Universidad de las Ciencias Informáticas

Facultad de Ciencias y Tecnologías Computacionales

Título: Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad para la predicción de propiedades biológicas en proteínas.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Juan Carlos Sánchez Rodriguez

Tutor: M.Sc. Ernesto Contreras Torres

La Habana, Junio 2017

"Año 59 de la Revolución"

Frase

"No existen más que dos medios para lograr un objetivo importante y para realizar grandes obras: La Fuerza y la Perseverancia."

Wolfgang Goethe

DECLARACIÓN DE AUTORIA

Declaramos ser autores de la presente tesis que tiene por título: "Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad para la predicción de propiedades biológicas en proteínas" y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Juan Carlos Sánchez Rodriguez

_____

Firma del Autor

Ernesto Contreras Torres

_____

Firma del Tutor

DATOS DE CONTACTO

Autor:

Juan Carlos Sánchez Rodriguez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Email: juansr@uci.cu

Tutor:

M.Sc. Ernesto Contreras Torres

Email: econtreras@uci.cu

AGRADECIMIENTOS

A mi familia, que siempre ha estado presente y me ha dado su apoyo incondicional. Que me han acompañado en este largo viaje para alcanzar mi objetivo.

A mi tutor, que con su asesoramiento durante todo el proceso creativo me ha permitido llegar a este momento.

A mis profesores que guiaron mi formación profesional.

A mis compañeros de aula con quienes he compartido estos años, “en las buenas y en las malas”.

A mis amigos del diurno, Adrián, Alberto, Ernesto y Randy, que siempre estuvieron presentes y me brindaron su apoyo.

A Juan Carlos Pérez por su valiosa ayuda al brindarme el laboratorio para realizar los cálculos de la investigación.

A Miguel Ángel por su preocupación constante por este trabajo.

A todos los que de una forma u otra han estado presentes durante estos años.

DEDICATORIA

A mis padres, que son mi guía y mi faro, mi orgullo y mi inspiración. Por creer en mí, por impulsarme a seguir adelante, por hacer de mi la persona que soy.

A mi hermana, por estar siempre a mi lado, por sus consejos y su apoyo incondicional.

RESUMEN

El desarrollo de fármacos es una tarea en extremo compleja, pero también muy apreciada por la sensibilidad que genera el impacto negativo de las enfermedades en la sociedad moderna. Dada la importancia que tiene para la industria farmacéutica la identificación de propiedades biológicas en proteínas, resulta necesario el desarrollo métodos que predigan sus atributos. En el presente trabajo se desarrolla un Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad calculados con el software ToMoCoMD-CAMPS, para predecir la clase estructural y la velocidad de plegamiento de las proteínas. Para ello se realizó un estudio a partir del cual fueron desarrollados un grupo de modelos, de los cuales fueron seleccionados 2 para integrar al Sistema Experto, uno de clasificación, obtenido con la técnica Random Forest, que presenta una exactitud global en la serie de entrenamiento de 100% y un 98 % en la serie de predicción y uno de regresión, obtenido con la técnica Regresión Lineal Múltiple, que presenta valores de Q^2_{loo}=0.7612 y Q^2_{ext}=0.7263. Ambos son modelos robustos y con alto poder predictivo, aventajando a otros modelos reportados en investigaciones precedentes. Se anticipa la potencial aplicación del sistema desarrollado como una herramienta complementaria a los enfoques precedentes en la predicción de propiedades biológicas en proteínas.

Palabras Clave: clases estructurales de proteínas, Random Forest, Regresión Lineal Múltiple, Sistema Experto, velocidad de plegamiento de proteínas.

Abstract

The development of drugs is an extremely complex task, and at the same time highly appreciated due to the sensitivity caused by the negative impact of diseases in modern society. Given the importance to the pharmaceutical industry of the identification of biological properties in proteins, it is necessary to develop methods that predict their attributes. In the present work an Expert System was developed, based on models of bilinear features of the proximity matrix calculated with the software ToMoCoMD-CAMPS, to predict the structural class and folding rate of the proteins. A study was carried out from which a group of models were developed, two of them were selected to integrate the Expert System, one model obtained by the Random Forest technique, which has a global accuracy in the training series of 100 % and 98% in the prediction series, and a regression model obtained with the Multiple Linear Regression technique with values of Q^2_{loo}=0.7612 and Q^2_{ext}=0.7263. Both are robust models with high predictive levels, and they get better results than others previously reported findings. It is anticipated the potencial application of the developed system as a complementary tool to the existing approaches for the prediction of proteins biological properties.

Key words: Expert System, Multiple Linear Regression, proteins folding rate, Random Forest, structural class of proteins.

Índice

Índice de Figuras

Índice de Tablas

Índice de Tablas

Introducción

El desarrollo de una terapia para determinada patología es un proceso comúnmente constituido por tres pasos. El primer paso es la identificación de la diana biológica o terapéutica, es decir, la identificación de una molécula biológica, principalmente proteínas, involucrada en algún mecanismo implicado en algún proceso patológico. Un estudio relativamente reciente desarrollado por el Boston Consulting Group y que implicó a 50 compañías e instituciones académicas, reveló que el proceso de desarrollo de un nuevo medicamento hasta su uso autorizado en terapéutica requiere, en promedio, la inversión de 880 millones de dólares (USD) y 15 años de investigación. Lo anterior evidencia la alta complejidad asociada a la tarea de desarrollar "un nuevo medicamento", pero también muy valorada por la sensibilidad que genera el impacto negativo de las enfermedades en la sociedad moderna (Marrero-Ponce et al. 2013).

El descubrimiento de nuevas (secuencias y estructuras 3D de proteínas) y el conocimiento de sus atributos ha creado una brecha que está en continuo incremento, por lo que constituye un reto para la comunidad científica desarrollar métodos computacionales que predigan sus atributos. Estudios realizados en diversos laboratorios de investigación alrededor del mundo han indicado que: el análisis matemático, el modelado computacional y la introducción de nuevos conceptos en la biología y la medicina, así como el análisis gráfico, permiten una mejor comprensión en el desarrollo de investigaciones básicas y el diseño de fármacos y como consecuencia son bien acogidos por la comunidad científica (Chou 2011; Chou 2015; Randić et al. 2012). Debido a la necesidad de explotar las cantidades masivas de datos generados por las tecnologías de alto rendimiento, los métodos computacionales se han ido implementando de manera creciente en el proceso de descubrimiento de fármacos (Xu and Hagler 2002).

En los últimos años, la industria farmacéutica ha encauzado sus investigaciones hacia aquellos métodos que permitan describir de manera eficiente y de bajo costo computacional la estructura química de las moléculas candidatas a medicamentos, así como, predecir la respuesta biológica de prácticamente cualquier compuesto orgánico con altas probabilidades de acierto (Tropsha 2010). Dichos métodos se enmarcan en los estudios QSAR (siglas del inglés *Quantitative Structure Activity Relationship*), estos permiten estimar con aceptable grado de precisión, la actividad/propiedad de nuevos compuestos, por lo que pueden aplicarse como estrategia de tamizaje virtual como alternativa a los costosos procesos de síntesis y bioensayos (González-Díaz et al. 2005; González-Díaz et al. 2007).

Los métodos QSAR han demostrado que las relaciones entre la estructura molecular y las propiedades químico-físicas o actividades biológicas de los compuestos se pueden cuantificar matemáticamente a

partir de parámetros estructurales simples, estos cuantificadores matemáticos se conocen como descriptores o índices. De acuerdo a la naturaleza en su definición y a la complejidad de los rasgos moleculares estructurales que se codifican, los descriptores moleculares (DMs) se clasifican de forma general según las dimensiones que abarcan en: DMs-0D (Descriptores Constitucionales), DMs-1D (Descriptores Unidimensionales), DMs-2D (Descriptores Bidimensionales o Invariantes de Grafos), DMs-3D (Descriptores Tridimensionales) y DMs-4D (Descriptores Tetradimensionales) (Todeschini and Consonni 2009).

El desarrollo de descriptores para pequeñas moléculas orgánicas ha sido intensamente abordado (Todeschini and Consonni 2009), sin embargo, se ha dedicado menos esfuerzo a codificar secuencias de proteínas y en mucho menor grado su estructura 3D como se evidencia en (Di Paola et al. 2012; González-Díaz et al. 2005; González-Díaz et al. 2007; Randić et al. 2012). Por otra parte, resulta altamente complejo modelar todas las interacciones biológicas con un único descriptor o un pequeño número de descriptores (Randić et al. 2012). Por lo tanto, la búsqueda de nuevos descriptores y nuevas representaciones de proteínas constituye un área importante en ciencia de proteínas, como se argumenta en recientes revisiones (Chou 2011; Chou 2015).

El desarrollo de métodos para la caracterización numérica de proteínas, también conocidos como descriptores biomacromoleculares y su aplicación combinada con técnicas estadísticas y/o de aprendizaje automático, ha demostrado ser efectiva en la predicción de propiedades biológicas de interés (Chou 2015; Marrero-Ponce et al. 2015). En recientes reportes se desarrolló un procedimiento basado en las formas algebraicas bilineales, para la obtención de descriptores 3D-proteicos (*Multi-Linear Maps based on N-Metric and Amino Acids Weightings* [MuLiMs-MCoMPAs]), los cuales se aplicaron con resultados prometedores en la identificación de las clases estructurales de proteínas (Marrero-Ponce et al. 2015). Por lo tanto, resulta importante en primera instancia, evaluar su desempeño en la modelación de otras propiedades biológicas y poner a disposición pública el nuevo método de predicción, como se sugiere en (Chou 2011).

Por lo anteriormente expresado se plantea el siguiente **problema a resolver:** ¿Cómo contribuir a la identificación de propiedades biológicas en proteínas?

Para dar respuesta al problema planteado se define como **objetivo general:** Desarrollar un Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad, para la predicción de la clase estructural y la velocidad de plegamiento de proteínas.

Se define como **objeto de estudio:** Sistema Experto basado en modelos, enmarcado en el **campo de acción:** Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad, para la predicción de la clase estructural y la velocidad de plegamiento de proteínas.

La **hipótesis** de la siguiente investigación se expresa a continuación: Si se implementa un Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad, entonces es posible predecir la clase estructural y la velocidad de plegamiento de proteínas.

Las **tareas de la investigación** definidas para cumplir con el objetivo general son:

1. Revisión bibliográfica de los Sistemas Expertos basados en modelos.
2. Desarrollo de estudios exploratorios (de variabilidad y redundancia) de los descriptores 3D-proteicos denominados (*Multi-Linear Maps based on N-Metric and Amino Acids Weightings* [MuLiMs-MCoMPAs]), para reducir el espacio de alta dimensión de rasgos moleculares.
3. Desarrollo de un modelo de clasificación basado en rasgos bilineales de la matriz de proximidad, para la identificación de las clases estructurales de proteínas.
4. Desarrollo de un modelo de regresión de rasgos bilineales de la matriz de proximidad, para la predicción de la velocidad de plegamiento de proteínas.
5. Realización del análisis y diseño del Sistema Experto.
6. Implementación del Sistema Experto.
7. Evaluación del Sistema Experto.

En el desarrollo del presente trabajo se utilizan los siguientes **métodos de investigación**:

Métodos teóricos

Se basan en la utilización del pensamiento en sus funciones de deducción, análisis y síntesis (Barchini 2005). En la presente investigación los métodos teóricos a utilizar son:

- ✓ **Método analítico - sintético**: A través de este método, se realiza el análisis de las distintas fuentes bibliográficas relacionadas con Inteligencia Artificial, Sistemas Expertos, así como, investigaciones en el campo de la bioinformática relacionadas con las propiedades biológicas en proteínas, con el fin de recabar toda la información necesaria que permita dar solución al problema de investigación planteado.

- ✓ **Modelación**: La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad. Se utilizará en la modelación de los diagramas dentro de la metodología de desarrollo de software seleccionada para llevar a cabo la solución.

Métodos empíricos

Se aproximan al conocimiento del objeto mediante sus conocimientos directos y el uso de la experiencia (Barchini 2005). En la presente investigación el método empírico a utilizar es:

- ✓ **Análisis documental**: Este método se utiliza para revisar los documentos, artículos, libros y revistas especializadas en los temas referentes a estudios sobre proteínas y Sistemas Expertos en el campo de la bioinformática, con el objetivo de obtener toda la información necesaria y actualizada que permita cumplir con el objetivo planteado.

El presente documento está estructurado en 3 capítulos, los cuales se describen a continuación:

Capítulo 1 Fundamentación Teórica para la Creación de un Sistema Experto.

En este capítulo se abordarán los aspectos teóricos más importantes que sirven de referente para el diseño e implementación del Sistema Experto PropPred-ES. Además, se presentan las herramientas y tecnologías a utilizar en el desarrollo del mismo.

Capítulo 2 Desarrollo del Sistema Experto para la Predicción de Propiedades Biológicas en Proteínas.

Este capítulo estará organizado de acuerdo a la Metodología de Ingeniería del Conocimiento según John Durkin, abordando en cada acápite los elementos correspondientes a las tres primeras fases de la misma: Evaluación, Adquisición del conocimiento y Diseño.

Capítulo 3 Evaluación del Sistema Experto para la Predicción de Propiedades Biológicas en Proteínas.

En este capítulo se mostrarán los principales elementos que se corresponden con la fase de prueba. Se evaluará el desempeño de los modelos obtenidos y se realizará la validación del sistema.

Capítulo 1 Fundamentación Teórica para la Creación de un Sistema Experto

En este capítulo se hace referencia a varios conceptos sobre Inteligencia Artificial, así como diferentes Sistemas Expertos, su arquitectura y aplicación en la rama de los estudios bioinformáticos. Se describen y comparan algunas metodologías para el desarrollo de Sistemas Expertos para elegir y argumentar la seleccionada. Además, se exponen los métodos estadísticos y de aprendizaje automático y las herramientas y tecnologías a utilizar para el desarrollo de PropPred-ES.

1.1 Inteligencia Artificial

La Inteligencia Artificial (IA) constituye un campo muy atractivo para los científicos hoy día, este término se refiere a la capacidad de emular las funciones inteligentes del cerebro humano. Algunos autores como los pioneros de la investigación en IA (Barr and Feigenbaum 1981) definen esta como: “La rama de la Ciencia que se ocupa del diseño de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento y la resolución de problemas”.

Las definiciones han cambiado con el curso del tiempo debido a la rápida evolución en este campo de estudio. En algunos libros más actuales se plantea que: “La Inteligencia Artificial estudia cómo lograr que las máquinas realicen tareas que, por el momento, son realizadas mejor por los seres humanos” (Rich et al. 2000), en otros que es: “El esfuerzo por hacer que las computadoras piensen, máquinas con mentes en el más amplio sentido literal, o la automatización de actividades que están asociadas con el pensamiento humano, actividades como la toma de decisiones, resolución de problemas y aprendizaje” (Russell et al. 2010).

A pesar de ser la Inteligencia Artificial un término muy debatido en la comunidad científica y no existir aún un consenso en cuanto a su concepto, todos coinciden en que la misma está orientada a conseguir que las máquinas realicen actividades que requieran de las capacidades de razonamiento y los conocimientos de un ser humano. El empleo de la IA es variado y actualmente se utiliza principalmente en: procesamiento de lenguaje natural, lenguajes de programación y software, robótica, deducción y prueba de teoremas, aplicaciones y Sistemas Expertos entre otros (Banda 2014).

1.2 Los Sistemas Expertos

Los Sistemas Expertos (SE) fueron desarrollados por la comunidad de Inteligencia Artificial a mediados de la década del 60 y su nombre deriva del término Sistema Experto Basado en Conocimiento. Según Durkin: “Un Sistema Experto puede definirse como un sistema informático, el cual se encuentra compuesto por

hardware y software, los cuales tienen la capacidad de simular a los expertos humanos en determinada área de especialización" (Durkin 1998).

Un SE emplea conocimiento humano capturado en una computadora para resolver problemas que normalmente requieran de expertos humanos. Los sistemas bien diseñados imitan el proceso de razonamiento que los expertos utilizan para resolver problemas específicos y pueden funcionar mejor que cualquier humano experto individualmente, tomando decisiones en una específica y acotada área de pericia denominada como dominio (Turban 1995).

Características de los Sistemas Expertos (Banda 2014).

- ✓ Solidez en el dominio de su conocimiento.
- ✓ Capacidad para resolver problemas.
- ✓ Fiabilidad en los resultados obtenidos.
- ✓ Habilidad para adquirir conocimiento.

1.2.1 Ventajas de los Sistemas Expertos

A pesar de que el desarrollo o la adquisición de un Sistema Experto es generalmente caro, la ganancia en términos monetarios, tiempo y precisión resultante de los mismos son muy altas, por lo que consideramos una serie de ventajas y razones para utilizarlos. Algunas de las ventajas más importantes que aportan los Sistemas Basados en el Conocimiento con respecto a otros sistemas de soporte para la decisión y sistemas de información pueden cifrarse en las siguientes (López Sánchez and Carretero Díaz 2016):

- ✓ Permiten almacenar, expresar y utilizar el conocimiento de los grandes expertos y toda su dilatada experiencia.
- ✓ Un solo Sistema Experto puede apoyar las decisiones de muchas personas a la vez, ya que puede estar disponible en todo momento y en varios lugares al mismo tiempo.
- ✓ Puede mejorar la productividad del sistema (menor tiempo de respuesta) y la cualificación de los decisores, estos aprenden con él muy fácilmente, pues una de sus cualidades es que explica los fundamentos de su decisión, lo que en multitud de ocasiones es más importante que la propia decisión.
- ✓ El Sistema Experto puede proporcionar estabilidad y consistencia al proceso de decisión en un área determinada, de forma que las decisiones posteriores son consistentes con las previas, lo que no siempre ocurre con los decisores humanos.
- ✓ Reduce la dependencia frente al personal. Los Sistemas Expertos vienen a terminar con la escasez de expertos disponibles, o a reducir el coste de acceso a dicho conocimiento.

- ✓ Es una excelente herramienta de entrenamiento, pues justifica sus decisiones. Así los decisores pueden analizar el porqué de la decisión y utilizarlo como técnica de formación.

1.2.2 Arquitectura de los Sistemas de Expertos

Los Sistemas Expertos son, a la vez, un sistema de ejecución y un sistema de transmisión del conocimiento. En la revisión bibliográfica realizada se encontró que los diferentes autores definen varias arquitecturas para un SE, pero todas coinciden en que los principales componentes de la misma son los descritos a continuación:

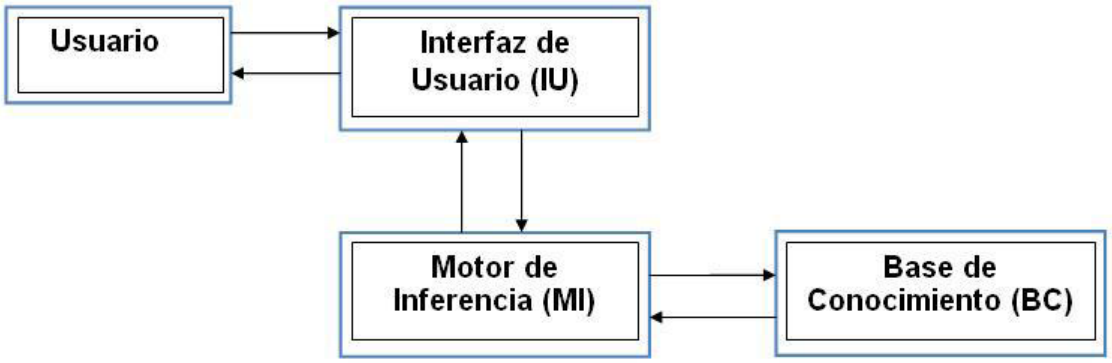


Fig.1 Arquitectura de un Sistema Experto según Jackson.

Interfaz con el usuario: Mediante ella el usuario plantea los problemas al Sistema Experto, recibe preguntas del mismo y ofrece las explicaciones necesarias. Sirve para diseñar, crear, actualizar y usar los SE. El propósito general de la interfaz del usuario es facilitar a los usuarios y a los que toman las decisiones el desarrollo y el uso de un Sistema Experto (Jackson 1998).

Motor de inferencia: Implementa algún método de solución de problemas que manipula el conocimiento almacenado en la base de conocimiento e informaciones sobre estados iniciales, estados actuales de la solución del problema, etc., las cuales procesan dinámicamente en una estructura que se le llama base de datos o memoria de trabajo. El propósito general de un motor de inferencias es buscar información y relaciones en la base de conocimientos y proporcionar respuestas, pronósticos y sugerencias en la misma forma en que lo haría un experto humano (Jackson 1998).

Base de Conocimiento: En ella se almacena el conocimiento sobre algún dominio de aplicación mediante alguna forma de representación del conocimiento, es el corazón del Sistema Experto. El propósito de una base de conocimientos es contener los hechos y la información pertinentes para el SE específico (Jackson 1998).

1.2.3 Tipos de Sistemas Expertos

El campo de la representación del conocimiento se refiere a los mecanismos para representar y manipular la información. Dependiendo de la forma de representación del conocimiento, se encontraron diferentes variantes de Sistemas Basados en el Conocimiento descritos por diferentes autores.

Sistemas Basados en Reglas: Un sistema basado en reglas es aquel que contiene los datos obtenidos a partir de un experto humano y representa esa información en forma de reglas con la estructura IF-THEN. Las reglas pueden usarse para realizar operaciones de inferencia sobre los datos y así lograr obtener una conclusión apropiada. Estas inferencias son esencialmente un programa de computadora que provee una metodología para razonar sobre la información en la base de reglas o base de conocimiento y formular conclusiones (Flasiński 2016).

Sistemas Basados en Casos: La idea básica del Razonamiento Basado en Casos (RBC) es adaptar soluciones que se usaron para resolver problemas previos y utilizarlas para solucionar nuevos problemas. En el RBC las experiencias descritas por los especialistas humanos y representadas como casos, son guardadas en una base de datos para su uso posterior cuando el usuario encuentre un nuevo caso con parámetros similares. El sistema trata de buscar en los casos almacenados con características problemáticas similares al nuevo aquel que más se ajuste y aplicar las soluciones del caso viejo al nuevo caso. Las soluciones exitosas son marcadas para el nuevo caso y ambos son almacenados conjuntamente con los otros casos en la base de conocimiento. Las soluciones infructuosas también son anexadas al caso de base junto con explicaciones en lo que se refiere a por qué las soluciones no surtieron efecto (Kolodner 2014).

Sistemas Expertos Difusos: Los Sistemas Expertos Difusos se desarrollan usando el método de Lógica Difusa, la cual trabaja con incertidumbre. Esta técnica emplea el modelo matemático de conjuntos difusos, simula el proceso del razonamiento normal humano permitiendo a la computadora comportarse menos precisa y más lógicamente que las computadoras convencionales. Este enfoque es utilizado porque la toma de decisiones no es siempre una cuestión de blanco y negro, verdadero o falso, a veces involucra áreas grises y el término “quizás” (Zimmermann 2012).

De acuerdo a la forma de representación del conocimiento en el área de estudio en que se enmarca el presente trabajo, se consideró utilizar el Sistema Basado en Modelos.

El Razonamiento Basado en Modelos se refiere al método de inferencia utilizado en los Sistemas Expertos basados en modelos concretos a partir de un área específica del conocimiento. En este enfoque, la elaboración del modelo es el núcleo principal de desarrollo de la aplicación. Durante la ejecución, el motor

de inferencia combina los modelos de conocimientos con los datos observados, para derivar conclusiones que lleven a un diagnóstico o predicción. Los modelos pueden ser cuantitativos, Ej. basado en ecuaciones matemáticas, o cualitativos, Ej. basado en modelos de causa /efecto. Pueden incluir representaciones de incertidumbre y con el paso del tiempo presentar cambios en su comportamiento (Russell et al. 2010).

1.2.4 Aplicaciones de los Sistemas Expertos

Los Sistemas Expertos constituyen una herramienta potencial para manejar grandes volúmenes de información con una gran velocidad de procesamiento, obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Razonan, pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad, por lo que su uso es especialmente recomendado cuando los expertos humanos en una determinada materia son escasos, en situaciones complejas donde la subjetividad humana puede llevar a conclusiones erróneas y cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión. Es por ello que los mismos se han implantado en diversas áreas del conocimiento donde los expertos humanos sean escasos, Ej. la medicina, la meteorología, la aeronáutica, la informática, biología entre otras (Liao 2005).

1.2.5 Sistemas Expertos en el campo de la Bioinformática y Biología Computacional

Tabla. 1 Ejemplos de SE para la predicción de propiedades biológicas en proteínas.

Sistema	Descripción	Método de Predicción
Euk-mPLoc 2.0 (http://www.csbio.sjtu.edu.cn/bioinf/euk-multi-2/)	Predice la localización subcelular de proteínas eucarióticas, incluyendo aquellas que se encuentran en múltiples lugares.	Grupo de clasificadores formado mediante la fusión de los clasificadores individuales OET-KNN (Evidencia Teórica Optimizada del Vecino Cercano).
Plant-mPLoc (http://www.csbio.sjtu.edu.cn/bioinf/plant-multi/)	Predice la ubicación subcelular de proteínas en plantas.	Grupo de clasificadores formado mediante la fusión de varios clasificadores individuales básicos, utilizando la regla de OET-KNN (Evidencia Teórica Optimizada del Vecino Cercano).
Gpos-mPLoc (http://www.csbio.sjtu.edu.cn/bioinf/Gpos-multi/)	Predice la ubicación subcelular de proteínas Gram-Positivo.	Grupo de clasificadores formado mediante la fusión de los clasificadores individuales OET-KNN (Evidencia Teórica Optimizada del Vecino Cercano).
K-Fold (http://gpcr2.biocomp.unibo	Predictor del mecanismo de velocidad de plegamiento de las proteínas.	Máquina de Soporte Vectorial.

.it/cgi/predictors/K-Fold/K-Fold.cgi)		
Fold-Rate (http://www.csbio.sjtu.edu.cn/bioinf/FoldRate/)	Predicción de la velocidad de plegamiento de las proteínas a partir de su secuencia de aminoácidos.	Conjunto predictor desarrollado fusionando tres predictores individuales, cada uno basado en el tamaño de la proteína, su efecto de hélice y su efecto de hoja respectivamente.
PPT-DB (http://www.pptdb.ca/)	Predicción de la estructura secundaria, parámetros de la estructura 3D y velocidad de plegamiento de las proteínas. Además, contiene bases de datos de proteínas que pueden ser empleadas por desarrolladores de software para predecir propiedades de las proteínas.	Mapeo de propiedades basadas en homología.
Prorate (http://sunflower.kuicr.kyoto-u.ac.jp/~sjn/folding/webserver.html)	Predicción de la velocidad de plegamiento de las proteínas en las topologías estructurales y las redes de propiedades complejas.	Regresión de soportes vectoriales empleando modelos de redes basados en dos escalas de longitud, la red de proteínas de contacto y la red de interacción de largo alcance.

Resulta importante destacar que en la bibliografía consultada, además de los Sistemas Expertos antes mencionados, se encontraron investigaciones en las que se emplean diferentes modelos de aprendizaje para la predicción de los atributos propuestos, donde se utilizan las técnicas: Random Forest, K-vecinos más Cercanos, Perceptrón Multicapa y Regresión Lineal Múltiple (Chaudhary et al. 2016; Ghafourian and Amin 2013; Hayat et al. 2016; Marrero-Ponce et al. 2015; Ruiz-Blanco et al. 2015; Sravani and Vani 2013; Suky S. and Selvakumar 2014). Dado que estos mostraron buen desempeño en la predicción de la clase estructural y velocidad de plegamiento, se consideró apropiado el empleo de los mismos en la presente investigación.

En recientes estudios realizados en Cuba (Contreras-Torres 2016), se desarrolló un software denominado *ToMoCoMD-CAMPS MuLiMs-MCoMPAs,* que automatiza el cálculo de descriptores 3D-proteicos basados en formas bilineales de la matriz de proximidad. Este software calcula descriptores totales (proteínas como un todo), locales (regiones de interés), así como descriptores totales y locales considerando cortes macromoleculares. Estos descriptores fueron evaluados en la caracterización de la estructura

macromolecular en estudios de predicción de clase estructural y velocidad de plegamiento, utilizando descriptores totales sobre la representación *Cα* (átomo de carbono alfa) mediante las técnicas Random Forest, K-vecinos más Cercanos, Perceptrón Multicapa y Regresión Lineal Múltiple, obteniendo modelos robustos y de buena capacidad predictiva. A partir de los resultados prometedores obtenidos en esta investigación, se considera importante utilizar otras representaciones, descriptores totales, locales y cortes sobre descriptores totales y locales para modelar estas propiedades.

La predicción de las clases estructurales es un tema de gran importancia en ciencia de proteínas. El conocimiento obtenido puede proveer información útil acerca de la estructura global de la proteína en estudio. Además, la práctica en sí misma, puede estimular técnicamente el desarrollo de nuevos predictores que pueden ser aplicados en otras áreas relevantes (Chou 2011). Por otra parte, la predicción de la velocidad de plegamiento es un paso importante hacia el entendimiento del proceso de plegamiento de una proteína, así como el establecimiento de relaciones secuencia-estructura-función. También es conocido que, alteraciones en el plegamiento de proteínas provocadas por factores cinéticos, puede derivar en enfermedades degenerativas como: priónicas y Alzheimer (Chen et al. 2011; Greenfield 2006). Por lo anteriormente expuesto se consideró importante predecir estas características, ya que resultan útiles en estudios bioinformáticos.

1.3 Metodologías para el desarrollo de Sistemas Expertos

Las metodologías son herramientas esenciales que dan pautas de cómo desarrollar un Sistema Experto y permiten detectar problemas para corregirlos a tiempo, evitando así el arrastre de los mismos. “Una metodología es un conjunto de métodos, prácticas, estilos, recursos y conocimientos, que permiten desarrollar de manera efectiva y eficiente cada una de las actividades que son necesarias para analizar, diseñar, producir, implantar y mantener un artefacto” (Cálad and Navarro 2001). En este caso, el concepto de artefacto se refiere a cualquier documento o software que se produzca.

El área de desarrollo de los Sistemas Expertos es relativamente reciente en comparación con la ingeniería de software convencional, no obstante, existen varias metodologías propuestas por expertos en este tipo de sistemas, estructuradas y adaptadas a sus necesidades. Al no disponer de una metodología estándar que unifique todas las ideas, cada autor propone una según el ámbito de aplicación de su interés. Existen algunas que han tenido más éxito que otras, lo cual ha llevado a su mayor difusión (Martinez et al. 2005).

De las diferentes metodologías encontradas en la bibliografía consultada se seleccionaron las de Grover (Grover 1983), Buchanan (Buchanan et al. 1983), Brulé (Brulé and Blunt 1989) y John Durkin (Durkin

1998) para realizar un análisis comparativo de sus ventajas y desventajas y determinar cuál se ajusta a las necesidades de la investigación. Este análisis se muestra en la Tabla 2.

Tabla. 2 Comparación de las metodologías estudiadas.

Metodología	Ventajas	Desventajas
Grover	Etapas bien detalladas que permiten una clara identificación del problema y de las personas que intervienen en el. Proporciona mejor documentación. Permite reemplazar en parte al experto de campo y brindar un medio de comunicación entre el usuario y el diseñador.	No poder identificar en primera instancia todos los puntos de la etapa “Definición de dominio”. Requiere de muchas entrevistas con el experto.
Buchanan	Con las críticas y sugerencias de los expertos permite una mejora y un control del funcionamiento del sistema. Propone con énfasis una documentación de los procesos.	Con la interferencia de los expertos (sugerencias de estos) puede tomar mucho tiempo en que ellos estén conformes.
Brulé	Rápida construcción del Sistema Experto (prototipo). Metodología recursiva, ya que después de realizar todas las etapas, si es que fuese necesario, se vuelven a analizar los requisitos. Adecuado para Sistemas Expertos de gran envergadura.	Metodología no apta para pequeños proyectos. Mayor tiempo requerido para realizar entrevistas.
John Durkin	Es aplicable para equipos de desarrollo pequeños. Cuenta con seis fases cada una bien detallada, de manera que no se necesita tener mucha experiencia con la misma.	No están pensadas para la construcción de grandes Sistemas Expertos. Genera poca documentación.

Una vez analizadas las ventajas y desventajas de las metodologías seleccionadas, se consideró que la más adecuada para desarrollar el Sistema Experto PropPred-ES es la de John Durkin, ya que las restantes anteriormente expuestas no se ajustan al contexto, pues unas están diseñadas para la construcción de Sistemas Expertos de gran envergadura, otras demandan un mayor tiempo para la elaboración de una correcta documentación requiriendo de equipos de desarrollo grandes o medianos, por

lo cual precisan de un personal más numeroso del que se dispone para la construcción del Sistema Experto.

En la Fig. 2. se muestra una descripción de la Metodología de Ingeniería del Conocimiento según John Durkin (Durkin 1998), en la misma se pueden apreciar las fases que componen dicha metodología y las relaciones entre ellas.

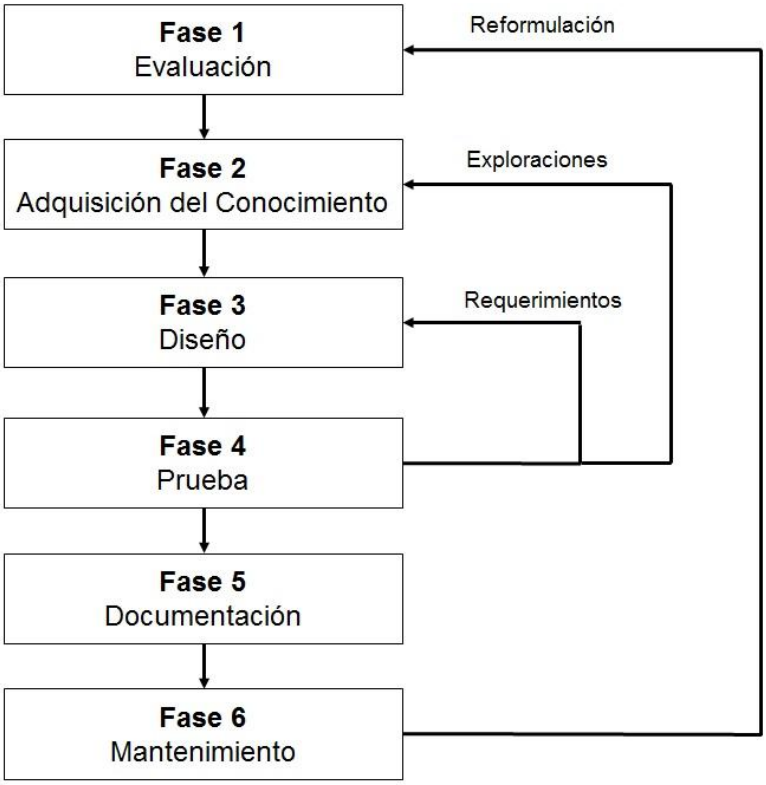


Fig. 2 Metodología de Ingeniería del conocimiento según John Durkin.

Una vez analizadas las fases y tareas de esta metodología, se encontró que no todas se adaptan a las necesidades propias de la investigación, por lo que solo se aplicarán aquellas que se ajustan a la misma. A continuación, se describen las fases y tareas que se ejecutarán para desarrollar el Sistema Experto.

FASE 1: Evaluación

1.1 Motivación para el esfuerzo: Consiste en determinar ¿Por qué está la organización motivada en crear el Sistema Experto? Algunas organizaciones están mirando resolver un problema particular, mientras que otras están interesadas en encontrar qué puede hacer la tecnología por ellos. De acuerdo a lo antes mencionado, existen dos posiciones que puede asumir una organización al incursionar en la tecnología de Sistemas Expertos: Conducida por el problema o Conducida por la solución.

1.2 Estudio de viabilidad: En esta tarea lo primordial es tratar de determinar si el proyecto tendrá éxito. Se consideran dos puntos a evaluar:

Primero: Una lista de ítems que debería reunir el proyecto es verificada. Estos ítems incluyen los recursos propios, un recurso de conocimiento y personal del proyecto. La siguiente lista de requerimientos debería ser verificada primero cuando se considera un problema para una aplicación de Sistema Experto. Disponibilidad de: conocimiento para la solución del problema, un ingeniero del conocimiento, software de desarrollo de sistema y facilidades de computador.

Segundo: Considerar asuntos que son importantes para el éxito del proyecto, pero son subjetivos de naturaleza y requieren algún juicio para determinar. Ellos incluyen: características del problema, de la gente involucrada del proyecto y asuntos de despliegue.

FASE 2: Adquisición del conocimiento

2.1 Recolección del conocimiento: Es la tarea de adquirir conocimiento del experto.

2.2 Interpretación: La interpretación de la información recolectada envuelve la identificación de piezas clave del conocimiento como: conceptos, reglas, estrategias, etc.

2.3 Análisis: Envuelve el estudio de las piezas clave del conocimiento que surgen durante la tarea de interpretación. Este esfuerzo proporciona la visión de formar las teorías en la organización del conocimiento y estrategias de solución de problemas.

FASE 3: Diseño

3.1 Seleccionar Técnica de Representación del Conocimiento: Consiste en seleccionar la técnica de representación del conocimiento que más se ajuste a las características que posee la información.

3.2 Desarrollo de la interfaz: Se implementa la interfaz mediante la cual el usuario interactúa con el sistema.

FASE 4: Pruebas

4.1 Validación del Sistema: Un Sistema Experto modela la decisión de un experto humano. Si se diseñó correctamente, el sistema deriva los mismos resultados que el experto y razona de una manera similar a este. Por ello, las pruebas deben dirigirse a validar los resultados del sistema y la base de conocimiento.

FASE 5: Documentación

5.1 Relación de temas que deben ser documentados: Como un proyecto de Sistema Experto maduro la cantidad de conocimiento recolectado del experto crece. Después de un tiempo la cantidad de información es abrumadora, para manejar esta situación se hace necesario utilizar un método para documentar la misma. Durante el esfuerzo de desarrollo se necesitará volver a menudo a esta documentación para registrar la nueva información, o estudiar previamente la información descubierta. Desde que muchos proyectos requieren un reporte final de proyecto, la información almacenada en la documentación sirve como una fuente valiosa para este esfuerzo.

Siguiendo el despliegue del Sistema Experto este necesitará ser mantenido. Para acomodar cada uno de estos esfuerzos debe documentar lo siguiente: conocimiento, gráficos de conocimiento, código fuente, pruebas, glosario de términos específicos del dominio y reportes.

5.2 Organización de la documentación: Además de contener la información listada en la sección anterior, la documentación debe ser organizada para facilitar el desarrollo del sistema. Para muchos proyectos de Sistemas Expertos se necesita escribir un reporte final cuyo contenido debe incluir: página del título, tabla de contenidos, resumen ejecutivo, visión global del proyecto, descripción del programa, resultados de las pruebas, resumen, referencias, bibliografías y apéndices.

1.4 Métodos estadísticos y de aprendizaje automático

1.4.1 Análisis de Variabilidad

El método de Análisis de Variabilidad (AV) (Godden and Bajorath 2002; Godden et al. 2000) cuantifica el contenido de información y por tanto, la variabilidad de los DMs. Este método no supervisado está basado en el cálculo de la Entropía de Shannon (Shannon 2001) bajo el principio de que, DMs apropiados para estudios quimio-métricos pudieran poseer altos valores de entropía como un indicador de su tendencia a cambiar gradualmente con la modificación de la estructura molecular; mientras que, DMs (casi constantes o constantes) pudieran tener valores bajos, siendo cero el límite para aquellos DMs que contienen el mismo valor para estructuras diferentes.

1.4.2 Análisis de Componentes Principales

El Análisis de Componentes Principales (ACP), es una técnica estadística de síntesis de la información o reducción de la dimensión (número de variables). Es decir, ante un banco de datos con muchas variables, el objetivo será reducirlas a un menor número, perdiendo la menor cantidad de información posible (Baró and Alemany 2000; Peña Sánchez de Rivera 1989). Este procedimiento es útil cuando al trabajar con

varias variables (probablemente con un gran número de variables) se cree que existe cierto grado de redundancia en las mismas, es decir, un ACP tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que existe información redundante y, por tanto, pocos factores explicarán gran parte de la variabilidad total.

1.4.3 Random Forest

Random Forest (RF) es un algoritmo para clasificación de amplio uso que tiene un rendimiento especialmente bueno para datos de alta dimensionalidad. Es una combinación de árboles predictores {h(**x**,Θ_k), *k=1*, ...}, en la que cada árbol depende de los valores de un vector aleatorio (Θ_k) probado independientemente y con la misma distribución para cada uno de estos. Cada árbol realiza el voto unitario a favor de la clase más frecuente de la entrada **x**. Para realizar la predicción de un nuevo caso este realiza un recorrido hacia los nodos terminales de cada árbol. Luego se le asigna la etiqueta (clase) correspondiente al nodo terminal. Este proceso se repite en cada uno de los árboles del ensamblado y la clase que obtenga el voto mayoritario es reportada como la predicción (Liaw and Wiener 2014).

1.4.4 K-vecinos más Cercanos

K-vecinos más Cercanos (K-NN) es una técnica de aprendizaje automático basada en instancias (casos). En este grupo de técnicas son almacenadas las instancias del conjunto de entrenamiento y se emplea una función de distancia para determinar qué caso(s) se encuentra(n) más cercano(s) al caso que se pretende clasificar. Es importante señalar que el parámetro *k* determina el número de vecinos. Frecuentemente, más de un vecino es usado (k>1) para realizar la clasificación, en estas situaciones la clase mayoritaria de los K-vecinos más Cercanos (o la distancia ponderada promedio, si la variable respuesta es numérica) es asignada al nuevo caso (Witten et al. 2016).

1.4.5 Perceptrón Multicapa

El Perceptrón Multicapa (MLP) es una red neuronal artificial formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple. El MLP puede ser totalmente o localmente conectado; en el primer caso, cada salida de una neurona de la capa *i* es entrada de todas las neuronas de la capa *i+1*, mientras que en el segundo, cada neurona de la capa *i* es entrada de una serie de neuronas (región) de la capa *i+1* (Haykin and Lippmann 1998).

1.4.6 Regresión Lineal Múltiple

La Regresión Lineal Múltiple (RLM) es un enfoque estadístico tradicional para derivar modelos QSAR. Esta técnica es utilizada para estudiar las relaciones entre una única variable dependiente u objetivo y un conjunto de variables independientes (predictores). La anotación matemática del modelo o ecuación de Regresión Lineal Múltiple puede ser expresada como:

$$y_i = b_o + \sum_{j=1}^{n} b_j x_{ij} + e_i$$

donde (y_i) es la variable dependiente o explicada, (b_o) es la intersección o término constante, (b_j) es el coeficiente de regresión que va a ser estimado, (x_{ij}) son las variables independientes usadas en el modelo de regresión y (e_i) es el error aleatorio también llamado error del modelo o residual (Gramatica et al. 2013).

1.5 Herramientas y Tecnologías

1.5.1 Lenguaje de modelado: UML 2.0

El Lenguaje Unificado de Modelado UML (del inglés *Unified Modeling Language*) es un lenguaje de modelado visual que se usa para visualizar, especificar, construir y documentar artefactos de un sistema de software. Además, es usado para la captura de decisiones y conocimiento sobre los sistemas que se deben construir y se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas (Mouheb et al. 2015). Este lenguaje da apoyo a la mayoría de los procesos orientados a objetos y está pensado para ser usado en herramientas interactivas de modelado visual que tengan generadores de informes y de código, por ejemplo, Visual Paradigm UML, que es utilizado en la fase de análisis y diseño del Sistema Experto para realizar modelos conceptuales, diagramas de paquetes, diagramas de clases y diseño de interfaces.

1.5.2 Herramienta CASE: Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE (del inglés *Computer Aided Software Engineering*) que utiliza UML como lenguaje de modelado bajo el paradigma Programación Orientada a Objetos, para la ayuda en el proceso de desarrollo de software. Brinda confiabilidad y estabilidad en el desarrollo orientado a objetos a ingenieros de software, analistas y arquitectos que están interesados en la construcción de sistemas a gran escala. La versión 8.0 soporta el estándar UML en su versión 2.0 y es compatible con equipos de desarrollo de software en la captura de requisitos, análisis de casos de uso, ingeniería de código, modelado de clase y el modelado de datos (Paradigm 2016).

1.5.3 Lenguaje de programación: Java 1.8

Java es un lenguaje de programación de propósito general, concurrente, basado en clases y orientado a objetos. Este se define como un lenguaje interpretado, ya que las aplicaciones desarrolladas son compiladas a código byte e interpretadas por la máquina virtual de Java, la cual permite que las soluciones sean ejecutadas en diferentes entornos de hardware y software (Gosling et al. 2014). Se escoge debido a que al estar WEKA implementado en Java se puede hacer uso de sus funcionalidades.

1.5.4 Entorno de desarrollo: NetBeans 8.1

NetBeans es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el NetBeans. Es un producto libre y gratuito sin restricciones de uso. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están: administración de las interfaces de usuario (ej. menús y barras de herramientas), administración de las configuraciones del usuario, administración del almacenamiento (guardando y cargando cualquier tipo de dato) y administración de ventanas (NetBeans 2016).

1.5.5 Herramienta para el cálculo de los descriptores 3D-proteicos: MuLiMs-MCoMPAS 1.0

MuLiMs-MCoMPAS es un software que facilita al usuario la configuración y el cálculo de descriptores 3D-proteicos basados en las formas algebraicas bilineales. Estos índices se calculan estableciendo relaciones entre pares de átomos aplicando varias métricas de proximidad, transformaciones matriciales, cortes, cálculo de locales y operadores de agregación (Contreras-Torres 2016).

1.5.6 Herramienta para el Análisis de Variabilidad de los DMs-3D: IMMAN 1.0

IMMAN es un programa computacional libre para el análisis quimio-métrico. Provee valiosas herramientas basadas en teoría de la información para realizar tareas de selección supervisada y no supervisada de características. Ofrece funcionalidades de pre-procesamiento de datos como: tratamiento de valores faltantes, particionado de datos y búsqueda. Además, provee opciones de ranking para parámetros simples o conjuntos multicriterios. Consecuentemente, este software es adecuado para tareas como: reducción de la dimensionalidad, rankeo de atributos y análisis comparativo de diversidad de matrices de datos (Urias et al. 2015).

1.5.7 Herramienta para el análisis de ortogonalidad de los DMs-3D: STATISTICA 8.0

STATISTICA es un producto de StatSoft Software, líder mundial en software para el análisis de datos y uno de los productores más grandes de software estadístico y analítico en el mundo. El paquete

STATISTICA implementó procedimientos para el análisis, administración, minería y visualización de datos (StatSoft 2008).

1.5.8 Herramienta para el desarrollo e integración de los modelos de clasificación: WEKA 3.7.10

WEKA, del inglés *Waikato Environment for Knowledge Analysis*, es una plataforma de software para el aprendizaje automático y la minería de datos escrita en Java y desarrollada en la Universidad de Waikato. Se encuentra distribuido bajo la licencia GNU-GPL. El paquete WEKA contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades (Witten et al. 2009).

1.5.9 Herramienta para el desarrollo de los modelos de regresión: MobyDigs 1.0

MobyDigs es un software para el cálculo de modelos de regresión usando algoritmos genéticos en la selección de variables, obteniendo un subconjunto óptimo de modelos predictivos desarrollado por Milano Chemometrics y el Grupo Investigativo QSAR (Todeschini et al. 2005).

Conclusiones parciales

Los estudios realizados sobre Inteligencia Artificial y Sistemas Expertos, así como la aplicación de estos en el campo de la bioinformática, proporcionaron un referente teórico para la construcción de PropPred-ES. Luego de realizar una comparación entre diferentes metodologías, se determinó que la de John Durkin es la que más se adapta a las características del problema. Después de analizar varias investigaciones concernientes al campo de estudio de la ciencia de las proteínas para la predicción de propiedades biológicas, se escogieron los métodos estadísticos y de aprendizaje automático que se emplearán en el desarrollo de los modelos de clasificación y regresión. Además, para el desarrollo del Sistema Experto se seleccionaron las siguientes herramientas y tecnologías: NetBeans 8.1 como entorno de desarrollo, lenguaje de programación Java 1.8, como herramienta CASE Visual Paradigm 8.0. Por otra parte, para el cálculo de los descriptores 3D-proteicos MuLiMs-MCoMPAS 1.0, para el Análisis de Variabilidad de los DMs-3D IMMAN 1.0, para el análisis de ortogonalidad de los índices 3D STATISTICA 8.0. Finalmente, se seleccionan WEKA 3.7.10 y MobyDigs 1.0 para el desarrollo de los modelos de clasificación y regresión respectivamente.

Capítulo 2 Desarrollo del Sistema Experto para la Predicción de Propiedades Biológicas en Proteínas

En el presente capítulo se lleva a cabo un estudio de viabilidad para determinar si el proyecto puede ser desarrollado, además, se realizan estudios exploratorios para la adquisición del conocimiento. Para guiar el proceso de análisis y diseño se parte de los principios generales del desarrollo de software planteados por Pressman y Sommerville para obtener los requisitos y generar artefactos como: modelo conceptual, diagrama de casos de uso, vista lógica del sistema para representar la arquitectura del mismo y modelo del diseño, haciendo uso de los patrones arquitectónicos y de diseño. Se selecciona la Técnica de Representación del Conocimiento basada en modelos y se desarrollan los modelos de clasificación y de regresión. Se implementa el Sistema Experto describiendo los diagramas de componentes, los principales elementos del estándar de codificación que se emplean y se desarrolla la interfaz.

2.1 Evaluación

2.1.1 Motivación para el esfuerzo

Actualmente nuevas enfermedades azotan la humanidad, por lo que resulta de vital importancia la búsqueda de tratamientos o nuevos fármacos para contrarrestarlas (Medina-Franco et al. 2015). El diseño y desarrollo de un medicamento es un proceso complejo, lento y puede incurrir en gastos millonarios. En los últimos años, la industria farmacéutica ha encauzado sus investigaciones hacia aquellos métodos que permitan describir de manera eficiente la estructura química de las moléculas candidatas a medicamentos, lo cual facilita que se puedan acortar los plazos de ejecución y los costos de la investigación (Marrero-Ponce et al. 2013). Resultados obtenidos en investigaciones precedentes, donde se combinan técnicas de aprendizaje automático y métodos para la caracterización numérica de proteínas, han demostrado su efectividad en la predicción de propiedades biológicas de interés. La motivación para el esfuerzo en el desarrollo del Sistema Experto es conducida por el problema, el cual es, contribuir a la identificación de propiedades biológicas en proteínas mediante el uso de descriptores 3D-proteicos, obtenidos en recientes estudios a partir de procedimientos basados en formas algebraicas bilineales (Marrero-Ponce et al. 2015).

2.1.2 Estudio de viabilidad

Una vez identificada la motivación para el esfuerzo es preciso determinar la viabilidad del desarrollo del proyecto para la construcción del Sistema Experto, para ello se siguieron los siguientes pasos:

El primer paso consiste en la verificación de la lista de requerimientos que debe poseer el proyecto, los cuales son: recursos propios, de conocimiento y personal del proyecto. Para el desarrollo del Sistema

Experto se cuenta con disponibilidad de datos para la solución del problema [RCSB *Protein Data Bank* (http://www.rcsb.org)], un ingeniero del conocimiento cuya función es implementar el SE, un software para el desarrollo del sistema (NetBeans 8.1) y facilidades de computador.

El segundo paso consiste en considerar los asuntos que son importantes para el éxito del Proyecto. Para ello se utilizará la técnica de estimación propuesta por John Durkin, formando una lista de temas importantes a considerar: viabilidad del problema, del personal y de despliegue.

A cada tema se le asignará un peso (entre 0 y 10), resultado de la experiencia de consulta de Durkin sobre los esfuerzos de determinación de proyectos anteriores que refleja la importancia de cada uno durante la evaluación del proyecto. Luego se le atribuirá a cada tema un valor (entre 0 y 10), que refleja el grado de creencia en el tema. Estos valores se calculan obteniendo puntajes que según Durkin darán una estimación de la viabilidad del proyecto, donde valores cercanos a 10 indican que es viable (Ver Anexo 1).

Una vez aplicada la técnica se obtuvo una viabilidad de 8.62, demostrando que el Sistema Experto puede ser desarrollado.

2.2 Adquisición del conocimiento

La adquisición del conocimiento es el desafío más difícil en el desarrollo de un Sistema Experto (Durkin 1998). Esta fase consta de tres tareas: recolección, interpretación y análisis del conocimiento. Dadas las características de la presente investigación no se consideró necesario valorar de forma independiente cada una de estas tres fases, por lo que a continuación se describe cómo se adquirió el conocimiento.

Por las características del sistema a desarrollar, la recolección del conocimiento no requiere de un experto humano como ocurre tradicionalmente, el mismo es obtenido mediante la utilización de software, es decir, de forma automática. Para ello se realizó un estudio no supervisado a partir de una base compuesta por 152 proteínas, para ver el comportamiento de descriptores respecto a (métricas para el cálculo de distancias inter-atómicas, cortes moleculares y locales), la cual se empleó previamente en la literatura (Estrada 2004). El software MuLiMs-MCoMPAs genera millones de descriptores, por lo que resulta necesario reducir este espacio de alta dimensionalidad de rasgos moleculares, para ello se utilizarán los métodos AV y ACP.

2.2.1 Estudios exploratorios de Análisis de Variabilidad basado en Entropía de Shannon

En este epígrafe se realiza un análisis de la variabilidad de los DMs-3D basado en Entropía Promedio de Shannon Estandarizada (EPSE), acorde al empleo de diferentes métricas para el cálculo de distancias inter-atómicas, el uso de cortes moleculares y locales.

Análisis comparativo de los DMs-3D acorde a las métricas para el cálculo de distancias inter-atómicas

El objetivo de este estudio es evaluar la variabilidad de los DMs-3D MuLiMs-MCoMPAs según las métricas para el cálculo de distancias inter-atómicas. Como puede observarse en la Fig. 3, los mayores grados de variabilidad se encuentran en un rango (superior a 0.65 de EPSE), son obtenidos por DMs que usan las métricas: Squared Euclidean (M19), [Minkowski p=3] (M7), [Minkowski p=2.5] (M6), [Minkowski p=3.5] (M9), Average Euclidean (M18), [Minkowski p=2] (M5), [Chebyshev/Lagrange] (M8), [Minkowski p=1.5] (M4), [Minkowski p=1] (M3), SL-like (M17) y [Minkowski p=0.5] (M2). Luego aparece un grupo de DMs con grados de variabilidad (entre 0.2 y 0.35 de EPSE): [Lance-Williams/Bray-Curtis] (M11), [Clark/Coefficient of Divergence] (M12), Soergel (M13), Canberra (M10), [Minkowski p=0.25] (M1) y Bhattacharyya (M14). Los DMs de menor variabilidad se encuentran en un rango (menor a 0.2 de EPSE): Chord (M22), Wave-Edges (M15), Cosine-Ochiai (M21), Angular Separation/[1-Cosine (Ochiai)] (M16), Dice (M29), Tanimoto (M24), Sokal-Sneath (M26), Kulcynzski (M25), Pearson (M20), Simpson (M27), Identity corrected (M30), Ruzicka (M28), Additivity (M31), Proportionality Corrected (M32) y Fossum (M23). Este análisis sugiere que los DMs con buen comportamiento de variabilidad son los que utilizan las métricas: Squared Euclidean (M19), [Minkowski p=3] (M7), [Minkowski p=2.5] (M6), [Minkowski p=3.5] (M9) y Average Euclidean (M18).

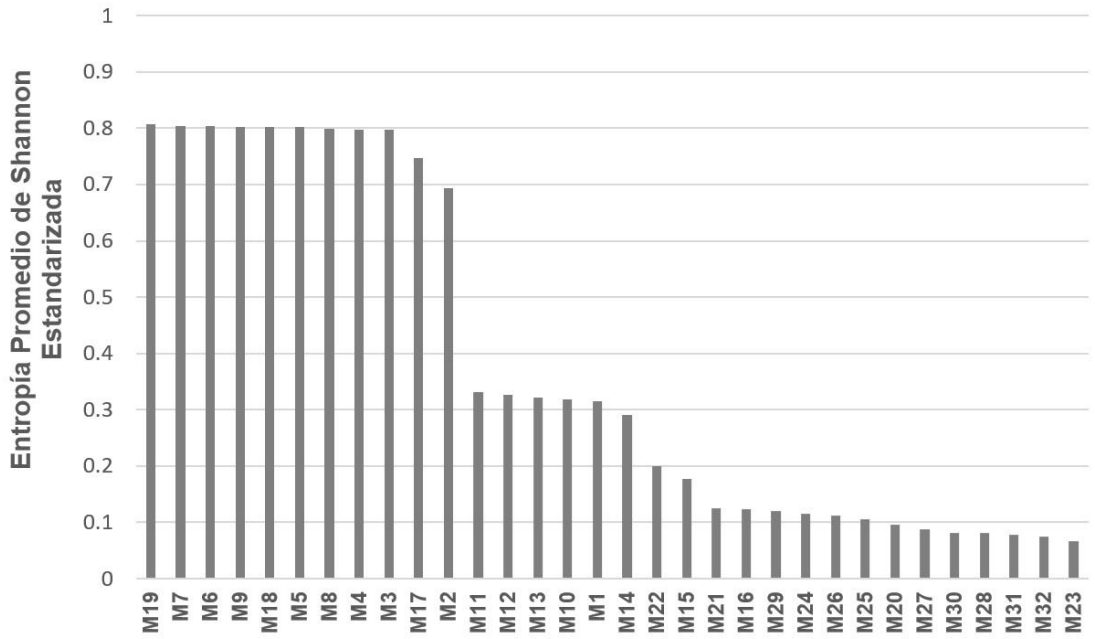


Fig. 3 Promedio de los DMs-3D acorde a las métricas para el cálculo de distancias inter-atómicas.

Análisis comparativo de los DMs-3D acorde a los cortes moleculares

El objetivo de este estudio es evaluar la variabilidad de los DMs-3D MuLiMs-MCoMPAs según los cortes topológicos (Lag P), los cortes geométricos (Lag L) y la combinación de ambos (Lag P- Lag L). Como puede observarse en la Fig. 4, los mayores grados de variabilidad se encuentran en un rango (superior a 0.45 de EPSE), son obtenidos por DMs que usan los cortes: Lag P [1-3]-Lag L [8.1-11], Lag P [2-3]-Lag L [8.1-11], Lag P [4-7]-Lag L [8.1-11], Lag P [+12], Lag P [+12]-Lag L [6-8], Lag L [6-8] y Lag P [1-3]. Luego aparece un grupo de DMs con grados de variabilidad (entre 0.40 y 0.45 de EPSE): Lag P [8-12]-Lag L [8.1-11], Lag P [+12]-Lag L [4-5.9], Lag L [8.1-11], Lag P [8-12]-Lag L [6-8], Lag P [+12]-Lag L [8.1-11], Lag L [4-5.9] y Lag P [4-7]-Lag L [6-8]. Los DMs de menor variabilidad se encuentran en un rango (menor a 0.40 de EPSE): KA, Lag P [8-12], Lag P [1-3]-Lag L [6-8], Lag P [2-3]-Lag L [6-8], Lag P [2-3], Lag P [4-7], Lag P [1-3]-Lag L [4-5.9], Lag P [2-3]-Lag L [4-5.9], Lag P [4-7]-Lag L [4-5.9] y Lag P [8-12]-Lag L [4-5.9]. En este análisis se obtienen DMs de variabilidad muy similar, no obstante, los de mejor comportamiento son: Lag P [1-3]-Lag L [8.1-11], Lag P [2-3]-Lag L [8.1-11], Lag P [4-7]-Lag L [8.1-11], Lag P [+12] y Lag P [+12]-Lag L [6-8].

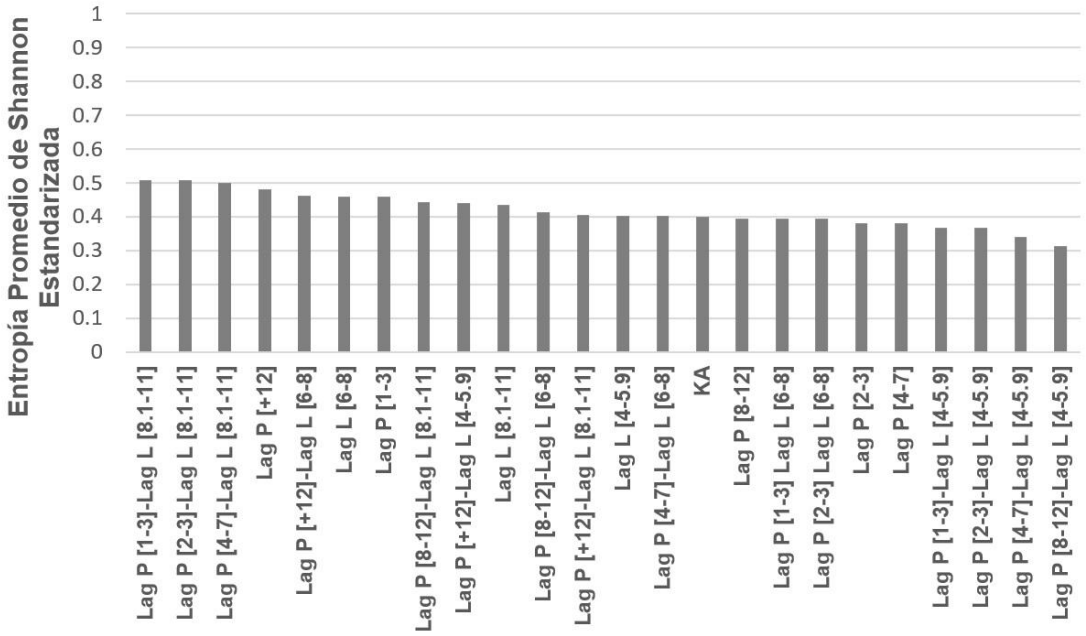


Fig. 4 Promedio de los DMs-3D acorde a los cortes moleculares.

Análisis comparativo de los DMs-3D acorde a los locales

El objetivo de este estudio es evaluar la variabilidad de los DMs-3D MuLiMs-MCoMPAs según los locales utilizados. Como puede observarse en la Fig. 5, los mayores grados de variabilidad se encuentran en un rango (superior a 0.45 de EPSE), estos son obtenidos por DMs que usan los locales: Glutamina (GLN),

Apolar (RAP), Tirosina (TYR), Cistina (CYS), Polar cargado positivamente (RPC), No comunes en hélice-alfa y hoja-beta (UFG), Arginina (ARG) y Serina (SER). Luego aparece un grupo de DMs (entre 0.40 y 0.45 de EPSE): Favorecedores de hoja-beta (FBS), Alifáticos (ALG), Polares no cargados (RPU), Glicina (GLY), Favorecedores de hélice-alfa (FAH), Glutamato (GLU), Prolina (PRO), Lisina (LYS) y Triptófano (TRP). Los DMs de menor variabilidad se encuentran en un rango (menor a 0.40 de EPSE): Isoleucina (ILE), Histidina (HIS), Arómatico (ARO), Alanina (ALA), Treonina (THR), Leucina (LEU), Valina (VAL), Fenilalanina (PHE), Metionina (MET), Asparagina (ASN), Total, Favorecedores de giros beta (AFT), Polar cargado negativamente (RNC) y Aspartato (ASP). Este análisis indica que se obtienen DMs de una variabilidad muy similar, sobresaliendo con un mejor comportamiento los locales: Glutamina (GLN), Apolar (RAP), Tirosina (TYR), Cistina (CYS) y Polar cargado positivamente (RPC).

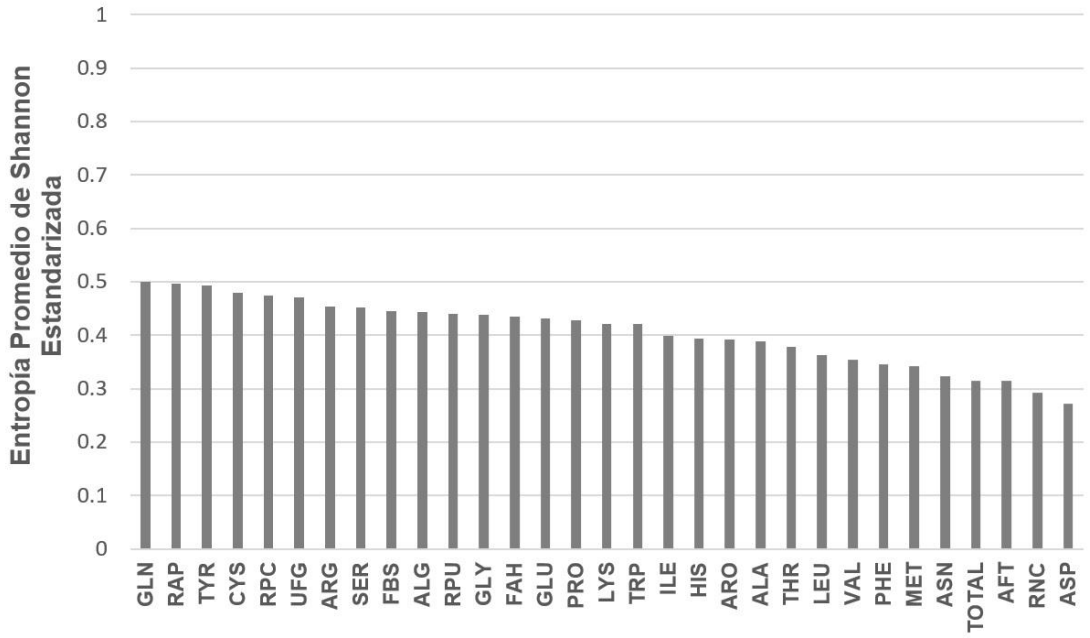


Fig. 5 Promedio de los DMs-3D acorde a los locales.

2.2.2 Estudios exploratorios de Análisis de Componentes Principales

En el presente acápite se realiza un estudio para evaluar la posible ortogonalidad de los DMs-3D mediante el método de Análisis de Componentes Principales. Para realizar el estudio se empleó el software STATISTICA 8.0 y la misma base de proteínas empleada en los Análisis de Variabilidad.

Independencia lineal de los DMs-3D acorde a las métricas para el cálculo de distancias inter-atómicas

Para realizar el estudio se calcularon 1218 DMs usando las diferentes métricas de distancia. Se determinaron 10 componentes, los cuales explican aproximadamente el 93.55% de la varianza acumulada. Al examinar estos es posible concluir que: Se encuentran correlacionados los DMs que usan las métricas [Minkowski (p=0.25)] (M1) y [Minkowski (p=0.5)] (M2) en el factor 5, Pearson (M20), Identity corrected (M30) y Additivity (M31) en el factor 6, Angular Separation/ [1-Cosine (Ochiai)] (M16) y Chord (M22) en el factor 7. Presentan carga exclusiva en los factores 10 y 8 respectivamente los DMs que utilizan las métricas Pearson (M20) y Proportionality corrected (M32), lo cual revela que codifican información ortogonal (complementaria).

Independencia lineal de los DMs-3D acorde a los cortes topológicos (Lag P)

Para realizar el estudio fueron calculados 390 DMs, obteniéndose 15 componentes que explican aproximadamente el 57.78% de la varianza acumulada. El análisis de los mismos reveló que: Los DMs que emplean los cortes Lag P [1-3] y Lag P [2-3], Lag P [1-3] y Lag P [8-12] están correlacionados en los factores 3 y 9 respectivamente. Los DMs que utilizan los cortes Lag P [1-3], Lag P [2-3], Lag P [4-7], Lag P [+12] y Lag P [8-12] presentan carga exclusiva en los factores 12, 14, 7, (10 y 15) y (5 y 13) respectivamente, todo esto indica que los mismos codifican información ortogonal.

Independencia lineal de los DMs-3D acorde a los cortes geométricos (Lag L)

Para realizar el estudio fueron calculados 260 DMs, se obtuvieron 15 componentes los cuales explican aproximadamente el 57.21% de la varianza acumulada. Al analizar estos se puede señalar que: Los DMs que usan los cortes geométricos Lag L [4-5.9], Lag L [6-8] y Lag L [8.1-11] se encuentran exclusivamente cargados en los factores (3,9,15), (6,10,13) y (4,8,11,14) respectivamente por lo que codifican información diferente.

Independencia lineal de los DMs-3D acorde a los cortes combinados (Lag P-lag L)

Para realizar el estudio fueron calculados 1040 DMs, se obtuvieron 15 componentes los cuales explican aproximadamente el 45.65% de la varianza acumulada. Un análisis realizado a estos muestra que: Existe correlación en los DMs que usan los cortes Lag P [1-3]-Lag L [6-8], Lag P [4-7]-Lag L [6-8] y Lag P [2-3]-Lag L [6-8] en el factor 4, los Lag P [+12]-Lag L [6-8], Lag P [1-3]-Lag L [8.1-11] y Lag P [2-3]-Lag L [8.1-11] en el factor 5, los Lag P [1-3]-Lag L [4-5.9], Lag P [8-12]-Lag L [8.1-11] y Lag P [2-3]-Lag L [4-5.9] en el factor 6, los Lag P [8-12]-Lag L [4-5.9] y Lag P [8-12]-Lag L [8.1-11] en el factor 9, los Lag P [1-3]-Lag L [8.1-11] y Lag P [2-3]-Lag L [8.1-11] en el factor 12. Se encuentran exclusivamente cargados los DMs que

utilizan los cortes Lag P [8-12]-Lag L [6-8] y Lag P [4-7]-Lag L [6-8] en los factores 11 y (13,15) respectivamente, lo cual sugiere que son linealmente independientes.

Independencia lineal de los DMs-3D acorde a las agrupaciones locales

Para realizar el estudio fueron calculados 572 DMs, de los cuales se obtuvieron 15 componentes que explican aproximadamente el 67.04% de la varianza acumulada. Una vez analizados estos, se pudo concluir que: Presentan correlación los DMs que usan las agrupaciones locales Alifáticos (ALG), Favorecedores de hélice-alfa (FAH) y Apolar (RAP) en el factor 4, Alifáticos (ALG), Apolar (RAP) y Polar cargado positivamente (RPC) en el factor 5, Favorecedores de giros beta (AFT) y Polares no cargados (RPU) en el factor 6, Favorecedores de hélice-alfa (FAH) y Favorecedores de hoja-beta (FBS) en el factor 10. Aparecen con carga exclusiva los DMs que utilizan la agrupación local Apolar (RAP) y No comunes en hélice-alfa y hoja-beta (UFG) en los factores 13 y 8 respectivamente, esto indica que los mismos codifican información complementaria.

Independencia lineal de los DMs-3D acorde a las agrupaciones locales de aminoácidos

Para realizar el estudio fueron calculados 1092 DMs, obteniéndose 15 componentes que explican aproximadamente el 49.34% de la varianza acumulada. El análisis de los mismos evidencia que: Los DMs que están correlacionados son los que emplean las agrupaciones locales de aminoácidos Lisina (LYS) y Metionina (MET) en el factor 5, Glutamina (GLN), Glutamato (GLU) y Isoleucina (ILE) en el factor 6, Glutamina (GLN), Histidina (HIS) y Triptófano (TRP) en el factor 7, Asparagina (ASN) y Tirosina (TYR) en el factor 8, Glutamato (GLU), Glicina (GLY) y Fenilalanina (PHE) en el factor 9, Isoleucina (ILE), Lisina (LYS) y Triptófano (TRP) en el factor 13. Los DMs que usan la agrupación local de aminoácidos Alanina (ALA) y Prolina (PRO) muestran carga exclusiva en los factores 14 y 12 respectivamente, por lo que es posible concluir que son linealmente independientes.

2.3 Modelo conceptual

Con el fin de representar los conceptos claves del domino y mostrar las relaciones existentes entre ellos, se muestra en la Fig. 6 un modelo conceptual que proporciona una visión estructural que servirá como base de conocimiento del problema.

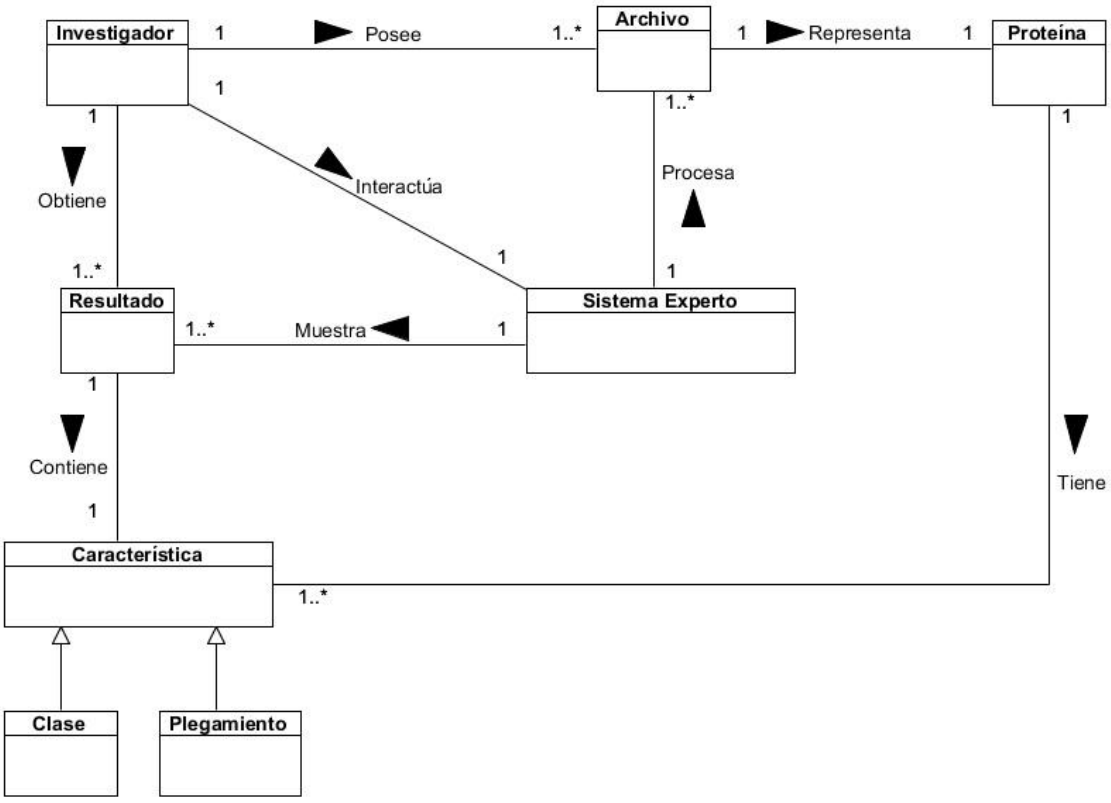


Fig. 6 Modelo conceptual.

Investigador: Es el especialista en el área de la bioinformática o la biología molecular, cuyo interés radica en la predicción de propiedades biológicas de las proteínas.

Archivo: Es un archivo en formato PDB o ENT que contiene detalles experimentales de la determinación de la estructura, características estructurales, químicas y bioquímicas de una proteína específica, así como, las coordenadas atómicas de la estructura.

Proteína: Compuesto orgánico que forma parte de las sustancias componentes de la materia fundamental de las células y de las sustancias vegetales y animales. Son biomoléculas formadas por cadenas lineales de aminoácidos.

Característica: Son características de la proteína (clases estructurales y velocidad de plegamiento).

- ✓ Clase: Clases estructurales de proteínas (Todo Hélice alfa, Todo Hoja plegada beta, Alfa/beta, Alfa + Beta).
- ✓ Plegamiento: Velocidad de plegamiento de cadenas polipeptídicas (Es la velocidad en que una proteína soluble alcanza su estructura tridimensional).

Sistema Experto: Es el software que permite al usuario realizar una predicción de las características de las proteínas de su interés.

Resultado: Es el resultado de la predicción de la clase estructural o la velocidad de plegamiento de la(s) proteína(s) de interés para el usuario.

2.4 Requisitos del sistema

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer, el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito. Los requerimientos del sistema de software se clasifican como funcionales y no funcionales. Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar a entradas particulares y de cómo debería comportarse en situaciones específicas. Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema (Sommerville 2011). En la presente investigación se identificaron los siguientes requisitos.

Requisitos funcionales

RF 1: Cargar archivos en formato PDB o ENT.

RF 2: Realizar predicción de la clase estructural de las proteínas.

RF 3: Realizar predicción de la velocidad de plegamiento de las proteínas.

RF 4: Visualizar resultados de la predicción.

RF 5: Guardar resultados de la predicción.

Requisitos no funcionales

Usabilidad.

RNF 1: En la interfaz principal debe haber un menú principal con todas las opciones que brinda el sistema.

RNF 2. La aplicación puede ser usada por especialistas en el área de la bioinformática o la biología molecular con conocimientos básicos en el manejo de las computadoras.

Software.

RNF 3: La computadora donde se despliegue la aplicación debe tener instalado el sistema operativo Windows y/o Linux y la Máquina Virtual de Java (1.7 como mínimo).

Hardware.

RF 4: Las estaciones de trabajo donde se despliegue la aplicación deben poseer como mínimo, 1 GB de memoria RAM, microprocesador Intel(R) Dual-Core.

2.5 Diagrama de Casos de Uso del sistema

Los Casos de Uso (CU) son una técnica de descubrimiento de requerimientos. En su forma más sencilla, un caso de uso identifica a los actores implicados en una interacción y nombra el tipo de interacción, esto se complementa con información adicional que describe la interacción con el sistema. Los casos de uso se documentan con el empleo de un diagrama de casos de uso. El conjunto de casos de uso representa todas las interacciones posibles que se describirán en los requerimientos del sistema (Sommerville 2011). A continuación, se definen los siguientes casos de uso a partir de los requisitos funcionales como se muestra en la Fig. 7, proporcionando una vista común a desarrolladores y clientes de las funcionalidades que proveerá el sistema.

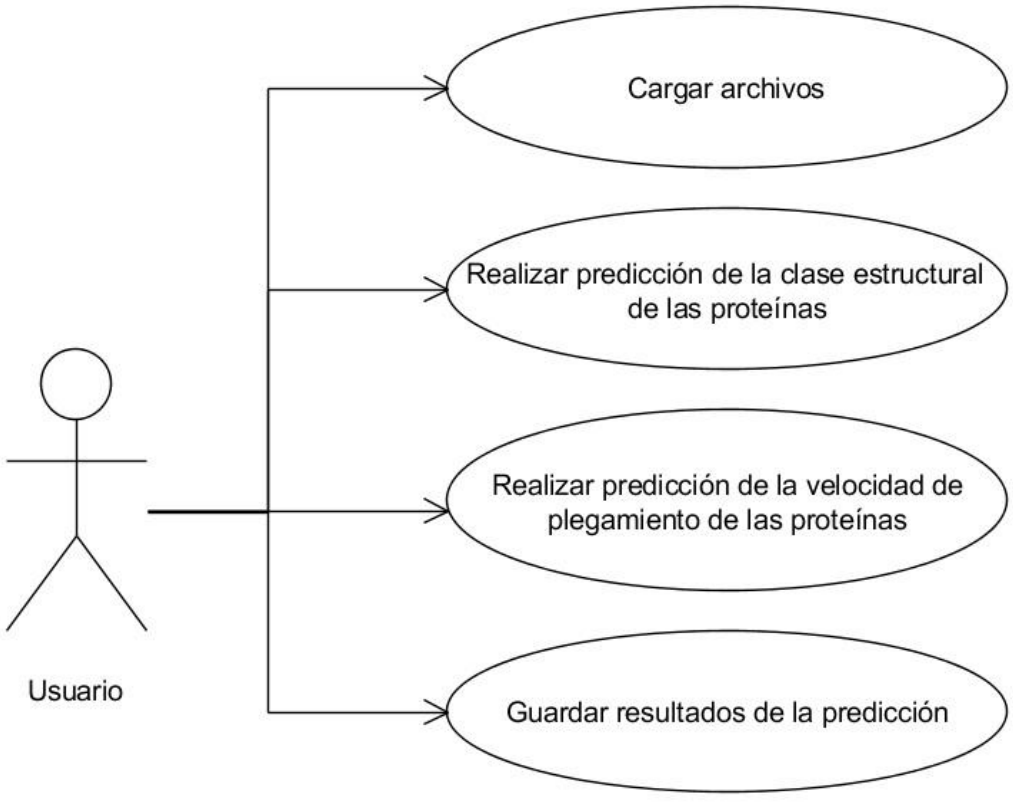


Fig. 7 Diagrama de Casos de Uso del sistema.

En el diagrama anterior el usuario inicia los Casos de Uso: Cargar archivos, Realizar predicción de la clase estructural de las proteínas, Realizar predicción de la velocidad de plegamiento de las proteínas y

Guardar resultados de la predicción. El primero representa la acción del usuario de cargar al sistema uno o varios archivos PDB o ENT. El segundo inicia cuando el usuario solicita al sistema realizar la predicción de la clase estructural de las proteínas. El tercero comienza cuando el usuario solicita al sistema realizar la predicción de la velocidad de plegamiento de las proteínas. El cuarto inicia cuando el usuario desea guardar los resultados de la predicción. En la Tabla 3 se realiza la especificación formal del CU "Realizar predicción de la clase estructural de las proteínas".

Tabla. 3 Especificación formal del CU "Realizar predicción de la clase estructural de las proteínas".

Actores	Usuario
Resumen	El caso de uso se inicia cuando el actor selecciona la característica "Clase Estructural" y la opción "Predecir", la cual consiste en determinar la clase estructural de la(s) proteína(s) cargadas en formato PDB o ENT. El caso de uso finaliza cuando el sistema muestra el resultado de la predicción de la clase estructural.
Complejidad	Alta
Prioridad	Crítico
Flujo de eventos	
Flujo básico "Realizar predicción de la clase estructural de las proteínas"	
Actor	**Sistema**
1. El caso de uso inicia cuando el actor selecciona la característica "Clase estructural" y la opción "Predecir".	2. El sistema verifica que al menos un archivo este cargado y se encuentre seleccionada la característica "Clase estructural".
	3. El sistema empieza la predicción mostrando una barra de progreso de la misma y le da la opción al actor de "cancelar" la predicción.
	4. El sistema calcula los descriptores a partir del o de los archivos filtrados y los evalúa en el modelo de clasificación para predecir la clase estructural de la proteína.
	5. El sistema muestra el resultado de la predicción, terminado así el caso de uso.

Prototipo de Interfaz

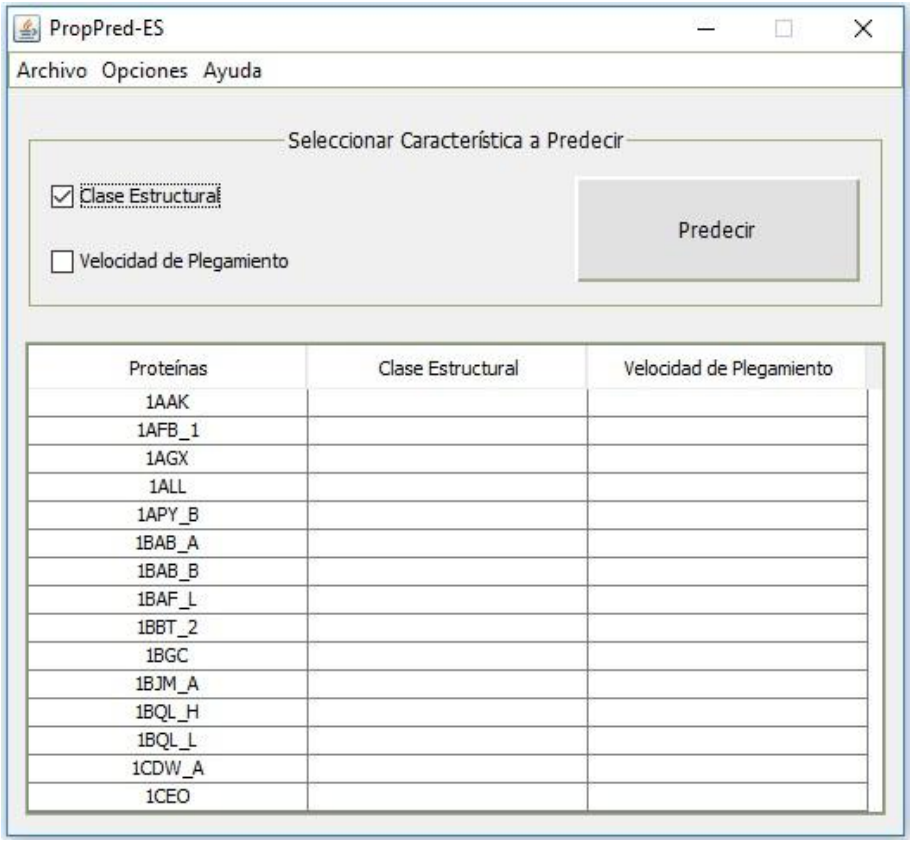


Fig. 8 Prototipo de interfaz para realizar predicción de la clase estructural.

Flujos alternos "Cargar archivo"

Nº Evento 2

Actor	Sistema
	2 a. El sistema muestra un mensaje de aviso informando que debe cargar al menos un archivo terminando así el caso de uso.

Prototipo de Interfaz

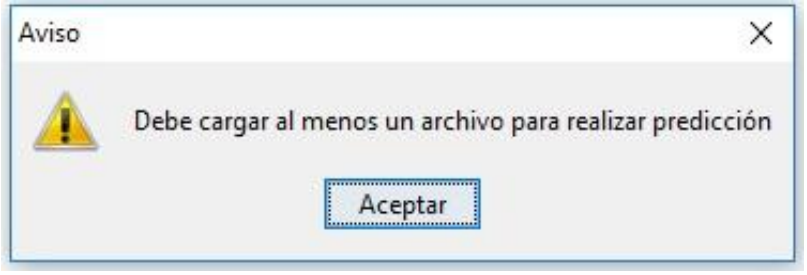


Fig. 9 Prototipo de interfaz de aviso cuando no se ha cargado un archivo.

Flujos alternos "Seleccione característica a predecir"	
Nº Evento 2	
Actor	**Sistema**
	2 b. El sistema muestra un mensaje de aviso informando que debe seleccionar una característica a predecir, terminando así el caso de uso.

Prototipo de Interfaz

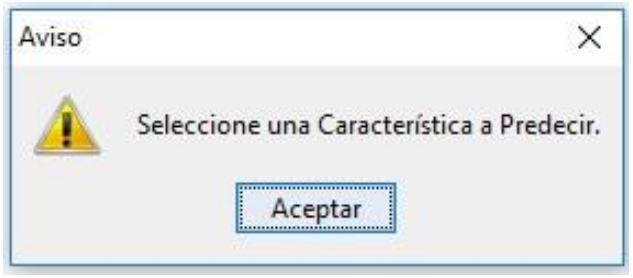


Fig. 10 Prototipo de interfaz de aviso cuando no se ha seleccionado la característica a predecir.

Flujos alternos "Cancelar predicción"	
Nº Evento 3	
Actor	**Sistema**
3. a. El Actor selecciona la opción "Cancelar".	3 b. El sistema cancela la predicción, terminando así el caso de uso.

Prototipo de Interfaz

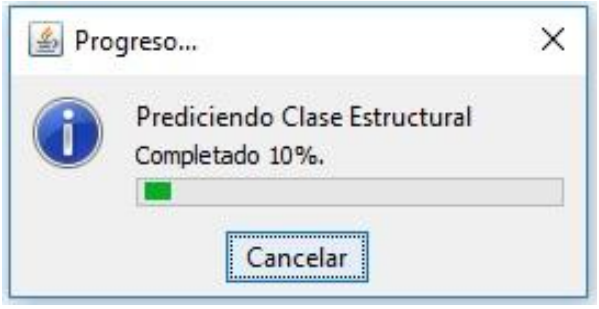


Fig. 11 Prototipo de interfaz de progreso de la predicción.

2.6 Diseño

El diseño de la arquitectura define la relación entre los elementos principales de la estructura del software, los estilos y patrones de diseño de la arquitectura que pueden usarse para alcanzar los requerimientos definidos por el sistema y las restricciones que afectan la forma en la que se implementa la arquitectura (Pressman 2010).

2.6.1 Patrón arquitectónico utilizado

Los patrones arquitectónicos se abocan a un problema de aplicación específica dentro de un contexto dado y sujeto a limitaciones y restricciones. El patrón propone una solución arquitectónica que sirve como base para el diseño de la arquitectura (Pressman 2010).

Patrón arquitectónico N-Capas.

En la estructura básica de una arquitectura en capas, se definen un número de capas diferentes; cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. En la capa externa, los componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen servicios de utilerías y funciones de software de aplicación (Pressman 2010). En la construcción del sistema se aplicó el patrón arquitectónico N-Capas, donde se definieron 2 niveles de abstracción (capas): Lógica y Presentación como se muestra en la vista lógica general del sistema (Fig. 12).

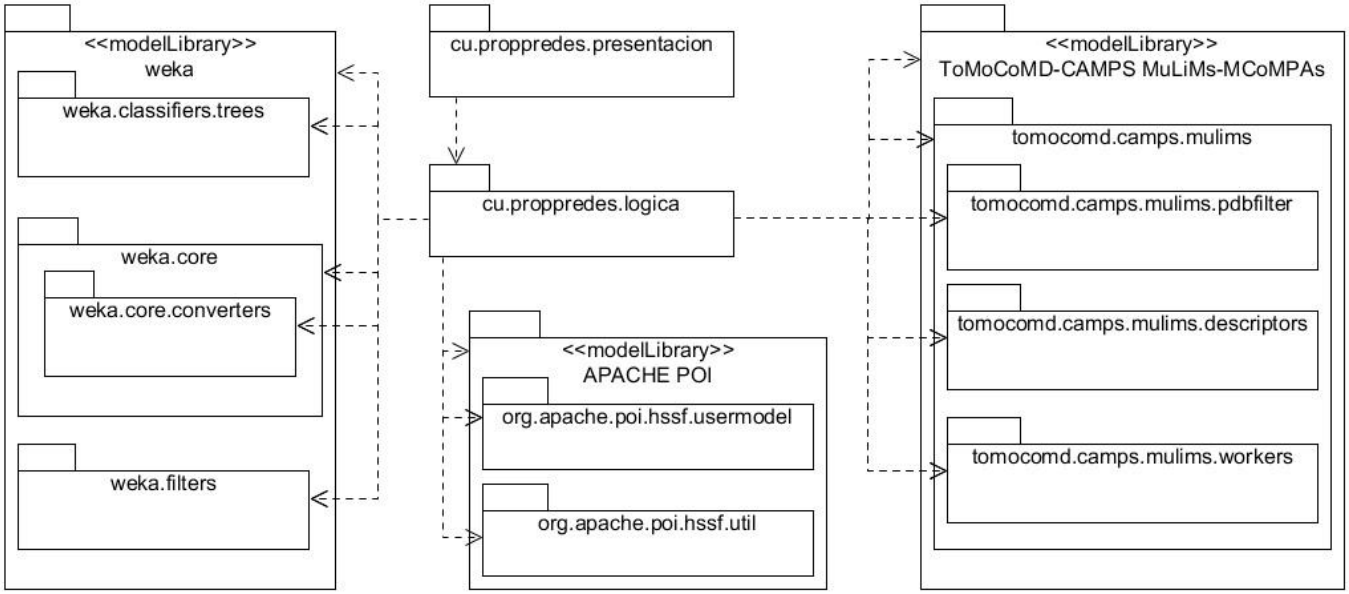


Fig. 12 Vista lógica general del sistema.

En la figura anterior fueron representados los siguientes elementos:

- ✓ Capa de Presentación (cu.proppredes.presentacion): En este nivel están organizadas las interfaces gráficas de usuario. Estas acceden a los procedimientos definidos por la capa inferior (Lógica) para obtener los servicios que la misma brinda y mostrarle al usuario las respuestas requeridas una vez realizadas las peticiones al sistema.

- ✓ Capa Lógica (cu.proppredes.logica): Esta capa contiene clases y procedimientos necesarios para dar cumplimiento a los requisitos. Utiliza las funcionalidades brindadas por las bibliotecas ToMoCoMD-CAMPS MuLiMs-MCoMPAs para el cálculo de los descriptores, WEKA para realizar el proceso de clasificación y APACHE POI para exportar los datos. Proporciona funcionalidades a la capa Presentación.
- ✓ Biblioteca WEKA: La biblioteca WEKA no constituye una capa en la estructura de la aplicación, pero se utiliza para realizar el proceso de clasificación para predecir las clases estructurales de las proteínas.
- ✓ Biblioteca ToMoCoMD-CAMPS MuLiMs-MCoMPAs: La biblioteca ToMoCoMD-CAMPS MuLiMs-MCoMPAs no constituye una capa en la estructura de la aplicación, pero es la que se encarga del cálculo de los descriptores.
- ✓ Biblioteca APACHE POI: La biblioteca APACHE POI no constituye una capa en la estructura de la aplicación, pero permite guardar el resultado de la predicción.

2.6.2 Diagrama de Clases del Diseño

Partiendo de los casos de uso del sistema, se realizaron los correspondientes Diagramas de Clases del Diseño (DCD), los cuales representan las clases que serán utilizadas dentro del sistema con sus operaciones y las relaciones que existen entre ellas para dar respuesta a los requisitos. En la Fig. 13 se muestra el DCD para el CU “Realizar predicción de la clase estructural de las proteínas”.

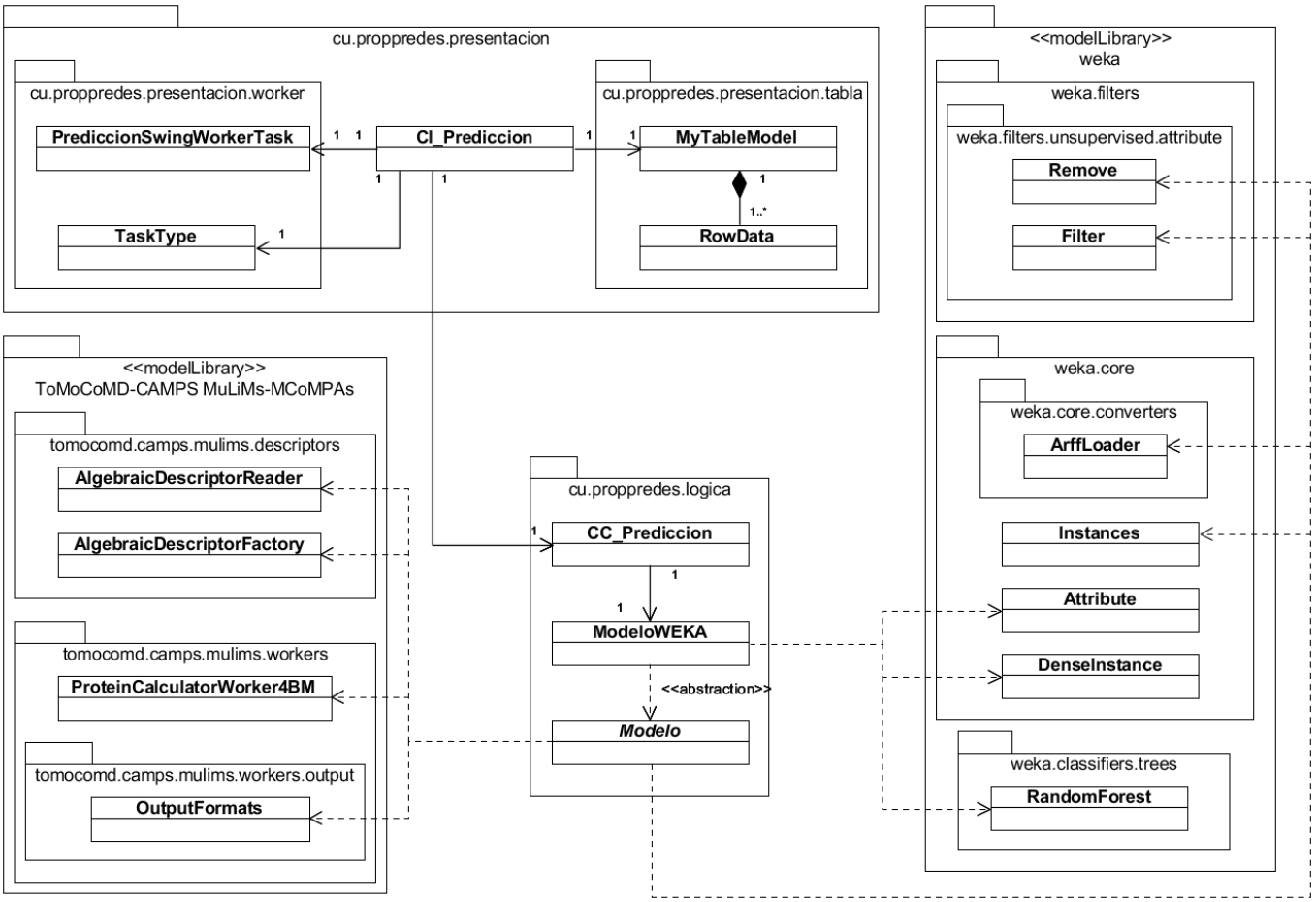


Fig. 13 DCD para el CU “Realizar predicción de la clase estructural de las proteínas”.

La clase interfaz *CI_Prediccion* invoca el procedimiento de la clase controladora *CC_Prediccion* para realizar la predicción de la clase estructural, esta invoca el método *makePrediction()* de la subclase *ModeloWEKA*. Este método calcula los descriptores necesarios para llevar a cabo la predicción mediante las clases implementadas en la biblioteca ToMoCoMD-CAMPS MuLiMs-MCoMPAs a partir de los archivos cargados y evalúa los descriptores obtenidos en el modelo de clasificación utilizando la biblioteca WEKA. Una vez obtenidas las predicciones, la clase *CC_Prediccion* devuelve una lista con los resultados a la clase *CI_Prediccion* donde son mostrados al usuario.

2.6.3 Patrones Aplicados

Patrones GRASP

Para la construcción de los diagramas es necesario asignar correctamente las responsabilidades. Para ello es importante auxiliarse de los patrones de diseño, garantizando un correcto diseño de la interacción de los objetos del sistema. En este caso se tuvieron en cuenta como buenas prácticas de desarrollo de

software los patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos expresados en forma de patrones (Tabares 2011).

- ✓ Experto: *CC_Prediccion* cuenta con la información de cada clase asignando a cada una sus responsabilidades.
- ✓ Creador: *CC_Prediccion* es creadora porque crea una instancia de *ModeloWEKA* y *ModeloRLM*, garantizando así un bajo acoplamiento.
- ✓ Bajo Acoplamiento: En el sistema se logra un bajo acoplamiento, ya que las responsabilidades asignadas a las diferentes clases minimizan el nivel de dependencia entre las mismas, gracias a la aplicación de los demás patrones relacionados (Experto, Creador).
- ✓ Alta Cohesión: En el sistema se logra una alta cohesión porque las responsabilidades asignadas a las diferentes clases se encuentran en la misma área de aplicación y están relacionadas con tareas afines.
- ✓ Controlador: *CC_Prediccion* es el que recibe las peticiones del usuario a través de la interfaz gráfica y coordina su realización delegando a otros objetos.

2.6.4 Técnica de Representación del Conocimiento

Existen diversas formas de representar el conocimiento, en este caso se seleccionó la Técnica de Representación del Conocimiento basada en modelos, ya que la misma es ampliamente utilizada en el campo de estudio de la ciencia de las proteínas para la predicción de propiedades biológicas, ejemplo de esto son los Sistemas Expertos estudiados a los cuales se hace referencia en el Capítulo 1 (ver epígrafe 1.2.5 Sistemas Expertos en el campo de la Bioinformática y Biología Computacional).

A partir de los resultados obtenidos en los estudios de Análisis de Variabilidad y de Componentes Principales, se seleccionaron, de los parámetros analizados, aquellos que mostraron mejor comportamiento. Estos parámetros se utilizaron para calcular los descriptores que se emplearon para desarrollar los modelos de clasificación y de regresión que se describen a continuación.

Modelos para la predicción de las clases estructurales de proteínas

Descripción del conjunto de datos

Para el desarrollo de los modelos de clasificación se utilizó el conjunto de datos propuesto en (Chou 1999), el cual está compuesto por 204 proteínas de las cuales 52 son *All-α*, 61 *All-β*, 45 *α/β* y 46 *α+β*. Con el propósito de obtener los modelos de clasificación y evaluar su capacidad predictiva se dividió el

conjunto original en dos, una serie de entrenamiento (*SE*) con 149 proteínas y otra de predicción (*SP*) con 55 como se procedió en (Marrero-Ponce et al. 2015).

Desarrollo de los modelos de clasificación

Para la obtención de los modelos de clasificación se utilizaron las técnicas: Random Forest, K-vecinos más Cercanos y Perceptrón Multicapa, las cuales están implementadas en el software WEKA 3.7.10, utilizando los descriptores obtenidos mediante el siguiente procedimiento:

1. Calcular mediante el software ToMoCoMD-CAMPS MuLiMs-MCoMPAs 1.0 los descriptores para cada una de las representaciones 3D-proteicas [*Cα* (átomo de carbono alfa), *Cβ* (átomo de carbono beta), *CEA* (átomo de carbono del enlace Amida), *AVG* pseudo átomo (Promedio o media aritmética de las coordenadas espaciales (x, y, z) de todos los átomos de aminoácidos)].

2. Seleccionar de cada fichero generado 1000 DMs de mayor variabilidad por Entropía de Shannon con el software IMMAN 1.0.

3. Adicionar a cada fichero obtenido en el paso 2, la variable respuesta (clase estructural).

4. Aplicar a cada fichero obtenido en el paso 3, la técnica de selección de atributos *Correlation Feature Selection* disponible en el software WEKA 3.7.10.

5. Mezclar todos los ficheros obtenidos en el paso 4.

6. Seleccionar de cada fichero generado en el paso 5, 100 DMs de mayor variabilidad por Entropía de Shannon con el software IMMAN 1.0 y repetir los pasos 3 y 4.

Modelos para la predicción de la velocidad de plegamiento de cadenas polipeptídicas

Descripción de los conjuntos de entrenamiento y prueba

Para el desarrollo y evaluación de los modelos de regresión, se emplearon dos conjuntos de datos, una serie de entrenamiento con 80 proteínas (Ouyang and Liang 2008) y una de predicción con 16 (Ruiz-Blanco et al. 2015). Resulta importante señalar que de la serie de entrenamiento fue excluida la proteína con identificador “2BLM” por contener únicamente las coordenadas espaciales de sus átomos *Cα*.

Desarrollo de los modelos de regresión

Los modelos de Regresión Lineal Múltiple se obtuvieron con el software Mobydigs 1.0, estos se calcularon utilizando la configuración: tamaño de la población (100), cantidad de variables permitidas en el modelo (2-8), intercambio entre reproducción/mutación (0.7). Lista de elementos no permitidos: cuarto orden mayor que (8), correlación entre x/x mayor que (0.95), Entropía Estandarizada menor que (0.3), empleando los descriptores obtenidos mediante el siguiente procedimiento:

1. Calcular mediante el software ToMoCoMD-CAMPS MuLiMs-MCoMPAs 1.0 los descriptores para cada una de las representaciones 3D-proteicas [*Cα*, *Cβ*, *CEA*, *AVG*].

2. Seleccionar de cada fichero generado 1000 DMs de mayor variabilidad por Entropía de Shannon con el software IMMAN 1.0.

3. Adicionar a cada fichero obtenido en el paso 2, la variable respuesta (ln_kf).

4. Aplicar a cada fichero obtenido en el paso 3, la técnica de selección de atributos *Correlation Feature Selection* disponible en el software WEKA 3.7.10.

5. Mezclar todos los ficheros obtenidos en el paso 4 y repetir desde el paso 2 al 4.

2.6.5 Diagrama de Componentes

Un Diagrama de Componentes (DC) describe los elementos físicos del sistema y las relaciones entre las partes (componentes), ya sean bibliotecas, componentes de código fuente o archivos, conteniendo además sus dependencias más significativas. Un diagrama no debe reflejar necesariamente todo el sistema, generalmente se divide por apartados (Mouheb et al. 2015). En el diagrama mostrado en la Fig. 14, se exponen los componentes vinculados a la ejecución de CU “Realizar predicción de la clase estructural de las proteínas”.

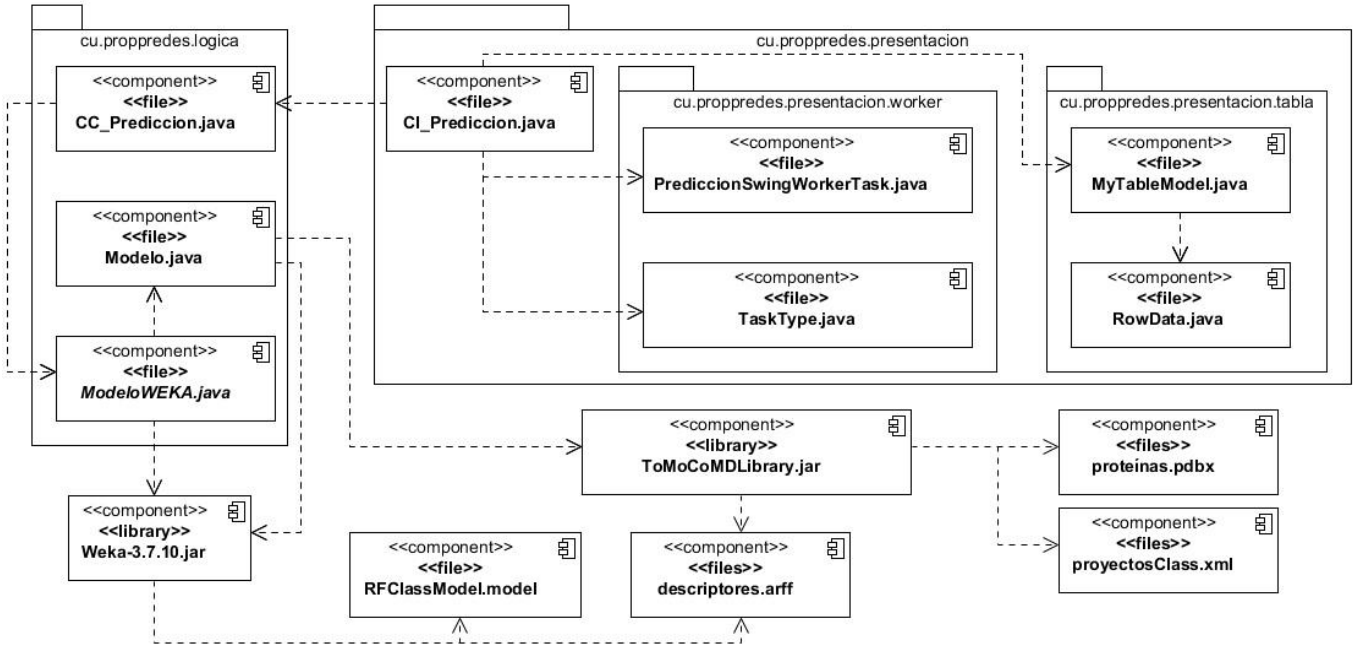


Fig. 14 DC para el CU “Realizar predicción de la clase estructural de las proteínas”.

En la figura anterior se muestran los siguientes elementos organizados por capas:

Capa Presentación (cu.proppredes.presentacion):

- ✓ CI_Prediccion.java: Contiene la clase interfaz *CI_Prediccion*.
- ✓ MyTableModel.java: Contiene la clase *MyTableModel*.
- ✓ RowData.java: Contiene la clase *RowData*.
- ✓ PrediccionSwingWorkerTask.java: Contiene la clase *PrediccionSwingWorkerTask*.
- ✓ TaskType.java: Contiene la clase *TaskType*.

Capa Lógica (cu.proppredes.logica):

- ✓ CC_Prediccion.java: Contiene la clase controladora *CC_Prediccion*.
- ✓ Modelo.java: Contiene la clase abstracta *Modelo*.
- ✓ ModeloWEKA.java: Contiene la clase *ModeloWEKA* que extiende de la clase abstracta *Modelo*.

Archivos y bibliotecas:

- ✓ ToMoCoMDLibrary.jar: Contiene la biblioteca ToMoCoMD-CAMPS MuLiMs-MCoMPAs encargada de realizar el cálculo de los descriptores.
- ✓ Weka-3.7.10.jar: Contiene la biblioteca WEKA que se encarga del manejo de los archivos arff y realizar la clasificación.

- ✓ proteínas.pdbx: Ficheros donde se almacenan los datos necesarios para calcular los descriptores.
- ✓ proyectosClass.xml: Ficheros que contienen las configuraciones de los parámetros para el cálculo de los descriptores.
- ✓ descriptores.arff: Ficheros que contienen los descriptores calculados.
- ✓ RFClassModel.model: Fichero que contiene el modelo de clasificación previamente serializado.

2.6.6 Estándar de codificación

Un estándar de codificación, establece las pautas y normas a seguir por los desarrolladores de un sistema para que su código fuente sea escrito en un formato unificado, logrando mayor organización y efectividad durante la fase de implementación. En la codificación de PropPred-ES, se utilizaron los estándares establecidos para la codificación en Java, reflejados en (Stoll 2014) de acuerdo a las particularidades del sistema. Algunos de los aspectos importantes a tener en cuenta son:

- ✓ Evitar las líneas de más de 80 caracteres.
- ✓ Siempre que sea posible inicializar las variables locales donde se declaran.
- ✓ Poner las declaraciones solo al principio de los bloques.
- ✓ Cada línea debe contener como máximo una sentencia simple (ej.: *c++).*
- ✓ El prefijo del nombre de un paquete se escribe siempre con letras ASCII en minúsculas y debe ser uno de los nombres de dominio de alto nivel (com, gov) o uno de los códigos ingleses de dos letras que identifican a cada país (cu, es, en). Los subsecuentes componentes del nombre del paquete varían de acuerdo a las convenciones de nombres internas de cada organización.
- ✓ Los nombres de las clases e interfaces deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas.
- ✓ Los nombres de los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula y la primera letra de las siguientes palabras que los forman en mayúscula.

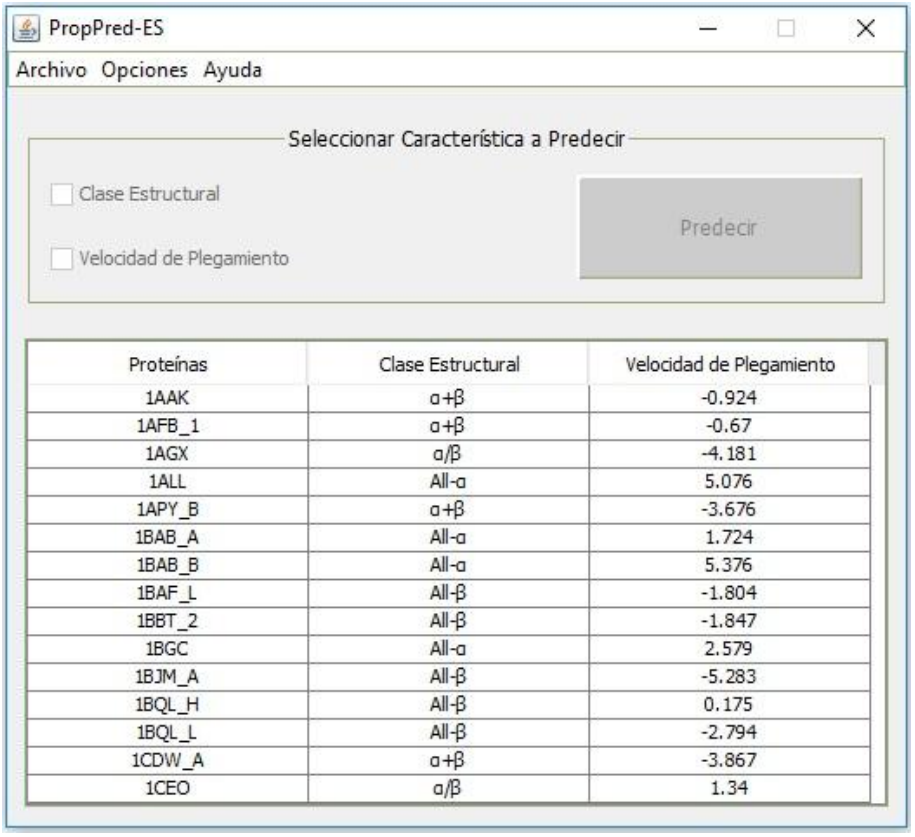
En la Fig. 15 se muestra un fragmento del código en lenguaje Java implementado, aplicando el estándar de codificación.

```
public Instances mergeArffFiles(File [] descriptorstomerge) throws IOException, Exception
{
   ArffLoader arffloader = new ArffLoader();
   arffloader.setSource(descriptorstomerge[0]);
   Instances fusionData = arffloader.getDataSet();
   for (int i = 1; i < descriptorstomerge.length; i++)
   {
      ArffLoader currenLoader = new ArffLoader();
      currenLoader.setSource(descriptorstomerge[i]);
      Instances currentInstances = currenLoader.getDataSet();
      currentInstances = removeDuplicatedAttributes(currentInstances, getAttributesNames(fusionData));
      if (currentInstances.numInstances() == fusionData.numInstances())
      {
         fusionData = Instances.mergeInstances(fusionData, currentInstances);
      }
   }
   return fusionData;
}
```

Fig. 15 Ejemplo de código fuente aplicando los estándares de codificación.

2.6.7 Desarrollo de la interfaz

En la Fig. 16 se muestra la interfaz para realizar la(s) predicción(es). La misma posee un menú que brinda las opciones: cargar el o los archivos que contienen la información de las proteínas, guardar resultado de la predicción y consultar ayuda sobre el funcionamiento del sistema. Además, la interfaz cuenta con dos paneles, uno donde el usuario puede seleccionar la(s) característica(s) a predecir de las proteínas cargadas y otro que muestra el resultado de la predicción.



Proteínas	Clase Estructural	Velocidad de Plegamiento
1AAK	α+β	-0.924
1AFB_1	α+β	-0.67
1AGX	α/β	-4.181
1ALL	All-α	5.076
1APY_B	α+β	-3.676
1BAB_A	All-α	1.724
1BAB_B	All-α	5.376
1BAF_L	All-β	-1.804
1BBT_2	All-β	-1.847
1BGC	All-α	2.579
1BJM_A	All-β	-5.283
1BQL_H	All-β	0.175
1BQL_L	All-β	-2.794
1CDW_A	α+β	-3.867
1CEO	α/β	1.34

Fig. 16 Interfaz del Sistema Experto.

Conclusiones parciales

El estudio de viabilidad realizado demostró que el Sistema Experto puede ser desarrollado exitosamente. Los estudios exploratorios de Análisis de Variabilidad y de Componentes Principales permitieron acotar el número de descriptores para desarrollar los modelos. El proceso de análisis y diseño para el desarrollo de software posibilitó una mejor comprensión del problema, permitiendo identificar 5 requisitos funcionales del sistema, los cuales fueron agrupados en 4 casos de uso. Se escogió el patrón arquitectónico 2-capas. Se seleccionó la Técnica de Representación del Conocimiento basada en modelos, partiendo de la misma se desarrollaron los modelos de clasificación y de regresión, los cuales constituyen el núcleo del Sistema Experto. Además, se implementó el Sistema Experto describiendo los diagramas de componentes y los principales elementos del estándar de codificación que se emplearon, así como el desarrollo de la interfaz.

Capítulo 3 Evaluación del Sistema Experto para la Predicción de Propiedades Biológicas en Proteínas

En este capítulo se evalúa el desempeño de los modelos de clasificación y de regresión obtenidos para determinar cuáles son los más predictivos. Además, se describen los procedimientos utilizados para validar el sistema y se presentan los resultados obtenidos.

3.1 Evaluación del desempeño de los modelos de clasificación y de regresión

Para evaluar el desempeño de los modelos de clasificación y de regresión seleccionados se utilizaron técnicas de validación interna y externa, siguiendo los criterios propuestos en el marco de la Organisation for Economic Co-operation and Development (OECD 2007), las cuales permitieron determinar el ajuste, robustez y capacidad predictiva de los modelos obtenidos.

3.1.1 Evaluación del desempeño de los modelos de clasificación

Para valorar el desempeño de los modelos de clasificación se utilizaron las siguientes medidas de evaluación:

- ✓ **Exactitud o precisión (Q):** Porcentaje de casos (proteínas) clasificadas correctamente (Baldi et al. 2000).
- ✓ **Sensibilidad (SEN):** Es la probabilidad de clasificar correctamente un caso positivo (Baldi et al. 2000).
- ✓ **Especificidad (ESP):** Es la probabilidad de que una predicción positiva sea correcta (Baldi et al. 2000).

En la Tabla 4 se muestran los porcentajes de proteínas clasificadas correctamente asociados a los modelos obtenidos, utilizando las técnicas Random Forest, K-vecinos más Cercanos y Perceptrón Multicapa para cada una de las representaciones 3D-proteicas.

Tabla. 4 Porcentajes de proteínas clasificadas correctamente.

Representaciones 3D-proteicas	Técnicas	Conjunto de datos	Exactitud global Q(%)	Clasificados correctamente
Cα	RF	SP	96	(53/55)
	K-NN	SP	91	(50/55)
	MLP	SP	78	(43/55)
Cβ	RF	SP	89	(49/55)
	K-NN	SP	89	(49/55)
	MLP	SP	63	(35/55)
CEA	RF	SP	98	(54/55)
	K-NN	SP	94	(52/55)
	MLP	SP	87	(48/55)
AVG	RF	SP	87	(48/55)
	K-NN	SP	89	(49/55)
	MLP	SP	73	(40/55)

Al analizar la exactitud global de las proteínas clasificadas correctamente en la serie de predicción se concluye que: El modelo obtenido a partir de la representación 3D-proteica CEA que utiliza la técnica Random Forest es el más predictivo de todos. En la Tabla 5 se muestran los parámetros estadísticos de este modelo. Para consultar los parámetros estadísticos de los restantes modelos de clasificación (Ver Anexo 2).

Tabla. 5 Parámetros estadísticos del modelo de clasificación seleccionado.

Conjunto	Exactitud global Q(%)	Sensibilidad (%)		Especificidad (%)	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	98	**clases**		**clases**	
		α+β	100	α+β	98
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	93	All-α	100

Evaluación del modelo de clasificación seleccionado

Validación interna: La exactitud global de casos clasificados correctamente en la serie de entrenamiento es del 100 % (149/149). Se aplicó la estrategia de validación interna *10-fold cross validation* obteniendo una precisión global de 97.98% (146/149), esto es indicativo de la robustez del modelo.

Validación externa: La exactitud global de casos clasificados correctamente en la serie de predicción es de 98 % (54/55), esto es un indicador del alto poder predictivo del modelo.

Por otra parte, como se evidencia en la Tabla 5, el modelo presenta altos valores de sensibilidad y especificidad. Además, los valores del parámetro *kappa statistic* (Witten et al. 2016) obtenidos en la serie de entrenamiento, *10-fold cross validation* y la serie de predicción son de 1, 0.972 y 0.975 respectivamente, estos revelan que la probabilidad de que exista correlación aleatoria con la clase es baja, lo cual es un indicativo de la robustez del modelo. Los resultados de la validación interna y externa del modelo de clasificación seleccionado demuestran que es un modelo robusto y con alto poder predictivo. Finalmente, resulta importante señalar que el modelo desarrollado supera el desempeño obtenido por el modelo reportado en la literatura, en el cual se obtuvieron los siguientes resultados Q(*SE*) 92.6%, Q(*SP*) 92.7% (Marrero-Ponce et al. 2015).

3.1.2 Evaluación del desempeño de los modelos de regresión

En la evaluación del desempeño de los modelos de regresión se utilizaron las siguientes medidas de evaluación:

- ✓ **Medida de ajuste:** Coeficiente de correlación cuadrado o coeficiente de determinación (R^2), para estimar la proporción de la variable respuesta explicada por la regresión. Es decir, si no existe relación lineal entre la variable dependiente y las variables independientes, entonces $R^2=0$; por otra parte, si existe un ajuste perfecto, entonces $R^2=1$ (OECD 2007).
- ✓ **Medidas de robustez:** Validación cruzada por re-muestreo (*bootstrapping*) (Wehrens et al. 2000), en la cual se determinan aleatoriamente conjuntos de entrenamiento y de prueba para determinar el poder predictivo promedio (Q^2_{boot}) y prueba de permutación de las respuestas o *Y-scrambling* para identificar modelos basados en correlación aleatoria y determinar la calidad del modelo a(Q^2).
- ✓ **Medidas de predicción externa:** Validación cruzada dejando uno fuera *leave-one-out* (LOO), en la cual se calculan *n* modelos reducidos para predecir la respuesta del compuesto excluido y determinar el poder predictivo (Q^2_{loo}) (OECD 2007). La validación externa permite evaluar si los modelos obtenidos son generalizables a nuevos compuestos químicos y de esta forma evaluar el verdadero poder predictivo de los mismos. El estadístico utilizado con este fin se denomina (Q^2_{ext}).

De los modelos obtenidos por cada una de las representaciones 3D-proteicas utilizando la técnica Regresión Lineal Múltiple, se seleccionaron aquellos que presentan mayor valor de (Q^2_{ext}), los cuales se muestran en la Tabla 6.

Tabla. 6 Modelos con mayor valor (Q^2_{ext}).

Representaciones 3D-proteicas	Dimensión	Q^2_{ext}
Cα	2	0.7199
Cβ	2	0.4324
CEA	3	0.7263
AVG	2	0.4729

Al analizar los valores (Q^2_{ext}) de los modelos seleccionados se determinó que: El modelo obtenido a partir de la representación 3D-proteica CEA con 3 variables es el más predictivo de todos. Resulta importante destacar, que el modelo fue obtenido a partir de la misma representación de proteínas con que se obtuvo el mejor modelo de clasificación seleccionado. Estos resultados evidencian la relevancia de esta representación, la cual exhibe un desempeño superior a la más comúnmente empleada basada en los *Cα*. En la Tabla 7 se muestran los parámetros estadísticos del modelo seleccionado. Para consultar los parámetros estadísticos de los restantes modelos de regresión (Ver Anexo 3).

Tabla. 7 Parámetros estadísticos del modelo de regresión seleccionado.

Dimensión	R^2	Q^2_{loo}	Q^2_{boot}	$a(Q^2)$	Q^2_{ext}
3	0.7832	0.7612	0.7584	-0.107	0.7263

Evaluación del modelo de regresión seleccionado

Validación interna: Un valor R^2 =1 es indicativo de un ajuste perfecto, en el modelo evaluado R^2 =0.7832, esto muestra que existe relación lineal entre la variable dependiente y las variables independientes, por lo que el modelo presenta un buen ajuste. El valor obtenido por la técnica "*bootstrapping*" (Q^2_{boot}) alcanza el 76% de la varianza total, por tanto, puede considerarse que el modelo es robusto o con buena capacidad de predicción interna. En el procedimiento "*Y-scrambling*" [$a(Q^2)$] se determinaron coeficientes con un valor de -0.107, lo que indica que la correlación entre las variables independientes y la variable modelada presenta un bajo grado de aleatoriedad.

Validación externa: Como la serie de entrenamiento es pequeña (79 proteínas), el parámetro (Q^2_{loo}) puede ser considerado como un indicador de la capacidad predictiva externa de los modelos (Todeschini and Consonni 2009; Tropsha et al. 2003), en este caso, el modelo obtenido explica el 76% de la varianza total. Al examinar el desempeño en la serie de predicción (16 proteínas), se puede decir, que el modelo

obtenido posee un buen poder predictivo ya que los valores del estadístico Q^2_{ext} =0.7263, lo que indica que el modelo obtenido explica aproximadamente el 73% de la varianza total.

La evaluación del modelo de regresión seleccionado reveló que es un modelo robusto y con buen poder predictivo. Además, resulta importante destacar que el modelo posee un coeficiente de correlación entre la variable respuesta (ln_kf) y la predicción obtenida por el modelo de 0.8333, el cual supera al modelo reportado en el artículo (Ruiz-Blanco et al. 2015), que posee un coeficiente de correlación de 0.70496.

3.2 Validación del sistema

Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como, descubrir defectos en el programa antes de usarlo. En la prueba de validación se espera que el sistema se desempeñe de manera correcta mediante un conjunto dado de casos de prueba que refleje el uso previsto del sistema (Sommerville 2011).

Para realizar la validación del PropPred-ES y comprobar el correcto funcionamiento del sistema, así como el cumplimiento de los requisitos, se realizaron los niveles de pruebas de unidad y de sistema.

Para las pruebas de unidad se utilizó el NetBeans IDE. Se crearon casos de prueba JUnit (del inglés, *JUnit Test Case*) (Vogel 2013), donde se verificó el correcto funcionamiento de los procedimientos que responden a los CU “Realizar predicción de la clase estructural de las proteínas” y “Realizar predicción de la velocidad de plegamiento de las proteínas”, mediante la clase *CC_Prediccion*. En estos casos se utilizaron diferentes archivos, para comprobar si las predicciones obtenidas coinciden con resultados obtenidos anteriormente y el sistema respondió satisfactoriamente (Ver Anexo 4).

A nivel de sistema se realizaron pruebas de funcionalidad, aplicando el método de Caja Negra. Este se basa en la realización de pruebas sobre la interfaz del programa, evaluando las entradas y salidas del mismo, de esta forma se comprueba el cumplimiento de todos los requisitos del sistema. Para su aplicación se elaboraron casos de prueba a partir de las funcionalidades descritas en los casos de uso del sistema y de la descripción de los mismos como apoyo para las revisiones. En cada caso de prueba se refleja la especificación de un caso de uso, dividido en secciones y escenarios, donde se detallan las funcionalidades descritas en él y se describen las variables utilizadas.

Para elaborar los casos de prueba se empleó la técnica partición de equivalencia, dividiendo los datos de entrada en conjuntos válidos o inválidos (clases de equivalencias), de esta forma se garantiza que se evalúen todos los casos posibles. En la Tabla 8 se describe el caso de prueba para el CU “Cargar archivos” para la sección con el mismo nombre y en la Tabla 9 se realiza la descripción de las variables.

Tabla. 8 Caso de prueba “Cargar archivos”.

Escenario	Descripción	Nombre del archivo	Respuesta del sistema	Flujo central
EC 1.1 El actor carga un archivo en formato PDB.	En este escenario el actor carga un archivo PDB que contiene la estructura de una proteína.	V 1ADW.pdb	El sistema carga correctamente el archivo seleccionado y lo muestra en la tabla.	1-Archivo/Cargar archivo. 2-Seleccionar el archivo. 3-Seleccionar la opción "Cargar".
EC 1.3 El actor carga un archivo corrupto en formato PDB.	En este escenario el actor carga un archivo PDB corrupto que no contiene la estructura de una proteína.	I 1CSP-corrupto.pdb	El sistema muestra un mensaje de notificación, informando al usuario que el archivo tiene problemas y no se pudo cargar.	1-Archivo/Cargar archivo. 2-Seleccionar el archivo. 3-Seleccionar la opción "Cargar".

Tabla. 9 Descripción de las variables del caso de prueba “Cargar archivos”.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre del archivo	Archivo	No	Archivo que solo puede estar en formato PDB o ENT.

Después de realizadas las pruebas funcionales, los errores encontrados fueron redactados como no conformidades para ser solucionados por el equipo de desarrollo. En total se realizaron tres iteraciones y se encontraron 4 no conformidades, las cuales fueron resueltas en la iteración correspondiente como se muestra en la Fig. 17. En la Tabla 10 se muestran algunas de las no conformidades encontradas.

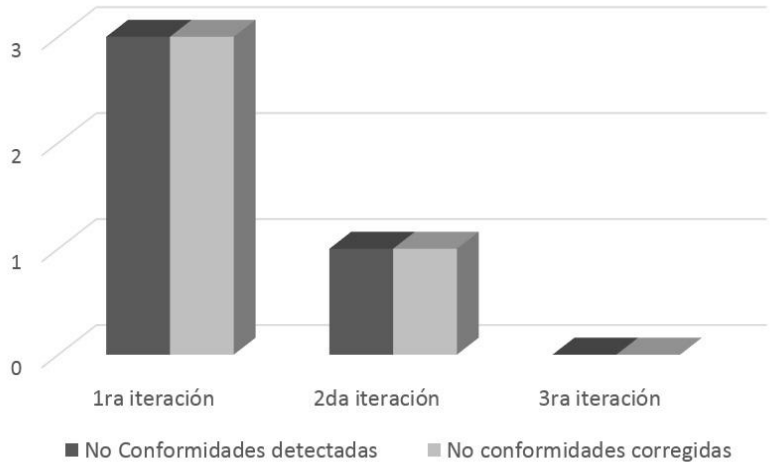


Fig. 17 No conformidades detectadas y corregidas.

Tabla. 10 Ejemplo de no conformidades encontradas

Elemento	No.	No Conformidad	Ubicación	Etapa de detección	Clasificación
Aplicación	1	La tabla donde se muestran los archivos cargados es editable.	Cargar archivos	Pruebas de funcionalidad	Significativa
Aplicación	2	No se pueden seleccionar archivos PDB y ENT al mismo tiempo.	Cargar archivos	Pruebas de funcionalidad	Significativa

Para evaluar el rendimiento del sistema se realizaron pruebas en una PC con las siguientes características: Procesador Intel(R) Dual-Core E5300 2.60 GHz y 1GB de RAM con Sistema Operativo Windows 7 Ultimate x86, además, a la máquina virtual de Java solo se le asignó 512 Megabytes de memoria RAM. Para realizar las pruebas se escogió la base de proteínas utilizada en los estudios exploratorios, esta es representativa en cuanto a la longitud (complejidad) de estos compuestos, ya que el número de residuos de aminoácidos de las proteínas que contiene varía de 50 a 753. Se midieron los tiempos de respuesta al realizar predicciones utilizando diferentes cantidades de proteínas. En la Tabla 11 se muestran los resultados obtenidos al realizar las predicciones de las características clase estructural y velocidad de plegamiento, donde se utilizaron 50, 100 y 150 proteínas, para comprobar los tiempos de respuesta en cada caso.

Tabla. 11 Comparación de los tiempos de respuesta del sistema.

Tiempo de respuesta en minutos para la predicción de:		Tiempo de respuesta en segundos para una predicción.	Promedio del tiempo de respuesta en segundos de una predicción.
50 Proteínas	3.74	4.488	4.406
100 Proteínas	7.59	4.554	
150 Proteínas	10.44	4.176	

De la tabla anterior se deduce que, en una PC con estas características el sistema funcionará brindando tiempos de respuesta razonables. Además, se realizaron cálculos para el caso hipotético donde fueran seleccionadas 10000 estructuras de gran tamaño (Ej. 753 aminoácidos). El tiempo de respuesta al realizar ambas predicciones es de aproximadamente 38.89 horas, lo cual evidencia un buen desempeño si se tiene en cuenta la cantidad y complejidad de proteínas a predecir.

3.3 Validación de la hipótesis

El análisis estadístico del mejor modelo de clasificación estructural obtenido con descriptores bilineales de la matriz de proximidad, muestra que sus niveles de previsibilidad interna y externa son elevados (superior a 98%), lo que puede considerarse como excelente según (Eriksson 2003), indicando que efectivamente es posible predecir la clase estructural de proteínas.

Por su parte, la evaluación estadística del mejor modelo obtenido con descriptores bilineales de la matriz de proximidad para predecir la velocidad de plegamiento de proteínas, demuestra que la varianza explicada en serie externa es alta (aproximadamente 73%), lo que puede ser considerado como bueno según (Eriksson 2003), probando que ciertamente es posible predecir la velocidad de plegamiento de proteínas.

Las pruebas de validación que se realizaron al Sistema Experto PropPred-ES demostraron el correcto funcionamiento del mismo. Por todo lo anteriormente expuesto se puede afirmar que el Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad implementado, predice la clase estructural y la velocidad de plegamiento de proteínas.

Consecuentemente, queda demostrada la hipótesis de investigación: “Si se implementa un Sistema Experto basado en modelos de rasgos bilineales de la matriz de proximidad, entonces es posible predecir la clase estructural y la velocidad de plegamiento de proteínas”.

Conclusiones parciales

Las técnicas de validación interna y externa utilizadas en la evaluación del desempeño de los modelos de clasificación y de regresión, permitieron seleccionar los modelos más robustos y con mejor poder predictivo, encontrándose que estos aventajan a otros modelos reportados en investigaciones anteriores. Se automatizaron las pruebas realizadas al sistema utilizando casos de prueba JUnit, los cuales permitieron verificar el correcto funcionamiento de los procedimientos correspondientes a los diferentes casos de uso. Se realizaron pruebas de funcionalidad, utilizando el método de Caja Negra donde se detectaron errores que fueron posteriormente solucionados. Finalmente, se valoró el cumplimiento de la validación de la hipótesis enunciada en la investigación.

Conclusiones

Con la realización del presente trabajo se arriban a las siguientes conclusiones:

- ✓ El estudio de los conceptos relacionados con la Inteligencia Artificial y los Sistemas Expertos basados en modelos sirvió de referente para el diseño e implementación del Sistema Experto PropPred-ES.
- ✓ Se realizaron estudios exploratorios, logrando reducir el espacio de alta dimensión de rasgos moleculares, obteniendo los parámetros de mejor comportamiento, los cuales se utilizaron para calcular los descriptores que se emplearon en el desarrollo de los modelos de clasificación y de regresión.
- ✓ Se desarrollaron modelos de clasificación para predecir la clase estructural de proteínas, utilizando las técnicas K-vecinos más Cercanos, Perceptrón Multicapa y Random Forest. Se seleccionó el modelo obtenido con esta última técnica para integrar al Sistema Experto por ser el más predictivo.
- ✓ Se desarrollaron modelos para predecir la velocidad de plegamiento de proteínas, utilizando la técnica Regresión Lineal Múltiple y se seleccionó el más predictivo para integrar al Sistema Experto.
- ✓ Partiendo de los principios generales del desarrollo de software se realizó el análisis y diseño del Sistema Experto, logrando definir su organización lógica, haciendo uso de patrones de diseño.
- ✓ Se implementó un Sistema Experto que predice la clase estructural y la velocidad de plegamiento de las proteínas, dando cumplimiento a los requisitos funcionales identificados.
- ✓ La validación de los modelos que integran el Sistema Experto demostró que estos presentan buen ajuste, alta robustez y buena capacidad predictiva. Además, se validaron de forma automatizada las funcionalidades implementadas en el sistema, lo cual permitió corroborar su utilidad en la predicción de la clase estructural y la velocidad de plegamiento en proteínas.

Recomendaciones

Una vez concluida la investigación y para contribuir al éxito en la continuidad de la misma existen diferentes aspectos que merecen ser tratados posteriormente, en tal sentido se propone:

- ✓ Extender el Sistema Experto para modelar otras propiedades biológicas de interés.
- ✓ Emplear los resultados obtenidos para implementar un web server y ponerlo a disposición de la comunidad científica.
- ✓ Emplear la estrategia utilizada para reducir el espacio de alta dimensión de rasgos moleculares en el estudio de otros descriptores.
- ✓ Utilizar otras técnicas implementadas en el software WEKA para la obtención de modelos.

Referencias Bibliográficas

1. **BALDI, P., S. BRUNAK, Y. CHAUVIN, C. A. ANDERSEN, et al.** Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 2000, 16(5), 412-424.
2. **BANDA, H.** *Inteligencia Artificial: Principios y Aplicaciones*. Edtion ed., 2014.
3. **BARCHINI, G. E.** Métodos I+ D de la Informática. Revista de Informática Educativa y Medios Audiovisuales, 2005, 2(5), 16-24.
4. **BARÓ, J. AND R. ALEMANY** Estadística II. Fundació per a la Universitat Oberta de Catalunya. Barcelona, 2000.
5. **BARR, A. AND E. FEIGENBAUM** The Handbook of Artificial Intelligence, vol. 1. Los Altos, CA: William Kaufmann, 1981.
6. **BRULÉ, J. AND A. BLUNT.** Knowledge acquisition: JF Brulé, A. Blunt. McGraw-Hill Publishing Co., New York (1989). xxvii+ 253 pp, ISBN 0-07-008600-1. In. New York: McGraw-Hill, 1989.
7. **BUCHANAN, B. G., D. BARSTOW, R. BECHTAL, J. BENNETT, et al.** Constructing an expert system. Building expert systems, 1983, 50, 127-167.
8. **CÁLAD, M. H. AND V. J. B. NAVARRO** *CommonKADS-RT: una metodología para el desarrollo de sistemas basados en el conocimiento de tiempo real*. Edtion ed.: Universidad Politécnica de Valencia, 2001.
9. **CONTRERAS-TORRES, E.** Procedimiento de extracción de rasgos 3D-proteicos basado en Álgebra Lineal: Aplicaciones en estudios bioinformáticos. Universidad Central “Marta Abreu” de Las Villas, 2016.
10. **CHAUDHARY, P., A. N. NAGANATHAN AND M. M. GROMIHA** Prediction of change in protein unfolding rates upon point mutations in two state proteins. Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics, 2016, 1864(9), 1104-1109.
11. **CHEN, K.-C., M. XU, W. J. WEDEMEYER AND H. RODER** Microsecond unfolding kinetics of sheep prion protein reveals an intermediate that correlates with susceptibility to classical scrapie. Biophysical journal, 2011, 101(5), 1221-1230.
12. **CHOU, K.-C.** A key driving force in determination of protein structural classes. Biochemical and biophysical research communications, 1999, 264(1), 216-224.
13. **CHOU, K.-C.** Some remarks on protein attribute prediction and pseudo amino acid composition. Journal of theoretical biology, 2011, 273(1), 236-247.
14. **CHOU, K.-C.** Impacts of bioinformatics to medicinal chemistry. Medicinal Chemistry, 2015, 11(3), 218-234.

15. **DI PAOLA, L., M. DE RUVO, P. PACI, D. SANTONI, et al.** Protein contact networks: an emerging paradigm in chemistry. Chemical reviews, 2012, 113(3), 1598-1613.
16. **DURKIN, J.** *Expert systems: design and development*. Edtion ed.: Prentice Hall PTR, 1998. ISBN 0023309709.
17. **ERIKSSON, L. J., JOANNA; P. WORTH, ANDREW; T.D. CRONIN, MARK; MCDOWELL, ROBERT M.; GRAMATICA,PAOLA** Methods for Reliability and Uncertainty Assessment and for Applicability Evaluations of Classification- and Regression-Based QSARs. Environmental Health Perspectives, 2003, 111, 1361-1375.
18. **ESTRADA, E.** A protein folding degree measure and its dependence on crystal packing, protein size, secondary structure, and domain structural class. Journal of chemical information and computer sciences, 2004, 44(4), 1238-1250.
19. **FLASIŃSKI, M.** Rule-Based Systems. In *Introduction to Artificial Intelligence.* Springer, 2016, p. 125-139.
20. **GHAFOURIAN, T. AND Z. AMIN** QSAR models for the prediction of plasma protein binding. Bioimpacts, 2013, 3(1), 21.
21. **GODDEN, J. W. AND J. BAJORATH** Chemical descriptors with distinct levels of information content and varying sensitivity to differences between selected compound databases identified by SE-DSE analysis. Journal of chemical information and computer sciences, 2002, 42(1), 87-93.
22. **GODDEN, J. W., F. L. STAHURA AND J. BAJORATH** Variability of molecular descriptors in compound databases revealed by Shannon entropy calculations. Journal of chemical information and computer sciences, 2000, 40(3), 796-800.
23. **GONZÁLEZ-DÍAZ, H., E. URIARTE AND R. R. DE ARMAS** Predicting stability of Arc repressor mutants with protein stochastic moments. Bioorganic & medicinal chemistry, 2005, 13(2), 323-331.
24. **GONZÁLEZ-DÍAZ, H., Y. PÉREZ-CASTILLO, G. PODDA AND E. URIARTE** Computational chemistry comparison of stable/nonstable protein mutants classification models based on 3D and topological indices. Journal of computational chemistry, 2007, 28(12), 1990-1995.
25. **GOSLING, J., B. JOY, G. STEELE JR, G. BRACHA, et al.** The Java R Language Specification–Java SE 8 Edition. Oracle America. In.: Inc, 2014.
26. **GRAMATICA, P., N. CHIRICO, E. PAPA, S. CASSANI, et al.** QSARINS: a new software for the development, analysis, and validation of QSAR MLR models. Journal of computational chemistry, 2013, 34(24), 2121-2132.
27. **GREENFIELD, N. J.** Analysis of the kinetics of folding of proteins and peptides using circular dichroism. NATURE PROTOCOLS, 2006, 1, 2891-2899.

28. **GROVER, M. D.** A Pragmatic Knowledge Acquisition Methodology. In *IJCAI.* Citeseer, 1983, vol. 83, p. 436-438.
29. **HAYAT, M., M. SOHAIL, H. KHAN AND M. NOMAN HAYAT** Identification of Outer Membrane Proteins Utilizing K-Nearest Neighbor. IJRCCT, 2016, 5(9), 485-489.
30. **HAYKIN, S. AND R. LIPPMANN** Neural networks, a comprehensive foundation. International journal of neural systems, 1998, 5(4), 363-364.
31. **JACKSON, P.** Introduction to expert systems 1998.
32. **KOLODNER, J.** *Case-based reasoning*. Edtion ed.: Morgan Kaufmann, 2014. ISBN 1483294498.
33. **LIAO, S.-H.** Expert system methodologies and applications-a decade review from 1995 to 2004. Expert systems with applications, 2005, 28(1), 93-103.
34. **LIAW, A. AND M. WIENER.** randomForest: Breiman and Cutler's Random Forests for Classification and Regression. Comprehensive R Archive Network. 2014. In., 2014.
35. **LÓPEZ SÁNCHEZ, J. I. AND L. E. CARRETERO DÍAZ** La inteligencia artificial y la ingeniería del conocimiento como soporte para las técnicas de decisión basadas en la gestión del conocimiento. Dirección y Organización, 2016, (23), 171-185.
36. **MARRERO-PONCE, Y., Ó. M. R. BORROTO, Y. H. DÍAZ, J. M. G. DE LA VEGA, et al.** Perspectiva general sobre el proceso de desarrollo de fármacos y las técnicas de cribado virtual basadas en la similitud molecular. In *Anales de la Real Academia Nacional de Farmacia.* 2013, vol. 79.
37. **MARRERO-PONCE, Y., E. CONTRERAS-TORRES, C. R. GARCÍA-JACAS, S. J. BARIGYE, et al.** Novel 3D bio-macromolecular bilinear descriptors for protein science: Predicting protein structural classes. Journal of theoretical biology, 2015, 374, 125-137.
38. **MARTINEZ, R. G., B. ROSSI AND P. BRITOS** METODOLOGIAS DE EDUCCIÓN DE CONOCIMIENTO PARA LA CONSTRUCCION DE SISTEMAS INFORMATICOS EXPERTOS 2005.
39. **MEDINA-FRANCO, J. L., E. FERNÁNDEZ-DE GORTARI AND J. J. NAVEJA** Avances en el diseño de fármacos asistido por computadora. Educación química, 2015, 26(3), 180-186.
40. **MOUHEB, D., M. DEBBABI, M. POURZANDI, L. WANG, et al.** Unified Modeling Language. In *Aspect-Oriented Security Hardening of UML Design Models.* Springer, 2015, p. 11-22.
41. **NETBEANS.** NetBeans [En línea]. https://netbeans.org. [2016.
42. **OECD.** GUIDANCE DOCUMENT ON THE VALIDATION OF (QUANTITATIVE) STRUCTURE-ACTIVITY RELATIONSHIP [(Q)SAR] MODELS. In.: OECD Environment Health and Safety Publications Series on Testing and Assessment No. 69, 2007.

43. **OUYANG, Z. AND J. LIANG** Predicting protein folding rates from geometric contact and amino acid sequence. Protein Science, 2008, 17(7), 1256-1263.
44. **PARADIGM, V.** Visual Paradigm. [En línea]. https://www.visual-paradigm.com. [2016.
45. **PEÑA SÁNCHEZ DE RIVERA, D.** Estadística, modelos y métodos. Volumen 2: Modelos lineales y series temporales, 1989.
46. **PRESSMAN , R. S.** *Ingeniería del Software Un enfoque Práctico* Edtion ed. Mexico, 2010.
47. **RANDIĆ, M., M. NOVIČ, A. R. CHOUDHURY AND D. PLAVŠIĆ** On graphical representation of trans-membrane proteins. SAR and QSAR in Environmental Research, 2012, 23(3-4), 327-343.
48. **RICH, E., K. RICH AND K. KNIGHT** *Inteligencia artificial*. Edtion ed.: McGraw-Hill, 2000. ISBN 8448118588.
49. **RUIZ-BLANCO, Y. B., Y. MARRERO-PONCE, P. J. PRIETO, J. SALGADO, et al.** A Hooke′ s law-based approach to protein folding rate. Journal of theoretical biology, 2015, 364, 407-417.
50. **RUSSELL, S. J., P. NORVIG, J. F. CANNY, J. M. MALIK, et al.** *Artificial intelligence: a modern approach*. Edtion ed.: Prentice hall Upper Saddle River, 2010.
51. **SHANNON, C. E.** A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1), 3-55.
52. **SOMMERVILLE, I.** Ingeniería del software. 9na. Edición. Madrid: Dearson, 2011.
53. **SRAVANI, T. D. AND K. S. VANI** Protein Fold Classification Using Sequence Features. International Journal, 2013, 3(6).
54. **STATSOFT,** I. STATISTICA (data analysis software system) version 8.0. http://www.statsoft.com. [Version for 8.0. 2008.
55. **STOLL, R.** Java Code Conventions 2014.
56. **SUKY S. , A. AND S. SELVAKUMAR** Protein Structural Class Prediction Using Feature Elicitation and Classification. International Journal of Innovative Research in Science, Engineering and Technology, 2014.
57. **TABARES, R. B.** Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación. Entre Ciencia e Ingeniería, 2011, (8), 161-173.
58. **TODESCHINI, R. AND V. CONSONNI** *Molecular descriptors for chemoinformatics, volume 41 (2 volume set)*. Edtion ed.: John Wiley & Sons, 2009. ISBN 3527628770.
59. **TODESCHINI, R., V. CONSONNI, A. MAURI AND M. PAVAN.** MOBYDIGS version 1.0. In.: Milano, 2005.
60. **TROPSHA, A.** Best practices for QSAR model development, validation, and exploitation. Molecular Informatics, 2010, 29(6-7), 476-488.

61. **TROPSHA, A., P. GRAMATICA AND V. K. GOMBAR** The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSPR models. Molecular Informatics, 2003, 22(1), 69-77.
62. **TURBAN, E.** *Decision Support and Expert Systems*. Edtion ed.: Prentice Hall PTR, 1995. ISBN 0024216631.
63. **URIAS, R. W. P., S. J. BARIGYE, Y. MARRERO-PONCE, C. R. GARCÍA-JACAS, et al.** IMMAN: free software for information theory-based chemometric analysis. Molecular diversity, 2015, 19(2), 305-319.
64. **VOGEL, L.** Unit Testing with JUnit Tutorial. In., 2013.
65. **WEHRENS, R., H. PUTTER AND L. M. BUYDENS** The bootstrap: a tutorial. Chemometrics and intelligent laboratory systems, 2000, 54(1), 35-52.
66. **WITTEN, I. H., E. FRANK, M. A. HALL AND C. J.** PAL *Data Mining: Practical machine learning tools and techniques*. Edtion ed.: Morgan Kaufmann, 2016. ISBN 0128043571.
67. **WITTEN, I. H., E. FRANK, L. E. TRIGG, M. A. HALL, et al.** Weka: Practical machine learning tools and techniques with Java implementations 2009.
68. **XU, J. AND A. HAGLER** Chemoinformatics and drug discovery. Molecules, 2002, 7(8), 566-600.
69. **ZIMMERMANN, H.-J.** *Fuzzy sets, decision making, and expert systems*. Edtion ed.: Springer Science & Business Media, 2012. ISBN 9400932499.

Anexos

Anexo 1. Estudio de viabilidad.

Tabla. 12 Estimación de la viabilidad del proyecto.

Categoría	Puntaje Total	Peso Total
Problema	661	71
Personal	423	54
Despliegue	330	39
Total	1414	164

$$\boldsymbol{Viabilidad\ del\ Proyecto} = \frac{1414}{164} = 8.62$$

Anexo 2. Parámetros estadísticos de los modelos de clasificación.

Modelos de clasificación de la Representación 3D-proteica Cα.

Tabla. 13 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica RF en la Rep. *Cα.*

Conjunto	Exactitud global Q(%)	Sensibilidad (%)		Especificidad (%)	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	96	**clases**		**clases**	
		α+β	92	α+β	98
		α/β	100	α/β	100
		All-β	92	All-β	98
		All-α	100	All-α	100

Tabla. 14 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica K-NN en la Rep. *Cα*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	91	**clases**		**clases**	
		α+β	58	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	98
		All-α	100	All-α	90

Tabla. 15 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica MLP en la Rep. *Cα*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	78	**clases**		**clases**	
		α+β	8	α+β	100
		α/β	100	α/β	92
		All-β	92	All-β	98
		All-α	100	All-α	80

Modelos de clasificación de la Representación 3D-proteica Cβ.

Tabla. 16 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica RF en la Rep. *Cβ.*

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	89	**clases**		**clases**	
		α+β	58	α+β	98
		α/β	100	α/β	100
		All-β	92	All-β	98
		All-α	100	All-α	90

Tabla. 17 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica K-NN en la Rep. *Cβ*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	89	**clases**		**clases**	
		α+β	58	α+β	98
		α/β	100	α/β	100
		All-β	100	All-β	98
		All-α	92	All-α	90

Tabla. 18 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica MLP en la Rep. *Cβ.*

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	99	**clases**		**clases**	
		α+β	97	α+β	100
		α/β	100	α/β	99
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	63	**clases**		**clases**	
		α+β	25	α+β	100
		α/β	100	α/β	71
		All-β	15	All-β	98
		All-α	100	All-α	80

Modelos de clasificación de la Representación 3D-proteica CEA.

Tabla. 19 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica K-NN en la Rep. *CEA.*

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	94	**clases**		**clases**	
		α+β	83	α+β	98
		α/β	94	α/β	100
		All-β	100	All-β	98
		All-α	100	All-α	98

Tabla. 20 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica MLP en la Rep. *CEA.*

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	87	**clases**		**clases**	
		α+β	42	α+β	100
		α/β	100	α/β	97
		All-β	100	All-β	93
		All-α	100	All-α	93

Modelos de clasificación de la Representación 3D-proteica AVG.

Tabla. 21 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica RF en la Rep. *AVG*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	87	**clases**		**clases**	
		α+β	50	α+β	98
		α/β	100	α/β	100
		All-β	100	All-β	95
		All-α	93	All-α	90

Tabla. 22 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica K-NN en la Rep. *AVG*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	100	**clases**		**clases**	
		α+β	100	α+β	100
		α/β	100	α/β	100
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	89	**clases**		**clases**	
		α+β	67	α+β	95
		α/β	100	α/β	100
		All-β	92	All-β	98
		All-α	93	All-α	93

Tabla. 23 Parámetros estadísticos del modelo de clasificación obtenido mediante la técnica MLP en la Rep. *AVG*.

Conjunto	**Exactitud global Q(%)**	**Sensibilidad (%)**		**Especificidad (%)**	
SE	99	**clases**		**clases**	
		α+β	97	α+β	100
		α/β	100	α/β	99
		All-β	100	All-β	100
		All-α	100	All-α	100
SP	73	**clases**		**clases**	
		α+β	83	α+β	100
		α/β	100	α/β	87
		All-β	69	All-β	100
		All-α	100	All-α	76

Anexo 3. Parámetros estadísticos de los modelos de regresión.

Tabla. 24 Parámetros estadísticos del mejor modelo de Regresión Lineal Múltiple por representación.

Rep.3D	Dim	R^2	Q^2_{loo}	Q^2_{boot}	a(Q^2)	Q^2_{ext}	$SDEP_{ext}$	Modelos
Cα	2	0.6875	0.6634	0.6641	-0.088	0.7199	1.581	**ln_kf**=-4.05863(±0.49196)*CA_I50_F_M32_SS-1_T_LGP[+12.0]_LGL[4-5.9]_KDS_MCoMPAs-0.09441(±0.01471)*CA_K_B_M32_MP-2_T_LGL[4-5.9]_MM-Z3_MCoMPAs+11.36218(±0.63655)
Cβ	2	0.7232	0.7016	0.7032	-0.084	0.4324	2.251	**ln_kf**=-9.54398(±2.30502)*CB_I50_Q_M32_SS-3_RAP_LGP[+12.0]_LGL[4-5.9]_PTT_MCoMPAs-0.01489(±0.0018)*CB_I50_B_M2_NS-1_FBS_KA_MM-KDS_MCoMPAs+12.77859(±0.67783)
CEA	3	0.7832	0.7612	0.7584	-0.107	0.7263	1.563	**ln_kf**=-0.12087(±0.01806)*AB_K_Q_M20_NS6_TYR_KA_Z1_MCoMPAs-00.00611(±0.00166)*AB_I50_B_M2_SS-6_RAP_LGL[6-8]_KDS-EPS_MCoMPAs +0.00001(±0)*AB_Q2_F_M2_NS2_T_LGL[6-8]_EPS_MCoMPAs+22.24081(±1.36658)
AVG	2	0.7041	0.6809	0.6808	-0.087	0.4729	2.169	**ln_kf**=-15.52436(±2.82005)*AVG_Q2_Q_M32_SS6_FAH_LGP[+12.0]_LGL[4-5.9]_PTT_MCoMPAs-0.02502(±0.00316)*AVG_I50_B_M5_NS-2_FBS_KA_MM-KDS_MCoMPAs+14.20601(±0.85033)

Rep. 3D: Representaciones 3D-proteicas

Dim.: Dimensión.

Anexo 4. Código de prueba *JUnit* para comprobar los métodos de la clase *CC_Prediccion.*

```
/**
 * Test of predictClass method, of class CC_Prediccion.
 */
@Test
public void testPredictClass() throws Exception {
    System.out.println("predictClass");
    CC_Prediccion instance = new CC_Prediccion();
    String[] expResult = {"α+β"};
    String[] result = instance.predictClass();
    assertArrayEquals(expResult, result);
}
```

Fig. 18 Código de prueba de prueba JUnit para comprobar el método *predictClass*.

```
/**
 * Test of PredictFoldRate method, of class CC_Prediccion.
 */
@Test
public void testPredictFoldRate() throws Exception {
    System.out.println("PredictFoldRate");
    CC_Prediccion instance = new CC_Prediccion();
    double[] expResult = {-0.92425157814004O7};
    double[] result = instance.predictFoldRate();
    assertDoubleEquals(expResult, result);
}
```

Fig. 19 Código de prueba JUnit para comprobar el método *predictFoldRate*.