
**UNIVERSIDAD DE TALCA**
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

# Desarrollo de un repositorio de patrones estructurales en proteínas

**CARLOS BERNARDO HERNÁNDEZ ROJAS**

Profesor Guía: RENZO ANGLES ROJAS

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Enero, 2019

# CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.


Curicó, 2022

*Dedicado a mis padres*

## AGRADECIMIENTOS

En primer lugar, me gustaría agradecer sinceramente a mi profesor guía, Dr. Renzo Angles que me entregó todo lo necesario para poder realizar este proyecto de memoria de forma completa. Además, por su paciencia y motivación frente a lo que se estaba desarrollando.

En segundo lugar, quiero agradecer al profesor de Ingeniería en Bioinformática, Dr. Mauricio Arenas, por su tiempo dedicado a las videollamadas concretadas para reunir requisitos necesarios y por ayudarme a organizar una sesión de evaluación de la aplicación con estudiantes suyos de Bioinformática.

En tercer lugar, a los profesores de Ingeniería Civil en Computación que me acompañaron a lo largo de estos años en la Universidad, que me enseñaron conocimientos importantes para que pueda desenvolverme en el futuro como ingeniero.

Por último, agradecer a mis compañeros con los que fui equipo de trabajo en los distintos módulos. A aquellos que hicieron que mi progreso dentro de la Universidad fuese más fácil y hasta más rápido.

# TABLA DE CONTENIDOS

página

# ÍNDICE DE FIGURAS

página

# ÍNDICE DE TABLAS

página

## RESUMEN

En las últimas décadas, gracias a los avances de la ciencia y la tecnología, muchos investigadores han utilizado variadas técnicas para determinar y entender la estructura de moléculas tales como las proteínas, ADN, ARN, entre otras. La información obtenida a partir de estos estudios, ha sido muy útil para describir las interacciones que se dan entre las moléculas, y así poder entender las estructuras y funciones que realizan muchos procesos biológicos importantes.

La cantidad de información disponible acerca de las moléculas es considerable, y va creciendo de forma exponencial día a día. En este sentido, se han creado diversas bases de datos que intentan unificar o integrar información a nivel global. Este es el caso del Protein Data Bank (PDB).

Poder analizar todos los datos de PDB es una tarea compleja y requiere tiempo. Actualmente, esto se lleva a cabo a través de software especializado, el cual suele estar diseñado para cumplir tareas específicas para usuarios expertos. En este sentido, dichas herramientas presentan restricciones de usabilidad y adaptabilidad.

El objetivo general del proyecto descrito en este documento consiste en desarrollar una herramienta de software para la gestión de patrones estructurales disponibles en PDB. Específicamente, se desarrolló una aplicación Web que permite almacenar, explorar y visualizar patrones estructurales proteína-ligando. La principal característica de la herramienta es que los patrones son representados como estructuras en forma de grafo, las cuales pueden ser exploradas a través de diversos metadatos.

El prototipo funcional desarrollado fue validado y evaluado positivamente por un grupo de usuarios del área de bioinformática. El resultado final es una herramienta de interfaz sencilla y fácil de usar, que cambia drásticamente el modo de interacción habitual para la exploración de patrones estructurales.



**Palabras clave** Patrones estructurales - PDB

## ABSTRACT

In recent decades, thanks to advances in science and technology, many researchers have used various techniques to determine and understand the structure of molecules such as proteins, DNA, RNA, among others. The information obtained from these studies has been very useful to describe the interactions that occur between the molecules, and thus be able to understand the structures and functions that many important biological processes perform.

The amount of information available about molecules is considerable, and it grows exponentially day by day. In this sense, various databases have been created that attempt to unify or integrate information globally. This is the case of Protein Data Bank (PDB).

Being able to analyze all PDB data is a complex and time-consuming task. Currently, this is done through specialized software, which is usually designed to perform specific tasks for expert users. In this sense, these tools have usability and adaptability restrictions.

The general objective of the project described in this document is to develop a software tool for managing structural patterns available in PDB. Specifically, a Web application was developed that allows storing, exploring and visualizing protein-ligand structural patterns. The main feature of the tool is that the patterns are represented as graph-shaped structures, which can be explored through various metadata.

The functional prototype developed was validated and positively evaluated by a group of users in the bioinformatics area. The end result is a simple and easy-to-use interface tool, which dramatically changes the usual mode of interaction for the exploration of structural patterns.



**Key words** Structural patterns - PDB

# 1. Introducción

En esta sección se describen los aspectos más generales del proyecto, incluyendo definición del problema, propuesta de solución, hipótesis, objetivos y la metodología de trabajo.

## 1.1. Definición del problema

A lo largo de muchos años, la comunidad científica ha estudiado de diferentes formas el comportamiento de las proteínas con ligandos asociados en los diferentes procesos celulares que se lleva a cabo en todos seres vivos. Es por esto que, conforme el avance de la tecnología y la computación, se han ido ideando nuevas formas de modelar la información obtenida a partir de los estudios previos de las estructuras de las proteínas, sus funciones y como éstas trabajan. Se han ido creando muchas herramientas computacionales que ayudan a los profesionales del área científica a mejorar el entendimiento de lo mencionado anteriormente, el modelado de estructuras y la interacción con los ligandos.

Existen herramientas desarrolladas altamente complejas que requieren de años de estudio para poder manejarlas con fluidez y entenderlas. Hasta la fecha, el uso de herramientas computacionales ha sido clave para el estudio de comportamientos de estas macromoléculas y resulta crucial para la investigación biológica y la búsqueda de nuevas formas de crear y/o modificar fármacos que requieran estrictamente conocer la estructura y comportamiento a nivel molecular de las proteínas junto con los ligandos.

Dicho lo anterior, es un hecho que el campo de estudio de la interacción de las proteínas y ligandos está activo y día a día crece debido a la gran cantidad de variables

a considerar para estas macromoléculas. Cabe destacar que existen fuentes de datos que almacenan la información con respecto a las proteínas que son accesibles para el público general, pero están carentes de herramientas que entreguen información estadística en detalle y no sólo a grandes rasgos.

Actualmente, lo más cercano a un repositorio de patrones estructurales proteína-ligando es Prosite. Prosite es un repositorio de dominios de proteínas, familias de proteínas y sitios funcionales.

En estos momentos, GSP4PDB [6] permite dibujar patrones estructurales, pero no da la opción de almacenar estos diseños para luego explorar y comparar. Por lo tanto, una vez que se cierra la página donde se tiene abierta la aplicación, éste se pierde para siempre. En la Figura 1.1 se muestra una ejemplo de un patrón estructural proteína-ligando en GSP4PDB.

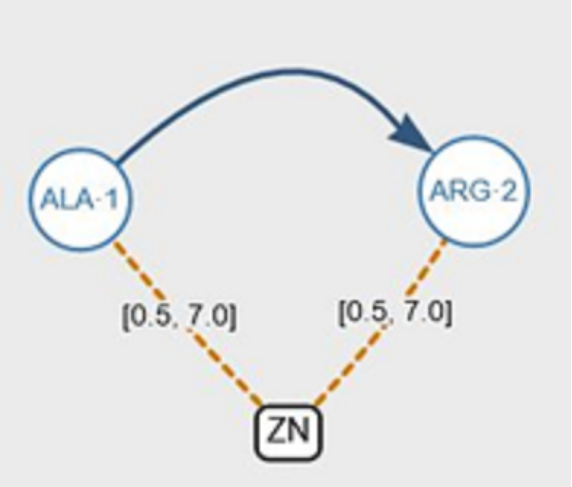



Figura 1.1: Representación basada en grafos de un patrón estructural proteína-ligando, dibujado con la aplicación GSP4PDB

No existe un repositorio en donde los patrones se representen usando grafos y que sea fácil almacenarlos, visualizarlos y explorarlos. No hay alguna aplicación que permita explorar patrones estructurales proteína-ligando, que sea capaz de dar a conocer sus similitudes y diferencias entre éstos, que entregue información acerca de su clasificación (según función), organismos (seres vivos) en los que están presentes,

que patrones presentan composición similar salvo por algunos detalles, sean estos ligandos, orden secuencial de amino ácidos, unidades de distancia para las uniones entre nodos del grafo que representa al patrón, etc.

## 1.2. Propuesta de solución

Dada la problemática expuesta anteriormente, se propone desarrollar una herramienta de software que permita almacenar y visualizar patrones estructurales, para que los profesionales del área de Bioinformática usen esta aplicación que automatiza el proceso de exploración y así disminuir el tiempo dedicado al estudio de los patrones estructurales proteína-ligando.

La exploración de patrones estructurales incluye las siguientes tareas:

- Listar: Dar a conocer la representación en grafos de todos los patrones estructurales proteína-ligando que estén almacenados en la base de datos, dado ciertos filtros de búsqueda proporcionados por el usuario.
- Filtrar: Filtros de búsqueda sean éstos por ligando y aminoácidos presentes, familia y clasificación de proteína a la cual pertenece el patrón, identificador de proteína, organismo de proteína y clasificación CATH.
- Buscar: Examinar en la base de datos si existen patrones estructurales proteína-ligando.

Dadas estas funcionalidades, se busca visualizar la representación de los patrones encontrados en forma de grafo (nodos unidos entre sí mediante aristas) y así, obtener información estadística de la frecuencia de aparición de patrones estructurales proteína- ligando dentro de las proteínas.

## 1.3. Hipótesis

Es posible desarrollar una aplicación de software web que permita almacenar, visualizar y explorar patrones estructurales proteína-ligando.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Diseñar e implementar un repositorio de patrones estructurales proteína-ligando.

### 1.4.2. Objetivos específicos

- Visualización: Visualizar patrones estructurales usando una interfaz gráfica.
- Exploración: Explorar patrones estructurales de acuerdo a filtros de búsqueda.
- Almacenamiento: Crear, leer, editar y eliminar patrones estructurales.
- Evaluación: Evaluación de usabilidad del software siguiendo el estándar ISO 9216.

## 1.5. Metodología de desarrollo del proyecto

Primero, se hizo una investigación del contexto referente a las proteínas, patrones de interacción proteína-ligando, interfaces de usuarios basada en facetas y de las herramientas de consulta de patrones estructurales proteína-ligando en internet.

Segundo, se identificaron y definieron los requisitos para el almacenamiento, exploración y visualización de patrones estructurales proteína-ligando. En primera instancia de consulta con un experto del área de la Bioinformática para poder desarrollar una aplicación que permita dichas tres grandes funcionalidades. Luego, se determinó una forma de modelar patrones estructurales de forma clara y que reúna la información más importante para visualizar.

Tercero, se diseñó la interfaz gráfica que tendrá la aplicación. Se empieza diseñando bosquejos de prueba (mock-ups) para luego determinar la interfaz definitiva de la aplicación. Se elige cómo serán los distintos elementos de la interfaz y cómo se distribuirán. Se deben cumplir todos los requisitos especificados empleando tecnologías web.

Cuarto, se implementa un prototipo funcional de la aplicación. Se crea una base de datos para la aplicación. Luego, se implementa la lógica de la aplicación, se integra con la interfaz y se conecta con la base de datos para llevar a la práctica el

almacenamiento y exploración de patrones estructurales. Seguido de esto, se desarrollan los distintos métodos necesarios para llevar a cabo la visualización de patrones estructurales en la interfaz de la aplicación.

Quinto, se realizan las pruebas pertinentes por parte del desarrollador para corroborar que los datos arrojados por la aplicación en funcionamiento son los correctos.

Sexto, se evalúa el funcionamiento y usabilidad de la aplicación. Se preparan los instrumentos de evaluación: una guía de actividades y un cuestionario de evaluación. Se aplica a un conjunto definido de usuarios. Éstos realizan las actividades y responden el cuestionario.

Por último, se analizan los resultados de la evaluación y se obtienen conclusiones.

# 2. Conceptos básicos

En este capítulo se describirán los conceptos o temas más importantes asociados al desarrollo del proyecto. Específicamente, Protein Data Bank, proteínas, aminoácidos, ligandos, patrones de interacción proteína-ligando, interfaces de usuario basada en facetas y herramientas de consulta de patrones estructurales.

## 2.1. Protein Data Bank (PDB)

El PDB o banco de información de proteínas es un repositorio que almacena información sobre las formas tridimensionales de las proteínas, ácidos nucleicos y ensamblajes complejos que ayuda a estudiantes e investigadores de todo el mundo a comprender todos los aspectos de la biomedicina y la agricultura, desde la síntesis de proteínas hasta la salud y la enfermedad.

Fue creado en 1971 por Walter Hamilton y Edgar Meyer como respuesta al gran interés mostrado por parte de cristalógrafos, bioinformáticos y químicos por visualizar y analizar estructuras de proteínas. En un principio, fue lanzado con tan sólo 7 proteínas, pero desde aquel entonces hasta la fecha, el número de proteínas ha crecido de forma exponencial, semana a semana va incrementando sus registros y ya se tiene registro alrededor de 142220 estructuras macromoleculares de proteínas.

Cada estructura macromolecular de proteína se identifica por un código único de 4 caracteres alfanumérico (PDB ID) y el resto corresponde a información primaria de la proteína. Además, PDB almacena información de ADN y ARN e híbridos. Por ejemplo, información como las coordenadas espaciales de los átomos de los aminoácidos y los átomos de los ligandos. Añade también información del título, autor, funciones biológicas, organismos en los que dicha proteína se encuentra presente.

El PDB dispone de una gran cantidad de información de proteínas a disposición de la población, expertos, profesionales de bioinformática, científicos y es un recurso clave en el estudio para campos como la bioquímica, medicina, física, bioinformática y estudiantes de todos los niveles que desean hacer uso de este banco para obtener información de fácil acceso acerca de las proteínas.

## 2.2. Proteínas, aminoácidos y ligandos

Las proteínas son las macromoléculas más versátiles en los sistemas vivos y cumplen funciones cruciales en prácticamente todos los procesos biológicos. Funcionan como biorreguladoras (forman parte de las enzimas), transportan y almacenan otras moléculas como el oxígeno, brindan protección para nuestro sistema inmunológico, generan movimiento e impulsos nerviosos, controlan el crecimiento, etc[1].

En términos generales, una proteína está compuesta de una secuencia de aminoácidos (Figura 2.1). Los aminoácidos son los bloques de construcción monoméricos de las proteínas. El átomo de carbono alfa (C$\alpha$), que está adyacente al grupo carboxilo, está unido a cuatro grupos diferentes: un grupo amino (NH2), un grupo carboxilo (COOH), un átomo de Hidrógeno (H) y un grupo variable, llamado cadena lateral. [8].

La mayor cantidad de proteínas conocidas actualmente están formadas únicamente por 20 aminoácidos diferentes. Existen otros 150 aminoácidos que no forman parte de las proteínas, pero para este proyecto solo se va a trabajar con los 20 primeros. Por lo general, el número de aminoácidos que conforman a una proteína se encuentra alrededor de 100 y 300. Los enlaces que son responsables de la unión entre éstos para formar las proteínas se les denomina enlace peptídico. Dicho enlace es un enlace amida que se forma entre el grupo carboxilo de un aminoácido con el grupo amino de otro, con la eliminación de una molécula de agua.

Los aminoácidos que son tratados en este proyecto son los siguientes: Alanina, Arginina, Asparginina, Ácido Aspártico, Cisteína, Glutamina, Ácido Glutámico, Glicina, Histidina, Isoleucina, Leucina, Lisina, Metionina, Fenilalanina, Prolina, Serina, Treonina, Triptófano, Tirosina y Valina.

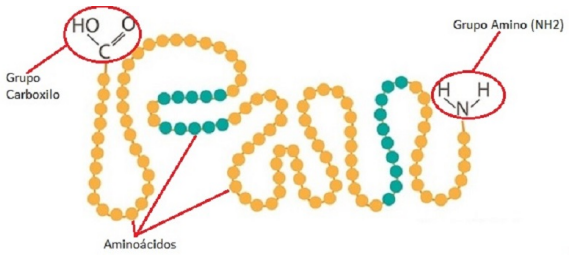



Figura 2.1: Proteína compuesta por cadena lineal de aminoácidos, un grupo carboxilo y un grupo amino.[3]

Las proteínas presentan cuatro niveles de organización de su estructura. La estructura primaria se refiere a la secuencia de aminoácidos que están unidos por enlaces peptídicos para formar cadenas polipeptídicas. Las cadenas polipeptídicas pueden plegarse en estructuras regulares tales como la hélice alfa y la lámina beta. Estas subestructuras conforman la estructura secundaria de la proteína. La estructura terciaria se refiere a la organización tridimensional completa de una cadena polipeptídica. Finalmente, si una proteína particular está formada por más de una cadena polipeptídica, la estructura completa se designa como la estructura cuaternaria.

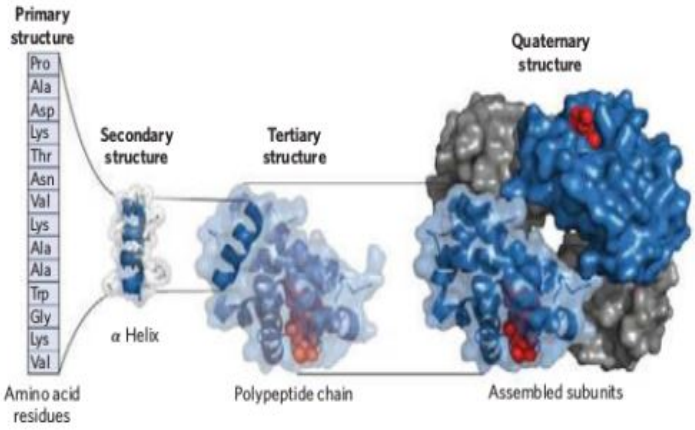



Figura 2.2: Estructura de las proteínas [15].

La función de las proteínas se ve definida según su estructura molecular. Las proteínas, para poder llevar a cabo su función dentro de los organismos, trabajan en conjunto con moléculas llamadas ligandos, los cuales, se unen a las cavidades de las proteínas y, una vez unidos, afectan la funcionalidad de una proteína.

Los ligandos son pequeñas moléculas (como el oxígeno, metales en general) que al interactuar con las proteínas determinan la función de estas últimas. Dicho de otra forma, la unión entre las proteínas y los ligandos son complementarias (Figura 2.3), ya que las proteínas requieren de éste último para poder cumplir sus funciones dentro del organismo. Los ligandos pueden interactuar, enlazar y controlar la función biológica de las proteínas. [14]

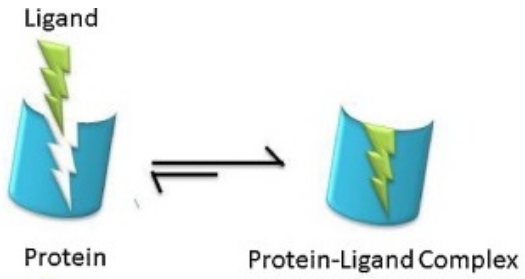



Figura 2.3: Representación de un ligando uniéndose a una proteína. [11]

## 2.3. Patrones de interacción proteína-ligando

La noción de patrón estructural se usa para describir una estructura tridimensional o forma que ocurre en la estructura secundaria de una proteína (Figura 2.2). El mismo patrón estructural puede ocurrir en un grupo de proteínas con una frecuencia dada y que satisfaga criterios específicos (por ejemplo, distancia atómica, composición, conectividad, etc.). Hay muchos tipos de patrones estructurales, pero para este contexto se limitará sólo a aquellos que son del tipo proteína-ligando.

Un patrón estructural proteína-ligando se define como la combinación de un ligando con un grupo de aminoácidos, cuya distribución de estos elementos podría estar determinada por tres tipos de relaciones de unión: distancia entre dos aminoácidos, distancia entre un aminoácido y un ligando, y el orden de precedencia (en la secuencia) de un aminoácido con respecto a otro aminoácido [14]. Por ejemplo, un tipo C2H2 Zinc Finger puede ser descrito por un patrón estructural proteína-ligando, en donde el átomo de Zinc (correspondiente al ligando) se encuentra rodeado por dos cisteínas y dos histidinas (correspondiente a los aminoácidos). La forma de representar en forma de grafo un patrón estructural proteína-ligando, usando el ejemplo de Zinc Finger es la siguiente:

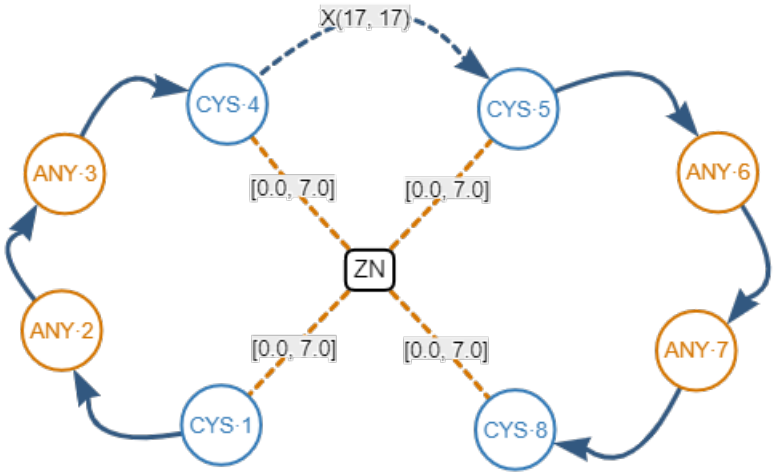



Figura 2.4: Patrón estructural del tipo C2H2 Zinc Finger representado en un grafo [14].

Los patrones estructurales proteína-ligando poseen su propia metadata tratada para este proyecto y son las siguientes:

- Protein ID: Identificador de proteína. Para un patrón estructural, es el identificador de proteína en la cual dicho patrón está contenido. Puede estar en más de una proteína.
- Protein Classification: Clasificación de proteína. Para un patrón estructural, corresponde a las funciones biológicas que cumple dicho patrón estructural en el accionar de una proteína.
- Pattern Family: Familia de patrón. La familia de patrones estructurales corresponde a conjuntos de patrones que comparten aspectos en común, pudiendo ser determinante la relación con ligando, su estructura.
- Protein Organism: Organismo de proteína. Corresponde a organismos (seres vivos o células procariotas[1] y eucariotas[2]) cuya estructura se basa en el patrón

[1]Es un organismo unicelular sin núcleo, es decir, cuyo material genético se encuentra disperso en el citoplasma

[2]Son células que tienen su material genético encerrado dentro de una doble membrana, la en-

estructural especificado.

- Ligand: Ligando. El ligando que está unido al grupo de aminoácidos del patrón estructural.
- Aminoacids: Aminoácidos. Grupo de aminoácidos que componen el patrón estructural.

## 2.4. Interfaces de usuarios basada en facetas

La búsqueda facetada se refiere a categorías utilizadas para caracterizar información o artículos de una colección. Para este proyecto, se utilizan facetas jerárquicas, que son un conjunto de etiquetas. Para estas facetas, existen etiquetas que están incluidas dentro de ellas, por ejemplo: la faceta jerárquica *aminoácido* contiene etiquetas como *Alanina*, *Histidina*, *Triptófano*, etc.

En una interfaz de búsqueda facetada, las etiquetas se asignan a los elementos de la colección. Entonces, cuando se selecciona una etiqueta, todos los elementos de la colección que se les ha asignado dicha etiqueta se *recuperan*. En otras palabras, el proceso de recuperación es equivalente a obtener una nueva colección, la cual solo se compone de elementos que tienen asignada la o las etiquetas seleccionadas. Cuando se seleccionan más de una etiqueta, el sistema crea un conjunto de elementos disyuntivos, vale decir, se recuperan elementos que se caractericen por una u otra etiqueta. [4]

Por ejemplo, en la plataforma **Hammings**[3] se venden, compran vehículos y partes de vehículos clásicos. Hammings dispone de una interfaz de usuario basada en facetas para realizar consultas de acuerdo a facetas jerárquicas. Las facetas jerárquicas para el sitio son: marcas de vehículo, modelos de vehículo, marca de partes de vehículo, entre otras. Las etiquetas seleccionadas de la jerarquía *Modelo de vehículo* para este caso (Figura 2.5) son *1900 SSC* y *2000*. Se puede observar que se han recuperado dos vehículos cuyo modelo corresponde o a *1900 SSC* o *2000*. Además, es claro ver que el sistema creó un conjunto de elementos disyuntivos, porque cada vehículo pertenece a un sólo modelo. Entonces, la colección recuperada corresponde

voltura nuclear

[3]Sitio web: https://www.hammings.com/

a vehículos cuyo modelo sea *1900 SSC* sumado de aquellos vehículos cuyo modelo es *2000*.

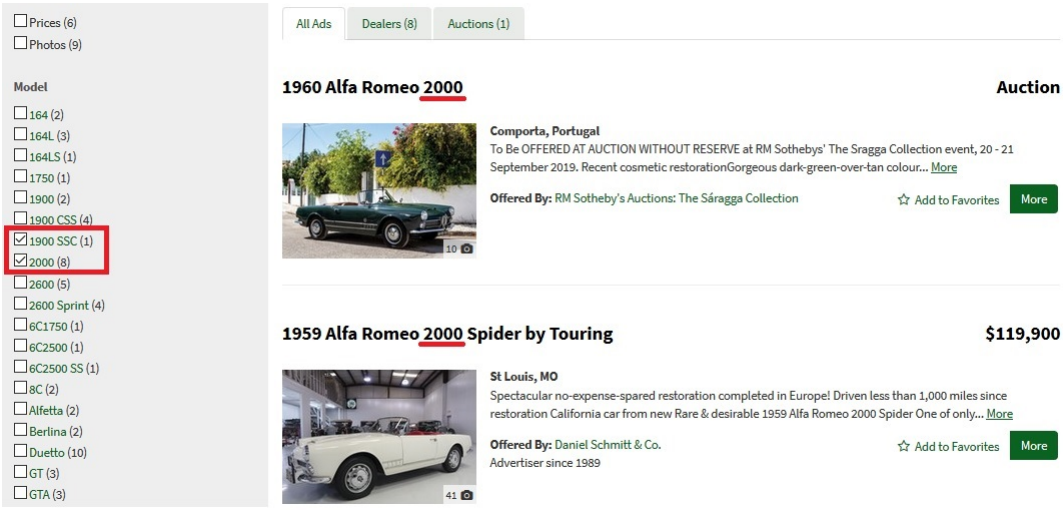



Figura 2.5: Ejemplo de búsqueda de contenido mediante el uso de interfaces de usuario basada en facetas.

Las interfaces de usuario basada en facetas describe interfaces de recuperación de información, la cual combina conceptos de *browse*[4] y *keyword*[5] en una única interfaz que permite al usuario la capacidad de restringir fácilmente los resultados de las consultas de búsqueda. La búsqueda facetada ofrece a los usuarios una visión general de los elementos disponibles de la colección y les ayuda a evitar conjuntos de resultados vacíos y también a lo que podría ser una abrumadora variedad de resultados.

Para lograr una interfaz facetada acorde con los elementos que se desea explorar, cualquiera sea la aplicación, desde el principio es extremadamente importante investigar cómo los usuarios tienen la intención de buscar y aplicar los datos que encuentran y usar esto como base para definir las etiquetas que mejor describan las características de los elementos.[2]

Para el caso de la aplicación web GSPRepository para almacenamiento, visualización y exploración de patrones estructurales proteína-ligando, la implementación

[4]Hojear. Explorar vistas de un mismo dominio.
[5]Palabra clave. Tiene un significado especial dado el contexto.

de interfaz de usuario basada en facetas resulta una herramienta fundamental en la exploración de patrones estructurales proteína-ligando, puesto que permite al usuario encontrar patrones estructurales que coincidan con aspectos especificados en los filtros, analizar cómo se relacionan los patrones estructurales, qué características comparten entre sí y todo aquello gracias a la búsqueda facetada.

En el siguiente ejemplo se puede observar cómo funciona esta búsqueda facetada para el repositorio:

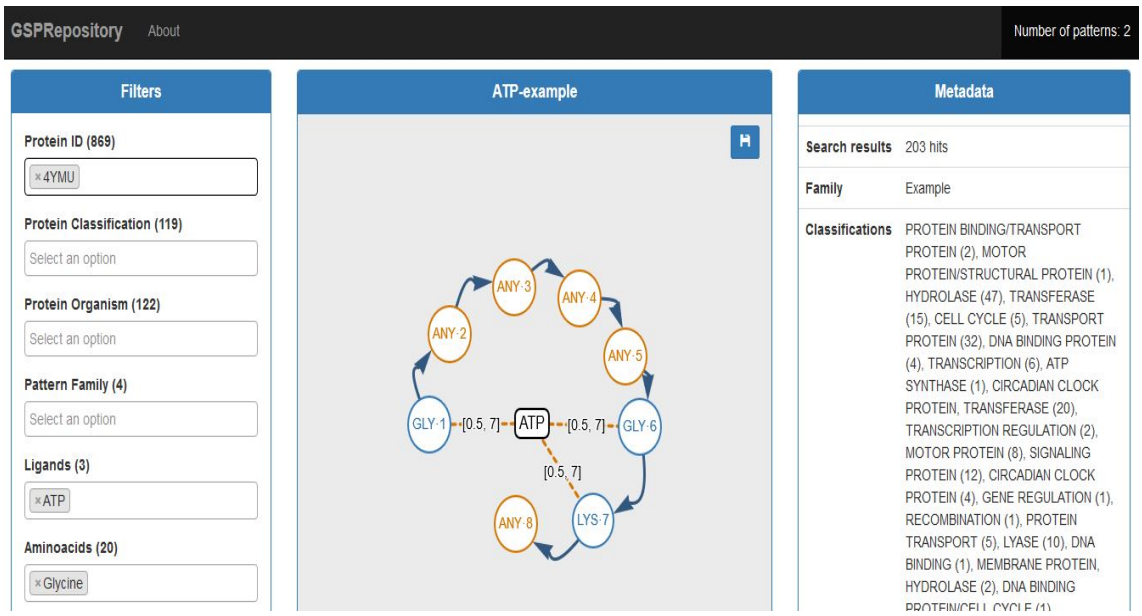



Figura 2.6: Implementación de interfaz de usuario basada en facetas para el repositorio de patrones estructurales proteína-ligando.

## 2.5. Herramientas de consulta de patrones estructurales

Luego de realizar una investigación de herramientas web que lleven a cabo procedimientos similares o medianamente similares en la búsqueda de patrones estructurales, se ha encontrado una serie de aplicaciones que tratan con patrones estructurales.

Dichas herramientas entregan datos estadísticos, además de metadatos e información acerca del patrón.

Existen diversas formas de representar patrones estructurales. Una de éstas corresponde a la herramienta **LigPlot**, la cual obtiene representaciones esquemáticas bidimensionales [12] (en otras palabras, en grafos) de patrones estructurales proteína-ligando a partir del ingreso como entrada de archivos con extensión PDB, para así obtener un resumen de la estructura molecular del patrón (Figura 2.6).

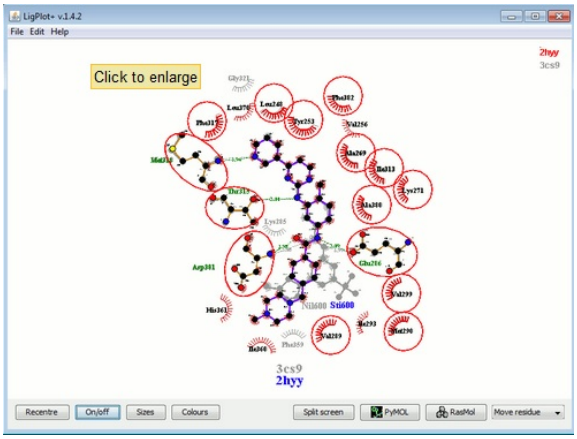



Figura 2.7: Ligplot usando una representación esquemática bidimensional de la proteína 2HYY

A pesar de que LigPlot sólo se limita a analizar y representar estructuras de proteínas una a la vez, existen otras herramientas disponibles en Internet que permiten hacer otro tipo de análisis para macromoléculas. Es por esto que a continuación se darán a conocer algunas más.

**LigPrep** es una herramienta que permite generar estructuras moleculares 3D, también aplica reglas sofisticadas para corregir estructuras de Lewis[6] y eliminar errores en los ligandos en el sitio de unión con los aminoácidos [7].

Otra aplicación que realiza representaciones bidimensionales de patrones estructurales proteína ligando es **PLIP**. Éste es un analizador de patrones estructurales

[6]Es una representación gráfica que muestra pares de electrones de enlaces entre los átomos de una molécula (para este caso un aminoácido) y los pares de electrones solitarios que puedan existir.

proteína-ligando totalmente automático [10]. Realiza una visualización y detecta patrones estructurales proteína-ligando dentro de una proteína especificada como archivo con extensión PDB que es cargado al software. La información que retorna corresponde netamente a diagramas de interacción 2D (grafos) y 3D, tablas con información en detalle acerca de la interacción entre aminoácidos y ligandos, además permite descargar archivos XML[7] con los valores retornados. A fin de cuentas, permite el procesamiento rápido de archivos PDB en el análisis de patrones estructurales proteína-ligando (Figura 2.7).

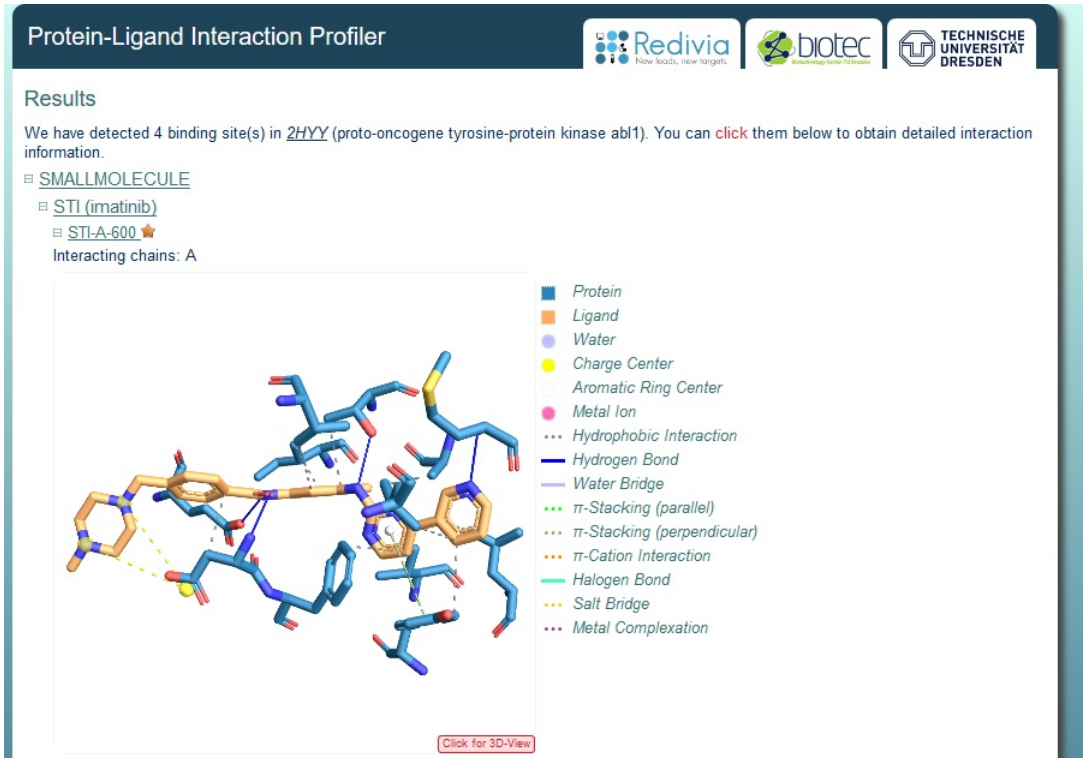
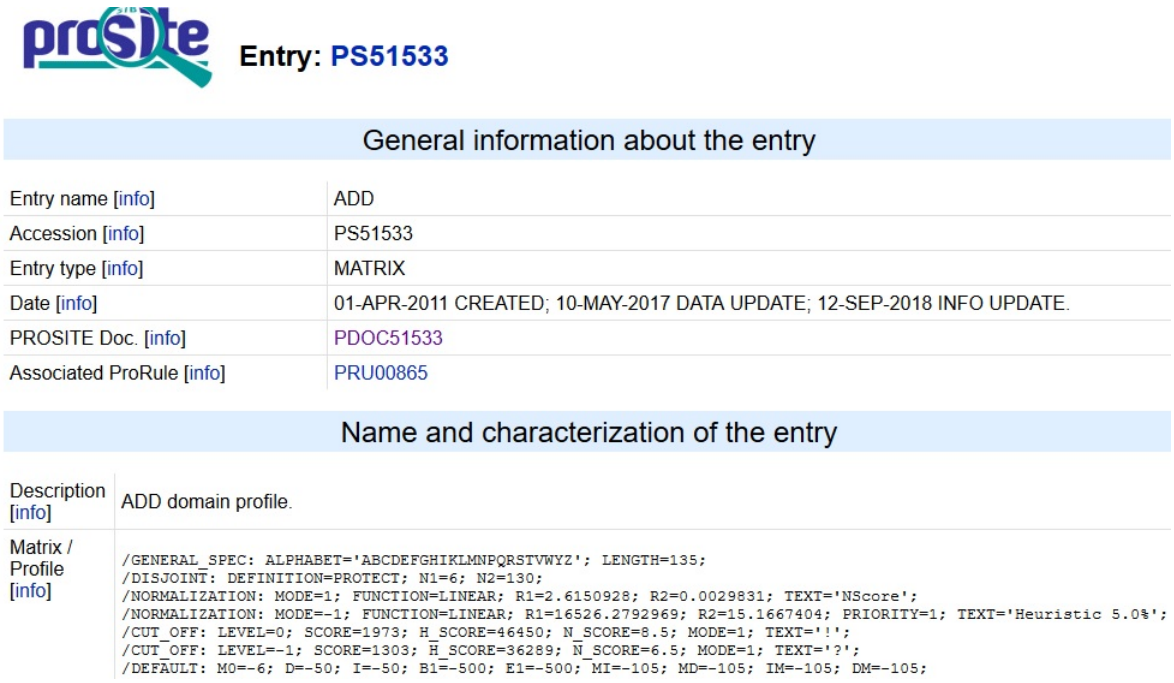



Figura 2.8: PLIP mostrando el resultado del análisis de la proteína 2HYY.

También, existe una aplicación de características similares a la que se desarrolla en este proyecto, la cual de por sí es un repositorio que almacena patrones estruc-

[7]Representación de información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.

turales proteína-ligando, llamada **Prosite**. Dicho repositorio también se encarga de almacenar familias y dominios de proteínas, como también documentación que proporciona información de antecedentes de las proteínas y las funciones de éstas [5]. Además, presenta diversas formas de representación para patrones estructurales, sean éstas: bidimensionales, archivo de texto plano, etc (Figura 2.8).



prosite **Entry: PS51533**

**General information about the entry**

| | |
|---|---|
| Entry name [info] | ADD |
| Accession [info] | PS51533 |
| Entry type [info] | MATRIX |
| Date [info] | 01-APR-2011 CREATED; 10-MAY-2017 DATA UPDATE; 12-SEP-2018 INFO UPDATE. |
| PROSITE Doc. [info] | PDOC51533 |
| Associated ProRule [info] | PRU00865 |

**Name and characterization of the entry**

| | |
|---|---|
| Description [info] | ADD domain profile. |
| Matrix / Profile [info] | /GENERAL_SPEC: ALPHABET='ABCDEFGHIKLMNPQRSTVWYZ'; LENGTH=135;<br>/DISJOINT: DEFINITION=PROTECT; N1=6; N2=130;<br>/NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=2.6150928; R2=0.0029831; TEXT='NScore';<br>/NORMALIZATION: MODE=-1; FUNCTION=LINEAR; R1=16526.2792969; R2=15.1667404; PRIORITY=1; TEXT='Heuristic 5.0%';<br>/CUT_OFF: LEVEL=0; SCORE=1973; H_SCORE=46450; N_SCORE=8.5; MODE=1; TEXT='!';<br>/CUT_OFF: LEVEL=-1; SCORE=1303; H_SCORE=36289; N_SCORE=6.5; MODE=1; TEXT='?';<br>/DEFAULT: M0=-6; D=-50; I=-50; B1=-500; E1=-500; MI=-105; MD=-105; IM=-105; DM=-105; |

Figura 2.9: Prosite mostrando información acerca del patrón Zinc Finger.

Por último, la herramienta **GSP4PDB** es una aplicación web fácil de usar que permite a los usuarios diseñar, buscar y analizar patrones estructurales dentro de biomacromoléculas de manera gráfica [16]. Esta aplicación utiliza el Protein Data Bank (PDB) para obtener información estructural detallada. El programa recibe un patrón basado en grafos formado por componentes gráficos que representan un ligando y los aminoácidos que lo rodean. El patrón también incluye algunas asociaciones que condicionan la estructura: distancias entre el ligando y los aminoácidos, distancias entre los aminoácidos y la secuencia de aminoácidos en la cadena.

El usuario puede determinar qué tipo de ligando desea analizar e incluir los aminoácidos fijos, pero también puede incluir los aminoácidos çualquiera". Pueden tomar el valor de cualquier aminoácido, aunque el usuario puede restringirlos a una clasificación específica de acuerdo con su polaridad. Finalmente, el programa busca coincidencias dentro de cada macromolécula en el PDB. Los resultados de la búsqueda se muestran en forma de texto y gráficamente por un visor 3D. Este análisis permite comparar proteínas, ADN o ARN donde existe la estructura y evaluar cómo los aminoácidos rodean al ligando en cada coincidencia.[17]

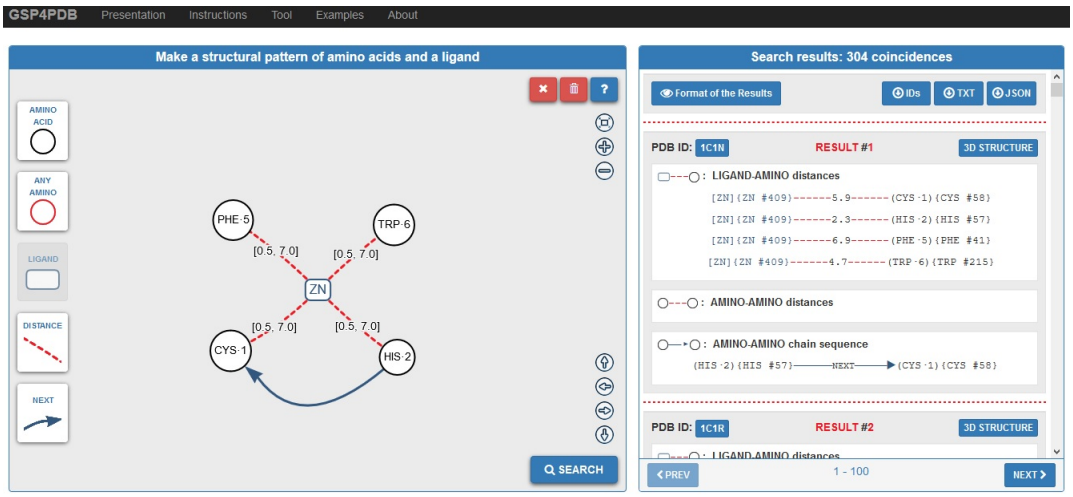



Figura 2.10: GSP4PDB mostrando un patrón estructural proteína-ligando dibujado y al costado derecho, las estadísticas obtenidas de la búsqueda de coincidencias del patrón en el PDB.

Cabe destacar que la herramienta **GSP4PDB** ha sido mejorada y se ha lanzado últimamente una segunda versión para ésta, denominada **GSP4PDB2**. Ésta última también permite diseñar patrones estructurales proteína-ligando usando un entorno gráfico con algunas modificaciones (Figura 2.10), considerándose como una extensión de la primera. Además permite realizar búsquedas en la base de datos, cumpliendo criterios de búsqueda utilizando interfaces basada en facetas de usuario para encontrar y visualizar patrones estructurales contenidos en distintas proteínas que cumplan dichos criterios (Figura 2.11).

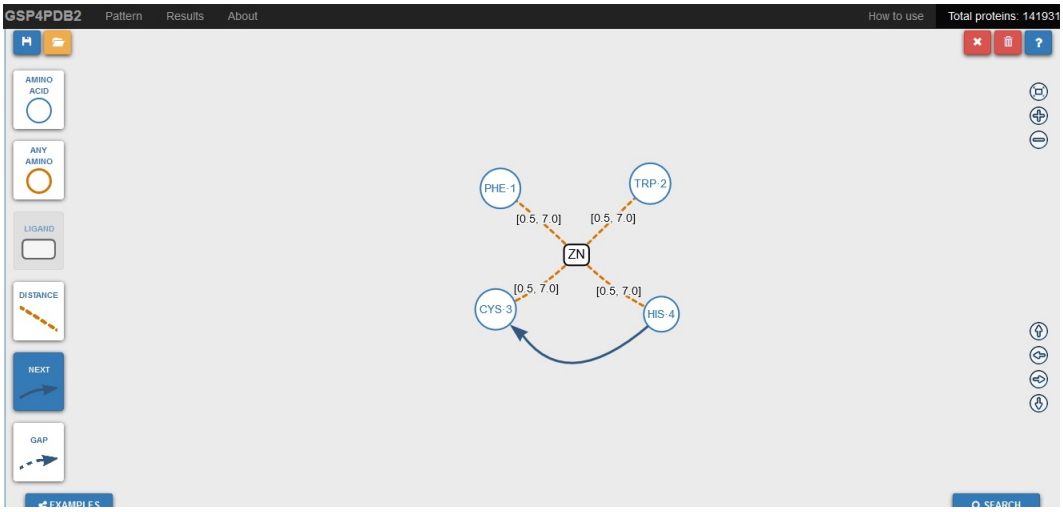



Figura 2.11: GSP4PDB2 mostrando un patrón estructural proteína-ligando dibujado.

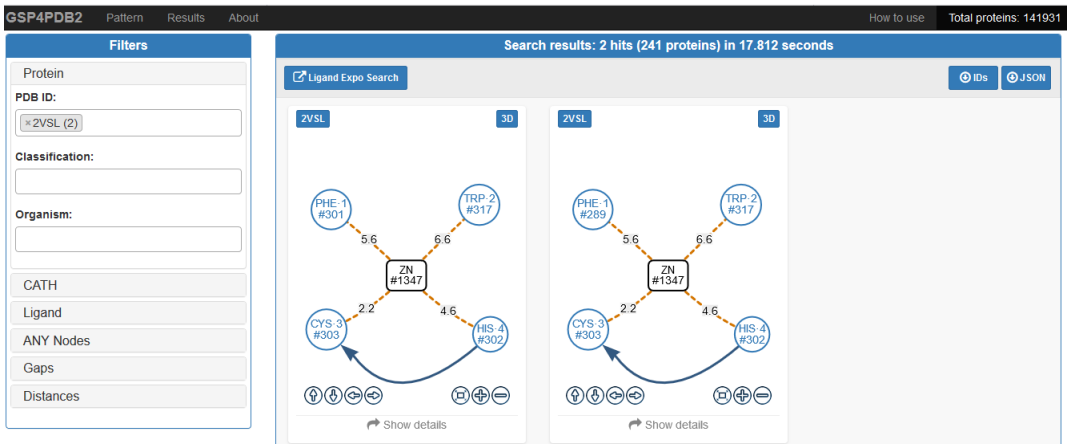



Figura 2.12: GSP4PDB2 visualizando resultados de búsqueda del patrón previamente diseñado (Figura 2.10).

En el Cuadro 2.1 se detalla cada una de las herramientas relacionadas, el tipo de manejo sobre las estructuras macromoleculares y su fuente.

| **Nombre** | **Tipo** | **Fuente** |
|---|---|---|
| LigPlot | Análisis | https://www.ebi.ac.uk/thornton-srv/software/LigPlus/ |
| LigPrep | Exploración y visualización | https://www.schrodinger.com/ligprep |
| PLIP | Análisis | http://plip.biotec.tu-dresden.de/plip-web/plip/index |
| Prosite | Exploración | https://prosite.expasy.org/ |
| GSP4PDB2 | Diseño y Exploración | https://structuralbio.utalca.cl/gsp4pdb/ |

Cuadro 2.1: Tabla

# 3. Diseño y Construcción

En este capítulo se describe el proceso de desarrollo del repositorio de patrones estructurales. Específicamente, se presentan los requisitos funcionales, los componentes de diseño, y las interfaces de usuario finales de la herramienta. Cabe destacar la arquitectura que soporta este sistema, como también el diseño del diagrama de clases para entender la interacción entre los objetos que componen el sistema. Además se incluye un diagrama de interfaz que explica a grandes rasgos los componentes de la herramienta. Todo lo anterior se detalla a continuación.

## 3.1. Requisitos generales

GSPRepository debe ser una aplicación que consta de 3 componentes que interactúan entre sí, los cuales son:

1. Filtros de búsqueda
2. Visualización del patrón estructural
3. Metadatos del patrón estructural

En primer lugar, los filtros de búsqueda es la opción que permite la interacción con el usuario. Estos filtros poseen valores para los cuales, el usuario podrá elegir un conjunto de éstos y, posteriormente, llevar a cabo la exploración de patrones estructurales de acuerdo a los valores especificados que coincidan con la información de los patrones almacenados en el repositorio.

En segundo lugar, encontramos la visualización de patrones estructurales. Para esto, la aplicación dispone de una representación basada en grafos para los patrones

usando una interfaz gráfica dinámica. Éstos serán mostrados de acuerdo a los valores especificados en los filtros de búsqueda, para los cuales, dicha información del patrón coincide con estos valores.

Y por último, la sección de metadatos del patrón estructural consta de un formulario que da a conocer las características del patrón e información general de éste. Cada metadato se ubica en paralelo con la visualización del patrón correspondiente y también depende de la aparición del patrón estructural respecto al ámbito del filtrado por exploración.

## 3.2. Requisitos específicos

Para este proyecto, la metodología de desarrollo implementada es Scrum. Por entonces la ceremonia que se lleva a cabo para definir los requisitos está sujeta a cambios durante el transcurso del proyecto. Sin embargo, los requisitos de aplicación se reflejan en historias de usuario, las cuales se transcriben como deseos del usuario que la aplicación deba satisfacer y, posteriormente, la transcripción a requisitos funcionales. A continuación se detallan las historias de usuario:

| ID | Descripción historia de usuario |
|---|---|
| **HU01** | Como usuario debo tener la opción de visualizar patrones estructurales proteína-ligando mediante una representación basada en grafos |
| **HU02** | Los patrones estructurales proteína-ligando deben ser conformados por ligandos, aminoácidos estándar y aminoácidos comodín, que pueden tomar cualquier valor de los 20 aminoácidos estándar definidos para el proyecto. |
| **HU03** | Como usuario debo tener la opción de explorar patrones estructurales de acuerdo a filtros de búsqueda, sean éstos por: ligando, aminoácidos, familia de patrón, identificador de proteína, proteína de organismos y por CATH. |
| **HU04** | Como usuario debo poder visualizar patrones estructurales proteína-ligando de tal forma que éstos cumplan con los valores especificados en los filtros de búsqueda. |
| **HU05** | Como usuario debo poder ver los metadatos de cada patrón estructural y ver que éstos cumplen con las condiciones de búsqueda según los filtros aplicados. |
| **HU06** | Como usuario debo tener la opción de poder cargar patrones estructurales directamente a la base de datos. Dicho procedimiento debe ser transparente. |
| **HU07** | Como usuario deseo poder proporcionar valores a los filtros de búsqueda sólo si éstos últimos me los proporciona, ya que éstos corresponderían a los cargados en base de datos. |

Cuadro 3.1: Historias de usuario

A continuación se detallan los requisitos transcritos para la herramienta de Repository:

- Visualizar patrones estructurales proteína-ligando mediante una representación basada en grafos. Éstos están compuestos de: ligandos, aminoácidos y aminoácidos comodín que pueden tomar cualquier valor de los 20 aminoácidos estándar definidos para el proyecto.
- Implementar opción de explorar patrones estructurales de acuerdo a filtros de búsqueda, sean éstos por: ligando, aminoácidos, familia de patrón, identificador de proteína, proteína de organismos y CATH.
- Visualizar patrones estructurales proteína-ligando que cumplan con valores especificados en los filtros de búsqueda.

- Mostrar los metadatos correspondientes de cada patrón estructural visualizado y que éstos cumplan con las condiciones de búsqueda según los filtros aplicados.
- Cargar a la base de datos patrones estructurales proteína-ligando en formato JSON.
- Los valores para cada filtro de búsqueda son únicamente provenientes de la base de datos de la aplicación.

Las historias de usuario que han sido mencionadas fueron emitidas y confirmadas por ambos clientes, de los cuales surgió la idea de desarrollar GSPRepository. Estos clientes son Ph. Renzo Angles de la carrera Ingeniería Civil en Computación y Mauricio Arenas docente de la carrera de Ingeniería en Bioinformática de la Universidad de Talca.

Los requisitos funcionales son transcripciones formales que se obtienen a partir de las historias de usuario. A partir de éstos, se desarrolla la aplicación GSPRepository.

## 3.3. Arquitectura de la aplicación

### 3.3.1. Arquitectura cliente-servidor

La arquitectura utilizada para este proyecto es del tipo cliente-servidor. Por el lado del servidor, se encuentra la base de datos correspondiente al almacenamiento de patrones estructurales proteína-ligando y el servidor web, ambos en la misma máquina.

Los componentes de la capa de presentación son HTML5, CSS3 y la utilización de una biblioteca de visualización Javascript llamada vis.js (versión 4.20.1). El servidor web almacena dos aplicaciones: GSP4PDB3 y GSPRepository. Para la base de datos se utiliza el gestor PostgreSQL.

Dicha arquitectura presenta diversas ventajas en lo que respecta a mantenimiento; al estar distribuidas las funciones y responsabilidades de las distintas máquinas, es posible introducir nuevas máquinas, reemplazarlas, efectuar actualizaciones, entre otras.

Es posible observar en la Figura 3.1 las tres capas de la arquitectura cliente-servidor que es aplicada para el proyecto. En primer lugar, la capa de presentación es

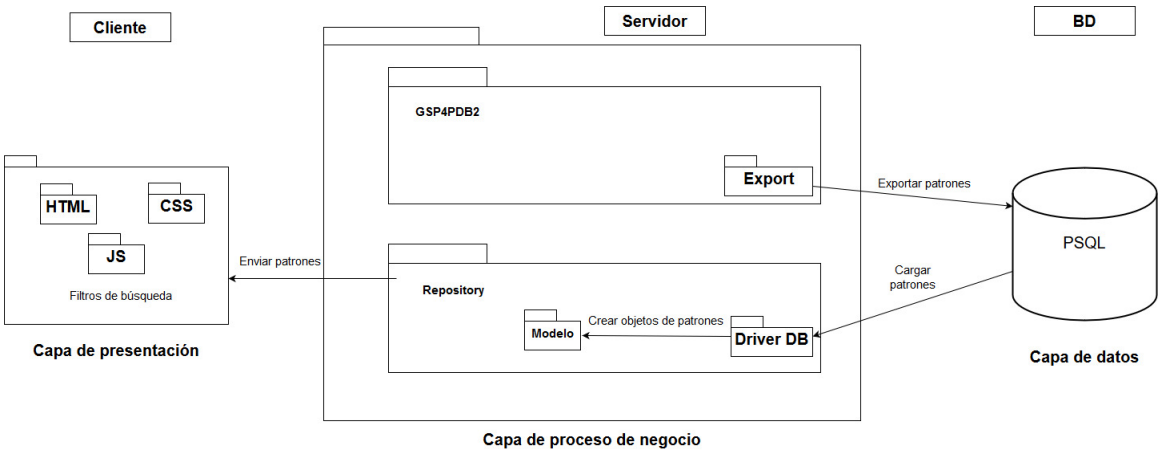



Figura 3.1: Arquitectura cliente-servidor de 3 capas.

la encargada de mostrar al cliente la representación basada en grafos de los patrones estructurales.

En la capa de proceso de negocio, correspondiente al servidor de la aplicación, consta de dos aplicaciones dentro de él: GSP4PDB2 y GSPRepository. La función que cumple GSP4PDB2 es que tiene la facultad de exportar patrones estructurales proteína-ligando directamente a la base de datos de GSPRepository.

La aplicación GSPRepository carga la información directamente desde la base datos, ubicada en la tercera capa, vale decir, en la capa de datos. Luego se encarga de construir instancias de objetos que representen al patrón estructural utilizando un modelo de objetos. Una vez construido el modelo, envía los patrones a la capa de presentación.

Los patrones estructurales son visualizados en su totalidad, dando la opción al usuario de interactuar con los filtros de búsqueda. Una vez que el usuario proporciona valores para los filtros, se procede a, valga la redundancia, filtrar los patrones y mostrar sólo aquellos que cumplan las condiciones especificadas.

## 3.4. Diagrama de clases

Para representar los patrones estructurales proteína-ligando en el servidor, se ha implementado un modelo de clases que permite definir las relaciones entre los componentes de un patrón estructural. Para ello, se definió el diagrama de clases presentado en la Figura 3.2.

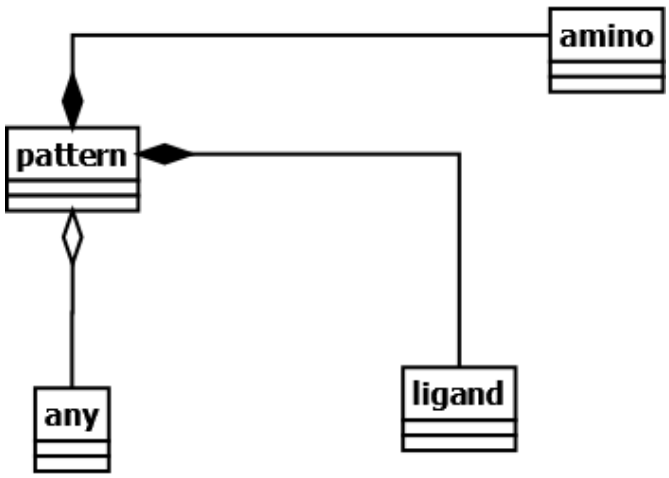



Figura 3.2: Diagrama de clases de alto nivel para la aplicación GSPRepository.

Este modelo de clases ha sido implementado en el lado del servidor, utilizando el lenguaje de programación PHP. La implementación del modelo consiste en que el patrón (pattern) está conformado por uno o muchos aminoácidos estándar (amino) y/o aminoácidos comodín (any) y por un ligando (ligand).

Si una estructura no posee un ligando, no se le considera patrón estructural proteína-ligando. Si una estructura posee un ligando a secas, tampoco se le considera patrón estructural proteína-ligando, puesto que consta tan sólo de un ligando sin aminoácidos. La secuencia de aminoácidos que forman parte de una proteína, estrictamente en la región que se considera patrón estructural proteína-ligando, se restringe a tener dentro de su estructura molecular a un ligando relacionado con uno o más aminoácidos, sean éstos estándar y/o comodín (Figura 3.3).

La clase **pattern** posee listas de aminoácidos estándar, aminoácidos comodín y

de ligando, como también listas que registra todas las relaciones (distancia y precedencia) entre las estructuras que componen al patrón y cada clase posee una serie de atributos que los definen como tal.

Los listados de relaciones que almacena la clase pattern son los siguientes:

- Relación de distancias entre dos aminoácidos.
- Relación de distancias entre un aminoácido corriente y un aminoácido comodín.
- Relación de distancias entre dos aminoácidos comodín.
- Relación de distancias entre un ligando y un aminoácido corriente.
- Relación de secuencia de precedencia (Next) entre un aminoácido corriente y un aminoácido comodín, respectivamente.
- Relación de secuencia de precedencia entre dos aminoácidos comodín.
- Relación de secuencia de precedencia entre un aminoácido comodín y un aminoácido corriente, respectivamente.
- Relación de aminoácidos comodines entre medio (Gap) de dos aminoácidos corrientes.
- Relación de aminoácidos comodines entre medio de un aminoácido corriente y un aminoácido comodín, respectivamente.
- Relación de aminoácidos comodines entre medio de un aminoácido comodín y un aminoácido corriente, respectivamente.
- Relación de aminoácidos comodines entre medio de dos aminoácidos comodines.

Las demás clases como **ligand**, **amino** y **any** son aquellas partícipes y las que están contenidas dentro de los listados mencionados anteriormente que forman parte de la clase pattern. Una vez construidos estos listados y, la relación entre un ligando con un aminoácido corriente o un aminoácido comodín, un patrón estructural proteína-ligando se reconoce como tal. En otras palabras, un patrón estructural lo es si tiene a lo menos una relación del ligando con otra estructura, que puede ser un aminoácido corriente o un aminoácido comodín.

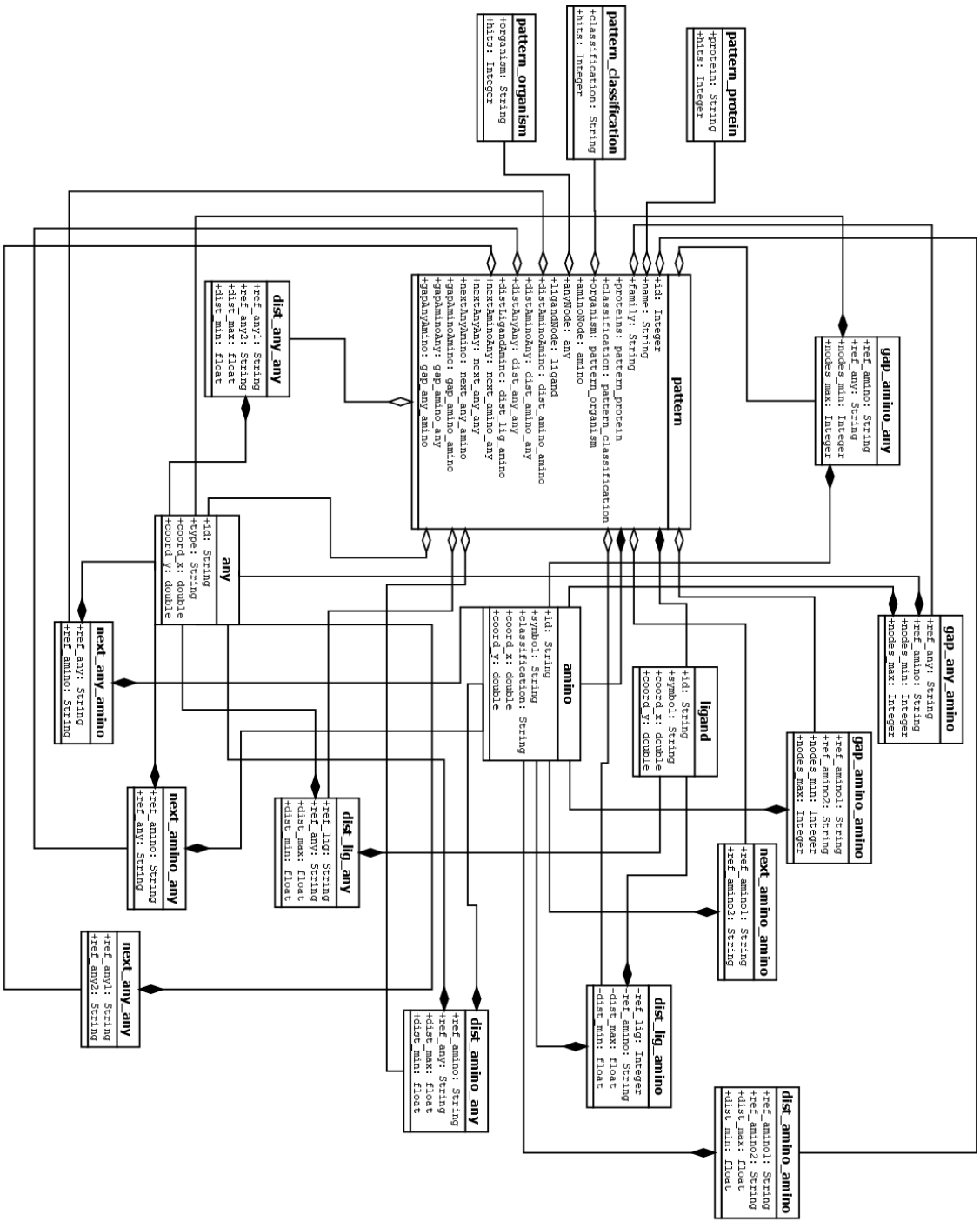



Figura 3.3: Diagrama de clases de Repository.

## 3.5. Esquema de base de datos

Los patrones estructurales proteína-ligando se almacenarán en una base de datos relacional que ha sido modelada únicamente para GSPRepository. Ésta es capaz de contener toda la información correspondiente al patrón estructural, tanto las relaciones de distancia, precedencia y gap que lo componen. En la Figura 3.3 se muestra el esquema completo de base de datos.

La tabla **Pattern** es la tabla principal y la que contiene a todas las demás. Es la tabla que contiene toda la información propia de un patrón estructural proteína-ligando. Cada una de las demás tablas del esquema de base de datos están ligadas a *pattern*, ya que cada de ellas pertenece a sólo un patrón específico.

Las tablas que representan los nodos que conforman el patrón estructural corresponden a **aminoácido estándar** (*Amino*), **aminoácido comodín** (*Nodo any*) y **ligando** (*Ligand*).

Se observa que todas las tablas tienen una referencia al patrón al cual pertenecen. Las tablas **amino** y **ligand** tienen un atributo llamado *symbol*, que representa la abreviatura con el cual se le denomina en la jerga del campo de la Biología a este nodo. Siendo así, por ejemplo, al aminoácido *Alanina* se le abrevia por *Ala*. Otro ejemplo de ello es para el ligando *Zinc*, que se le denomina por su abreviatura en la tabla periódica de elementos químicos por *Zn*.

También se encuentran las tablas que representan las relaciones de distancia **Dist** entre los componentes del patrón. Cada tabla de distancia se compone de una referencia al patrón al cual pertenecen, dos referencias a los que se relacionan y dos atributos para registrar la distancia mínima y la distancia máxima de lejanía entre los dos nodos en cuestión.

Otro tipo de tablas que componen a un patrón estructural son las **Next**. Éstas únicamente relacionan dos nodos en orden secuencial sin almacenar una distancia entre éstos a diferencia de las relaciones de distancia. La tabla next almacena la referencia de un nodo que va antes (secuencialmente) que un segundo nodo, además de la referencia al patrón al que pertenecen.

Por último, las tablas cuyo prefijo es **Gap** que consiste en abreviar un conjunto secuencial de aminoácidos comodín que se encuentren entre, ya sea: dos aminoácidos estándar, un aminoácido estándar y un aminoácido comodín o bien, entre un aminoácido comodín y un aminoácido estándar, respectivamente. El objetivo de este

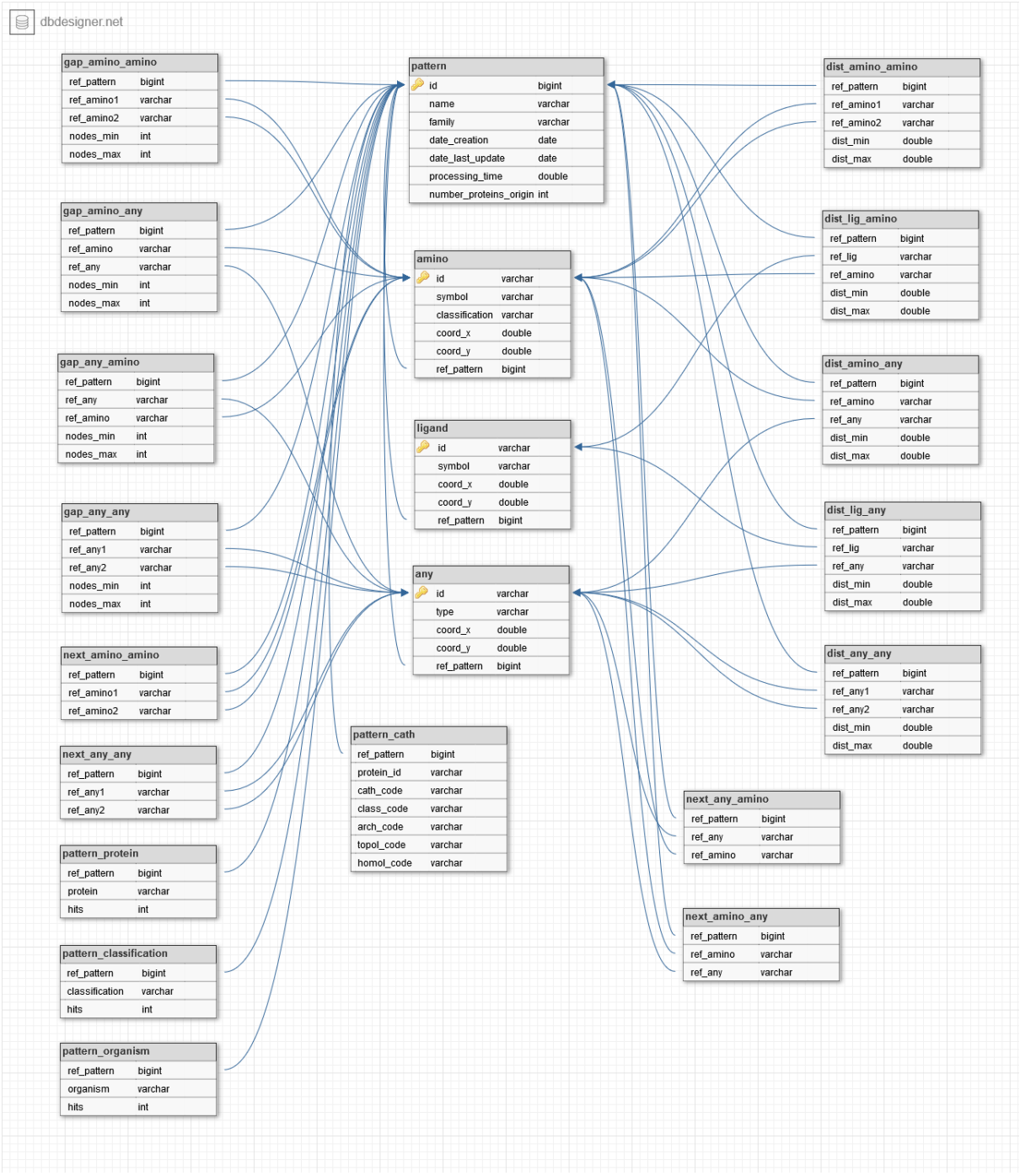



Figura 3.4: Base de datos relacional de GSPRepository.

tipo de relación son aquellos casos donde el patrón presenta una secuencia extensa de aminoácidos comodínes (any) y, por comodidad, se opta por la implementación de esta relación.

Además, la tabla Gap incluye dos atributos para denotar la cardinalidad para los aminoácidos comodínes existentes entre los dos nodos antes mencionados, según como esté construído el patrón. Esta cardinalidad consta de un par ordenado de valores: cantidad mínima y cantidad máxima de nodos entre medio.

Todas las tablas descritas conforman el esquema de base de datos y almacenan toda la información que debe tener un patrón estructural proteína-ligando en la aplicación GSPRepository.

## 3.6. Implementación

Para desarrollar la herramienta GSPRepository se implementó una serie de herramientas que permiten a ésta, disponer de filtros de búsqueda, paneles de visualización de patrones basados en grafos y, por último, un panel de visualización de metadatos de cada patrón que adjunta la información tanto de la estructura de éste como de sus características. A continuación, se dará a conocer en detalle todo lo relacionado en la implementación del repositorio.

### 3.6.1. Filtros de búsqueda

Los filtros de búsqueda, como se mencionó en capítulos anteriores, sirven para, valga la redundancia, filtrar patrones estructurales. Para poder implementar ésta herramienta es necesario la utilización de una librería de JavaScript llamada *select2*. La librería *select2* permite trabajar con etiquetas, las cuales pasan a ser los valores que se especifican en el filtro de búsqueda (Figura 3.5) y permiten al usuario interactuar con estos valores, con el objetivo de filtrar patrones estructurales y decidir qué patrones se van a visualizar siguiendo el cumplimiento de estos valores.

Se puede observar que los valores especificados en los filtros son: *5VMU* para Protein ID, *TRANSCRIPTION/DNA* para Protein Classification, *HOMO SAPIENS* para Protein Organism y *BR* y *ZN* para Ligands.

Cada uno de estos filtros corresponde a datos almacenados en la base de datos de la aplicación. La aplicación realizará búsquedas utilizando los valores disponibles en los filtros y no otros proporcionados por el usuario mediante teclado.

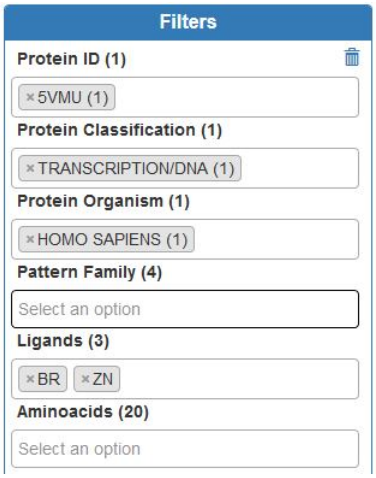



Figura 3.5: Interfaz de filtros de búsqueda.

En la Figura 3.6 se observa el diseño de interfaces de la aplicación como también una explicación clara de los elementos que componen la interfaz de usuario.

Los círculos de borde color azul representan a los aminoácidos (ALA y CYS). El rectángulo de borde color negro representa al ligando (ZN). Las líneas discontinuas de color naranjo representan las relaciones de unión distancia entre los nodos ubicados a ambos extremos (ZN con ALA y ZN con CYS). Para las distancias, se haya un rango etiquetado entre [0.5, 7.0] La flecha que une a los aminoácidos (ALA y CYS) es una relación de precedencia. En el panel derecho, en la parte superior de color azul se indica el nombre que tiene el patrón estructural (Cys-Ala Zn). Además, en el extremo superior derecho se ubica dos opciones para los patrones: exportar y actualizar la información del patrón.

Al costado izquierdo se observa la interfaz de filtros consiste en el identificador del filtro en cuestión (Protein ID, Protein Classification, Protein Organism, Pattern Family, Aminoacid, Ligand, etc) y el campo que permite ingresar valores para explorar patrones estructurales.

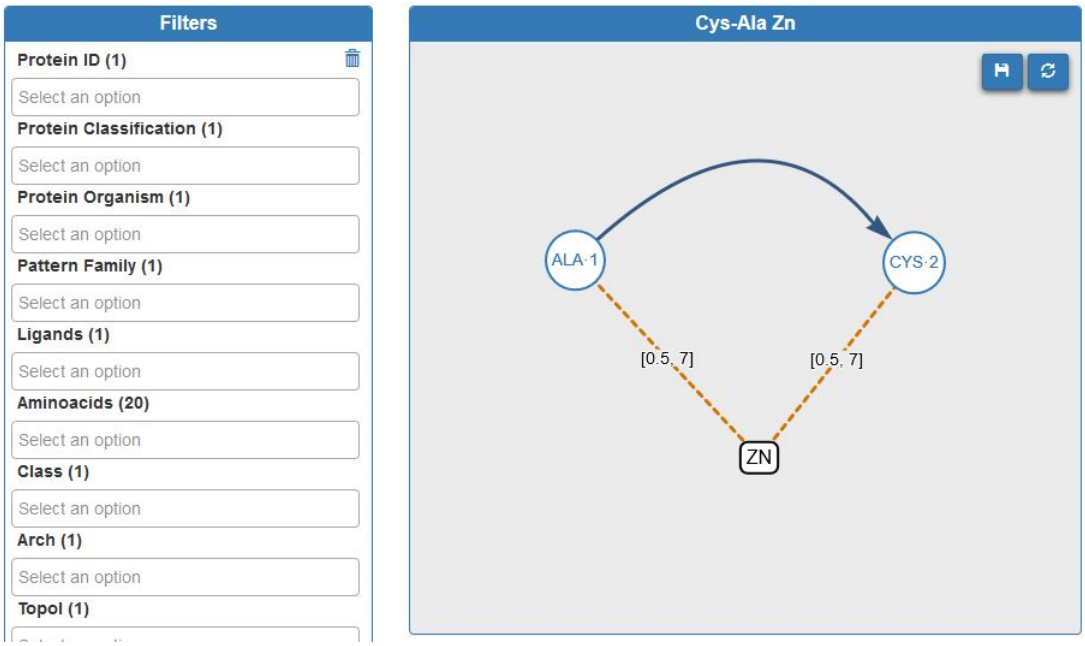



Figura 3.6: Paneles de filtro y visualización de patrón donde se realiza la exploración de patrones estructurales.

## 3.7. Exploración de patrones

La exploración de patrones estructurales consiste en un listado descendente que muestra todos los patrones cuyos datos coinciden con los valores especificados en los filtros de búsqueda. Además, en esta sección se explica otras funciones que hay para la exploración de patrones como los botones de navegación como también la metadata (información del patrón). Para que sea más fácil el entendimiento de las funcionalidades, se han enumerado las distintas propiedades del patrón como las características de su metadata (Figura 3.7).

Un patrón estructural se representa así mismo mediante un nombre único (1). La funcionalidad **exportar patrones** (2) consiste en transformar la estructura del patrón a un formato *JSON* y, posteriormente, descargarlo en un archivo con extensión .txt que contenga dicho JSON.

El objetivo de esta función de descarga (Figura 3.8) es poder contar con el archivo

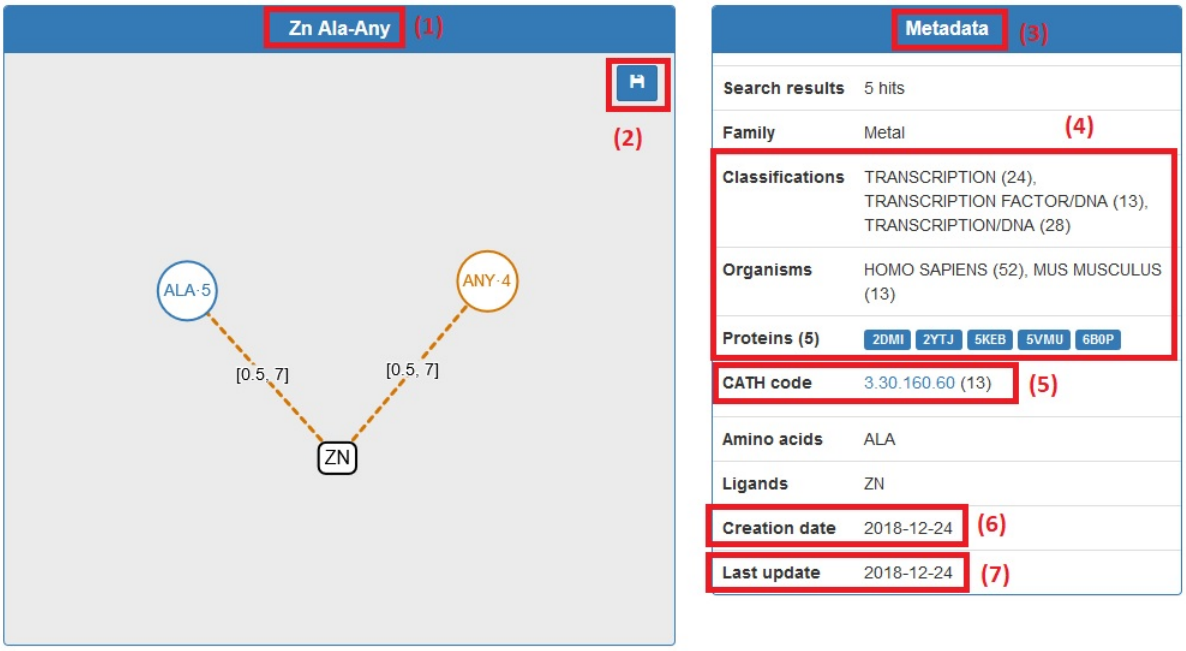



Figura 3.7: Función de exportación de patrón estructural y descarga de archivo en formato *JSON*.

que contenga la estructura basada en grafos del patrón estructural y que pueda ser cargado en GSP4PDB2 para su posterior edición.

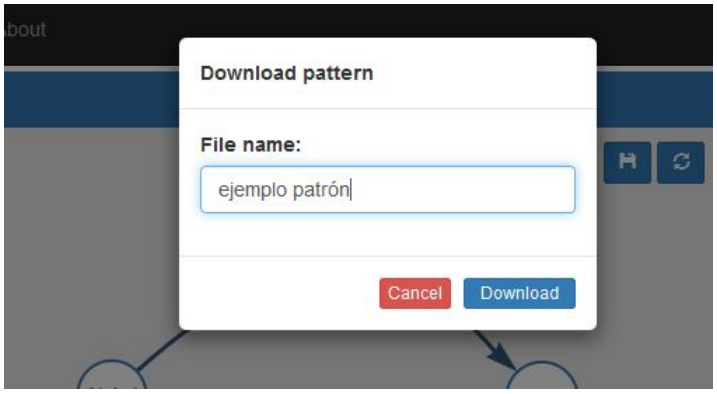



Figura 3.8: Función de descarga del patrón estructural proporcionando un nombre al archivo que contiene la estructura en formato JSON.

Una vez que se decide descargar el patrón estructural (2), se descarga el archivo en formato JSON.

Los patrones estructurales poseen información acerca de su estructura, familia, clasificación (4), organismos (4), proteínas (4), fecha de creación (6) y de la más reciente actualización (7), frecuencia en la base de datos y *CATH* (5). Al conjunto de estos datos se le denomina *Metadata* (3) (Figura 3.11).

Para la **estructura** en la Metadata se muestra un listado de *Amino acids* (aminoácidos) y otro de *Ligand* (ligando). Estos listados corresponden a los nodos del grafo que componen al patrón estructural.

La **familia** corresponde al conjunto de patrones que están regidos por alguna característica en común, por ejemplo, aquellos patrones que contengan un ligando que sea un elemento químico correspondiente a un metal (Zinc), se denomina que su familia es **Metal**.

Las proteínas cumplen una serie de funciones en las células (biológicas) de los seres vivos. Por ello, existen muchos patrones estructurales que forman parte de la estructura de éstas proteínas que cumplen dichas funciones. Al conjunto de estas funciones se les denomina **clasificación de proteína** (*Classification*). La forma de representación es la siguiente: *clasificación* acompañada de la frecuencia de aparición de dicha clasificación en la base de datos.

Las proteínas en las que un patrón estructural forma parte de su estructura se le denomina **proteínas** (*Proteins*). La forma de representacion es la siguiente: *Identificador de proteína.*

Por otro lado, existen patrones estructurales que están presentes en las proteínas que forman parte de la composición y son partícipes de muchas funciones a nivel celular de los seres vivos. A estas funciones se les denomina **organismos de proteína** (*Organisms*). La forma de representación es la siguiente: *Organismo* acompañado de la frecuencia de aparición de dicho organismo en la base de datos.

Por último, *CATH* es una clasificación jerárquica similar a un árbol que comienza en el "tronco"del árbol, agrupando los dominios de proteínas en categorías amplias, las cuales se describen a continuación: [6]:

- Clase: Las estructuras de clase se clasifican según su composición de estructura secundaria (principalmente alfa, mayoritariamente beta, mixta alfa/beta o estructuras secundarias). Se le denomina para la aplicación como *Class code.* [6]

- Arquitectura: Las estructuras se clasifican según su forma general según lo determinado por las orientaciones de las estructuras secundarias en el espacio 3D, pero ignora la conectividad entre ellas. Se le denomina para la aplicación como *Arch code.* [6]

- Topología (familia de pliegues): Las estructuras se agrupan en grupos de pliegues en este nivel según la forma general y la conectividad de las estructuras secundarias. Se le denomina para la aplicación como *Topol code.* [6]

- Superfamilia homóloga: Este nivel agrupa los dominios de proteínas que se cree que comparten un ancestro común y, por lo tanto, se pueden describir como homólogos. Se le denomina para la aplicación como *Homol code.* [6]

Para un mejor entendimiento, se ha creado una tabla que explica la distribución de cada una de las jerarquías y cómo es que se aplican en GSPRepository.

Para la exploración de patrones estructurales utilizando filtros, éstos disponen de valores numéricos, vale decir, el valor del código (Tabla 3.2). La representación del código CATH en la metadata del patrón es de la forma: *Class code.Arch code.Topol code.Homol code* seguido de la cantidad de veces que dicho código CATH aparece

| Letra | Jerarquía | Código | Criterio de agrupamiento |
|---|---|---|---|
| C | Class code | 3 | Alpha Beta |
| A | Arch code | 30 | 2-Layer Sandwich |
| T | Topol code | 160 | Double Stranded RNA Binding Domain |
| H | Homol code | 60 | Classic Zinc Finger |

Cuadro 3.2: Ejemplo de distribución de códigos de jerarquía del árbol CATH.

en las proteínas de las cuales el patrón forma parte. Por ejemplo: El código CATH 3.30.160.60 (13) (Figura 3.9).

Esto quiere decir que, dado las proteínas en las cuales el patrón estructural se encuentra, para algunas de éstas, existe un código CATH asociado (Figura 3.9). Cabe destacar que no todas las proteínas poseen un código CATH.

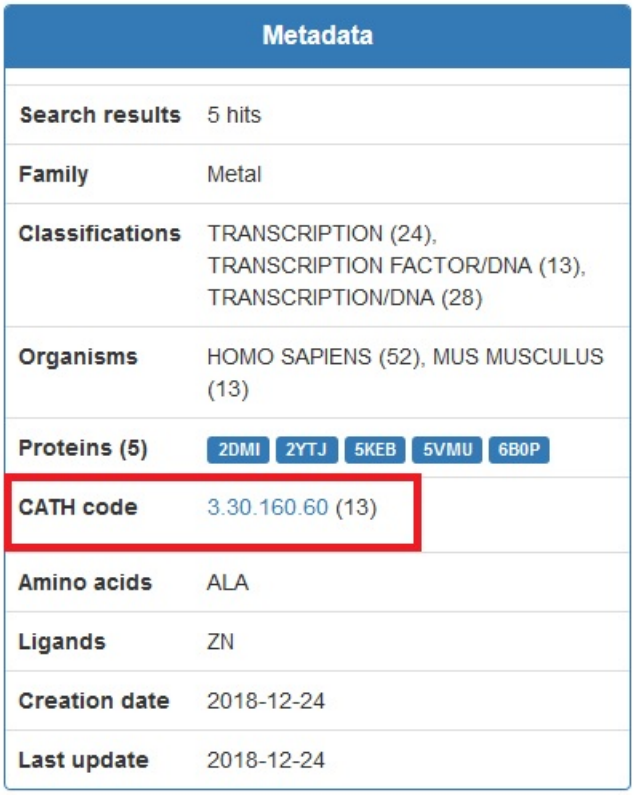



Figura 3.9: Representación de código CATH en metadata del patrón estructural.

## 3.8. Archivo en formato JSON del patrón estructural

La estructura del archivo en formato JSON del patrón estructural consiste en dos listados principales: *nodes* y *edges* (nodos y aristas, respectivamente).

Para los nodos, se crea un objeto cuyos primeros atributos son: $x$ e $y$, los cuales corresponden a las coordenadas cartesianas de dicho nodo en el espacio 2D donde se grafica el patrón en la interfaz gráfica. Seguido de un identificador único dentro de la tabla de base de datos que lo almacena. El atributo *group* el cual agrupa los nodos para como deben ser representados en la interfaz. Posee también *label* que corresponde al texto contenido en el nodo dibujado en la interfaz. Otro atributo *type* que corresponde al tipo de nodo (*amino*, *ligand* o *any*) y *symbol* que corresponde a su abreviatura según la jerga en la Biología.

```
{"nodes": [
      {
          "x": -297,
          "y": -153,
          "id": "5",
          "group": "amino",
          "label": "ALA·5",
          "type": "amino",
          "symbol": "ALA"
      },
      {
          "x": -172,
          "y": -14,
          "id": "999999999",
          "group": "ligand",
          "label": "ZN",
          "type": "ligand",
          "symbol": "ZN"
      },
      {
          "x": -47,
          "y": -161,
          "id": "4",
          "group": "any",
          "label": "ANY·4",
          "type": "any",
          "symbol": "ANY"
      }
    ],
    "edges": [
    {
      "id": "4",
      "from": 999999999,
      "to": 5,
      "label": "[0.5, 7]",
      "type": "distance"
    },
    {
      "id": "999999999_5",
      "from": 999999999,
      "to": 4,
      "label": "[0.5, 7]",
      "type": "distance"
    }
```

```
  ]
  }
}}
```

Por ejemplo, si es el aminoácido *Alanina*, entonces *symbol* será **ALA**. En cambio, si el nodo es un ligando, por ejemplo, *symbol* sería **ZN**.

Para las aristas, la representación consiste en atributos como *id* que es un identificador único dentro de la tabla de base de datos que lo almacena. Posee atributos *from* y *to*, los cuales significan *desde* y *hacia*, respectivamente. Especifica desde qué nodo y hasta qué nodo se establece la relación, siendo éstas: distancia, next (precedencia) o gap. Dichas relaciones se establecen dado el atributo *type*. Si es de *distancia*, entonces su *label* especifica en un intervalo la distancia mínima y máxima que alcanzan ambos extremos (nodos). Si la relación es de **precedencia**, no posee atributo *label*. Por último, si la relación es *gap*, el atributo *label* es de la siguiente forma: X(*cantidad mínima de aminoácidos comodín*, *cantidad máxima de aminoácidos comodín*). Dicho de otro modo, el intervalo para *Gap* abrevia una cantidad mínima y máxima de aminoácidos comodín entre dos nodos del patrón estructural.

```
{"nodes": [
      {
      "x": -240,
      "y": -260,
      "id": "1",
      "group": "amino",
      "label": "CYS·1",
      "type": "amino",
      "symbol": "CYS"
    },
    {
      "x": -107,
      "y": -261,
      "id": "2",
      "group": "amino",
      "label": "CYS·2",
      "type": "amino",
      "symbol": "CYS"
    },
    {
      "x": -107,
      "y": -95,
      "id": "4",
      "group": "amino",
      "label": "HIS·4",
      "type": "amino",
      "symbol": "HIS"
    },
    {
      "x": -241,
      "y": -96,
      "id": "6",
      "group": "amino",
      "label": "HIS·6",
      "type": "amino",
      "symbol": "HIS"
    },
    {
```

```
      "x": -175,
      "y": -179,
      "id": "999999999",
      "group": "ligand",
      "label": "ZN",
      "type": "ligand",
      "symbol": "ZN"
    }
    ],
    "edges": [
    {
      "id": "999999999",
      "from": 999999999,
      "to": 1,
      "label": "[0.5, 7]",
      "type": "distance"
    },
    {
      "id": "999999999_1",
      "from": 999999999,
      "to": 2,
      "label": "[0.5, 7]",
      "type": "distance"
    },
    {
      "id": "999999999_2",
      "from": 999999999,
      "to": 4,
      "label": "[0.5, 7]",
      "type": "distance"
    },
    {
      "id": "999999999_4",
      "from": 999999999,
      "to": 6,
      "label": "[0.5, 7]",
      "type": "distance"
    },
    {
      "id": "999999999_6",
      "from": 1,
      "to": 2,
      "label": "X(2,4)",
      "type": "gap"
    },
    {
      "id": "1-2",
      "from": 2,
      "to": 4,
      "label": "X(12,12)",
      "type": "gap"
    },
    {
      "id": "2-4",
      "from": 4,
      "to": 6,
      "label": "X(2,6)",
      "type": "gap"
    }
  ]
  }
}}
```

# 4. Evaluación del Software

En este capítulo se explica la metodología de evaluación, la cual consiste en definir bajo qué elementos de la *usabilidad* va a ser evaluado GSPRepository. En concreto, define actividades del proceso de experimentación con la aplicación: definición, diseño, ejecución y análisis. [9]

## 4.1. Metodología de evaluación

La evaluación de la aplicación consiste en organizar una sesión con un numero determinado de personas, preparar un conjunto de actividades de interacción con la aplicación, elaborar un cuestionario de evaluación que permitirá obtener estadísticas y analizar los datos obtenidos.

El objetivo de la evaluación medirá la *usabilidad* que, según la ISO 9126[1], se refiere a un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.[13]

Para la evaluación de la aplicación se midió la *usabilidad* en términos de:

- Aprendizaje: Atributos de la aplicación que se relacionan al esfuerzo de los usuarios para aprender a usarla [13].
- Comprensión: Atributos de la aplicación que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus funcionalidades.[13]
- Atractividad: Atributos de la aplicación que se relaciona a que tan atractivo resultan ser los elementos de la interfaz y que tan representativa es la abstracción de los objetos de la aplicación [13].

[1]Estándar internacional para la evaluación de la calidad del software

### 4.1.1. Diseño

En esta etapa se especifica un protocolo de los tratamientos a los sujetos (usuarios que interactúan con la aplicación), tipo de sujetos a emplear así como se preparan preparan los instrumentos o materiales a realizar o ejecutar el experimento. [9]

### 4.1.2. Protocolo de tratamiento de sujetos

El protocolo describe detalladamente la presentación de la aplicación a los sujetos, la definición de actividades a realizar al interactuar con la aplicación y la entrega de un cuestionario a los sujetos para que éstos entreguen su opinión y percepción de la aplicación para así obtener estadísticas.

### 4.1.3. Presentación de la aplicación

En la etapa de **presentación** el alumno tesista Carlos Hernández y el docente Mauricio Arenas de la carrera de Ingeniería en Bioinformática de la Universidad de Talca realizan una breve descripción de no más de 5 minutos acerca de la aplicación a estudiantes de Ingeniería en Bioinformática.

El resto del tiempo da lugar a las etapas de *definición de actividades* y *entrega de cuestionario*.

### 4.1.4. Definición de actividades

Para la **definición de actividades** se le entrega a cada estudiante una guía de actividades preparadas con anterioridad por el alumno tesista. Las actividades son interacciones con las distintas funcionalidades y elementos de la interfaz de la aplicación. Por un lado, las actividades son para la interacción con la aplicación GSPRepository, lo que es la exploración y visualización de patrones estructurales y, por otro lado, interactuar con GSP4PDB3 para la exportación de patrones estructurales desde ésta hacia la base de datos de GSPRepository.

Todo esto con el fin de que el usuario pueda interactuar con ambas aplicaciones, ver cómo estas se conectan y crear una percepción propia de éstas.

Para ver las actividades definidas en esta sección, dirigirse a Anexos.

### 4.1.5. Entrega de cuestionario

Y por último, en la **entrega de cuestionario** se le proporciona a cada estudiante un cuestionario digital para que refleje su percepción de la aplicación de acuerdo a las preguntas que aparecen en el cuestionario. El cuestionario consta de dos fases: fase de afirmaciones con alternativas cuyas respuestas (apreciaciones) utilizan la técnica *Escala Likert*[2] y una segunda fase de preguntas de desarrollo.

La fase de preguntas consiste en una serie de bloques. Cada bloque puede contener desde dos a cinco afirmaciones, seguidas de las siguientes apreaciaciones:

- Totalmente acuerdo: No hay lugar a dudas que la afirmación es totalmente verdadera.
- De acuerdo: Se está de acuerdo con que la afirmación es correcta pero no convence del todo. Pueden haber variables minúsculas (o casi insignificantes) que hagan dudar de ello.
- En desacuerdo: No se está de acuerdo con la afirmación. Existen variables que hagan dudar de la veracidad de la afirmación.
- Totalmente en desacuerdo: No hay lugar a dudas que la afirmación es falsa.

Para ver la guía de actividades y el cuestionario, dirigirse a Anexos.

### 4.1.6. Ejecución

El experimento se lleva a cabo en la sala de computadores de la Escuela de Ingeniería en Bioinformática en la Universidad de Talca, Campus Lircay y se cuenta con nueve estudiantes de de Ingeniería en Bioinformática que rinden el cuestionario de evaluación del software (Figura 4.1 y 4.2).

Los requisitos para poder realizar el experimento son sólo dos: disponer de un computador o notebook y estar conectados a la red de la Universidad *utalca*.

Formalmente no existe un tiempo límite para realizar la experimentación. Pero los estudiantes deberían poder (lo que se espera) completar todas las actividades propuestas entre 15 a 30 minutos.

[2]Es una escala psicométrica comúnmente utilizada en cuestionarios y es la escala de uso más amplio en encuestas para la investigación, principalmente en ciencias sociales. Al responder a una pregunta de un cuestionario elaborado con la técnica de Likert, se especifica el nivel de acuerdo o desacuerdo con una declaración (elemento, ítem o reactivo o pregunta).

Figura 4.1: Sala de computación de la Escuela de Ingeniería en Bioinformática, Universidad de Talca, Campus Lircay.

## 4.2. Resultados de la evaluación

En esta sección se hará analiza los resultados obtenidos producto del experimento que se ha hecho con la interacción de los estudiantes con las aplicaciones GSPRepository y GSP4PDB3. En concreto, se interpretarán las estadísticas obtenidas y la medición de cada bloque afirmaciones se hace de manera cualitativa.

Los bloques de afirmaciones son:

1. Interfaz de la aplicación

2. Filtrado de patrones estructurales

Figura 4.2: Experimentación de un estudiante de Ingeniería en Bioinformática con la aplicación GSP4PDB3.

3. Visualización de patrones
4. Metadatos sobre patrones
5. Visualización de patrones
6. Funcionalidad de exportar patrones
7. Correctitud de la información visualizada
8. Utilidad de la aplicación
9. Usabilidad de la aplicación

10. Percepción general de la aplicación

### 4.2.1. Interfaz de la aplicación

Las afirmaciones en términos de interfaz de la aplicación son los siguientes:

1. Es fácil identificar las tres secciones de la interfaz: filtros, patrón estructural, metadata del patrón
2. La distribución de las tres secciones de la interfaz es adecuada
3. La interfaz muestra información relevante

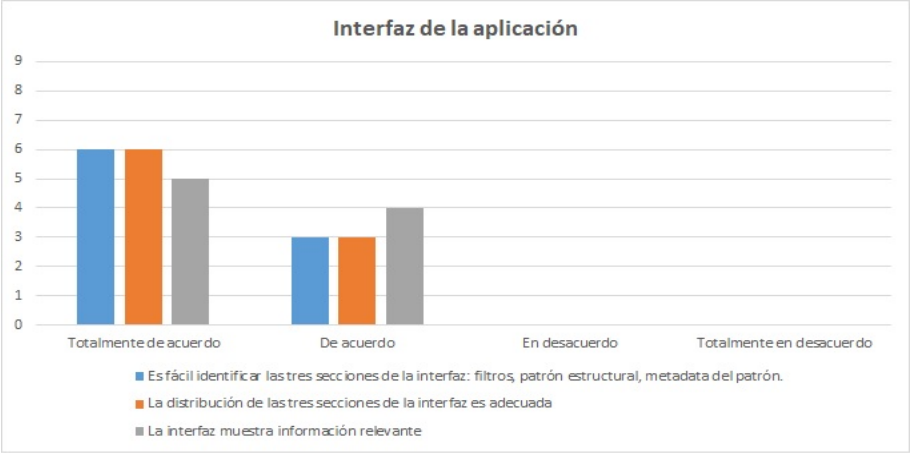



Figura 4.3: Percepción de la interfaz de la aplicación GSPRepository.

Se observa que la percepción más dominante es *Totalmente de acuerdo* seguido muy de cerca de *De acuerdo* (Figura 4.5). Por lo tanto, los usuarios no necesitan demasiado tiempo para aprender a usar la aplicación, logran identificar claramente los elementos de la interfaz (comprensión) y les resulta atractiva respecto a la información mostrada.

### 4.2.2. Filtrado de patrones estructurales

La aplicación GSPRepository proporciona filtros de búsqueda para patrones estructurales y lo hace de la siguiente manera:

1. El sistema ofrece fluidez en la interacción con elementos de la interfaz de filtros
2. El procedimiento para filtrar patrones es la adecuada
3. El despliegue de opciones de filtrado es intuitivo
4. La interfaz de filtros es atractiva

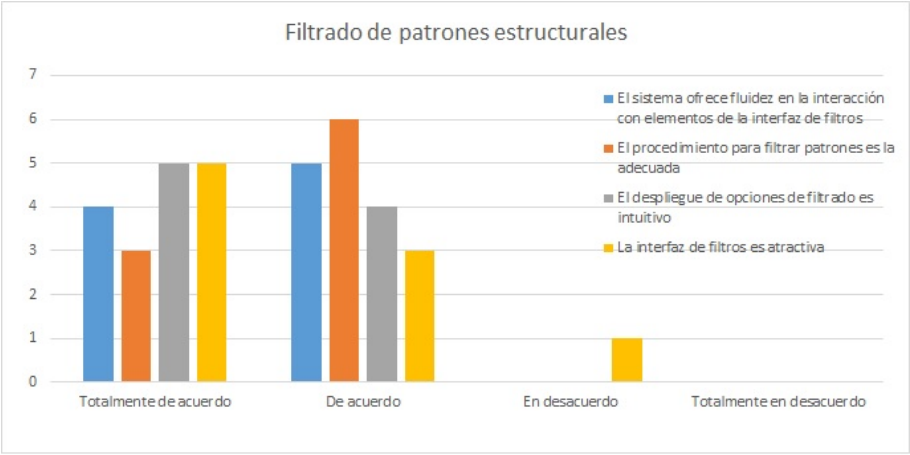



Figura 4.4: Percepción del filtrado de patrones de GSPRepository.

Se observa que en el filtrado de patrones estructurales (Figura 4.6) hay un gran porcentaje de la percepción que recae en la apreciación de *Totalmente de acuerdo* y

*De acuerdo* en proporciones similares. Sin embargo, existe una pequeña proporción que recae en el desacuerdo, pero como es de frecuencia igual a 1, se interpreta como *despreciable*[3].

### 4.2.3. Visualización de patrones

Consiste en qué tan fácil de comprender son los patrones basados en grafos. Las afirmaciones para este bloque son:

- La forma en que se visualizan los patrones permite entender de inmediato lo que se busca
- La visualización de los patrones basado en grafos es clara y se comprende

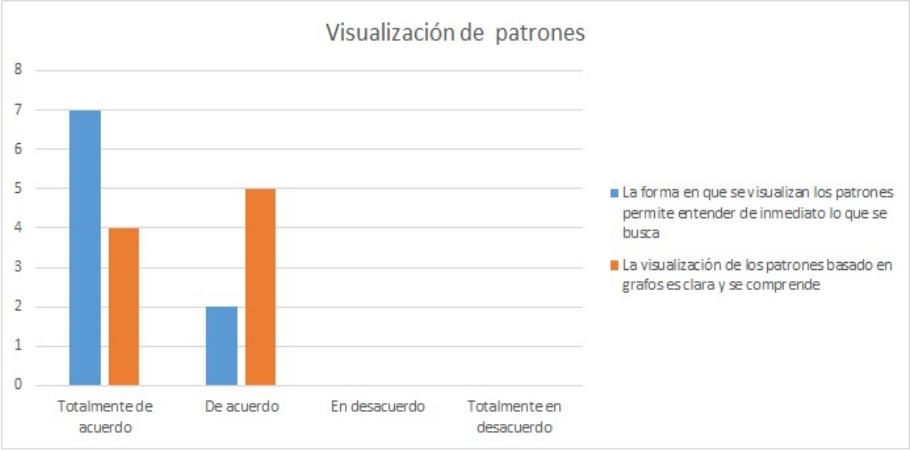



Figura 4.5: Percepción de la visualización de patrones estructurales basados en grafos de GSPRepository.

Se observa que el despliegue de los patrones estructurales (Figura 4.7) en la interfaz es claro (listado hacia abajo) y existen algunas variables a considerar que hacen que el patrón basado en grafos no sea comprensible a simple vista. Dichas variables van desde las relaciones *Gap* a los nodos *Any*. No se especifica explícitamente en la

[3]Que no es lo bastante grande, numeroso o importante como para ser tenido en cuenta

aplicación qué son cada uno, pero al ser experimentado por estudiantes del campo de la Bioinformática no les es tan difícil.

### 4.2.4. Metadatos sobre patrones

Consiste en cómo la información contenida en la metadata de los patrones es mostrada y si resulta comprensible para el usuario, además de destacar ciertos datos que son de importancia.

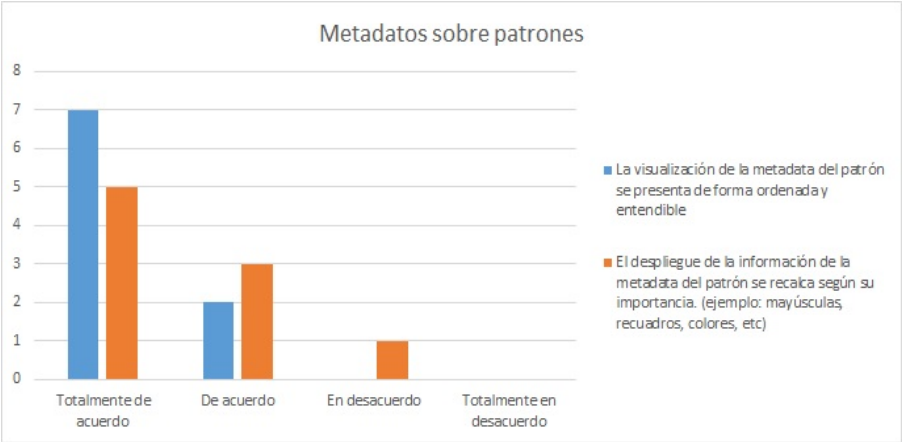



Figura 4.6: Percepción de los metadatos de patrones estructurales.

Se observa que la representación de los metadatos (Figura 4.8) de los patrones es entendible y se despliega de forma ordenada. Por otro lado, existen apreciaciones diversas respecto a los datos destacados, puesto que puede variar según el usuario que interactúe con la aplicación.

### 4.2.5. Funcionalidad de exportar patrones

Consiste en qué tan fácil es reconocer la funcionalidad de exportar (descargar en formato JSON) un patrón estructural y proceder a la descarga misma.

Se observa que los resultados (Figura 4.9) apuntan hacia la dificultad de lograr reconocer la opción de descarga del patrón. No resulta intuitiva y requiere de tiempo para llegar a concretar una descarga.

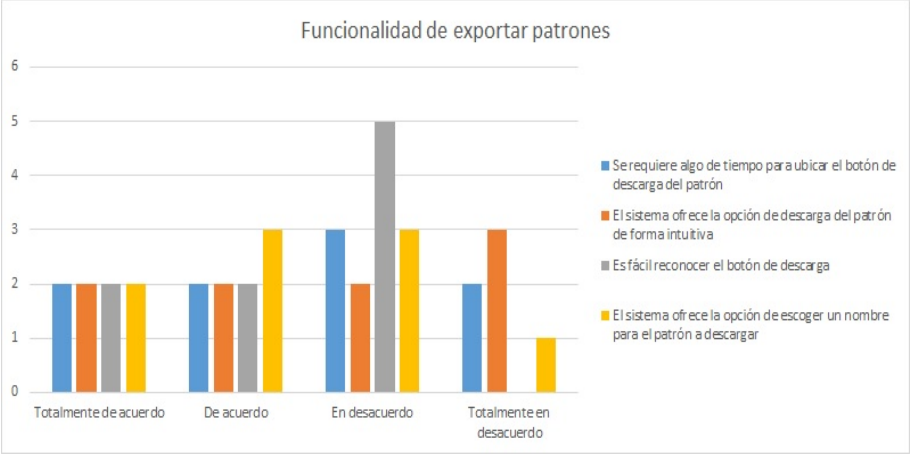



Figura 4.7: Percepción de la funcionalidad de exportar patrones estructurales.

### 4.2.6. Correctitud de la información visualizada

Consiste en qué tan correctos son los patrones visualizados con su respectiva metadata. En estricto rigor, si los valores especificados en los filtros de búsqueda corresponden exactamente a los valores contenidos tanto en la estructura basada en grafos del patrón como en la información contenida en su metadata.

Se observa en la correctitud de la información visualizada (Figura 4.10) que los patrones si son visualizados de acuerdo a que coincide su metadata y elementos de su estructura con los valores especificados en los filtros de búsqueda. Además, se observa un pequeño margen de desacuerdo que corresponde a la definición de una búsqueda usando filtros. Cabe destacar que la visualización de patrones está definida por la *unión de conjuntos*[4]. Dicho esto, hay desacuerdos por parte de los usuarios en cuanto a los patrones que se deben visualizar según ciertos valores de filtros especificados. Por un lado, puede querer interpretarse tanto la *unión* como también la *intersección de conjuntos*[5].

[4]Es una operación que resulta en otro conjunto, cuyos elementos son los mismos de los conjuntos iniciales

[5]Es una operación que resulta en otro conjunto que contiene los elementos comunes a los conjuntos de partida

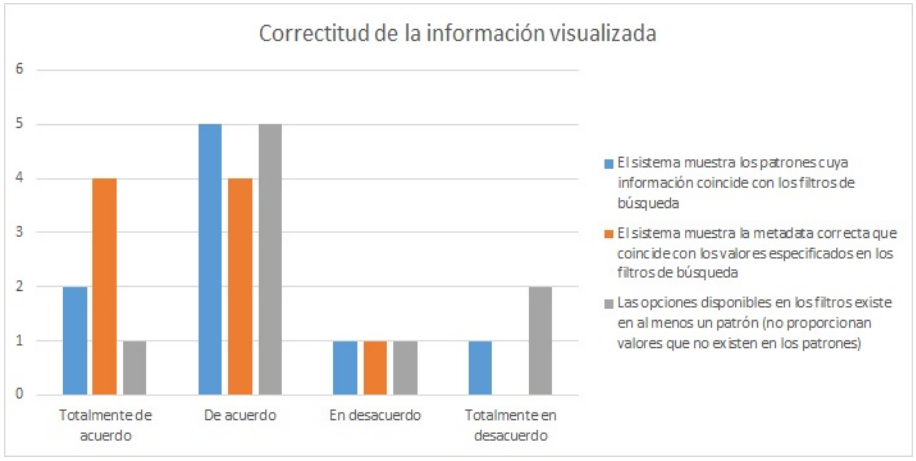



Figura 4.8: Percepción de la correctitud de la información visualizada.

### 4.2.7. Utilidad de la aplicación

Consiste en qué tan útil es GSPRepository para el campo de estudio de la Bioinformática. En estricto rigor, si la búsqueda y visualización de patrones estructurales proteína-ligando potencia el estudio de profesionales del área de Bioinformática.

Se puede observar que la utilidad de la aplicación (Figura 4.11) hay una percepción amplia acerca de la búsqueda de similitudes entre patrones. También, la utilidad con que los filtros son capaces de variar resultados (reales y ficticios) para visualizar. En resumen, la aplicación entrega información correcta acerca de los patrones filtrados y da la opción de valores de filtros correctos para una exploración exitosa.

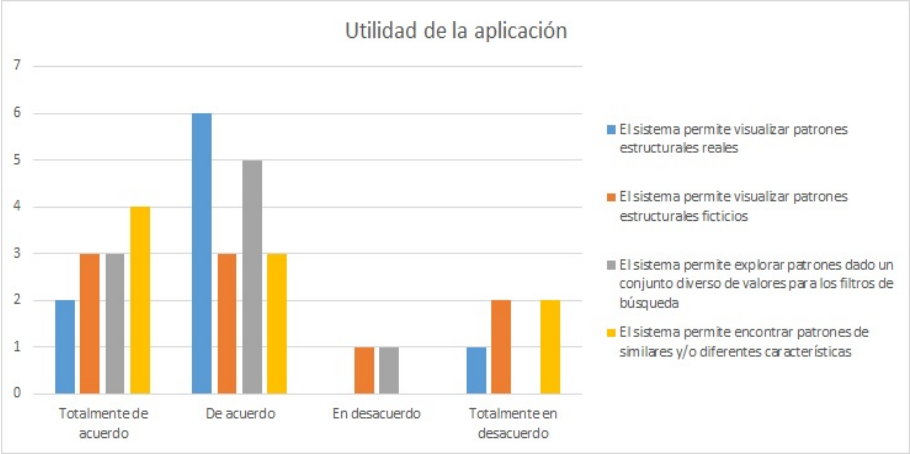



Figura 4.9: Percepción de la utilidad de GSPRepository.

### 4.2.8. Usabilidad de la aplicación

Consiste en qué tan fácil es utilizar la aplicación, con qué rapidez se aprende a usar los elementos funcionales de la interfaz, sean éstos: elección de valores para filtros de búsqueda, la descarga (asignación de nombre y hacer efectivo el procedimiento) del patrón estructural.

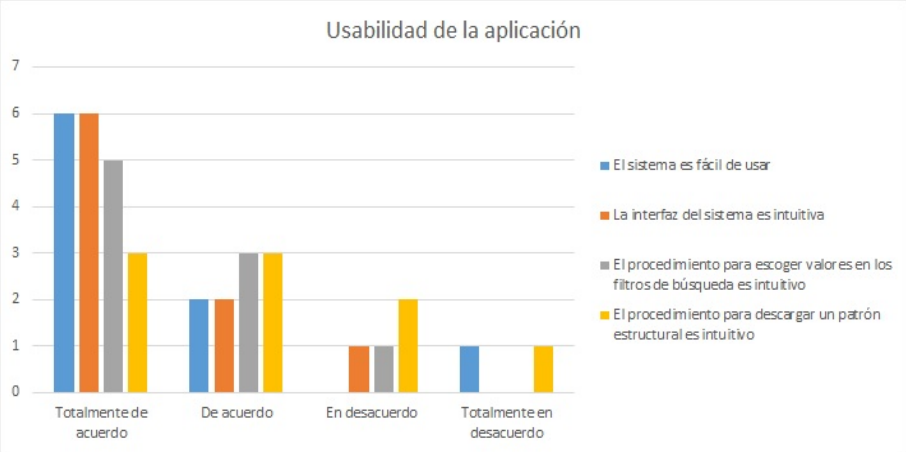



Figura 4.10: Percepción de la usabilidad de GSPRepository.

Se observa una amplia percepción sobre la facilidad (Figura 4.12) con que se utiliza la aplicación, lo rápido que se aprende a utilizar los filtros de búsqueda (inserción y descarte de valores) puesto que resulta intuitivo.

### 4.2.9. Percepción general de la aplicación

Corresponde al grado de aporte que presta la aplicación para el ámbito y estudio de la Bioinformática, la correctitud de los valores disponibles en los filtros de búsqueda y la facilidad y fluidez empleada en el proceso de filtrado, exploración y visualización de patrones en la interfaz.

Es claro que, para los usuarios, la percepción general de GSPRepository (Figura 4.13) proporciona un alto grado de aporte al estudio de la Bioinformática. Al usuario le resulta fácil navegar a través de los filtros de búsqueda y no requiere de tiempo para reconocer o dedicar tiempo a aprender a usar la interfaz de filtros. La interacción es fluída porque a medida que se escogen valores para los filtros, la interfaz visualiza inmediatamente los patrones estructurales, cuya estructura y metadata coincide con estos valores. Le proporciona al usuario una mayor claridad de lo que busca.

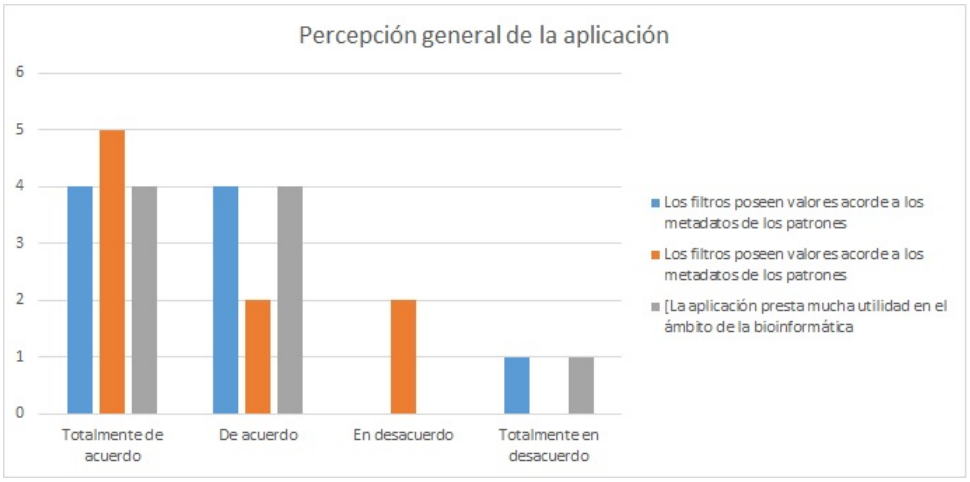



Figura 4.11: Percepción general de GSPRepository.

# 5. Conclusiones

Los patrones estructurales proteína-ligando son estructuras moleculares complejas. El estudio de éstas, utilizando aplicaciones computacionales permite a los profesionales del área de Bioinformática estudiar similitudes y diferencias entre dichas estructuras. Existen muchas aplicaciones disponibles en Internet que permiten a científicos estudiar estas estructuras. Como también se disponen de manera gratuita muchas aplicaciones que almacenen los patrones estructurales proteína-ligando.

Para lo que comprende el desarrollo de GSPRepository, se aprende más allá como una opción de almacenamiento de patrones estructurales. Aplica una solución nueva al problema de la visualización de los patrones estructurales. Debido a que existen diversas formas (o nomenclaturas) de expresar un patrón estructural proteína-ligando, pero no diversas formas de representar gráficamente, utilizando una interfaz gráfica definida basada en grafos: nodos y aristas.

Se busca una forma intuitiva y fácil de aprender a usar que permitirá explorar y visualizar patrones estructurales y obtener información (metadatos) para así dedicar sus esfuerzos en nuevos diseños de patrones para la ciencia, farmacéutica, etc. En otras palabras, desarrollar una aplicación cuyo fin sea potenciar el estudio de la ciencia, en busca del desarrollo de nuevos fármacos, siempre ha sido una motivación. Explorar patrones similares, diferenciados en tan sólo un ligando o algunos aminoácidos para aislar un efecto secundario de un fármaco específico o bien, aumentar los efectos de una droga cuyo estudio se basó en la utilización de herramientas como GSPRepository, resulta ser gratificante.

Para ello, se debe tener en consideración el estudio de las moléculas de proteínas, su estructura, clase, homología, topología, arquitectura, funciones, clasificaciones, familias, etc. En conjunto, son una rama tan grande que inquieta que la exploración

de patrones estructurales sea manual. Actualmente, la cantidad de personas con enfermedades a nivel mundial aumenta a nivel exponencial y los científicos trabajan a diario en busca de nuevos fármacos para curar las enfermedades. Para este caso, la gran motivación fue desarrollar una aplicación que automatice el estudio de exploración de patrones para acelerar la búsqueda y síntesis de nuevas drogas.

GSPRepository es una aplicación gratuita. Está disponible actualmente solamente para quienes estén conectados a la red de la Universidad. Pronto lo estará para cualquier usuario a nivel mundial. La idea de potenciar el estudio, debe ser para todo estudio dedicado en cualquier parte del mundo. Con el tiempo, la aplicación va a disponer de una gran cantidad de patrones estructurales (de cientos a miles) para que los usuarios puedan estudiarlos y analizarlos.

Finalmente, la opción de utilizar GSPRepository no se limita a profesionales, si no que también puede usarse para estudiantes. No se requiere un alto nivel de conocimiento en biología y bioinformática. Se requiere un grado aceptable de conocimientos y es suficiente para entender la aplicación. Junto con ello, tanto profesionales como estudiantes pueden llegar a obtener resultados muy útiles para el estudio y así estar cada vez más cerca de concretar avances en la cura de enfermedades.

# Bibliografía


[1] Stryer L. Berg JM, Tymoczko JL. Biochemistry. 5th edition. *NCBI*, 1, 2002.

[2] Steve Berry. Design considerations for faceted search: Literature review and case study. *University of Texas School of Information*, pages 1–8.

[3] Joseph Constance. Macromolecules: Polysaccharides, proteins and nucleic acids. news-medical. https://www.news-medical.net/life-sciences/Macromolecules-Polysaccharides-Proteins-and-Nucleic-Acids.aspx, 2019.

[4] Marti A. Hearst. Design recommendations for hierarchical faceted search interfaces. *School of Information*, pages 1–5.

[5] Bulliard V Cerutti L Cuche BA de Castro E Lachaize C Langendijk-Genevaux PS Sigrist CJA Hulo N, Bairoch A. New and continuing developments at prosite. *Nucleic Acids Res. 2012*.

[6] T. Lewis D. Lee J. Lees C. Orengo I. Sillitoe, N. Dawson. Cath: Protein structure classification database. http://www.cathdb.info, 2018.

[7] LLC New York NY LigPrep, Schrödinger. Schrödinger release 2019-2. https://www.schrodinger.com/ligprepo, 2019.

[8] Zipursky SL Lodish H, Berk A. Molecular cell biology. 4th edition. 2000.

[9] Gerzon E. Gómez Omar S. Gómez, Juan P. Ucán. Aplicación del proceso de experimentación a la ingeniería de software. *Abstraction Application*, 8:26–37, 2013.

[10] S Salentin. Plip: fully automated protein-ligand interaction profiler. *Nucleic Acids Res.*, pages 443–447.

[11] Mohsin Vahid Khan Maroof Ali Gulam Rabbani Mohd Ishtikhar y Rizwan Hasan Khan Tajalli Ilm Chandel, Masihuz Zaman. A mechanistic insight into protein-ligand interaction, folding, misfolding, aggregation and inhibition of protein aggregates: An overview. *International Journal of Biological Macromolecules*, 106:1115–1129, 2018.

[12] Thornton J M Wallace A C, Laskowski R A. Ligplot: a program to generate schematic diagrams of protein-ligand interactions. *Protein Eng*, 8:127–134, 1996.

[13] Bee Bee Chua y Laurel Evelyn Dyson. Applying the iso 9126 model to the evaluation of an e-learning system. http://www.ascilite.org.au/conferences/perth04/procs/chua.html, 2004.

[14] Renzo Angles y Mauricio Arenas. A graph-based approach for querying protein-ligand structural patterns. September 2017.

[15] David L. Nelson y Michael M. Cox. Lehninger principles of biochemistry: 6th edition. 2012.

[16] Diego Cisterna y Ph.D. Renzo Angles. Diseño de una interfaz gráfica para búsqueda de patrones estructurales en el protein data bank. https://structuralbio.utalca.cl/gsp4pdb1/, Julio 2017.

[17] Diego Cisterna y Ph.D. Renzo Angles. Graph-based structural patterns for pdb. https://structuralbio.utalca.cl/gsp4pdb1/, 2017.

# ANEXOS

# A. Documentos para evaluación de usabilidad

## A.1. Guía de actividades

A continuación, se adjunta la guía de actividades a realizar para interactuar con la aplicación GSPRepository:

| Ejercicios para la evaluación de la aplicación GSPRepository: Repositorio de almacenamiento, exploración y visualización de patrones estructurales proteína-ligando. |
|---|

Esta guía consiste en un conjunto de actividades a realizar para conocer e interactuar con la aplicación web GSPRepository, diseñada para la exploración de patrones estructurales proteína-ligando. Además, se especifican otras actividades en la aplicación GSP4PDB3, la cual permite diseñar patrones estructurales y realizar búsquedas de patrones de similares características a los diseñados en la herramienta.

El objetivo de completar estas actividades es que las personas que lo realicen, aprendan a usar ambas aplicaciones web. Si no les es posible completar alguna de las actividades o no las entienden, proceder con la siguiente.

Al final de este proceso, se les proporcionará a cada persona un cuestionario de evaluación de la aplicación en el que se les pide que respondan para así, valga la redundancia, evaluar ambas aplicaciones.

Dirigirse al sitio de GSPRepository **172.17.17.50/gsprepository** y realice las siguientes actividades:

1. Ir a filtros por ligando (ligands) y filtrar por el ligando Zinc (**ZN**) y verificar si dicho valor coincide con alguno de lo especificado en la metadata de los patrones visualizados y en el grafo representativo del patrón.

   **(Desde el ejercicio 1 al 9 se deben borrar los valores de los filtros de búsqueda utilizados para continuar con el siguiente ejercicio).**

2. Ir a filtros por aminoácido (amino acids) y filtrar por el aminoácido **Glycine** y verificar si dicho valor coincide con lo especificado en la metadata de los patrones visualizados y en el grafo representativo del patrón.

3. Ir a filtros por identificador de proteína (Protein ID) y filtrar por la proteína **2DMD** y verificar si dicho valor coincide con lo especificado en la metadata de los patrones visualizados.

4. Ir a filtros por clasificación de proteína (Protein classification) y filtrar por la clasificacion **TRANSCRIPTION** y verificar si dicho valor coincide con lo especificado en la metadata de los patrones visualizados.

5. Ir a filtros por organismo de proteína (Protein organism) y filtrar por la clasificación **HOMO SAPIENS** y verificar si dicho valor coincide con lo especificado en la metadata de los patrones visualizados.

6. Ir a filtros por familia de patrón (Pattern family) y filtrar por la familia **Zinc Finger** y verificar si dicho valor coincide con lo especificado en la metadata de los patrones visualizados.

7. Ir a filtros por CATH class y filtrar por la clasificación **30** y verificar si el primer valor (si ocurre lo siguiente: **30**.$x.x.x$ con $x$ siendo cualquier número) coincide con lo especificado en la metadata de los patrones visualizados.

8. Ir a filtros por ligando y filtrar por los ligandos **ATP** y **ZN** y verificar si se visualizan los patrones cuyo ligando sea ATP sumado de los patrones cuyo ligando sea ZN.

9. Ir a filtros por aminoácido y filtrar por el aminoácido **Cysteine**. Luego ir al filtro por identificador de proteína y seleccionar la proteína **2DMI** y verificar si se visualizan los patrones los cuales alguno de sus aminoácidos sea Cysteine (Cys) y en su metadata se registre la proteína 2DMI. **(No borrar valores especificados en los filtros para este ejercicio).**

   A continuación se detallarán los pasos para construir un patrón estructural proteína-ligando en GSP4PDB y que será guardado en la base de datos de GSPRepository.

10. Dirigirse a **172.17.17.50/gsp4pdb3** que es la aplicación GSP4PDB3. Iniciar sesión con los siguientes datos:

    Username: **admin**
    Password: **12345**

    Una vez dentro, dibuje un patrón estructural simple (de no más de 2 o 3 nodos) utilizando los botones de navegación que proporciona GSP4PDB3.


Figura A.1: Botones de navegación de GSP4PDB3.

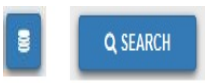



Figura A.2: Exportar patrones (izquierda), realizar búsqueda (derecha)

**Nota:** Recuerde que para establecer una relación (unir elementos) entre dos nodos, debe dibujar los nodos en el canvas, escoger un tipo de arrista (distance, next), clickear en un primer nodo y un segundo click con el nodo que se quiere conectar.

11. Una vez diseñado el patrón, realice la búsqueda de patrones estructurales para encontrar patrones con similares características haciendo en *Search* (Figura A.1).

12. Una vez completada la búsqueda (aunque no arroje resultados), proceder a exportar patrón estructural junto con su metadata haciendo en *Exportar* (Figura A.2).

## A.2. Cuestionario de evaluación

Se detalla el cuestionario que va ser respondido por estudiantes de Ingeniería en Bioinformática:

| Encuesta de evaluación de GSPRepository |
|---|

Se ha desarrollado un sistema que almacena patrones estructurales proteína-ligando y además permite explorar y visualizarlos a través de una interfaz gráfica. En concreto, el sistema permite, a través de filtros de búsqueda, explorar estos patrones de acuerdo a etiquetas determinadas por el usuario y visualizar aquellos patrones que cumplan con dichas etiquetas especificadas. Una vez visualizados los patrones, éstos muestran su metadata (información).

Esta encuesta se realiza para evaluar el aprendizaje, comprensión y atractividad.

**Datos Generales**

Ingeniería Civil en Computación - Universidad de Talca

**Edad**

Marca una opción

- ○ menor a 18 años
- ○ 18 a 21 años
- ○ 22 a 25 años
- ○ mayor a 25 años

**Género**

Marca una opción

- ○ Masculino
- ○ Femenino

**Ocupación**

Marcar su principal ocupación para caracterizar a los participantes.

- ○ Estudiante de Ingeniería en Bioinformática
- ○ Docente de Ingeniería en Bioinformática
- ○ Empleado independiente
- ○ Otro

Especifique cual:

**Actividad económica**

Indicar la principal actividad que realiza de acuerdo a su profesión u ocupación (ejemplo: Microempresario, Ingeniero, Motoboy, Estudiante, etc).

Respuesta:

En esta sección, debe seleccionar el grado de *de acuerdo* o *desacuerdo* referente a la utilización del sistema.

**Interfaz de la aplicación**

1. Es fácil identificar las tres secciones de la interfaz: filtros, patrón estructural, metadata del patrón.
2. La distribución de las tres secciones de la interfaz es adecuada.
3. La interfaz muestra información relevante.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Filtrado de patrones estructurales**

1. El sistema ofrece fluidez en la interacción con los elementos de la interfaz de filtros.
2. El procedimiento para filtrar patrones es el adecuado.
3. El despliegue de opciones de filtrado es intuitivo.
4. La interfaz de filtros es atractiva.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Visualización de patrones**

1. La forma en que se visualizan los patrones permite entender de inmediato lo que se busca.
2. La visualización de los patrones basado en grafos es clara y se comprende.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Metadatos sobre patrones**

1. La visualización de la metadata del patrón se presenta de forma ordenada y entendible.
2. El despliegue de la información de la metadata del patrón se recalca según su importancia (ejemplo: mayúsculas, recuadros, colores, etc).

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Visualización de patrones**

1. Se requiere algo de tiempo para ubicar el botón de descarga del patrón.
2. El sistema ofrece la opción de descarga del patrón de forma intuitiva.
3. Es fácil reconocer el botón de descarga.

4. El sistema ofrece la opción de escoger un nombre para el patrón a descargar.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Funcionalidad de exportar patrones**

1. Requiere tiempo aprender a construir un patrón para luego realizar una búsqueda.
2. Existe fluidez al momento de querer exportar un patrón al cabo de realizar una búsqueda.
3. Se reconoce qué funcionalidad cumple cada botón de navegación.
4. El sistema ofrece la ventaja de importar un patrón (en vez de diseñarlo) y realizar búsqueda de patrones similares.
5. Resulta atractivo cargar un patrón en vez de diseñarlo desde cero.
6. La visualización del patrón basado en grafo una vez cargado es igual en término de distribución de nodos.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Correctitud de la información visualizada**

1. El sistema muestra los patrones cuya información coincide con los filtros de búsqueda.

2. El sistema muestra la metadata correcta que coincide con los valores especificados en los filtros de búsqueda.
3. La opción disponible en los filtros existe en al menos un patrón (no proporcionan valores que no existen en los patrones).

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Utilidad del sistema**

1. El sistema permite visualizar patrones estructurales reales.
2. El sistema permite visualizar patrones estructurales ficticios.
3. El sistema permite explorar patrones dado un conjunto diverso de valores para los filtros de búsqueda.
4. El sistema permite encontrar patrones de similares y/o diferentes características.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Usabilidad del sistema**

1. El sistema es fácil de usar.
2. La interfaz del sistema es intuitiva.

3. El procedimiento para escoger valores en los filtros de búsqueda es intuitivo.
4. El procedimiento para descargar un patrón estructural es intuitivo.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Cómo le ha parecido la aplicación web respecto a:**

1. Facilidad y fluidez en el procedimiento de filtrado, exploración y visualización de patrones.
2. Los filtros poseen valores acordes a los metadatos de los patrones.
3. La aplicación presta mucha utilidad en el ámbito de la bioinformática.

○ Totalmente de acuerdo

○ De acuerdo

○ En desacuerdo

○ Totalmente en desacuerdo

**Preguntas** En esta sección se solicita responder las siguientes preguntas:

- ¿El sistema presentó algún error al momento de utilizarse? Respuesta:
- ¿Qué cambios o mejoras le haría usted al sistema? Respuesta:
- ¿Utilizaría la aplicación GSPRepository para futuras investigaciones? Respuesta: