás debería ponerlo después de la metodología y antes de gestión de la configuración

3.6. Análisis de riesgos

El análisis de riesgos desempeña un papel fundamental en la planificación y ejecución exitosa de proyectos. A veces surgen imprevistos que pueden afectar en mayor o menor medida a la correcta evolución de estos. Por ello, la aplicación de este proceso resulta esencial para minimizar la incertidumbre, intentar evitar la aparición de esos riesgos, y en caso de que se materialicen, paliarlos o mitigarlos de manera efectiva mediante unos planes de actuación.

En este apartado por tanto, se analizarán los riesgos potenciales del proyecto, incluyendo sus causas y el plan de acción para resolverlos o mitigar su impacto al máximo (Figura 3.1). Además, se realizará una evaluación de la probabilidad de ocurrencia y del impacto asociado a cada riesgo, que se puede ver en la Figura 3.2. Esto se basa en una matriz con 2 dimensiones: la probabilidad de ocurrencia de un riesgo y el impacto que tendría en el proyecto si se materializa. Se valorarán los riesgos según aspectos técnicos, de recursos humanos o complejidad y naturaleza del proyecto, y preguntas del tipo, ¿qué tan difícil sería recuperarse del riesgo?, ¿cuál es el resultado más negativo que podría originarse como consecuencia? o ¿ha sucedido este riesgo o alguno similar anteriormente?

3.6.1. Riesgos materializados

completar esto al final del trabajo, o conforme se vayan ocurriendo

Los riesgos materializados han estado relacionados principalmente con aspectos técnicos. Primeramente, el R.5. Tuve problemas para instalar OpenBabel en mi máquina por problemas de versiones, por lo que acabé utilizando Google Colab como entorno virtual e instalar ahí algunas librerías y para las primeras experimentaciones con moléculas. Esto tampoco era muy útil a largo plazo puesto que tenía que poder acceder al código fuente para modificarlo, añadir las funcionalidades y compilarlo manualmente para probar los cambios. Por lo que finalmente con ayuda de las guías, se pudo ejecutar localmente. Instantáneamente después, se materializó el riesgo R.3. La falta de conocimiento ante una librería tan grande ya existente retrasó un poco el proceso de modificación del código.

Finalmente, se consiguieron solventar los riesgos manifestados mediante

ID	Riesgo	Causa	Plan de acción
R.1	Pérdida de documentación y/o código	Eliminación accidental de ficheros clave	Realizar copias de seguridad y herramientas de control de versiones para su recuperación.
R.2	Fallo en el hardware de trabajo	Fallo irreparable por edad de uso, sobrecalentamiento, golpe o rotura.	Adquisición de nuevo portátil. Instalación y configuración del entorno de trabajo cuanto antes.
R.3	Falta de conocimiento de la librería de código Openbabel	Nula experiencia con toolkits químicos y sus prestaciones, en especial con OpenBabel	Invertir tiempo en el estudio y comprensión del código base.
R.4	Cambios inesperados y tareas nuevas no contempladas	Planificación poco adecuada a las necesidades del proyecto	Replanificación de las tareas existentes y nuevas, con una adecuada asignación de tiempo
R.5	Problemas en la instalación y configuración de las herramientas	Incompatibilidad de versiones, la máquina en donde se va a realizar la instalación o complejidad del proceso	Consulta de las guías de instalación, búsqueda de herramientas alternativas viables o uso de máquinas virtuales/entornos de desarrollo distintos
R.6	Falta de feedback directo con el tutor	Incompatibilidades horarias, o enfermedad de alguna de las partes	Agendar una reunión en otra fecha, y si fuera necesario o viable, enviar el material a revisar/preguntas por correo para feedback asíncrono.
R.7	Pérdida muy prolongada del contacto con el tutor	No hay respuesta a los correos enviados	Buscar otra forma de comunicación para retomar el contacto
R.8	Los objetivos del proyecto no son realistas	Se ha infravalorado el alcance del proyecto	Realizar una planificación coherente con el tiempo y recursos disponibles. Si fuera necesario, adaptar el contenido del trabajo original y proponer los cambios
R.9	Los resultados del cambio en el dibujado finalmente no son satisfactorios o los esperados	A la tutora (y los químicos colaboradores) no les gusta el resultado	Recibir feedback periódicamente de los cambios realizados y opiniones sobre los resultados intermedios.
R.10	Dificultades para crear un sistema canónico viable para cualquier molécula de entrada	La tarea de programación para llevar esto a cabo es más compleja de lo que se creía	Proponer, aunque sea un inicio de canonización y detallar formalmente las reglas que lo definen. O, sabiendo cómo funcionan internamente las librerías, buscar un sistema canónico más sencillo en términos de programación.
R.11	Imposibilidad de acceso al servicio de Overleaf para la redacción de la memoria	Caída temporal de los servidores	Si es muy urgente, uso de Visual Studio Code para compilado local de la documentación.
R.12	Finalización del uso gratuito de alguna herramienta utilizada	Conclusión de las licencias gratuitas o periodos de prueba	Extensión de la licencia en caso de ser posible o búsqueda de herramientas alternativas parecidas/compatibles.

Figura 3.1: Riesgos del proyecto, causas, y planes de actuación

	Probabilidad				
Impacto	Muy improbable (0,1)	Poco probable (0,3)	Moderada (0,5)	Probable (0,7)	Casi seguro (0,9)
Muy bajo					
Bajo	R.11	R.12	R.4		
Medio	R.6	R.9	R.8		
Alto	R.1, R.7			R.3	
Muy alto	R.2		R.5, R.10		

Figura 3.2: Matriz de probabilidad-impacto de riesgos

los planes de actuación descritos en cada uno de ellos.

Capítulo 4

Diseño e Implementación

Aqui meter los diagramas de clases y todos los metodos que he ido añadiendo/modificando En otro apartado, explicar el sistema de canonicacion (y los cambios en el dibujado, esto no se si es mejor directamente en resultados, puesto que tendré que poner fotos de cómo ha quedado, y explicarlo sin fotos es medio raro) al que he llegado y sus reglas. Ya en la seccion de experimentacion, expondré los resultados.

O quizas mejor, dejar esta seccion solamente para el diseño. Y la que se llama ahora mismo 'experimentacion', llamarla 'Implementacion y resultados'

Capítulo 5

Experimentación

Aquí la idea es ir poniendo las pruebas que vaya haciendo de las moléculas, y lo que vaya descubriendo. Probablemente exponer aquí tb el método o reglas de canonización a las que llegue.

Bibliografía

- [1] William J. Wiswesser. “107 Years of Line-Formula Notations (1861-1968)”. en. En: *Journal of Chemical Documentation* 8.3 (ago. de 1968), págs. 146-150. ISSN: 0021-9576, 1541-5732. DOI: 10.1021/c160030a007. URL: <https://pubs.acs.org/doi/abs/10.1021/c160030a007>.
- [2] Reiner Luckenbach. “The Beilstein Handbook of Organic Chemistry: the first hundred years”. en. En: *Journal of Chemical Information and Computer Sciences* 21.2 (mayo de 1981), págs. 82-83. ISSN: 0095-2338. DOI: 10.1021/ci00030a006. URL: <https://pubs.acs.org/doi/abs/10.1021/ci00030a006>.
- [3] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. en. En: *Journal of Chemical Information and Modeling* 28.1 (feb. de 1988), págs. 31-36. ISSN: 1549-9596. DOI: 10.1021/ci00057a005. URL: <https://pubs.acs.org/doi/abs/10.1021/ci00057a005> (visitado 20-11-2022).
- [4] Johann Gasteiger y Thomas Engel. *Chemoinformatics: A Textbook*. en. John Wiley & Sons, dic. de 2006. ISBN: 978-3-527-60650-4.
- [5] Andrew R. Leach y V. J. Gillet. *An Introduction to Chemoinformatics*. en. Google-Books-ID: 4z7Q87HgBdwC. Springer, sep. de 2007. ISBN: 978-1-4020-6291-9.
- [6] Johann Gasteiger. “The Scope of Chemoinformatics”. en. En: *Handbook of Chemoinformatics*. Ed. por Johann Gasteiger. Weinheim, Germany: Wiley-VCH Verlag GmbH, mayo de 2008, págs. 3-5. ISBN: 978-3-527-61827-9 978-3-527-30680-0. DOI: 10.1002/9783527618279.ch1. URL: <https://onlinelibrary.wiley.com/doi/10.1002/9783527618279.ch1>.
- [7] Nathan Brown. “Chemoinformatics—an introduction for computer scientists”. en. En: *ACM Computing Surveys* 41.2 (feb. de 2009), págs. 1-38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/1459352.1459353. URL: <https://dl.acm.org/doi/10.1145/1459352.1459353>.

- [8] Stephen R. Heller et al. "InChI, the IUPAC International Chemical Identifier". En: *Journal of Cheminformatics* 7.1 (mayo de 2015), pág. 23. ISSN: 1758-2946. DOI: 10.1186/s13321-015-0068-4. URL: <https://doi.org/10.1186/s13321-015-0068-4>.
- [9] Thomas Engel y Johann Gasteiger. *Applied Chemoinformatics. Achievements and Future Opportunities*. en. Wiley-VCH, 2018. ISBN: 978-3-527-34201-3.
- [10] Mario Krenn et al. "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation". en. En: *Machine Learning: Science and Technology* 1.4 (dic. de 2020), pág. 045024. ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba947. URL: <https://iopscience.iop.org/article/10.1088/2632-2153/aba947>.
- [11] Kevin Darza. *¿Cómo calcular la depreciación de un equipo de cómputo?* Ago. de 2022. URL: <https://www.oliversoft.mx/depreciacion-de-un-equipo-de-computo/> (visitado 25-05-2023).
- [12] Mario Krenn et al. "SELFIES and the future of molecular string representations". en. En: *Patterns* 3.10 (oct. de 2022). arXiv:2204.00056 [physics], pág. 100588. ISSN: 26663899. DOI: 10.1016/j.patter.2022.100588. URL: <http://arxiv.org/abs/2204.00056> (visitado 20-11-2022).
- [13] Alston Lo et al. *Recent advances in the Self-Referencing Embedding Strings (SELFIES) library*. arXiv:2302.03620 [physics]. Feb. de 2023. DOI: 10.48550/arXiv.2302.03620. URL: <http://arxiv.org/abs/2302.03620> (visitado 26-02-2023).
- [14] ICIQ. Prof. Mónica H. Pérez-Temprano, Research Group. en. URL: https://www.iciq.org/research/research_group/dr-monica-h-perez-temprano/section/research_overview/ (visitado 10-03-2023).
- [15] PubChem. *PubChem*. en. URL: <https://pubchem.ncbi.nlm.nih.gov> (visitado 01-02-2023).
- [16] SciFinder. *SciFinder-help*. en. URL: https://scifinder-n.cas.org/help/#t=Working_with_Search_Results%5C%2FAll_Answer_Types_screen.htm (visitado 01-02-2023).
- [17] SigmaAldrich. *SigmaAldrich - About us*. es. URL: <https://www.sigmaaldrich.com/ES/es/life-science/about-us/expertise> (visitado 01-02-2023).

Apéndice A

Tabla comparativa del set completo de moléculas

Tabla A.1: Tabla de índices con los nombres de las moléculas de la Tabla A.2

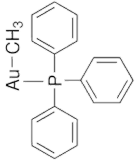
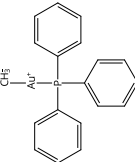
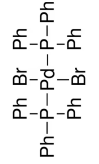
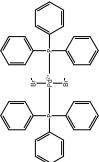
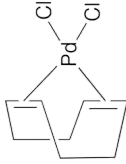
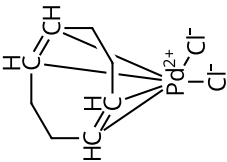
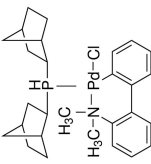
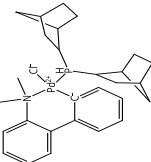
Id	Nombre del compuesto
1	Methyl(triphenylphosphine)gold(I)
2	trans-Dibromobis(triphenylphosphine)palladium(II)
3	Dichloro(1,5-cyclooctadiene)palladium(II)
4	SK-CC 01A
5	Bis[μ-chloro[5-hydroxy-2-[1-(hydroxyimino)ethyl]phenyl]palladium]
6	(SP-4-3)-(3,5-Dichloro-2,4,6-trifluorophenyl)iodobis (triphenylarsine)palladium
7	Palladium,(2,2'-bipyridine- <i>k</i> N1, <i>k</i> N1')[(2,2-dimethyl-1,2-ethanediyl)-1,2-phenylene]fluoro(4-methylbenzenesulfonamidato- <i>k</i> N)-, (OC-6-35)
8	Dibromo(1,2-dimethoxyethane)nickel(II)
9	Bis(triphenylphosphine)ruthenium(II) dicarbonyl chloride
10	Chloro(trimethylphosphine)gold(I)
11	Chloro[tris(para-trifluoromethylphenyl)phosphine]gold(I)
12	Chloro(dimethylsulfide)gold(I)

Continúa en la siguiente página

Tabla A.1 – Continuación de la página anterior

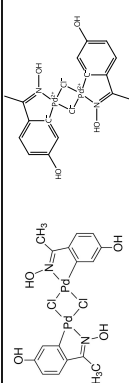
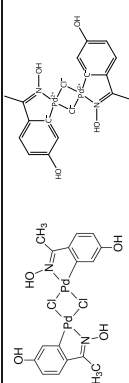
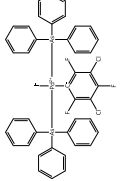
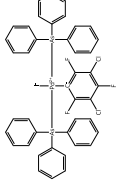
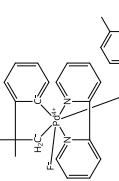
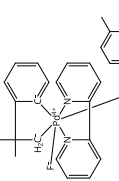
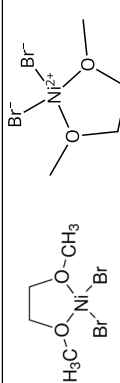
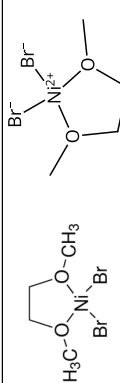
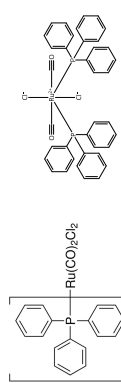
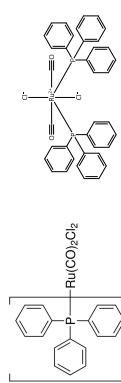
Id	Nombre del compuesto
13	Chloro(methyldiphenylphosphine)gold(I)
14	Chloro[diphenyl(o-tolyl)phosphine]gold(I)
15	Chloro[(1,1'-biphenyl-2-yl)di-tert-butylphosphine]gold(I)
16	(Acetonitrile)[(2-biphenyl)di-tert-butylphosphine] gold(I)hexafluoroantimonate
17	Chloro[2-di-tert-butyl(2',4',6'-triisopropylbiphenyl)phosphine] gold(I)
18	Chloro[2-dicyclohexyl(2',4',6'-trisopropylbiphenyl)phosphine]gold(I)
19	BisPhePhos XD gold(I) chloride
20	Chloro[di(1-adamantyl)-2-dimethylaminophenylphosphine]gold(I)
21	Dichloro(DPPE)digold(I)
22	Dichloro[(±)-BINAP]digold(I)
23	Bis(chlorogold(I)) [1,1'-bis(diphenylphosphino)ferrocene]
24	[(IMes)AuCl]
25	[(IPr)AuCl]
26	IPrAuNTf ₂
27	DPPF
28	Dicarbonylcyclopentadienyliodoiron(II)
29	(OC-6-11')-Bis[2,6-di(2-pyridinyl- <i>k</i> N)phenyl- <i>k</i> C]iron

Tabla A.2: Tabla extendida para el set de datos de 30 moléculas. Contiene la cadena SMILES extraída de Sigma-Aldrich (SA), la cadena SMILES extraída de SciFinder (SF), y las imágenes de las respectivas bases de datos (SA y SF)

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
1	<chem>C[Au].c1ccc(cc1)P(c2ccccc2)c3ccccc3</chem>	<chem>[Au+](([CH3-])[P](C=1C=CC=CC1)(C=2C=CC=CC2)C=3C=CC=CC3</chem>		
2	<chem>Br[Pd]Br.c1ccc(cc1)P(c2ccccc2)c3ccccc3.c4ccc(cc4)P(c5ccccc5)c6ccccc6</chem>	<chem>[Br-][Pd+2]([Br-])([P](C=1C=CC=CC1)(C=2C=CC=CC2)C=3C=CC=CC3)[P](C=4C=CC=CC4)(C=5C=CC=CC5)C=6C=CC=CC6</chem>		
3	<chem>Cl[Pd]Cl.C1CC=CCCC=C1</chem>	<chem>[Cl-][Pd+2]123([Cl-])[CH]=4CC[CH]3=[CH]2CC[CH]41</chem>		
4	<chem>C1C[C@@H]2C[C@H]1CC2PC3C[C@@H]4CC[C@H]3C4.CN(C)c5ccccc5c6ccccc6[Pd]Cl</chem>	<chem>[Cl-][Pd+2]1([C-]=2C=CC=CC2C=3C=CC=CC3[N]1(C)C)[PH](C4CC5CCC4C5)C6CC7CCC6C7</chem>		

Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
5	<chem>C\C(=N/O)c1ccc(O)cc1[Pd]Cl1</chem> <chem>.C\C(=N/O)c2ccc(O)cc2[Pd]Cl</chem>	<chem>OC=1C=CC=2C(=[N](O)[Pd+2]3</chem> <chem>([Cl-])[Pd+2]4([Cl-]3)[C-]=5</chem> <chem>C=C(O)C=CC5C(=[N]4O)C)[C-]2C1)C</chem>		
6	No se encontró el compuesto en Sigma-Aldrich	<chem>FC=1C(Cl)=C(F)[C-](=C(F)Cl)Cl)[Pd+2]</chem> <chem>([F-])([As])(C=2C=CC=CC2)(C=3C=CC=C</chem> <chem>C3)C=4C=CC=CC4)[As](C=5C=CC=CC5)</chem> <chem>(C=6C=CC=CC6)C=7C=CC=CC7</chem>		
7	No se encontró el compuesto en Sigma-Aldrich	<chem>O=S(=O)([NH-][Pd+4]12([F-])([C-</chem> <chem>]=3C=CC=CC3C(C)(C)[CH2-]1)</chem> <chem>[N]=4C=CC=CC4C=5C=CC=C[N]</chem> <chem>52)C6=CC=C(C=C6)C</chem>		
8	<chem>Br[Ni]Br.COCCOC</chem>	<chem>[Br-][Ni+2]1([Br-])O(C)CCO1C</chem>		
9	<chem>Cl[Ru](Cl)(C#[O])(C#[O])([PH](c1ccc</chem> <chem>cc1)(c2cccc2)c3cccc3)[PH](c4ccc</chem> <chem>cc4)(c5cccc5)c6cccc6</chem>	<chem>O#C[Ru+2]([Cl-])([Cl-</chem> <chem>])(C#O)([P](C=1C=CC=CC1)</chem> <chem>(C=2C=CC=CC2)C=3C=CC=CC3)[P]</chem> <chem>(C=4C=CC=CC4)(C=5C=CC=CC5)</chem> <chem>C=6C=CC=CC6</chem>		

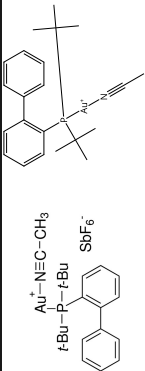
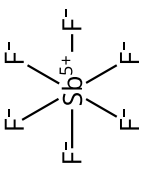
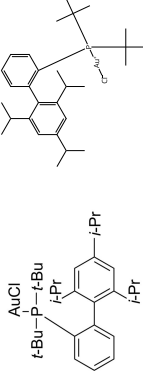
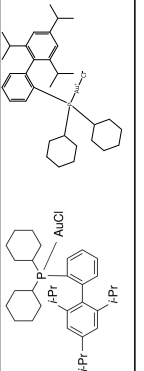
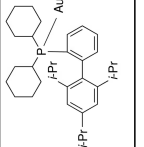
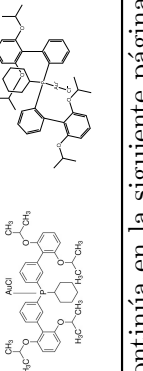
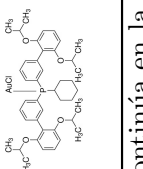
Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
10	<chem>Cl[Au].CP(C)C</chem>	<chem>[Cl-][Au+][P](C)(C)C</chem>		
11	<chem>Cl[Au].FC(F)(F)c1ccc(cc1)P(c2ccc(cc2)C(F)(F)F)c3ccc(cc3)C(F)(F)F</chem>	<chem>FC(F)(F)C1=CC=C(C(C=C1)[P]([Au+][Cl-])(C2=CC=C(C(C=C2)C(F)F)F)C3=CC=C(C(C=C3)C(F)(F)F)F</chem>		
12	<chem>Cl[Au].CSC</chem>	<chem>[Cl-][Au+][S](C)C</chem>		
13	<chem>Cl[Au].CP(c1cccc1)c2cccc2</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1)(C=2C=CC=CC2)C</chem>		
14	<chem>Cl[Au].Cc1cccc1P(c2cccc2)c3cccc3</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1)(C=2C=CC=CC2)C=3C=CC=CC3C</chem>		
15	<chem>Cl[Au].CC(C)(C)P(c1cccc1-c2cccc2)C(C)(C)C</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1C=2C=CC=CC2)(C(C)(C)C)C(C)(C)C</chem>		

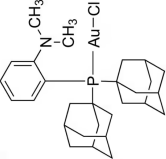
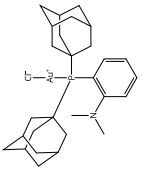
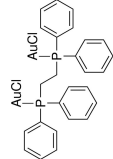
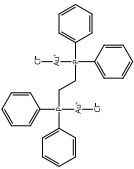
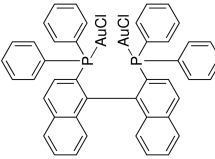
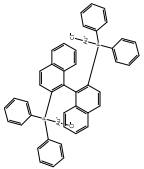
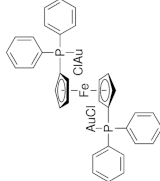
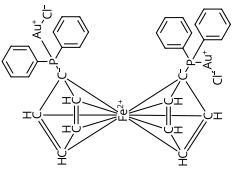
Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
16	<chem>[Au+].CC#N.F[Sb-](F)(F)(F)F. CC(C)(C)P(c1cccc1- c2cccc2)C(C)(C)C</chem>	<chem>[F-][Sb+5]([F-])([F-])([F-])([F-])([F-])[F-]. C([#N])[Au+][P](C=1C=CC=CC1C= 2C=CC=CC2)(C(C)(C)C(C)C(C)C)C</chem>		
17	<chem>CC(C)c1cc(C(C)C)c(c(c1)C(C)C)-c2 cccc2[PH]([Au]Cl)(C(C)(C)C)C(C)C</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1C=2C (=CC(=CC2C(C)C)C(C)C(C)C)C(C)C (C)C)C(C)C)C</chem>		
18	<chem>Cl[Au].CC(C)c1cc(C(C)C)c(c(c1)C(C) C)-c2cccc2P(C3CCCCC3)C4CCCCC4</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1C=2C (=CC(=CC2C(C)C)C(C)C(C)C)C(C)C CCCC3)C4CCCCC4</chem>		
19	<chem>CC(C)OC(C=CC=C1OC(C)C)=C1C2 =CC(P(C3CCCCC3)O4=CC(O5=C(O C(C)C)C=CC=C5OC(C)C)=CC=C4) =CC=C2.[Au]Cl</chem>	<chem>[Cl-][Au+][P](C=1C=CC=CC1C2=C(OC(C) C)C=CC=C2OC(C)C)(C=3C=CC=CC3C4 =C(OC(C)C)C=CC=C4OC(C)C)C5CCCCC5</chem>		

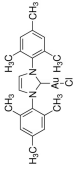
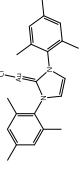
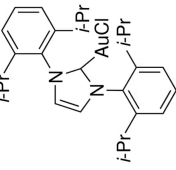
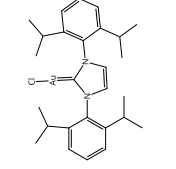
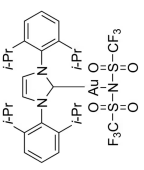
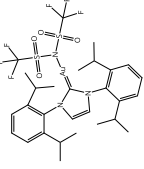
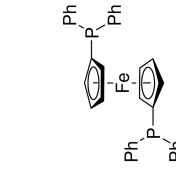
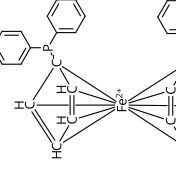
Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
20	<chem>Cl[Au].CN(C)c1cccc1P(C23C C4CC(CC(C4)C2)C3)C56CC7CC(C C(C7)C5)C6</chem>	<chem>[Cl-].[Au+].[P](C=1C=CC=CC1N (C)C)(C23CC4CC(CC(C4)C2)C3) C56CC7CC(CC(C7)C5)C6</chem>		
21	<chem>Cl[Au].Cl[Au].C((CP(c1cccc1) c2cccc2)P(c3cccc3)c4cccc4</chem>	<chem>[Cl-].[Au+].[P](C=1C=CC=CC1) (C=2C=CC=CC2)CC[P]([Au+][Cl-]) (C=3C=CC=CC3)C=4C=CC=CC4</chem>		
22	<chem>Cl[Au].Cl[Au].P(C1=CC=CC=C1)(C2 =C(C3=C(C=CC4=C3C=CC=C4)P (C5=CC=CC=C5)C6=CC=CC=C6)C7 =C(C=CC=C7)C8=CC=CC=C8</chem>	<chem>[Cl-].[Au+].[P](C=1C=CC=CC1)(C=2C =CC=CC2)C3=CC=C4C=CC=CC4=C3 C=5C=6C=CC=CC6C=CC5[P]([Au+][Cl-])(C=7C=CC=CC7)C=8C=CC=CC8</chem>		
23	<chem>[Fe].Cl[Au].Cl[Au].[CH]1[CH] [CH][C]([CH]1)P(c2cccc2)c3 ccccc3.[CH]4[CH][CH][C]([CH]4) P(c5cccc5)c6cccc6</chem>	<chem>[Cl-].[Au+].[P](C=1C=CC=CC1)(C=2C=CC =CC2)[C-]34[CH]5=[CH]6[CH]7=[CH]3[Fe+2] 6789%10%1154[CH]=%12[CH]%11=[CH]%10 [C-]9([CH]%128)[P]([Au+][Cl-])(C=%13 C=CC=CC%13)C=%14C=CC=CC%14</chem>		

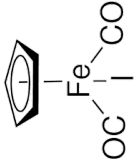
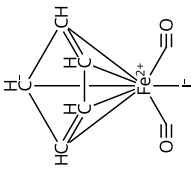
Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
24	<chem>Cl[Au].Cc1cc(C)c(N2[C]N(C=C2)c3c(C)cc(C)cc3C)c(C)c1</chem>	<chem>Cl[Au]=C1N(C=CN1C=2C(=CC(=CC2C(C)C)C=3C(=CC(=CC3C)C)C</chem>		
25	<chem>CC(C)c1cccc(C(C)C)c1N2C=CN(C2[Au]Cl)c3c(cccc3C(C)C)C(C)C</chem>	<chem>Cl[Au]=C1N(C=CN1C=2C(=CC=CC2C(C)C)C(C)C=3C(=CC=CC3C(C)C)C(C)C</chem>		
26	<chem>CC(C)c1cccc(C(C)C)c1N2C=CN(C2=[Au]N(S(=O)(=O)C(F)(F)F)S(=O)(=O)C(F)(F)F)c3c(cccc3C(C)C)C(C)C</chem>	<chem>O=S(=O)(N([Au]=C1N(C=CN1C=2C(=CC=CC2C(C)C)C(C)C=3C(=CC=CC3C(C)C)C(C)C=4C(=CC(=CC4C)C(=O)C(F)(F)F)C(F)(F)F</chem>		
27	<chem>[Fe].[CH]1[CH][CH][C]([CH]1)P(c2cccc2)c3cccc3.[CH]4[CH][CH][C]([CH]4)P(c5cccc5)c6cccc6</chem>	<chem>C=1C=CC(=CC1)P(C=2C=CC=CC2)[C-]34[CH]5=[CH]6[CH]7=[CH]3[Fe+2]6789%10%1154[CH]=%12[CH]%11=[CH]%10[C-]9[P(C=%13C=CC=CC%13)C=%14C=CC=CC%14][CH]%128</chem>		

Continúa en la siguiente página

Tabla A.2 – Continuación de la página anterior

Id	SMILES SA	SMILES SF	Imagen SA	Imagen SF
28	<chem>[Fe]I.[C-]#[O+].[C-]#[O+].[CH]1[CH][CH][CH][CH]1</chem>	<chem>O#C[Fe+2]1234([I-])(C#O)[CH]=5[CH]4=[CH]3[CH-]2[CH]51</chem>		
29	No se encontró el compuesto en Sigma-Aldrich	<chem>C=1C=C[N]2=C(C1)C3=CC=CC=4C=5C=CC=C[N]5[Fe+2]672([C-]34)[C-]=8C(=CC=CC8C=9C=CC=C[N]96)C=%10C=CC=C[N] %107</chem>		