ESTRATEGIA CO - EVOLUTIVA PARA LA IDENTIFICACIÓN DE MODELOS DE AUTÓMATA CELULAR EN TRAYECTORIAS DE SIMULACIÓN DE PLEGAMIENTO DE PROTEÍNA

Trabajo de Grado

Adriana Victoria Gómez Buitrón

Sayra Mildreth Ocoró Rosero

Director: MsC. Néstor Milciades Diaz Mariño

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo de I+D en Tecnologías de la Información

Modelado y Simulación de Sistemas Complejos

Popayán, Julio de 2012

TABLA DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

LISTA DE ACRONIMOS

2D: Dos dimensiones

3D: Tridimensional

µs: Microsegundo

AA: Aminoácido.

AC: Autómata Celular.

AG: Algoritmo Genético

Å: Angstrom

CAIF – PFT: Framework para la Identificación de Autómatas Celulares en Trayectorias de Plegamiento de Proteína.

C-α: Carbono alfa.

DM: Minería de Datos.

DMD: Dinámica Molecular Discreta.

FPR: Tasa de Falsos Positivos.

Fs: Femtosegundo.

ILAS: Aprendizaje Iterativo con Alternación de Estrato (Iterative Learning with Alternating Strata)

MCC: Coeficiente de Correlación de Matthews.

MD: Dinámica Molecular.

NMR: Espectroscopia por resonancia magnética nuclear.

Ns: nanosegundo

Ps: picosegundo.

Rc: Rango de contacto.

ROC: Receptor de Característica Operativa (Receiver Operating Characteristic)

SAA: Secuencia de aminoácidos.

TPR: Tasa de Verdaderos Positivos

FN: Falsos Negativos.

FP: Falsos Positivos.

VN: Verdaderos Negativos

VP: Verdaderos Positivos.

DCT: Tarea de Clasificación de Densidad.

IC: Configuraciones Iníciales.

1 INTRODUCCIÓN

El contexto que dio lugar a la propuesta descrita en el presente documento, se centra, por una parte, en el fenómeno de plegamiento de proteínas, que se define como el proceso mediante el cual una proteína alcanza su estructura terciaria o estado nativo.[1]. Este proceso resulta complejo, ya que la proteína, o secuencia de amino ácidos, debe adoptar una estructura tridimensional estable, partiendo desde su estructura primaria o no plegada.

Las técnicas de simulación, que constituyen otro de los ejes de la propuesta, permiten imitar el comportamiento de sistemas del mundo real. Para utilizarlas es necesario conocer el fenómeno o el sistema de tal manera que sea posible abstraer sus principales características y plasmarlas en un modelo. En el caso específico del fenómeno de plegamiento de proteína se presentan diversas limitaciones para la construcción de tales modelos, las cuales se derivan de la falta de entendimiento del proceso.

El enfoque más utilizado en la simulación de plegamiento de proteínas es la Dinámica Molecular [2]. Esta técnica de simulación *in silico* permite obtener trayectorias de plegamiento de proteínas, que no son más que un conjunto sucesivo de coordenadas 3D para cada uno de los átomos de una proteína. Los conceptos fundamentales de la física son la base de este enfoque; razón por la que resulta extremadamente demandante en tiempo de cómputo debido a la necesidad de manejar sistemas de nivel atómico con gran cantidad de partículas que interactúan entre sí.

Entre las técnicas de simulación que permiten modelar sistemas de alta complejidad como fenómeno de plegamiento de proteínas, están los Autómatas Celulares (ACs), que tienen una arquitectura más simple que la de otras técnicas de simulación de uso general, como la Dinámica de Sistemas y los Sistemas Multiagente. Adicionalmente, los ACs han tenido mayor desarrollo en cuanto a técnicas de aprendizaje de máquina para la identificación de modelos a partir de evidencia disponible para un fenómeno.

Con respecto a lo anterior, una de las técnicas de aprendizaje de máquina más utilizadas para la identificación de modelos de autómata celular, son los Algoritmos Genéticos (AG), los cuales han permitido hasta la fecha obtener modelos de AC que ofrecen una precisión de aproximadamente 90% [2], en la reproducción de una trayectoria de simulación de plegamiento de proteína.

A pesar de la precisión obtenida en trabajos previos [2], se hace necesario encontrar técnicas que permitan obtener modelos de AC que repliquen trayectorias de plegamiento de proteína con mayor precisión, ya que el porcentaje restante que no se está alcanzando, corresponde a características relevantes que no están siendo consideradas en el modelo. Es por esto que surge la idea de explorar las estrategias co – evolutivas ,

las cuales suelen presentar mejor rendimiento que los algoritmos genéticos tradicionales en tareas como la clasificación de densidad [3][4].

El objetivo de la propuesta se centra en la construcción de una técnica capaz de contribuir en la exploración de un camino hacia la obtención de un modelo simple, soportado en una representación discreta, que considere la dinámica del plegamiento de proteínas como sistema complejo. Este modelo simple se basará en el paradigma de los Autómatas Celulares (AC).

De manera similar a lo planteado en [3][4], aquí se plantea, que una estrategia co-evolutiva podría arrojar mejores resultados en la obtención de modelos de autómata celular que representen el fenómeno de plegamiento de proteínas. Es por eso que la pregunta que se formuló durante el planteamiento del anteproyecto, y que dio lugar al presente trabajo de investigación, fue ¿Cómo puede lograr una estrategia co-evolutiva mejorar los resultados obtenidos mediante Algoritmos Genéticos, para el problema de identificación de modelos de AC a partir trayectorias de plegamiento de proteínas?

Los resultados que se obtendrán, contribuirán en la búsqueda de la solución al problema del plegamiento de proteínas, ya que debido a su complejidad requiere de la utilización de diferentes enfoques que permitan avanzar en su entendimiento. Dichos resultados estuvieron sujetos al cumplimiento de los objetivos que se describen a continuación:

- En el objetivo general se planteó proponer una estrategia co - evolutiva que permita identificar modelos de autómata celular que reprodujeran con alta precisión (superior al 90%) trayectorias de simulación de plegamiento de proteínas. Para darle cumplimiento a éste objetivo, se satisficieron cada uno de los objetivos específicos, como se muestra a continuación.

- Al definir una estrategia co - evolutiva que permitió identificar el conjunto de reglas para modelos de autómata celular a partir de la información proporcionada por trayectorias de simulación de plegamiento de proteínas, se cumplió con el primer objetivo específico.

- Se implementó un prototipo de la estrategia co – evolutiva definida, con lo que se dio cumplimiento al segundo objetivo específico.

- Al evaluar la estrategia propuesta sobre un conjunto de datos de trayectorias de plegamiento de proteínas y constatar que permitía encontrar un modelo de AC que reproducía con alta precisión (superior al 90%) trayectorias de simulación de plegamiento de proteínas, se dio por cumplido el tercer objetivo específico, que en conjunto con los dos primeros contribuyeron a alcanzar el objetivo general del proyecto de investigación.

La siguiente parte del documento se estructura de la manera como se describe a continuación. El Capítulo Dos presenta el marco teórico y los trabajos previos relacionados con la propuesta de investigación. El Capítulo Tres, describe el proceso de

minería de datos que se siguió para el diseño de la solución al problema de investigación. En los Capítulos Cuatro y Cinco se describen los detalles correspondientes a la implementación de la estrategia diseñada y el análisis de los resultados obtenidos. En el capítulo final de sintetizan los aportes y conclusiones del trabajo de investigación, y adicionalmente, se presentan orientaciones para desarrollo de trabajos futuros.

2 MARCO CONCEPTUAL

2.1 CONTEXTO GENERAL

2.1.1 Autómatas Celulares (AC)

Los Autómatas celulares están ampliamente relacionados con conceptos como ecuaciones diferenciales, sistemas dinámicos y autómatas finitos[5]. Los AC son modelos matemáticos robustos que permiten representar sistemas dinámicos de forma más clara que los sistemas representados por ecuaciones diferenciales no lineales. Adicional a esto, el paradigma de autómatas celulares ofrece una arquitectura, compuesta de elementos muy simples, que permite representar de forma discreta fenómenos de la vida real; además, poseen la capacidad de representar de manera visual cada paso en la evolución del sistema, y permiten que el cambio de estado de cada celda pueda ser representado a manera de reglas de transición como en los autómatas finitos. Éste paradigma, es por lo tanto, una herramienta útil para predecir sucesos en sistemas de alta complejidad, sean éstos determinísticos o probabilísticos.

Los ACs se componen de cinco elementos, con los que se puede representar el comportamiento de un sistema: lattice (malla n – dimensional), conjunto de estados, reglas de evolución, condiciones de frontera y vecindad. Un AC está constituido por un espacio discreto n dimensional dividido en celdas generalmente de geometría regular; la cantidad de dimensiones de la malla o lattice determinan la complejidad de la representación. Cada una de las celdas en la malla, puede adoptar un estado, de un conjunto definido de posibles estados, en cada paso de tiempo. Cada celda tiene asociada una vecindad y puede pertenecer al vecindario de otras celdas. La vecindad de una celda se compone por el conjunto de celdas adyacentes, que ejercen influencia en los cambios de estado de ésta.

Los dos elementos restantes que conforman la arquitectura de un AC, son el conjunto de reglas y las condiciones de frontera. En cada paso de tiempo, el AC evoluciona según lo establecido en el conjunto de reglas, las cuales determinan la manera cómo interacciona cada celda con sus vecinas, y define por lo tanto el estado que tiene en cada paso de simulación.

Las condiciones de frontera, determinan lo que ocurre con las celdas que se ubican en los bordes de la malla. Las condiciones de frontera pueden variar según el fenómeno que se pretenda representar. Si se considera una frontera abierta, las celdas de los bordes tomarán valores fijos. Pero si el tipo de frontera que se va a considerar es reflectora, las celdas de los bordes tomarán valores de celdas dentro de la malla, como si se tratara de un espejo. En la frontera periódica, una celda ubicada en el borde, interactúa con sus vecinos inmediatos y con las celdas que están en el extremo opuesto de la malla. Con

condiciones de frontera infinita, la malla del AC no tiene límites, lo que es igual a decir, que no existe frontera [5].

Los AC se pueden clasificar en uniformes y no uniformes. En los AC uniformes tanto el tiempo, como el espacio son discretos, la actualización de celdas es sincronizada, y puede presentarse el paralelismo masivo. En general, los componentes básicos de los AC se caracterizan por la simplicidad, además los cálculos pueden llevarse a cabo en entornos distribuidos. Los AC no uniformes[6], ofrecen características adicionales, como un mayor rendimiento computacional, además, cada celda puede poseer sus propias reglas, y esto no implica un mayor consumo de recursos lo que compensaría hasta cierto punto, una de sus principales desventajas: el arduo trabajo que implica diseñarlo, ya que los espacios de búsqueda son más amplios que los de los ACs uniformes[6].

En lo referente al plegamiento de proteínas, hasta la fecha existen algunas propuestas de modelos de AC para representar fenómenos como el paso de ligando a través de la superficie de una proteína[7], y la predicción de estructura secundaria de una secuencia de aminoácidos[8]. Existe, además un modelo en 2D para la representación de colapso de polímeros, el cual se espera que a futuro, según lo planteado por sus autores, pueda ser aplicado en polímeros biológicos como las proteínas y el ADN[9].

2.1.2 Plegamiento de Proteínas

El plegamiento de proteínas es el proceso mediante el cual una proteína alcanza su estructura terciaria o estado nativo, en el cual está lista para realizar la función biológica para la que fue diseñada [10]. Los primeros aportes en plegamiento de proteínas fueron realizados por el científico estadounidense Anfinsen en el año de 1961, en donde éste ya planteaba que la “conformación nativa es determinada por la totalidad de interacciones atómicas y por lo tanto, por la secuencia de aminoácido.” [11].

En la actualidad, existen diversos enfoques utilizados para predecir el plegamiento de proteínas, uno de ellos son las técnicas in-vitro, donde se hace uso de la difracción de rayos X [1][12] y la NMR (Resonancia Magnética Nuclear)[1][13], para identificar las coordenadas en tercera dimensión de los átomos presentes en una Secuencia de Aminoácidos (SAA); éstas técnicas son experimentales, y resultan costosas en tiempo, equipos, recursos humanos y económicos [1].

Otro enfoque son las técnicas Ab-Initio, las cuales inician desde la estructura más básica de la proteína que es la misma SAA, en éste tipo de técnicas, se encuentran la simulación por Dinámica Molecular (DM) [14], y las técnicas de Monte Carlo (MC) [15]. La DM es considerada, en teoría, la técnica más precisa, con ella se exploran posibles configuraciones tridimensionales hasta encontrar la estructura más estable, para ello es necesario aplicar las ecuaciones de movimiento de la mecánica clásica a todos los átomos de la SAA, y por basarse en un enfoque de primeros principios, es decir en conceptos fundamentales de la física, resulta extremadamente demandante en tiempo de cómputo derivado de la necesidad de manejar sistemas de nivel atómico con gran cantidad de partículas que interactúan entre sí [1]. Las técnicas de MC pertenecen a un conjunto de técnicas estadísticas que se usan para encontrar la solución a diferentes

problemas, como los que incluyen la física de nivel atómico. En un concepto más general, fenómenos susceptibles de abstracción en términos de comportamientos estocásticos y muestreo aleatorio se pueden representar con las simulaciones de MC. Los por menores en el uso de simulaciones MC en plegamiento de proteínas se encuentran en el libro de Schlick[16] y de forma más práctica en [17].

Un tercer enfoque se basa en homologías, y consiste en buscar proteínas con estructuras 3D (nativa) conocidas, cuyas secuencias sean similares a la de la proteína de la que se desea conocer; éste enfoque computacional tiene sus bases en los siguientes axiomas [18]: 1) la estructura es más conservada que la secuencia, y 2) secuencias similares dan origen a estructuras similares. Se parte de una secuencia objetivo, y de un conjunto de estructuras y secuencias conocidas, llamadas plantillas. Se seleccionan aquellas plantillas que posean un mayor grado de similitud y se crea un modelo inicial que se aproxima a las características de la secuencia objetivo y que finalmente se refina, mediante funciones de energía. El principal inconveniente es que se hace necesario que exista una plantilla que sea idéntica a la secuencia objetivo, algo que no siempre es posible, limitando así la aplicación de éste enfoque para la predicción del plegamiento de un gran número de proteínas, sobre todo cuando no se conoce una proteína con una estructura similar previamente conocida.

Existe, además, un enfoque híbrido [19][20]que posee aspectos que son similares a los usados en el enfoque computacional por homologías y a los de DM.

2.1.3 Proteínas

Una proteína es una macromolécula formada por una secuencia de aminoácidos (SAA) unidos por enlaces peptídicos. Un aminoácido es una molécula que posee un carbón alfa, un átomo de hidrógeno, un grupo amino, un grupo carboxilo y una cadena lateral como se muestra en la Figura 1.

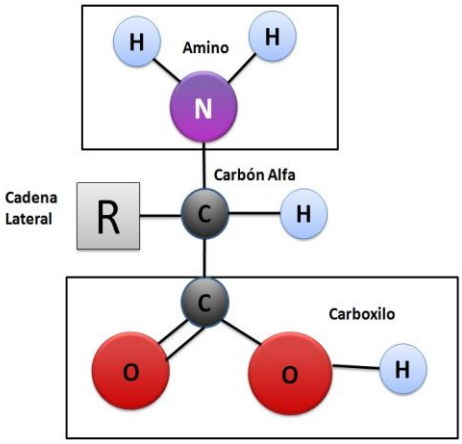

Figura 1. Estructura básica de un aminoácido.

La estructura primaria es la forma de organización más básica en una proteína. El conjunto de aminoácidos (AA) presentes en la cadena proteica, es muy importante cuando la proteína alcanza su forma tridimensional.[21] ya que de esto depende su desempeño. La estructura terciaria, es también conocida como el estado nativo de una proteína, y esta es la única con relevancia desde el punto de vista biológico.

2.1.4 Trayectorias de Simulación de Plegamiento de Proteína

Una trayectoria de simulación de plegamiento de proteína, es el conjunto de los estados por el cual pasa una proteína en cada paso de la simulación, iniciando desde una configuración no plegada, o estructura primaria, hasta su estructura nativa o hasta un punto de tiempo determinado.

Cada paso de simulación en una trayectoria contiene las coordenadas de cada átomo presente en el sistema de la proteína, más el solvente. Una trayectoria de plegamiento de proteína para el caso de esta investigación, se compone de un conjunto de modelos PDB [22].Un modelo PDB es una representación estándar para una estructura de datos macromoleculares como proteínas u otras moléculas y equivale a un paso de simulación. Los modelos PDB son derivados de estudios de difracción de rayos X, NMR (Resonancia Magnético Nuclear) y simulación por dinámica molecular discreta.

2.2 ANTECEDENTES

2.2.1 Co – Evolución

La co – evolución es un proceso dinámico, en el cual intervienen dos o más organismos o especies, ejerciendo presión de selección mutua y asíncrona entre si [23]; dicho de otra forma, los cambios evolutivos se producen de manera recíproca, pues las especies interactúan causando que la otra u otras se adapten.

Llevados a términos computacionales, los algoritmos co-evolutivos vendrían a ser una extensión de los algoritmos genéticos [3], donde una de las diferencias más marcadas es la forma de calcular qué tan apta es la solución encontrada, ya que se basa en la competencia entre individuos de la misma población, y de otras poblaciones.

Existen diferentes mecanismos o estrategias co - evolutivas, entre las que cabe mencionar la **Co - evolución Competitiva de una Población, Co - evolución Competitiva de Dos Poblaciones, Co - evolución Cooperativa de N Poblaciones** y **Niching.** La elección de la estrategia depende de la naturaleza del problema que se abordará, y de la forma como se aplicará la función de evaluación de los individuos [23].

La estrategia **co - evolución Competitiva de una Población,** es usada generalmente para implementar estrategias de competencia, y para optimizar soluciones candidatas.

El punto de partida de la estrategia **co - evolución competitiva de una población**, es un mal individuo, el cual es probado contra individuos no aptos, obteniendo así, inicialmente, un fitness igual a cero. Con respecto a la forma de estimar qué tan apto es un individuo,

existen dos alternativas en este tipo de mecanismo co - evolutivo: La primera consiste en probar un conjunto de individuos contra el conocimiento del experto, como en el caso de las aplicaciones en juegos de estrategia. La segunda manera, es probar un individuo contra otros de generaciones previas, ya que compararlos con los individuos de la generación actual, aunque es lo ideal, no es posible[23].

Otra estrategia es **Co - evolución Competitiva de Dos Poblaciones,** la cual consiste en dividir la población en dos sub – poblaciones: una población de soluciones y otra de pruebas. En la primera se mantienen los mejores individuos, mientras que en la segunda contiene casos de prueba, es decir, es una población colaborativa que cambia a los individuos de la población de soluciones forzándola a encontrar soluciones candidatas robustas.

Una tercera estrategia es **Co – evolución Cooperativa de N Poblaciones**, donde un problema se divide en n sub-problemas, posteriormente se halla la solución para cada sub-problema y se la evalúa. Al final, se integran las sub - soluciones y se determina el fitness de la solución general. Esta estrategia, es una manera eficaz de reducir la complejidad de grandes problemas.

El **Niching**[23]es una cuarta estrategia co – evolutiva con la cual se busca hallar similitudes entre individuos y localizar, diversificar y mantener alternativas de solución. Este mecanismo es utilizado para la optimización de técnicas de aprendizaje de máquina y en general para la optimización de algoritmos que involucran estructuras complejas no lineales como redes neuronales [24].

La elección de la estrategia co – evolutiva al igual que la construcción del test que se le aplicará a los individuos de las generaciones, depende en gran medida de la naturaleza del problema que se está abordando, así como del tipo de solución que se esté buscando. En cuanto a la valoración de los individuos, existen dos formas de hacerlo: una es la valoración del fitness relativo interno, y otra es la valoración del fitness externo.

El **Fitness Interno** es una medida usada para optimizar los individuos de la población a través de la selección. Mientras que el **Fitness Externo** es una medida que se utiliza para examinar la calidad de un individuo bajo las condiciones del contexto real, por lo que en última instancia, mide el progreso del algoritmo.

2.2.2 Diseño Inverso de Autómatas Celulares

El paradigma de AC es una herramienta de simulación poderosa, considerando la simpleza de su arquitectura. Es ideal para representar sistemas y fenómenos complejos. Sin embargo, cuando no se conoce a fondo el fenómeno a representar surge la necesidad de diseñar de manera inversa el modelo de AC, lo que consiste en tomar como base el comportamiento del sistema que se quiere representar, y utilizar técnicas de aprendizaje de máquina para identificar uno o más elementos del AC y así obtener un modelo completo[25].

El concepto de diseño inverso de modelos de AC que se considerará, retoma lo planteado por Bäck y Breukelaar en [26]. Luego, deberá entenderse como la obtención de la vecindad y de las reglas de transición para un modelo de AC, a partir de configuraciones globales conocidas. En un sentido estricto la definición de diseño inverso aborda dos problemas complejos, por una parte la determinación de la vecindad y por la otra la identificación de las reglas, los cuales serán abordados en este trabajo.

En la mayoría de las investigaciones que corresponden al diseño inverso automatizado de AC [27][28][29][30][31], los rasgos más comunes encontrados se centran en alfabetos reducidos, que son los que representan los distintos estados posibles en cada celda del autómata celular, y el uso de mallas de una y dos dimensiones [32]. Las técnicas comúnmente utilizadas para establecer los parámetros del AC, son tomadas de la Lógica Difusa, los Árboles de Clasificación y de Decisión, la Computación Evolutiva, y algunas técnicas de Minería de Datos.

En [26] se muestra un trabajo investigativo en el que se utilizan AC para manejar el problema de plegamiento de proteínas. Dos modelos de AC son elaborados con fines diferentes. El primero tiene como objetivo encontrar subestructuras, utilizando un AC de una dimensión con frontera cíclica, un grupo de reglas con una vecindad de radio tres, y el conjunto de estados posibles se refieren a las direcciones izquierda, derecha, arriba y abajo del aminoácido. El segundo modelo incluye una representación simplificada para los veinte aminoácidos, asociándolo a hidrofóbico o polar, además de una dirección de manera análoga al caso del primer modelo. Mediante algoritmos genéticos se identificaron los parámetros de los modelos. La predicción de la estructura proteica no fue tan importante en los resultados de estas investigaciones, como lo fue el éxito de hallar un modelo de AC por medio de algoritmos genéticos, con lo cual, dicho trabajo se convierte en un excelente antecedente que muestra que es posible utilizar el diseño inverso para encontrar representaciones de un fenómeno complejo relacionado con el plegamiento de proteínas.

2.2.3 Diseño Inverso de Autómatas Celulares con Algoritmos Co – Evolutivos

La tarea de clasificación de densidad, o DCT (por sus siglas en inglés), es un problema donde debe responderse a la pregunta de si hay más ceros o unos en una cadena de bits de longitud impar dada [3][4][26]. Utilizando algoritmos co - evolutivos en identificación de modelos de ACs que solucionen dicho problema, los resultados obtenidos han sido significativamente mejores que los alcanzados con los algoritmos evolutivos tradicionales [4]. Con algoritmos co-evolutivos es posible obtener ACs donde cada celda puede contener una regla diferente, de manera que el fitness es asignado por celda, y no por AC, por lo tanto cada celda podrá mantener su propio genotipo, es decir, la representación de la información del individuo [33].

Los modelos co-evolutivos, encuentran soluciones que usan estrategias de alta calidad, en comparación con los modelos evolutivos [4]. En [6] se documenta la experiencia obtenida al llevar a cabo experimentos con ACs unidimensionales de 149 celdas, realizando variaciones en el tamaño del radio de la vecindad, con el objetivo de solucionar el problema de clasificación de densidad. En este caso se utilizaron AG para encontrar las

reglas que permitieran solucionar el problema, pero el fitness de las reglas encontradas con este tipo de algoritmos, nunca superó el 75% de precisión, mientras que el puntaje máximo para el fitness de una regla encontrada con un algoritmo co-evolutivo, fue del 92%.

En[34], se define un framework para búsqueda co-evolutiva, y uno de los principales objetivos es descubrir el mejor ambiente de entrenamiento de individuos aprendices. Se resuelve la tarea de clasificación de densidad con un AC unidimensional de 149 celdas, cada una con 2 posibles estados y una vecindad de radio 3. Para obtener las reglas que permitieron solucionar el problema, se implementó una relación competitiva entre reglas y configuraciones iníciales (IC), alcanzando reglas con una precisión de hasta el 80%.

En [4] se utilizó un AC con un lattice unidimensional de 149 celdas para resolver el problema de la clasificación de densidad. Para obtener el conjunto de reglas que permitieron resolver éste problema, se aplicaron estrategias evolutivas y co-evolutivas. Para el primer caso, las ICs evolucionaban, mientras que en el segundo caso, evolucionaban los modelos de AC, y las ICs eran establecidas con cada generación de acuerdo con un procedimiento preestablecido. Con AG es posible obtener tres tipos de estrategias de clasificación:

- Estrategia por defecto, clasifica todas las cadenas de bits en una clase de densidad 0 ó 1.

- Estrategia partícula, usa la interacción entre patrones de meso-escala, es decir entre conjuntos de configuraciones locales, para clasificar cadenas de bits.

- Estrategia expansión de bloque, es una mejora a la estrategia por defecto, las cadenas de bits son clasificadas en una clase de densidad, la cual es 0 a menos que exista en la cadena un bloque suficientemente grande de 1s, en este caso el bloque se expande hasta cubrir todo el lattice.

De acuerdo a lo descrito, en [4] la eficacia de los modelos estaba determinada por la cantidad de estrategias partícula obtenidas y en los resultados se observó que los modelos co-evolutivos tienen una mayor capacidad adaptativa, ya que podían hacer transiciones entre las estrategias de expansión de bloque y partícula, mientras que los modelos evolutivos no.

En los algoritmos co – evolutivos se consideran los mismos parámetros que en los AG, los cuales deben afinarse evitando siempre que la técnica co – evolutiva desarrollada exhiba comportamientos indeseables, como el desacople, la dinámica cíclica y la sobre – especialización [35], que impiden encontrar buenas soluciones, ya que se pierden características obtenidas en generaciones anteriores.

2.2.4 Elección del Lenguaje de Programación

Para resolver el problema que se plantea en éste proyecto, es necesario implementar rutinas que permitan probar que la estrategia planteada cumple con los objetivos

establecidos. Por lo tanto, se debe seleccionar un lenguaje de programación que se ajuste a las necesidades del proyecto.

Se conoce por los antecedentes de simulación de plegamiento de proteínas, que el tiempo necesario para llevar a cabo esta tarea es considerablemente grande como para asignársela a un único equipo de cómputo, por lo que implementar un algoritmo distribuido que aproveche la capacidad de cálculo de múltiples equipos y agilice la ejecución de la tarea, sería lo ideal. De acuerdo a esto, se requiere que el algoritmo implementado se ejecute en diferentes máquinas, por lo tanto sería ideal utilizar un lenguaje que funcione en cualquier plataforma, como lo hacen los lenguajes de scripting.

Python es un lenguaje de programación multiparadigma de alto nivel [36], que permite obtener programas más compactos. Al ser un lenguaje interpretado, Python es menos eficiente en cuanto a tiempo de ejecución, pero es multiplataforma, por lo que funciona sobre cualquier arquitectura. Por lo tanto, es el lenguaje de programación seleccionado para este proyecto, puesto que hace posible explotar los recursos computacionales de diferentes máquinas.

3 METODOLOGÍA DE MINERÍA DE DATOS PARA LA IDENTIFICACIÓN DE MODELOS DE AC EN TRAYECTORIAS DE PLEGAMIENTO DE PROTEÍNA

Debido a que el proceso de identificar modelos de AC que permita minar las evidencias disponibles sobre el fenómeno de plegamiento de proteínas, se asemeja a un proceso de minería de datos, la metodología utilizada para alcanzar los objetivos propuestos en este proyecto es CRIPS-DM (Cross Industry Standard Process for Data Mining) [37].

En esta sección se muestra una descripción de la metodología en un alto nivel, como lo indica la Figura 2, donde se muestran las 6 fases del modelo de proceso de CRIPS-DM. La primera fase, Entendimiento del Negocio (Business Understanding) .Fase dos, de Entendimiento de los Datos (Data Understanding). La tercera fase, Preparación de los datos (Data Preparation).Fase cuatro, Modelado (Modeling). Quinta fase Evaluación (Evaluation). Sexta fase, Despliegue (Deployment).La Guía de Usuario de CRISP-DM[38], es el documento que se tomó como base para la utilización de la metodología mencionada en este proyecto.

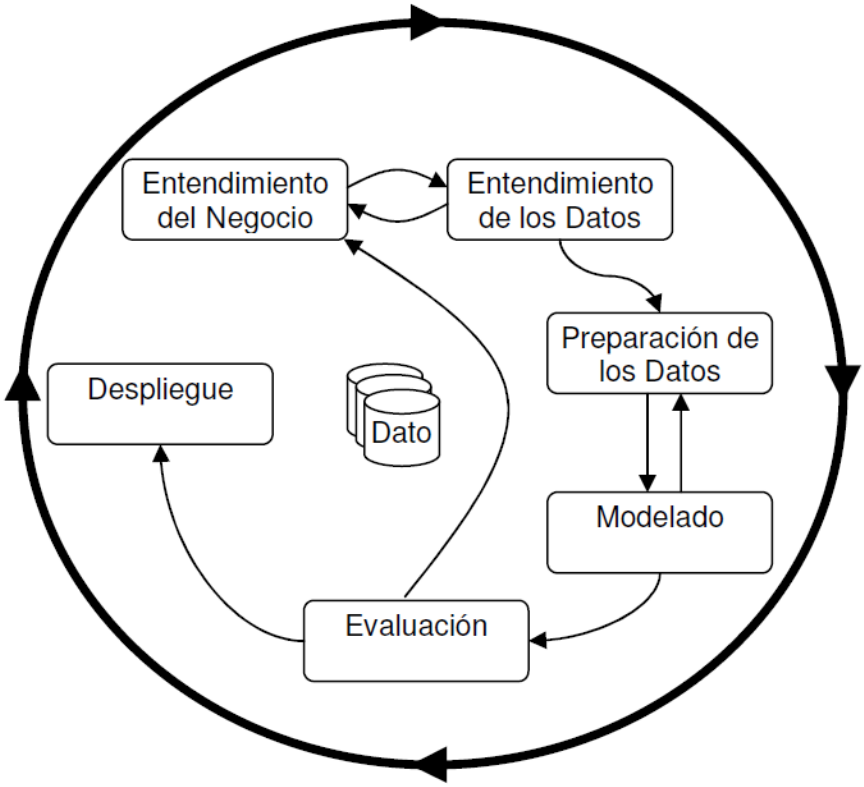

Figura 2. Fases del Modelo de CRIPS-DM. (Tomado de **[2]**)

3.1 FASE 1: ENTENDIMIENTO DEL NEGOCIO

En esta sección se muestran los objetivos del proyecto desde el punto de vista del negocio, para después pasar a la definición del problema en términos de minería de datos y generar un plan inicial diseñado para alcanzar los objetivos.

3.1.1 Determinar los Objetivos del Negocio

El objetivo del negocio se encuentra relacionado con el fenómeno de plegamiento de proteínas y modelos de simulación. Por lo tanto, el objetivo a lograr con la aplicación del proceso de minería de datos, es la obtención de un modelo de AC, que replique, con una calidad aceptable, una trayectoria simulada de plegamiento de una proteína.

3.1.1.1 Background

Los modelos de AC son eficientes computacionalmente. Adicional a esto, comparados con otras estrategias de simulación, resulta más sencillo obtener modelos de AC para representar sistemas con comportamiento complejo, como el fenómeno de plegamiento de proteínas, utilizando técnicas de aprendizaje de máquina [39].

Las trayectorias obtenidas mediante simulaciones de dinámica molecular, están compuesta por instantáneas de cada paso de simulación; donde cada instantánea proporciona información sobre algunas características físicas, además de la ubicación espacial de cada átomo involucrado en el sistema proteína – solvente. Estas trayectorias junto con una técnica de aprendizaje de máquina serán utilizadas para obtener un modelo de AC que represente el fenómeno de plegamiento de proteínas.

3.1.1.2 Objetivo del Negocio

Obtener un modelo de simulación basado en Autómatas Celulares, que replique una trayectoria de simulación de plegamiento de una proteína con una precisión superior al 90%.

3.1.1.3 Criterios de Éxito del Negocio

El modelo de Autómata Celular obtenido, replica de forma aproximada la trayectoria de simulación de plegamiento de proteínas.

El conjunto de reglas de AC obtenido, es legible y es posible extraer de él conocimiento, que permita explicar el comportamiento del fenómeno.

3.1.2 Evaluación de la Situación Actual

En esta etapa se realiza una investigación más detallada sobre todos los recursos, restricciones, presunciones, y otros factores que deberían ser considerados en la determinación del objetivo de análisis de datos y el plan de proyecto.

3.1.2.1 Inventario de Recursos

Los recursos computacionales para analizar el volumen de datos se convierte en factor de consideración, debido a la cantidad de información derivada de la simulación de plegamiento de proteínas. Con respecto a este factor se utilizaran veintitrés (23) computadores de escritorio con procesador Intel Core 2 Duo de 2.0 GHz de frecuencia y 2 GB de memoria RAM y un equipo de cómputo con procesador Phenom II X6 de 2.8 GHz de frecuencia y 4 GB de memoria RAM.

La generación de trayectorias se descarta por el considerable tiempo de CPU requerido para lograr una trayectoria de plegamiento razonablemente completa. Además, los repositorios públicos de datos de trayectorias de plegamiento de proteína son escasos. Por estas razones, solo se consideran las trayectorias de simulación provistas por el repositorio público SimTK[40] y disponibles en el sitio Web http://www.simtk.org.

Se usarán trayectorias no muy extensas, porque para trabajar con trayectorias que consideran demasiados pasos de simulación, se necesitaría unos equipos de computo con características superiores a los anteriormente descritos. Las coordenadas y características físicas de los átomos de las proteínas que se utilizaran estarán en formato PDB [22].

En cuanto a herramientas software para minería de datos para la evaluación del modelo, se considera el uso de Orange[41], que es una herramienta que implementa algoritmos de minería de datos en Python, haciendo posible su integración con las rutinas propias de la estrategia defina. Se utilizaran puntualmente los árboles de decisión aplicados al análisis del conjunto de reglas de los mejores modelos de autómata celular encontrados.

Debido a que el lenguaje de programación seleccionado para implementar la estrategia es Python, para ejecutar la estrategia en un entorno distribuido, se utilizará PYRO (Python Remote Object)[42] que maneja objetos remotos de Python, y se apoyará la implementación de la lógica del algoritmo con CAIF – PFT (Framework para la identificación de modelos de autómata celular en trayectorias de plegamiento de proteínas)[2].

3.1.2.2 Requerimientos, Asunciones y Restricciones

- Al finalizar el proyecto se espera obtener un modelo de autómata celular que simule trayectorias de plegamiento de proteínas y las reproduzca con una precisión superior al 90%.

- La información de las trayectorias que se empleará en el proyecto, está disponible en un repositorio público de datos SimTK[40], por cual no se considerarán asuntos de seguridad, además, el producto software que se espera obtener, será ejecutado en un entorno distribuido con acceso restringido.

- Simular el plegamiento de una proteína, es una actividad que consume una cantidad de tiempo considerablemente grande ($1,2 * 10^{17}$ segundos aproximadamente) [43], es por eso que asumir que las trayectorias de simulación de las que se dispone abarcan todo el proceso, desde la estructura primaria hasta la estructura nativa de la proteína, es poco razonable. Sin embargo, para el desarrollo del proyecto se asumirá que la información contenida en estas trayectorias parciales, involucra suficiente detalle del proceso de plegamiento.

- La cantidad y el tamaño de los datos que se obtienen de una trayectoria de simulación limitan de dos maneras el proyecto. Por una parte, si se tienen demasiados datos, son pocos los experimentos que se pueden realizar, ya que se requeriría mucho tiempo para procesarlos. En segunda instancia, no es posible construir un predictor, ya que la cantidad de datos de la que se dispone no es suficiente.

- Se cuenta con 24 equipos de cómputo, dispuestos en una red de 1 Gbps, para llevar a cabo la simulación, además el tiempo de uso de éstos equipos es limitado, pues sólo se puede acceder a ellos en horarios en los que no se realicen labores académicas.

3.1.2.3 Riesgos y Contingencias

En esta sección de la metodología se consideraron los posibles riesgos que podrían afectar el desarrollo del proyecto. La tabla de riesgos y contingencias, se puede apreciar en el capítulo 1 de los anexos, en la sección 1.1.

3.1.2.4 COSTOS Y BENEFICIOS

Los beneficios que se obtendrán al finalizar el proyecto, apoyarán el entendimiento de las reglas que determinan el fenómeno de plegamiento de proteínas, ya que con él se abre camino a trabajos futuros para la obtención de modelos más precisos. Esto se lograría a un costo considerablemente menor, en cuanto a esfuerzo, recursos humanos y recursos computacionales, comparado con otras alternativas de solución disponibles en el momento. La tabla de costos se puede apreciar con mayor detalle en capítulo 1 de los anexos, en la sección 1.2.

3.1.3 Determinar Objetivos de la Minería de Datos

3.1.3.1 Objetivos de Minería de Datos

Identificar un modelo de Autómata Celular que clasifique el estado de cada una de las celdas del lattice del autómata para el siguiente paso de tiempo, de acuerdo a la configuración actual de la vecindad de la celda.

3.1.3.2 Criterios de Éxito de la Minería De Datos

Se compararán los resultados obtenidos con la aplicación de las reglas encontradas con la estrategia co – evolutiva frente al conjunto de datos de trayectorias de plegamiento de proteínas disponibles. Las medidas que se considerarán para contrastar los resultados, serán SENSITIVIDAD[1], PRECISIÓN[2], COEFICIENTE DE CORRELACIÓN DE MATTHEWS[3], ESPECIFICIDAD[4], y ROC[5]. En cuanto a la precisión, deberá ser igual a superior a 0.9 para que se pueda afirmar que se alcanzó exitosamente el objetivo de minería de datos, y que la estrategia co – evolutiva que se utilizó en el diseño inverso del AC que reproduce trayectorias de plegamiento de proteínas es buena.

3.1.4 Construir Plan del Proyecto

3.1.4.1 Plan Del Proyecto

Hasta el momento, se cuenta con la información, que aunque limitada, es suficiente para alcanzar el objetivo planteado, además se poseen las herramientas software y hardware necesarias que exige la ejecución del proyecto.

---

1 Sensitividad: Mide la proporción de estados positivos correctamente actualizados.

2 Precisión (Accuracy): Indica la proporción de estados de AC correctamente actualizados; es sensible a sesgos de clase, luego es una medida útil para determinar que tan correctamente son actualizados los estados positivos y negativos. Al igual que la sensitividad, puede variar en el rango de 0 a 1.

3 Coeficiente de Correlación de Matthews, puede variar de -1 a 1, los valores cercanos a 1 indican buenas predicciones, luego los verdaderos positivos y los verdaderos negativos, son clasificados de manera satisfactoria. No se ve afectado por los sesgos de muestreo.

4 Especificidad: Es la probabilidad de predecir correctamente un negativo.

5 ROC: Receiver Operating Characteristics, facilita la evaluación de resultados ubicándolos en un espacio que permite identificar con claridad los mejores modelos.

Para el desarrollo del proyecto, se adoptarán dos metodologías, una metodología para desarrollo, “eXtreme Programming” [16], de la cual se van a tener en consideración la elaboración de algunos artefactos como: Historias de usuario y Pruebas de unidad. Por otra parte, y teniendo presente el objetivo principal de la propuesta, se ha elegido la metodología CRISP-DM [37], debido a su genericidad y adaptabilidad a las necesidades del problema.

La aplicación de la metodología de desarrollo XP se hará en doce iteraciones, en las cuales se encapsularán las actividades de cada fase de CRISP - DM: El diagrama de Gantt del cronograma del proyecto se puede ver en el capítulo 1 de los anexos, en la sección 1.3).

3.1.4.2 Evaluación Inicial de Técnicas y Herramientas

Las herramientas y técnicas que se consideraron en esta parte de la metodología, fueron los algoritmos de Minería de Datos para efectuar tareas de clasificación, llegando a la conclusión, que no existe una técnica de Minería de Datos que permita minar modelos de autómata celular, luego, se hace necesario explorar técnicas que faciliten la obtención de modelos de AC a partir de la información que se tiene de las trayectorias de plegamiento de proteínas.

Deben evaluarse herramientas y técnicas en el campo del diseño inverso de modelos de AC que resulten más apropiadas para el desarrollo de éste proyecto. De las técnicas existentes para abordar el diseño inverso de modelos de AC, se descartaron aquellas que presentan restricciones en cuanto a la cantidad de configuraciones globales (evidencias) utilizadas en el proceso. La mayor parte de las técnicas que cumplen con esta característica, fundamentan el proceso de minería de datos en considerar solo dos configuraciones globales del modelo de AC, una configuración inicial y una final deseada.

Se descartaron las técnicas que no involucran la identificación de la vecindad como uno de sus procesos, es decir que utilizan una vecindad predefinida, dado que para el tipo de modelos que se buscará identificar, se desconoce la vecindad a utilizar y por lo tanto es parte de los objetivos a cumplir.

Las técnicas que resultan más viables para llevar a una buena terminación de los objetivos del proyecto de minería de datos, son aquellas que simultáneamente, involucran la identificación de la vecindad y las reglas del modelo de AC, y que adicionalmente soportan mallas multidimensionales en 2D o 3D.

Las técnicas que específicamente se consideraron son las técnicas co – evolutivas aplicadas a la identificación de modelos de AC que resuelven el problema de la mayoría (i.e. [34],[44] y[45]), ya que constituyen una alternativa a tener en cuenta en el momento de mejorar la búsqueda asociada a técnicas basadas en algoritmos genéticos.

3.2 FASE 2: ENTENDIMIENTO DE LOS DATOS

3.2.1 Reporte Inicial de la Recolección de Datos

En el repositorio Simtk se encuentra disponible la trayectoria de simulación de la proteína HP35, llamada “Villin Headpiece”. Esta proteína es considerada como el poli péptido natural más pequeño que se pliega autónomamente en una estructura globular [47].

Actualmente, en el repositorio de Simtk se encuentran las trayectorias de una variante de Villin Headpiece, que es un subdominio de la proteína, llamado HP-35 Nle Nle [43]. Los archivos que se utilizará en este proyecto son de acceso público y se encuentran en formato PDB.

En la Figura 3 se muestra la estructura básica de un Modelo PDB, el cual contiene el nombre del átomo (ATOM NAME), nombre de la cadena lateral de cada AA presente en la SAA (RES NAME), además de las coordenadas tridimensionales (X, Y, Z) de cada carbono alfa, factor de ocupación y la temperatura de vibración de cada átomo.

```
ATOM      1  N   NLEU    1      29.810  32.380  19.260  1.00  0.00           N
ATOM      2  CA  NLEU    1      29.560  33.160  17.990  1.00  0.00           C
ATOM      3  C   NLEU    1      28.630  32.360  17.050  1.00  0.00           C
ATOM      4  O   NLEU    1      27.620  32.990  16.680  1.00  0.00           O
ATOM      5  CB  NLEU    1      30.870  33.460  17.320  1.00  0.00           C
ATOM      6  CG  NLEU    1      30.900  34.820  16.640  1.00  0.00           C
ATOM      7  CD1 NLEU    1      29.920  34.840  15.440  1.00  0.00           C
ATOM      8  CD2 NLEU    1      30.920  35.960  17.660  1.00  0.00           C
ATOM      9  HA  NLEU    1      29.130  34.130  18.240  1.00  0.00           H
ATOM     10  HB1 NLEU    1      31.110  32.720  16.560  1.00  0.00           H
ATOM     11  HB2 NLEU    1      31.590  33.520  18.130  1.00  0.00           H
ATOM     12  HG  NLEU    1      31.870  34.780  16.150  1.00  0.00           H
ATOM     13 HD11 NLEU    1      30.200  33.940  14.900  1.00  0.00           H
ATOM     14 HD12 NLEU    1      28.890  34.880  15.810  1.00  0.00           H
ATOM     15 HD13 NLEU    1      30.140  35.690  14.800  1.00  0.00           H
ATOM     16 HD21 NLEU    1      30.030  35.950  18.290  1.00  0.00           H
ATOM     17 HD22 NLEU    1      31.870  36.020  18.190  1.00  0.00           H
ATOM     18 HD23 NLEU    1      30.880  36.830  17.010  1.00  0.00           H
ATOM     19  H1  NLEU    1      30.230  31.490  19.040  1.00  0.00           H
ATOM     20  H2  NLEU    1      30.590  32.790  19.760  1.00  0.00           H
ATOM     21  H3  NLEU    1      29.010  32.110  19.810  1.00  0.00           H
ATOM     22  N   SER     2      29.120  31.260  16.440  1.00  0.00           N
ATOM     23  CA  SER     2      28.080  30.310  15.700  1.00  0.00           C
ATOM     24  C   SER     2      27.630  29.090  16.560  1.00  0.00           C
ATOM     25  O   SER     2      26.490  28.600  16.430  1.00  0.00           O
ATOM     26  CB  SER     2      28.860  29.860  14.500  1.00  0.00           C
ATOM     27  OG  SER     2      29.070  30.920  13.580  1.00  0.00           O
ATOM     28  H   SER     2      30.030  30.930  16.710  1.00  0.00           H
ATOM     29  HA  SER     2      27.260  30.870  15.250  1.00  0.00           H
ATOM     30  HB1 SER     2      28.560  28.930  14.010  1.00  0.00           H
ATOM     31  HB2 SER     2      29.830  29.750  14.990  1.00  0.00           H
ATOM     32  HG  SER     2      29.900  31.350  13.810  1.00  0.00           H
ATOM     33  N   ASP     3      28.620  28.580  17.290  1.00  0.00           N
ATOM     34  CA  ASP     3      28.570  27.250  18.070  1.00  0.00           C
```

Figura 3: Esquema del Modelo PDB.

3.2.1.1 Descripción y exploración de los Datos

Los datos de partida para el proyecto provienen de un experimento realizado en [43], llevado a cabo con 9 conformaciones no plegadas desnaturalizadas a 373°K, a partir de las cuales se obtuvieron trayectorias de plegamiento después de simular el proceso con dinámica molecular.

Cada trayectoria de simulación esta almacenada en formato PDB. Una trayectoria está dividida en 100 marcos de simulación. Cada marco agrupa 401 pasos de simulación, ya que el último paso de simulación, corresponde al primer paso del siguiente marco. El tiempo transcurrido entre cada paso de simulación es de 50 pico segundos (ps).

3.2.1.2 Verificar la Calidad de los Datos

Se implementó una rutina en Python para realizar un conteo sobre la cantidad de modelos PDB presentes en la trayectoria de simulación. Esta rutina también contaba la cantidad de carbones alfa presentes en cada modelo. A partir de los resultados de los conteos, se llegó a la conclusión que los datos de los que se estaba partiendo eran aptos para ser usados en la etapa de pre – procesamiento ya cumplían con las condiciones de completitud esperadas.

Parte del reporte de Calidad de los datos se muestra en la Figura 4 donde se puede observar que existían 35 carbones alfa en cada uno de los 401 modelos presentes en cada uno de los 100 frames.

```
D:\tesis\Iteracion2\RutinasPreprocesamiento\CAlpha/frame00.pdb.gz

modelo 0 tiene 35
modelo 1 tiene 35
modelo 2 tiene 35
modelo 3 tiene 35
.
.
.
modelo 397 tiene 35
modelo 398 tiene 35
modelo 399 tiene 35
modelo 400 tiene 35

El numero de modelos  es :401
.
.
.

D:\tesis\Iteracion2\RutinasPreprocesamiento\CAlpha/frame99.pdb.gz

modelo 0 tiene 35
modelo 1 tiene 35
.
.
.
```

Figura 4: Reporte de calidad de los datos.

Una vez confirmada la completitud de los datos, se utilizó la herramienta VMD (Visual Molecular Dynamics) [46], que es un programa de visualización molecular que sirve para mostrar, animar y analizar sistemas biomoleculares usando gráficos 3-D. Esta herramienta toma como insumos modelos PDB, los cuales deben estar correctamente estructurados. Partiendo de esta premisa, se pudo constatar que los datos que se utilizarían cumplían con el estándar de los formatos PDB, ya que fue posible utilizarlos para generar las imágenes correspondientes a la SAA.

En las Figuras 5 y 6 se muestran imágenes de las configuraciones iníciales, se dos diferentes ejecuciones de la proteína seleccionada. Tales imágenes fueron generadas a partir de los modelos PDB, utilizando la herramienta VMD.

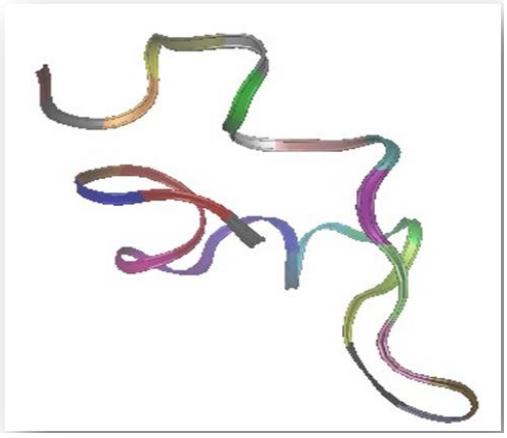

Figura 5: Conformación inicial de HP- 35 Nle Nle, Run 04.

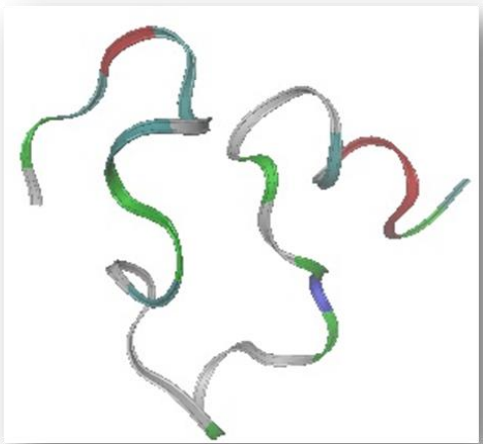

Figura 6: Conformación inicial de HP- 35 Nle Nle, Run 07

3.3 FASE 3: PREPARACIÓN DE LOS DATOS

3.3.1 Selección de Datos

El tamaño del sistema de simulación utilizado para generar las trayectorias, es de aproximadamente 9.700 átomos, entre los que se incluyen los 576 átomos pertenecientes a la SAA y los 9.108 átomos de las 3.036 moléculas de agua (el solvente). El efecto del solvente será considerado como implícito, por lo tanto, la cantidad de datos a analizar se reduce a los 576 átomos de la SAA.

En un modelo PDB se incluyen campos como los datos de identificación, ubicación espacial de los átomos de la SAA, además el factor de ocupación y la temperatura de vibración de cada átomo.

Es necesario eliminar los átomos que no hacen parte de la SAA y excluir las columnas de datos correspondientes al factor de ocupación y de temperatura de vibración debido a su comportamiento constante, ya que toman valores de 1.0 y 0.0 respectivamente. Además, se deben organizar los datos en archivos separados, uno por cada modelo, con el objetivo de facilitar los procesos construcción y formateo de datos.

3.3.2 Limpieza de Datos

Como resultado de la limpieza de datos, un frame de la trayectoria de simulación que se compone por 40.100 pasos o modelos PDB, se transforma en 40.100 archivos, donde cada uno contiene un único paso de simulación. Adicionalmente, se eliminó el último modelo de cada frame, ya que corresponde al primer modelo del siguiente, quedando en total 100 frames con 400 pasos de simulación o modelos PDB.

El formato de los datos también se cambia, dado que en el archivo PDB los datos no se encuentran separados por caracteres especiales, sino que se encuentran ubicados en posiciones específicas de la línea de archivo. Los campos que se incluyen en el modelo PDB “limpio” se separan por tabuladores.

Se seleccionaron sólo cinco columnas, el nombre del átomo (ATOM NAME), nombre de la cadena lateral de cada AA presente en la SAA (RES NAME), además de las coordenadas tridimensionales (X, Y, Z) de cada carbono alfa. La estructura básica de uno de los modelos PDB limpios obtenidos se observa en la Figura 7.

ATOM	NAME	RES NAME	X	Y	Z
CA	NLE	24.660	30.640		36.370
CA	SER	22.250	28.460		38.450
CA	ASP	20.580	25.660		36.590
CA	GLU	22.170	26.610		33.320
CA	ASP	24.720	29.310		32.390
CA	PHE	23.220	30.160		28.940
CA	LYP	19.960	31.140		30.840
CA	ALA	21.190	34.890		31.200
CA	VAL	21.190	35.040		27.310
CA	PHE	17.800	33.090		27.280
CA	GLY	15.630	32.020		30.290
CA	MET	14.090	29.250		28.400
CA	THR	15.770	26.010		27.240
CA	ARG	19.020	26.450		25.230
CA	SER	17.510	24.380		22.320
CA	ALA	14.650	26.940		21.680
CA	PHE	17.120	29.920		21.160
CA	ALA	18.530	29.240		17.680
CA	ASN	18.190	26.840		14.740
CA	LEU	21.020	25.340		12.980
CA	PRO	20.610	22.320		10.490

Figura 7: Esquema del Modelo PDB Limpio.

3.3.3 Construcción, Integración y Formateo de Datos

En el objetivo de minería de datos se planteó la construcción de un modelo de AC para clasificación, por lo tanto se hace necesario que los datos de cada modelo, adopten una forma discreta, semejante a una configuración global de un modelo de AC. Por este motivo, se construyeron mapas de contacto [47], uno por cada modelo PDB. Con esto, quedan definidas algunas de las características del modelo de AC como la dimensión del lattice y el conjunto de estados.

Un mapa de contacto es una matriz cuadrada de N x N, donde N es la cantidad de AA presentes en la proteína. Cada celda de la matriz, puede tomar dos posibles valores (0 ó 1), dependiendo de si los carbones alfa representados por la fila y la columna que se intersectan, se encuentran a una distancia euclidiana (calculada con las posiciones 3D de cada C-α) en el rango de contacto definido, que para este caso es [4Å, 7Å].

Al final de la fase de pre- procesamiento de datos, se obtuvieron 40.000 matrices de 35 x 35 bits. Un ejemplo de dichas matrices se muestra en la Figura 8, donde se ilustra un mapa de contacto como representación de la estructura 3D de una SAA de 35 AA, en un paso de simulación.Cada matriz o mapa de contacto, se separó en un archivo de texto. Al representar a manera de matriz 2D la estructura 3D de una SAA se puede manipular los datos de manera más general, además no hay pérdida definitiva de información, ya que es posible reconstruir la proteína a partir de un mapa de contacto y de la SAA que le dan origen usando técnicas como las planteadas en [47][48][49].

Figura 8: Mapa de contacto.

3.4 FASE 4: MODELADO

3.4.1 Selección de la Técnica de Modelado

En ésta sección se describe la técnica de modelado definida, que en este caso es una estrategia co – evolutiva que se planteó tomando como referente las técnicas documentadas en el estado del arte.

Una de las estrategias del estado del arte que se consideró importante fue [4] donde uno de los principales inconvenientes presentados fue que los modelos co – evolutivos aplicados, generalmente exhibieron comportamientos como la dinámica de Reina Roja [50] debido a la inicialización aleatoria de las configuraciones iniciales. Los resultados obtenidos son un claro indicio que el uso configuraciones iniciales cambiantes, generadas de manera aleatoria no son convenientes, y que por lo tanto no se considerarán en el desarrollo de la presente investigación. Otro aspecto interesante de este trabajo es la manera cómo abordaron la técnica co – evolución competitiva entre dos poblaciones, utilizando una población de configuraciones iniciales para evaluar la población de soluciones conformada por los modelos de AC identificados.

Otro ejemplo es la estrategia documentada en [6] donde el fitness de cada regla se calculó como la puntuación media de la fracción de estados correctos de las celdas en la última iteración. La selección se llevó a cabo por torneo. La manera como se calculó el

fitness es un ejemplo a imitar ya que en el caso de esta investigación interesa conocer que tan correctamente son actualizados los estados de cada una de las celdas.

Debido a la naturaleza del fenómeno que se quiere simular, y considerando la información que se posee, se podría estudiar la posibilidad de utilizar simultáneamente los enfoques de co – evolución competitiva y cooperativa como se hizo en el trabajo descrito en [51].

3.4.1.1 Manejo del Volumen de los Datos.

El conjunto de datos de partida que se seleccionó es de 40.000 mapas de contacto, el cual es muy grande como para que sea procesado de manera secuencial por un conjunto limitado de recursos hardware. Lo ideal sería aprovechar el paralelismo [52] y la escalabilidad [53] que ofrece un entorno distribuido. Por lo tanto, el dataset a utilizar será distribuido, como se indica en la figura 9.

Teniendo en cuenta lo anterior, y debido a que en [2] se determinó que los modelos de AC obtenidos en algunas secciones de la trayectoria de plegamiento de proteínas, presentan buen comportamiento en otras secciones de la trayectoria diferentes a las utilizadas para entrenamiento, se tomó la decisión de dividir el conjunto de datos, obtenido en la etapa de pre – procesamiento, en dos grandes grupos, uno para utilizarlo en la fase de entrenamiento, y otro para efectuar pruebas.

Por otra parte, partiendo del hecho que las estrategias co – evolutivas son métodos computacionales que trabajan con poblaciones de individuos [23], se hace necesario definir un número de poblaciones adecuado que 1) contribuya en la búsqueda de una solución que no se sobre ajuste a los datos de entrenamiento y que adicionalmente sea producto de una búsqueda intensiva en el espacio conformado por las posibles soluciones y 2) que sea de fácil manipulación considerando la disponibilidad de hardware.

En el contexto descrito, se decidió utilizar un conjunto de nueve poblaciones, a cada una de cuales se le asignó una sección del bloque de datos destinado para la fase de entrenamiento, estableciendo de esta manera una relación directa entre el número de poblaciones y el número de secciones de datos.

En proyectos de investigación donde se implementan técnicas de minería de datos, se utiliza el 70% de los datos para la fase de entrenamiento y el 30% para las pruebas. El 70% del Dataset utilizado en la presente investigación equivale a 28.000 mapas de contacto, que al ser repartidos entre 9 poblaciones, a cada población le correspondería una cantidad no entera de MCs, razón por la cual los porcentajes definidos quedan así: 67,5% de los datos para obtener los modelos de AC en la fase de entrenamiento y el 32,5% restante para realizar pruebas del mejor modelo de AC encontrado.

El entrenamiento será llevado a cabo bajo el esquema ILAS (Aprendizaje Iterativo con Alternancia de Estratos) [58], el cual se adaptará de la siguiente manera:

- El conjunto de los 27.000 mapas de contacto para entrenamiento, equivalentes al 67.5% del Dataset obtenido después del pre – procesamiento de datos, se dividirá en nueve bloques.
- Cada bloque de datos estará conformado por 3000 mapas de contacto.
- Cada población presente en el espacio co – evolutivo estará encargada de procesar un bloque de datos.
- Debido a que el enfoque ILAS utiliza estratos, se dividirá cada bloque de datos en 15 estratos, cada uno compuesto por 200 evidencias o mapas de contacto.
- En cada iteración, cada población procesará un estrato del bloque designado, de tal manera que cada 15 generaciones se completará una iteración ILAS.

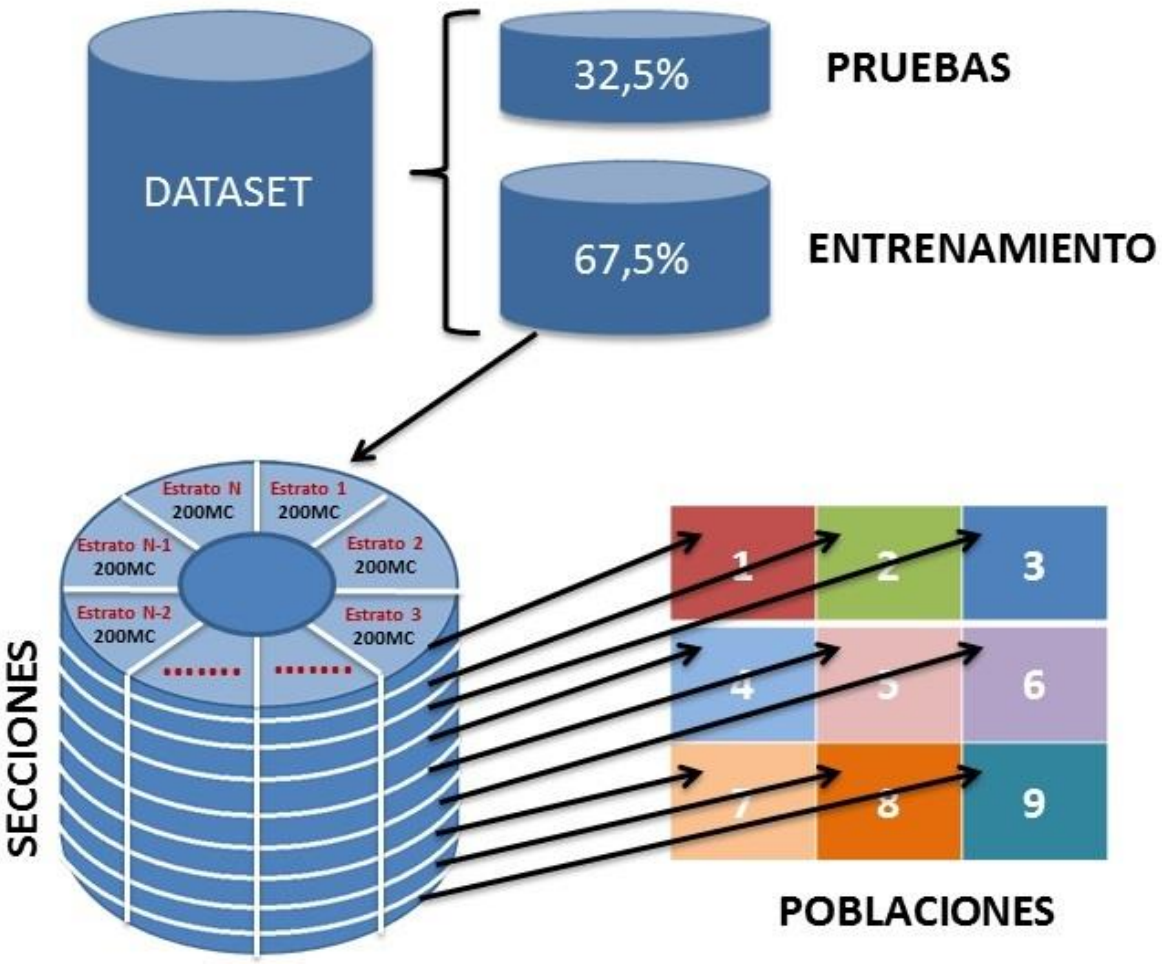

Figura 9: División del Dataset en Secciones y Estratos**.**

3.4.1.2 Espacio de la Estrategia Co – evolutiva

Partiendo de la necesidad de organizar las poblaciones de individuos para facilitar su manipulación, y tomando como referente trabajos de investigación similares como [4] [23], se tomó la decisión de distribuir las poblaciones en una estructura semejante a la abstracción de una malla bidimensional en la que los individuos estarían espacialmente embebidos, lo que implicaría que cada celda en la malla albergaría a un individuo.

Si en la malla 2d se ubicaran nueve poblaciones, cada una conformada por veinticinco individuos, el espacio co - evolutivo a utilizar seria semejante al ilustrado en la Figura 10.

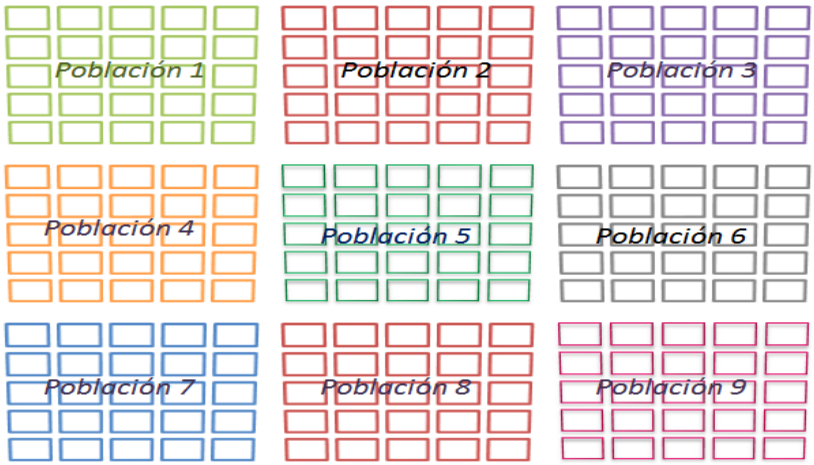

Figura 10: Espacio Co Evolutivo.

La malla 2D definida, tiene algunas características propias de un modelo de Autómata Celular, tales como las condiciones de frontera, lattice y vecindad. El objetivo de incluir estas características en el espacio co – evolutivo es facilitar la interacción entre los individuos que conforman las diferentes poblaciones en el momento de calcular el fitness externo.

La interacción entre los individuos del espacio co – evolutivo estará determinada por las condiciones de vecindad. Con la intención de ampliar este rango de interacción entre los individuos de las diferentes secciones que ocuparán cada celda en la malla, se utilizara frontera cíclica, luego la malla ilustrada en la Figura 10 se comportará de acuerdo a la condición de frontera establecida como el toroide ilustrado en la Figura 11.

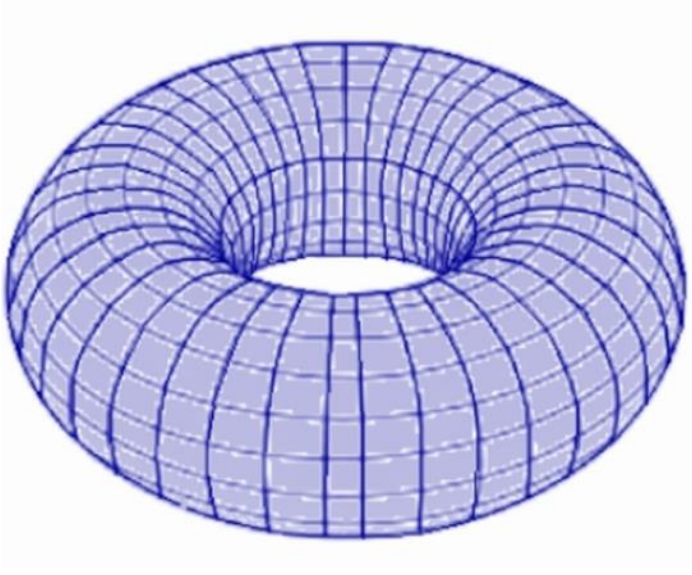

Figura 11: Toroide

3.4.1.3 Descripción de los Individuos

Un modelo de Autómata Celular está constituido por cinco elementos, lattice, conjunto de estados, condiciones de frontera, vecindad y reglas. Debido a la falta de conocimiento del fenómeno que se modelará con el paradigma de Autómatas Celulares, se hace necesario encontrar algunos de los elementos que componen la arquitectura utilizando una técnica de aprendizaje de máquina.

Se parte de un modelo de AC parcialmente definido, de acuerdo a las características de los datos obtenidos en la etapa de pre – procesamiento de datos:

- Lattice bidimensional, de n x n (con n = número de AA en la proteína).
- Cada celda en el lattice podrá tomar un valor del conjunto de estados {0, 1}, donde 1 indica contacto y 0 indica no contacto.
- Frontera finita.

Con la estrategia Co – evolutiva se buscarán los dos elementos faltantes del modelo de AC, la vecindad y el conjunto de reglas.

Debido a que el conjunto de reglas puede ser extraído a partir del conjunto de mapas de contacto (evidencias) una vez se tenga definida la vecindad, la tarea de diseño inverso de Autómatas Celulares se reduce a encontrar una vecindad adecuada al problema. Por lo tanto, los individuos que conformarán las poblaciones serán vecindades que tendrán asociado un modelo de AC.

3.4.1.4 Elementos generales de un algoritmo genético

Hasta el momento se ha definido la forma de cómo serán manipulados los datos, los individuos que se utilizarán y el espacio en el que evolucionaran. A continuación serán descritos los elementos del esquema general de un algoritmo genético

Partiendo de que computacionalmente los algoritmos co – evolutivos son definidos como una extensión de los algoritmos genéticos, es de esperar que posean elementos en común, los cuales se definieron como se describe a continuación.

- ***Definición del fenotipo***

  Los individuos en un AG se codifican generalmente como cadenas binarias. Haciendo uso de la terminología biológica, las cadenas binarias hacen parte del genotipo del AG y el espacio de búsqueda del problema se refiere a su fenotipo.

  El espacio de búsqueda que se explorará se reducirá las posibles vecindades que se puedan conformar dentro de una vecindad de Moore de radio 3, como la ilustrada en la Figura 12, donde la celda central que es la considerada para la evolución puede o no estar presente. Con esto, el fenotipo se determinará a partir de la posición de los genes que tome el valor binario de 1, lo que significará que la celda forma parte de la vecindad y que el estado del contacto entre un par de AA depende del estado de la vecindad pero no de su propio estado facilitando la aplicación de los operadores genéticos.

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 12. Espacio de búsqueda del problema (Vecindad de Moore de Radio 3).

La representación seleccionada hace referencia a configuraciones topológicas sobre los mapas de contacto, donde los estados de contacto y no contacto afectan a una celda de estudio en el contexto local de la vecindad codificada por el cromosoma de longitud 49 cuyo fenotipo es una vecindad de Moore de radio 3. Adicionalmente, [2] se demostró que los modelos de AC asociados a esta representación son aplicables a trayectorias en mapas de contacto de SAA de longitudes mayores o menores. Definición del genotipo

El genotipo para representar la vecindad consiste en una cadena binaria de longitud 49, donde a cada posición en la cadena le corresponde una de las 49 celdas en la vecindad de Moore de radio 3 del fenotipo.

Es importante que cada posición del cromosoma tenga un significado en la solución del problema, para este caso, los genes con valor igual a 1 indican que la celda a la que representan según su posición en el cromosoma, se incluyen en la vecindad. Por ejemplo, el genotipo que se muestra en la Figura 13, representa la vecindad de la Figura 14.

1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0

Figura 13: Ejemplo de genotipo

Figura 14. Espacio Ejemplo de Vecindad Codificada

- ***Operador Genético de Selección***

Después de que se ha evaluado el desempeño de cada uno de los cromosomas o individuos de la población, se conforma un conjunto de cromosomas que serán candidatos para cruzarse. En la mayoría de los casos la probabilidad de que un cromosoma sea seleccionado para cruzarse es proporcional a su desempeño. Con esto se intenta replicar el proceso de selección natural donde los individuos mejor adaptados tienen mayor posibilidad de sobrevivir.
El objetivo del operador genético de selección es escoger los individuos más aptos para conformar una nueva generación. Para este caso de estudio, se utilizará el mecanismo de selección por torneo[54][55], que tiene asociado un parámetro que le otorga a los mejores individuos la probabilidad de ser seleccionados para el proceso de reproducción.

- ***Operador Genético De Cruce***

Con el operador genético de cruce en dos puntos,[56] se eligen dos cromosomas padre los cuales intercambian entre ellos material genético para generar dos cromosomas hijo. Los operadores genéticos poseen una probabilidad de aplicación, en este caso probabilidad de cruce, de manera que el operador sólo se aplica si un número generado aleatoriamente (entre 0 y 1) está por encima de la probabilidad especificada. La probabilidad de cruce usualmente se establece entre 0,4 y 0,9.

- ***Operador Genético De Mutación***

Después de haber realizado el proceso de cruzamiento, los hijos resultantes pueden sufrir mutaciones como consecuencia de errores naturales en la copia del material genético. La forma más sencilla de mutación que puede aplicarse a un cromosoma

hijo es la mutación por sustitución puntual en la que se elige de manera aleatoria un gen y se cambia el valor del alelo para el mismo, de esta forma si en el gen se encuentra un alelo con valor 1 se cambia a 0 y viceversa. A este tipo de mutación se le conoce también como inversión de bit [57], que fue el seleccionado para la presente investigación.

Para aplicar el operador de mutación se realiza un recorrido por todos los cromosomas de la nueva población generando un número al azar entre 0 y 1 para cada cromosoma, si el número generado es mayor a la probabilidad de mutación establecida, el cromosoma actual se muta, en caso contrario permanece inalterado. La probabilidad de mutación no debería superar el valor de 0,1, ya que indica la frecuencia con la que los genes del cromosoma de un individuo son modificados. Si la probabilidad de mutación es 0, no habrá cambios, pero si por el contrario es 1 (ó 100%) la totalidad del cromosoma se cambia.

3.4.1.5 Algoritmo Co - Evolutivo.

En esta sección se describe cada una de las partes del algoritmo que se desarrolló para implementar la estrategia co – evolutiva. El diagrama de flujo de dicha estrategia se ilustra en la Figura 15.

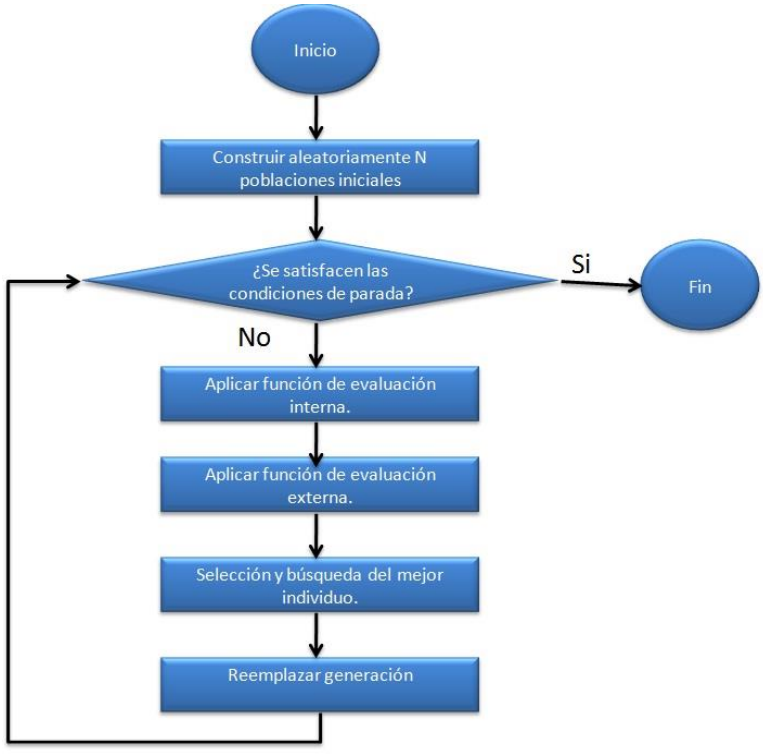

Figura 15: Diagrama de flujo algoritmo Co-Evolutivo

- ***Inicialización***

La inicialización del algoritmo hace referencia a definir todos los parámetros involucrados en el algoritmo, incluyendo la creación aleatoria de la primera generación de cromosomas. Más específicamente, se deben inicializar los siguientes parámetros:

- Cantidad de poblaciones (9)
- Tamaño de cada población (25)
- Longitud del cromosoma (49)
- Población inicial (valor de los genes para los N cromosomas)
- Probabilidad de cruce
- Probabilidad de mutación
- Criterios de parada.
- Probabilidad de Selección
- Probabilidad de Cruce.
- Probabilidad de Mutación

Los cromosomas que hacen parte de la población inicial se generan de manera completamente aleatoria, esto significa que se genera cada gen del cromosoma mediante una función que retorna un 1 o un 0 con igual probabilidad, lo que dota a la población de suficiente variedad para permitir explotar todas las zonas del espacio de búsqueda, garantizando de esta manera el buen funcionamiento del algoritmo.

- ***Función de desempeño***

Este paso es necesario para evaluar el desempeño del cromosoma como solución al problema considerado. La función de desempeño permite asignarle valores numéricos a los cromosomas, con dicho valor se indica qué tan bueno es el individuo como solución al problema. El resultado de la evaluación es útil para realizar el proceso de selección.

Debido a la naturaleza de la estrategia implementada, se consideró el cálculo de dos funciones de desempeño en dos momentos diferentes.

- **Función de Evaluación Interna**

La medida que arroja esta función se utiliza para determinar cuáles son los individuos más aptos que se considerarán en el proceso de selección. La evaluación se hace calculando el MCC (Coeficiente de correlación de Mathews) menos una penalización proporcional a la cantidad de celda incluidas en la vecindad. En el diagrama de flujo de la Figura 16 se ilustra el algoritmo aplicado para calcular la aptitud de un individuo medida por la función de evaluación interna.

La función se aplica para cada uno de los individuos embebidos en el espacio co – evolutivo. Antes de aplicarla se debe *encontrar el conjunto de reglas* del modelo de Autómata Celular asociado a la vecindad encontrada. Para esto, es necesario realizar un proceso de minería de patrones sobre los vecindarios presentes en las configuraciones globales representadas por los mapas de contacto obtenidos en la etapa de pre – procesamiento de datos. Además de los patrones, se debe determinar la frecuencia con la que estos se presentan.

Una vez se tiene completamente definido el modelo de AC con la vecindad y el conjunto de reglas, se procede a *simular*, y se guarda la trayectoria resultante. Posteriormente, se *compara la trayectoria* generada tras la simulación con la trayectoria en mapas de contacto obtenida después del pre – procesamiento de datos, con el objetivo de *calcular el puntaje* del individuo dependiendo de el porcentaje de contactos y no – contactos correctamente obtenidos, menos un factor que penaliza a las vecindades con demasiadas celdas.

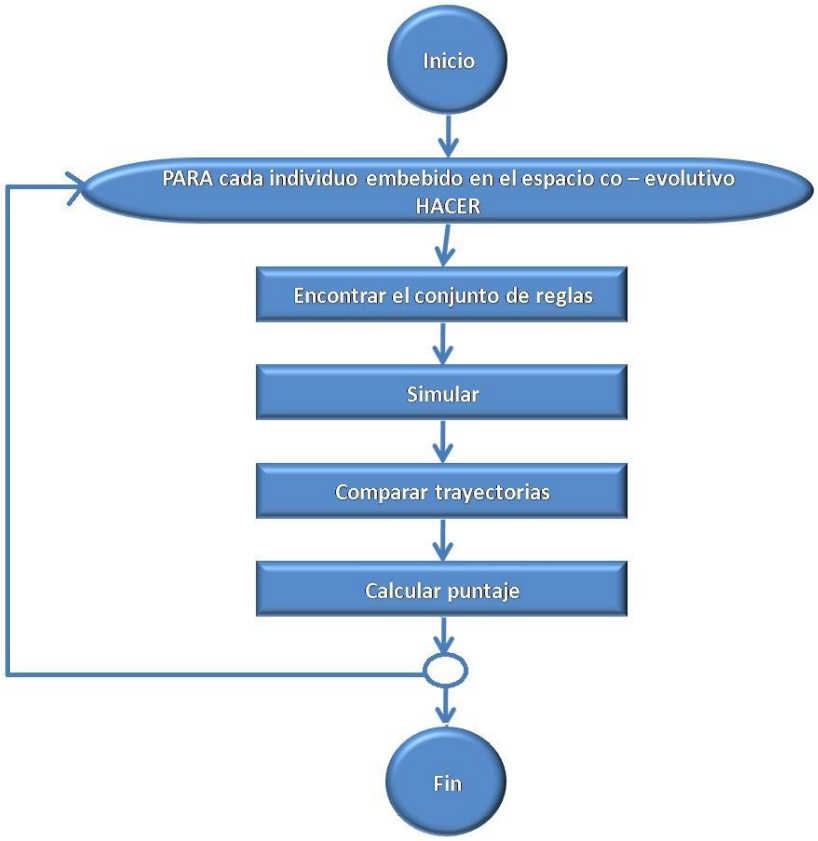

Figura 16: Diagrama de Flujo, Función de evaluación Interna

- ***Función de Evaluación Externa***

La medida que arroja la función se utiliza para examinar la calidad de un individuo en el contexto de otros individuos. La función de evaluación externa consiste en determinar que tan buen predictor es el modelo de AC, *calculando la distancia con el predictor perfecto en el espacio ROC*. Esta medida se calcula después de aplicar la función de evaluación interna de los individuos de cada población.

Después de calcular el fitness externo, se procede a *comparar el desempeño de cada individuo presente en la frontera de cada sección de la malla* con el desempeño de los individuos incluidos en una vecindad de von Newman de radio 1. Al realizar la comparación se verifica cuál de los modelos de AC vecinos está más cerca del predictor perfecto, y se realiza el *remplazo* en caso de encontrarse uno mejor que el que se está tomando como referente en la comparación.

- ***Condiciones de Parada***

Una vez calculada la aptitud de los individuos, se evalúan las condiciones de parada del algoritmo, las cuales se determinaron así:

- Se completa el número máximo de iteraciones.
- Se presenta decrecimiento en el proceso de búsqueda de mejores soluciones medida por el promedio acumulado de las mejores soluciones en cada iteración versus la media móvil de las últimas 40 generaciones.
- La diversidad de población se mantiene inferior al 20% a lo largo de 5 generaciones.

- ***Reemplazar generación***

Si después de evaluar la aptitud de los individuos y de verificar que no se cumple ninguna de las condiciones de parada, se procede a generar una nueva población de descendientes aplicando los procesos de selección, cruce y mutación.

El proceso de reemplazo no debe confundirse con el de selección, puesto que en este último se manifiesta una competencia entre los miembros de la misma población. Por el contrario el reemplazo se trata de un proceso de supervivencia poblacional, similar a lo

que ocurre en la naturaleza donde no todos los descendientes sobreviven para formar la nueva generación.

De las alternativas de reemplazo existentes, se seleccionó el reemplazo generacional, que consiste en reemplazar la población actual en su totalidad, con lo que sus descendientes pasarían a ser directamente la población actual para el siguiente ciclo del algoritmo, el cual itera mientras no se cumpla al menos una condición de parada.

3.4.1.6 Técnicas Co – evolutivas Aplicadas

En términos de técnicas co – evolutivas, la estrategia aquí planteada puede ser vista de dos diferentes formas:

- Como co – evolución de n poblaciones competitivas [23], ya que se comparan los individuos de las diferentes poblaciones y se intercambian los que se desempeñan mejor de acuerdo al Fitness Externo.
- Co – evolución competitiva de dos poblaciones P y Q [23], ya que los mapas de contacto pueden ser vistos como la población de prueba, que se toma como referente para calcular el fitness interno de los modelos de Autómata Celular, la población de soluciones.

3.4.2 Diseño de la Prueba.

Siguiendo los lineamientos de la metodología CRISP- DM[38], se diseñó la prueba que será aplicada durante la evaluación de los modelos de AC obtenidos con la aplicación de la estrategia Co – evolutiva. La prueba diseñada constituye el mecanismo para probar la calidad y validez del modelo. Para su diseño, se consideraron las medidas planteadas en los criterios de éxito de minería de datos, a saber, Sensitividad, Precisión, Coeficiente de Correlación de Matthews, Especificidad, y ROC.

Se evaluó el mejor modelo de AC obtenido tras diferentes ejecuciones de la técnica co - evolutiva implementada. Para ello se consideraron los reportes generados, los cuales se ordenaron de acuerdo al desempeño de los individuos, medido por las funciones de evaluación interna y externa.

Con la medida de Sensitividad, se medirá la proporción de contactos correctamente actualizados por el modelo de AC, teniendo la trayectoria en mapas de contacto generada después del proceso de simulación. La medida de sensitividad se calcula como se muestra a continuación.

Sensitividad = Verdaderos Positivos / (Verdaderos Positivos + Falsos Negativos)

La Precisión, es una medida sensible a sesgos de clase, que indica la proporción de estados correctamente actualizados; por eso servirá para determinar que tan correctamente son actualizados los contactos y no contactos.

Precisión = (Verdaderos Positivos + Verdaderos Negativos) / (Positivos + Negativos)

Con la Especificidad o tasa de verdaderos negativos, se determina la probabilidad de predecir correctamente un no contacto.

Especificidad = Verdaderos Negativos / (Verdaderos Negativos + Falsos Positivos)

El Coeficiente de Correlación de Matthews (MCC), es una medida que no se ve afectada por los sesgos de muestreo y puede variar de -1 a 1. EL valor de MCC se calcula como se muestra en la ecuación 1, los resultados cercanos a 1 indican buenas predicciones, es decir que los verdaderos positivos y los verdaderos negativos, son clasificados de manera satisfactoria. La manera como se calcula el valor del MCC se muestra a continuación., donde VP = Verdaderos Positivos, VN = verdaderos Negativos, FP = Falsos Positivos y FN = Falsos Negativos.

$$MCC = (VP * VN - FP * FN) / \sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}$$

Ecuación 1 - Coeficiente de Correlación de Matthews

En la Figura 17 se ilustra el Espacio ROC, que facilita la evaluación de resultados ubicándolos en un espacio que permite identificar con claridad los mejores modelos. En este caso, los mejores modelos serán aquellos que estén más cercanos al predictor perfecto.

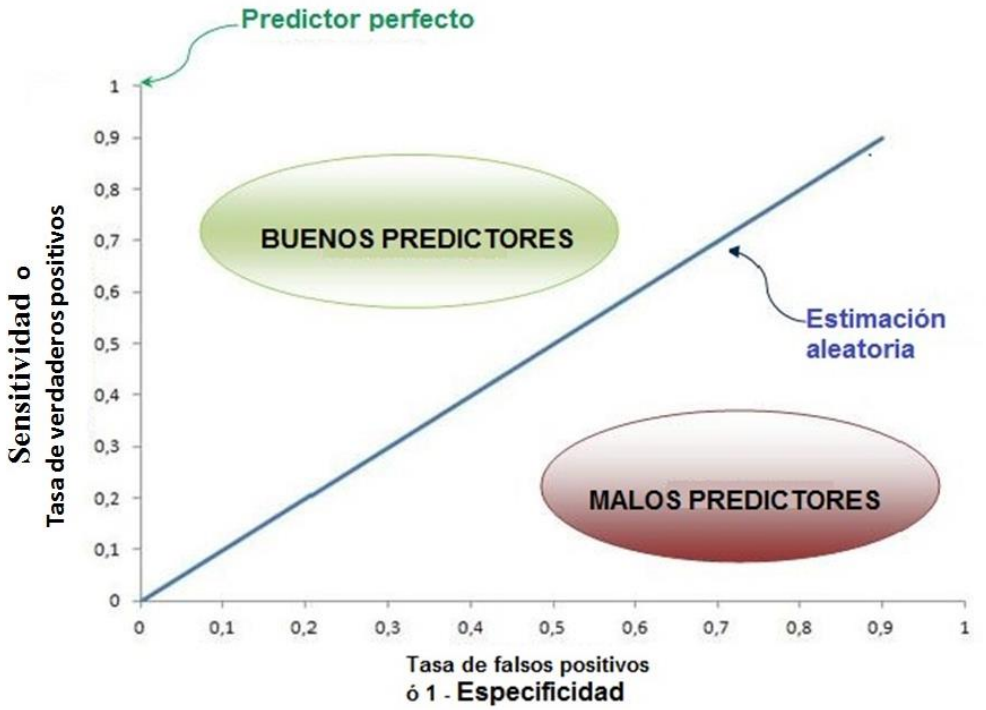

Figura 17: Espacio ROC (Receiver Operating Characteristics)

Adicionalmente, se generará un Árbol de Decisión con los patrones de las reglas del mejor modelo de Autómata Celular obtenido con la técnica. Este árbol se utilizará para determinar si el modelo captura la dinámica del fenómeno considerado.

La prueba diseñada se aplicará una vez se seleccione el mejor modelos de AC de una ejecución, para esto se utilizará la porción de evidencias destinada para las pruebas como se indicó en la sección de la definición de la estrategia co – evolutiva.

3.4.3 Construcción del Modelo

En esta fase de la metodología CRISP – DM, se ejecuta la herramienta de modelado, que en este caso es la estrategia co – evolutiva, sobre el conjunto de datos pre - procesados para encontrar uno o más modelos de autómata celular.

3.4.3.1 Ajuste de parámetros

Al igual que con cualquier herramienta de modelado, existen ciertos parámetros que deben ser ajustados. A continuación se documenta el proceso que se llevó a cabo para ajustar los parámetros de la estrategia co – evolutiva implementada. Además, se

presentan y describen las graficas de los reportes estadísticos generados tras cada ejecución.

Partiendo de los parámetros utilizados en [2], donde se obtuvieron resultados que superan el umbral de calidad establecido, se realizaron una serie de ejecuciones con el fin de encontrar unos parámetros que posibilitaran obtener resultados superiores.

Debido a que la estrategia que se implementó requiere parámetros adicionales a los que se fijan en un Algoritmo Genético, fue necesario afinarlos durante el desarrollo de la estrategia teniendo en cuenta el tiempo que se tardaba en distribuir, procesar, retornar los resultados en el entorno distribuido.

Las características del algoritmo que se mantuvieron como constantes en las diferentes ejecuciones fueron las siguientes:

- Longitud del Cromosoma: 49 genes
- Alelos: {0,1}
- Operador de Selección: Torneo.
- Operador de Cruce: Dos puntos.
- Operador de Mutación: Inversión de bit
- Función de Evaluación Interna: Coeficiente de Correlación de Matthews – penalización por cantidad de reglas.
- Función de Evaluación Externa: Distancia con el predictor perfecto en el espacio ROC
- Tamaño del estrato ILAS: 200 pasos de simulación.
- Número máximo de iteraciones: 100.
- Número de secciones: 9
- Distribución de las secciones en el espacio Co – evolutivo: 3 Filas x 3 Columnas.
- Tamaño de la Población: 25 Individuos.
- Distribución de los individuos en cada sección: 5 Filas x 5 Columnas.

En esta actividad de la cuarta fase de la metodología CRISP – DM, se llevó a cabo el ajuste de 3 parámetros, las probabilidades de selección, cruce y mutación tomando como referente estudios realizados sobre el ajuste de parámetros como los registrados en [58][59] que indican que lo más aconsejable es ajustar dos parámetros en cada ejecución variándolos en una o dos centésimas según el desempeño estadístico [60] del algoritmo. Estos estudios indican además, que se deben utilizar una cantidad de datos menor o igual al 12% en el proceso, y que es válido ajustar los parámetros de manera manual, y que su ajuste impacta profundamente el desempeño del algoritmo.

Tras cada iteración, la aplicación implementada guarda en un reporte estadístico los datos correspondientes al fitness interno y externo del mejor individuo, la media acumulada, la

media móvil y la diversidad de población. Estos datos hacen posible verificar el desempeño del algoritmo, una tarea crucial en la actividad de ajuste de parámetros.

En el objetivo del proyecto se definió que se busca un modelo de autómata celular que actualice los contactos y no contactos con una precisión superior al 90%, esto, sumado a la convergencia del algoritmo medida por la media acumulada y la diversidad de población, son criterios útiles para determinar si se detiene una ejecución.

Uno de los comportamientos que se espera en los resultados arrojados por las funciones de evaluación, es que se presenten variaciones en los valores de los fitness calculados, derivados de la manera como se implementó el enfoque ILAS explicado en la sección *Manejo del Volumen de los Datos.*

- ***Ejecución No. 1 para ajuste de parámetros***

En la Tabla 1 se muestran los parámetros de la primera ejecución del Algoritmo los cuales, como ya se mencionó, fueron tomados de [2] y constituyen el punto de partida para el proceso de ajuste de parámetros. Y en la Figura 18 ilustra el Fitness interno y externo de los mejores individuos que se obtuvieron al realizar la ejecución.

Tabla 1. Parámetros Ejecución No. 1.

Parámetros	Valores
Probabilidad de selección	0,05
Probabilidad de cruce	0,9
Probabilidad de mutación	0,01

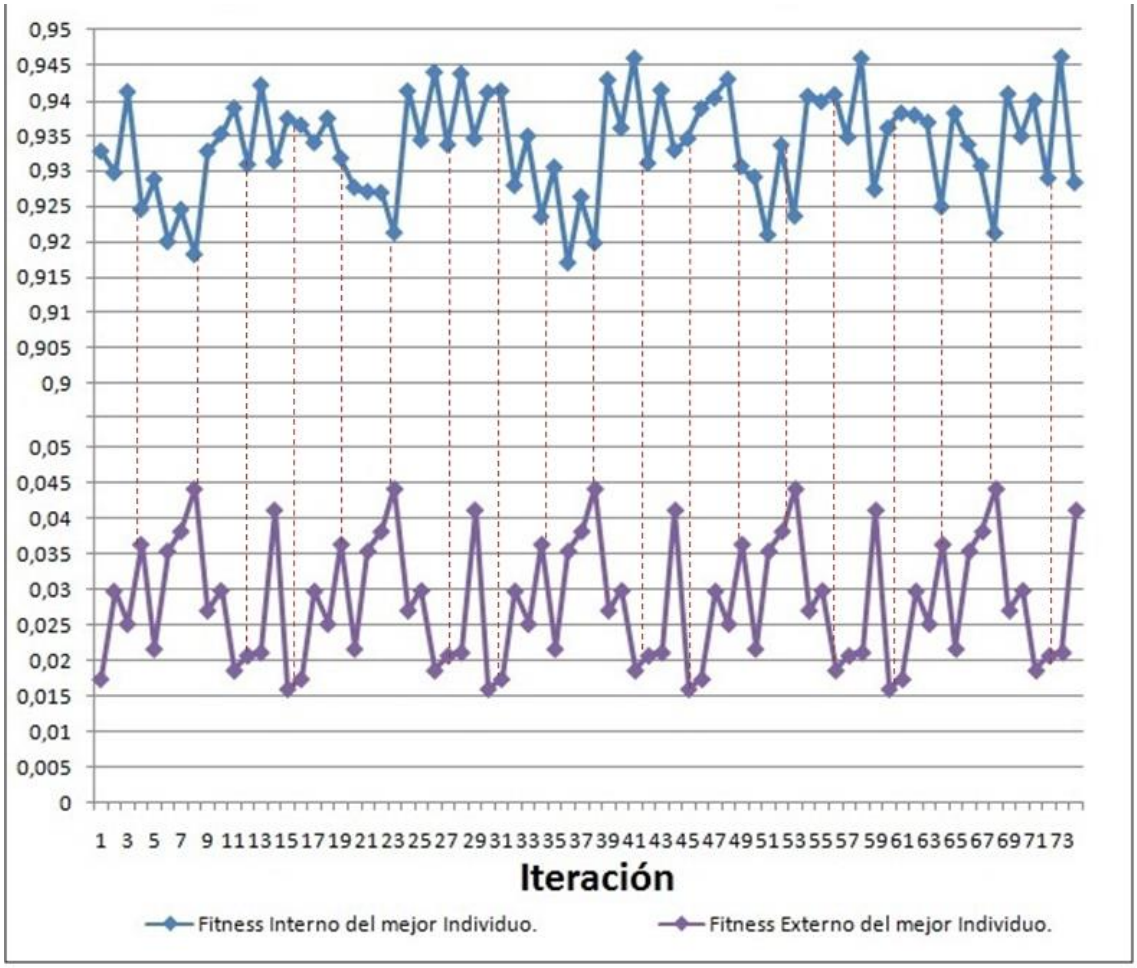

Figura 18: Fitness Interno vs Externo. Ejecución No. 1**.**

En esta gráfica es posible observar que el algoritmo encuentra individuos con un MCC mayor a 0.9, lo que implica que es posible que los modelos encontrados tengan una precisión para actualizar los contactos y no contactos mayor al 90%. Pese a esto, al observar la Figura 19 donde se ilustran la Media acumulada vs Diversidad de población, es posible observar que aunque la diversidad de población no decrece, el algoritmo no se estabiliza con el paso del tiempo, y que a medida que se itera, el desempeño de los individuos empeora pues la media acumulada del fitness interno decrece, razón suficiente para detener la ejecución y reajustar los parámetros.

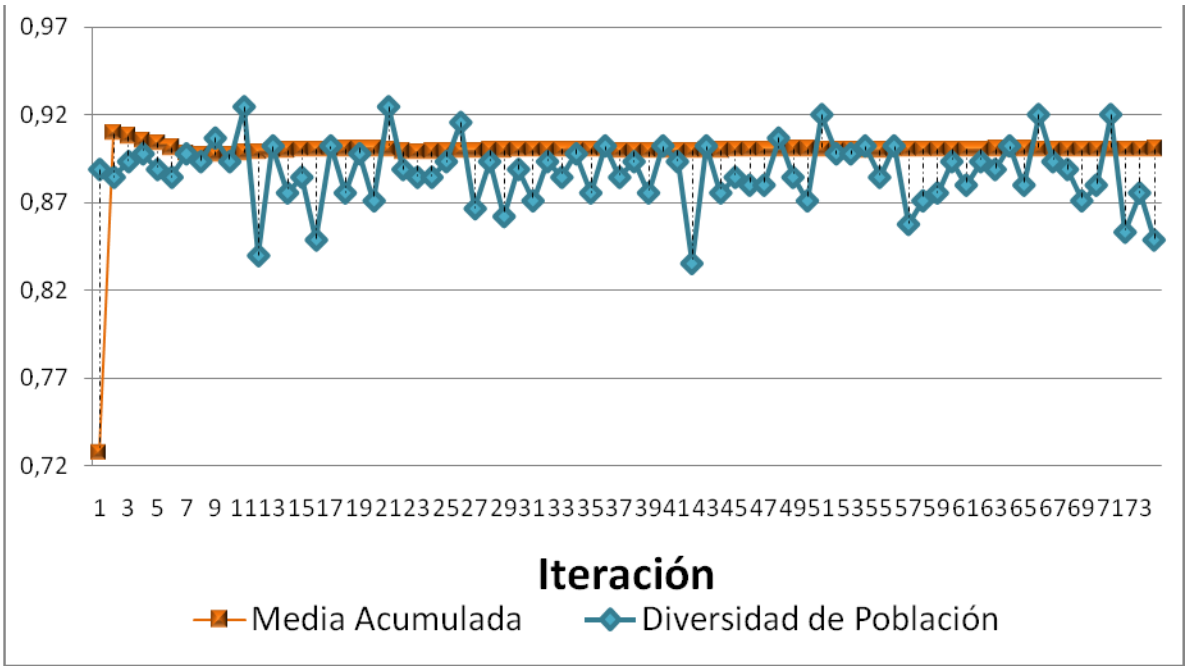

Figura 19. Media Acumulada y Diversidad de Población, Ejecución 1

- ***Ejecución No. 2 para ajuste de parámetros***

La segunda ejecución, se llevó a cabo con los parámetros mostrados en la Tabla 2. Los parámetros considerados, son cercanos a los utilizados en la primera ejecución. Se decremento en 0.001 la probabilidad de selección, y se incrementó en 0.002 la probabilidad de mutación, con esto es más probable que se efectúen modificaciones sobre los cromosomas resultantes después de seleccionar y cruzar los mejores individuos, adicionalmente se amplía la posibilidad de que los individuos seleccionados no sean los mejores.

Tabla 2. Parámetros Ejecución No. 2

Parámetros	Valores
Probabilidad de selección	0,049
Probabilidad de cruce	0,9
Probabilidad de mutación	0,012

La gráfica de la Figura 20, muestra el desempeño de los individuos, medido por el fitness interno y externo a lo largo de 26 iteraciones. A pesar del buen comportamiento de los mejores individuos obtenidos en cada iteración, se aprecia que la medida MCC calculada no supera el umbral definido en los objetivos.

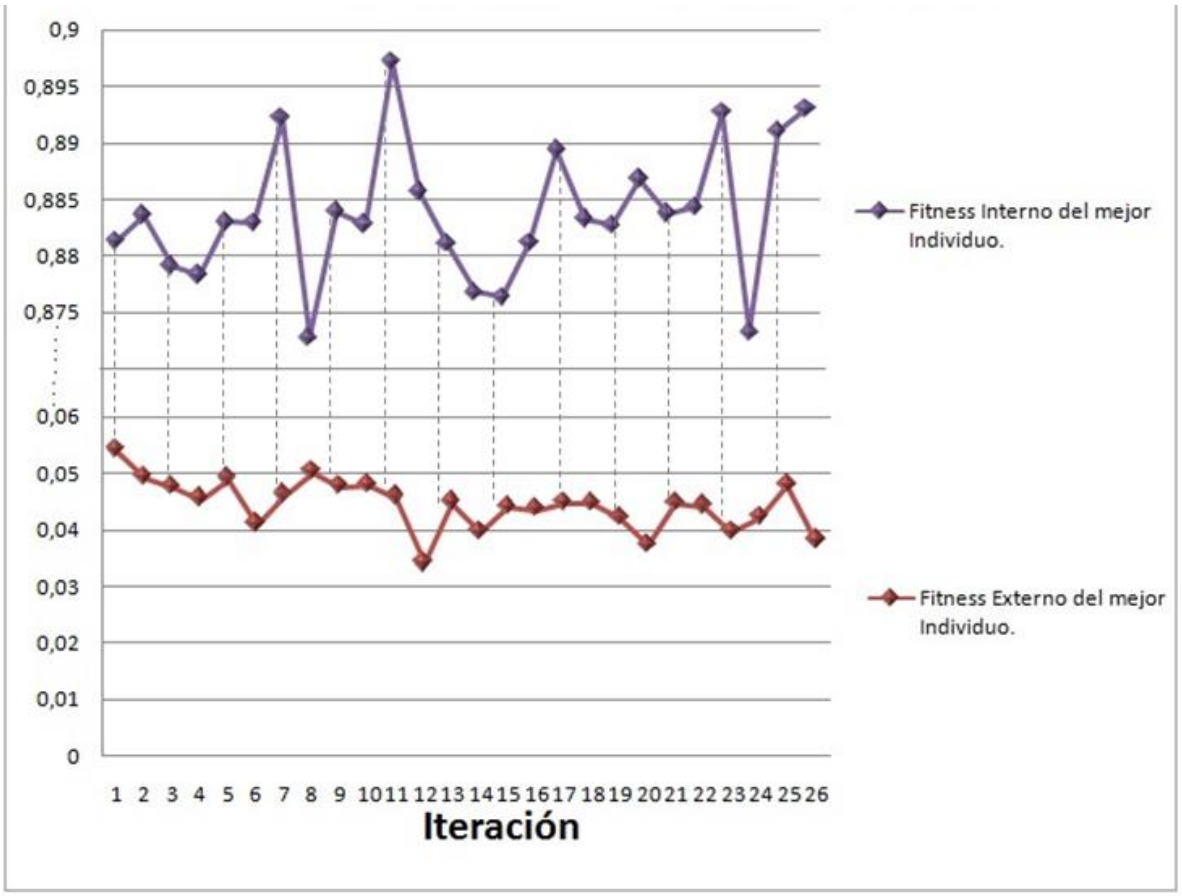

Figura 20: Fitness Interno y Externo, Ejecución No. 2.

Por otra parte, en la Figura 21 es posible observar que la diversidad de población se mantiene por encima del 80% y que a diferencia de lo acontecido en la primera ejecución, la media acumulada del fitness interno tiende a estabilizarse.

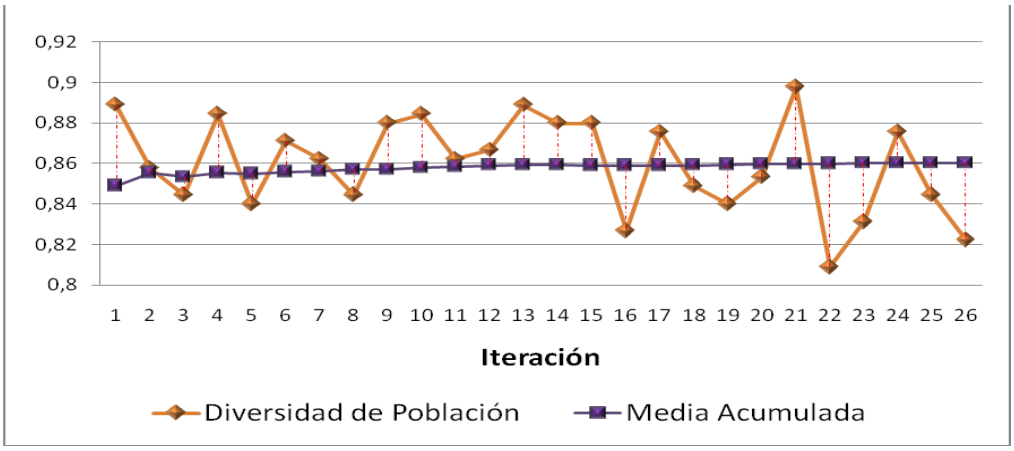

Figura 21: Media Acumulada y Diversidad de población, Ejecución No. 2

Adicionalmente, aunque el comportamiento del algoritmo mejoró, los modelos obtenidos no permitieron alcanzar el objetivo de encontrar un modelo con un MCC mayor al 90%, razón por la que se decidió detener la ejecución y reajustar los parámetros, pues al seguir iterando era posible que no se obtuvieran mejores resultados, desperdiciando de esta manera tiempo y recursos hardware.

- ***Ejecución No. 3 para ajuste de parámetros***

Para esta ejecución se reajustaron los parámetros como se muestra en la Tabla 3. El ajuste se hace respecto a los parámetros de la primera ejecución. La probabilidad de cruce se mantiene y se incrementa en 0.02 la probabilidad de mutación y selección.

Tabla 3. Parámetros Ejecución No. 3

Parámetros	Valores
Probabilidad de selección	0,07
Probabilidad de cruce	0,9
Probabilidad de mutación	0,03

En la Figura 22 se ilustra el fitness interno y externo de cada uno de los individuos de las 17 iteraciones. A diferencias de las anteriores ejecuciones, el desempeño de los individuos presenta una notoria mejoría visible principalmente en la tendencia creciente del fitness interno.

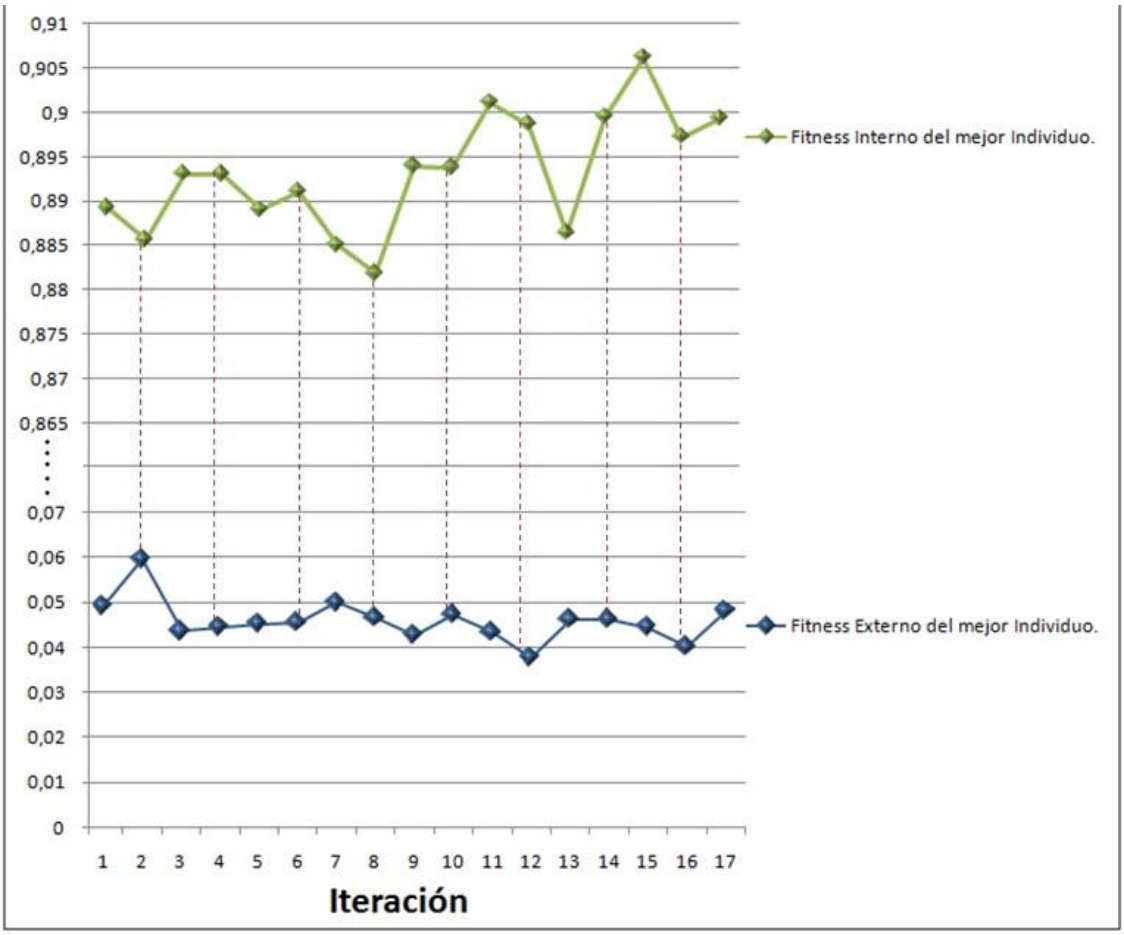

Figura 22: Fitness Interno y externo, Ejecución 3.

Al analizar la Figura 23 donde se grafican la media acumulada del fitness interno vs diversidad de población, se confirma que a pesar de las escasas iteraciones, el desempeño del algoritmo fue mejor que en las ejecuciones anteriores, ya que la diversidad de población se mantiene superior al 75% y la media acumulada del fitness interno tiende a estabilizarse a partir de la cuarta iteración, manteniéndose con valores superiores a 0,86.

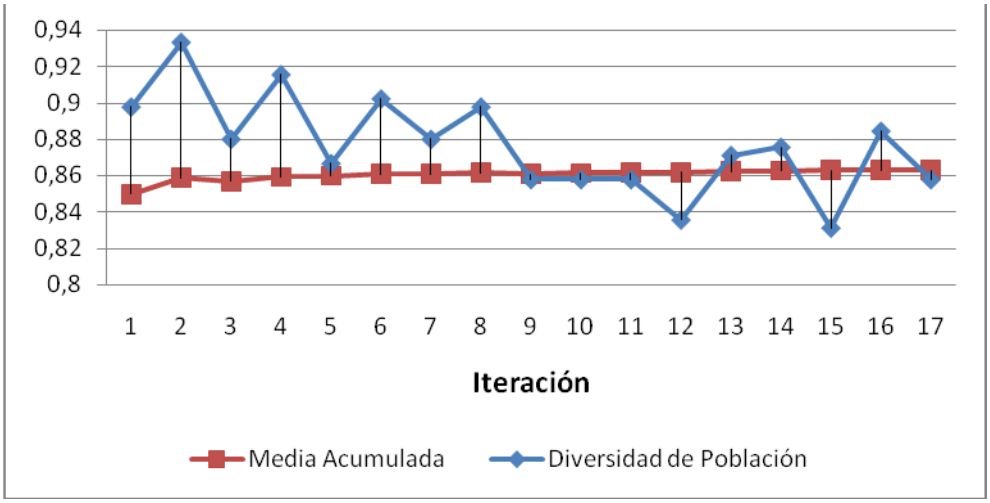

Figura 23: Media acumulada y Diversidad de población, Ejecución No. 3

Basados en las medidas arrojadas por las funciones de evaluación y en los reportes estadísticos que demuestran que la diversidad de población, es superior a 0,82 durante todas las iteraciones y que adicionalmente la media acumulada del fitness interno tiene un comportamiento convergente, se decide utilizar los parámetros encontrados para llevar a cabo el entrenamiento con el volumen de datos destinado para ello, ya que para la el ajuste de parámetros se utilizó únicamente el 10% de los datos.

3.4.4 Evaluación de Modelos

En esta parte de la metodología CRISP – DM, se lleva a cabo la evaluación del modelo encontrado después de la construcción con el objetivo de asegurar que se encontró un modelo que satisface los criterios de éxito de la minería de datos y que cumple satisfactoriamente con los criterios de la prueba diseñada.

La evaluación realizada está basada en el resultado de la tarea de clasificación realizada por el modelo que tuvo mejor desempeño en la etapa de construcción. Para seleccionarlo se consideraron las estadísticas obtenidas en las ejecuciones llevadas a cabo durante el ajuste de parámetros, las cuales se tabularon con los parámetros utilizados (ver tabla 4).

Tabla 4. Parámetros Ejecución y Fitness

No. Ejecución	Probabilidad de Selección	Probabilidad de Cruce	Probabilidad de Mutación	Fitness Interno del Mejor Individuo	Fitness Externo del Mejor Individuo
1	0.05	0.9	0.01	0.946	0.021
2	0.049	0.9	0.012	0.897	0.049
3	0.07	0.9	0.03	0.906	0.045

Aunque el individuo con mejor desempeño se obtuvo en la primera ejecución, no debe ignorarse que el algoritmo se estabilizó y mostró mejor desempeño en la tercera ejecución, razón por la cual se decidió aplicar la prueba diseñada al mejor modelo de autómata celular encontrado en esta.

El mejor individuo encontrado en la tercera ejecución pertenecía a la séptima sección del espacio Co - Evolutivo, la cual tenía asignados los pasos de simulación de la trayectoria comprendidos entre los mapas de contacto 21000 a 24000.

Para probar el modelo, se utilizó la sección de los datos destinada para la fase de pruebas. Esta sección de la trayectoria está compuesta por los últimos13.000 mapas de contacto obtenidos en la etapa de pre – procesamiento.

Después de simular, se calculó la cantidad de contactos y no contactos correctamente actualizados, y se calcularon las medidas especificadas en la tabla 5.

Tabla 5. Medidas de Calidad del Mejor Modelo de AC

Medida Calculada	Valor
Precisión	0.974
Especificidad	0.982
Sensitividad	0.876
MCC	0.821
Distancia con el predictor perfecto en el espacio ROC	0.124

Hasta este punto de la evaluación, se puede observar claramente, que al simular una sección de la trayectoria para la cual el modelo de autómata celular no encontró reglas específicas, los resultados arrojados tienen una precisión superior al 95%, lo que indica que se cumple satisfactoriamente el criterio de éxito definido tanto en los objetivos de minería de datos, como en el proyecto en general.

Por otra parte, el fenotipo de la solución que se muestra en la Figura 24, representa la siguiente vecindad codificada:

1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 1 0 1 1 1 0 0 0 1

Esta vecindad constituye una de las 2^{49} posibles configuraciones de vecindad que se podían conformar en una vecindad de Moore de radio 3 y aunque incluye el 40% del total del número de celdas que conforman la vecindad que fue definida como el espacio de búsqueda, se compensa con el buen desempeño del modelo de AC a la que está asociada.

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 24: Fenotipo de la vecindad del mejor modelo de AC construido.

Otro de los objetivos de éxito definidos, está relacionado con la manera cómo el modelo de autómata celular captura la dinámica del fenómeno de plegamiento de proteínas, hecho que se puede determinar a partir del conjunto o tabla de de reglas del modelo de AC.

La mejor manera de representar la tabla de reglas de un modelo de autómata celular, es un árbol de decisión, debido a que permite clasificar el estado de la celda central de la vecindad en el siguiente paso de simulación a partir de la configuración actual de los vecinos. Es por eso que después de confirmar que el modelo construido con ayuda de la estrategia co – evolutiva implementada arrojaba buenos valores de acuerdo a las medidas de calidad presentadas en el *Diseño de la Prueba*, se procedió a construir el árbol de decisión a partir del conjunto de reglas del AC. Una sección de dicho árbol se ilustra en la figura 26 y está directamente relacionada con la conformación de las hélices alfa.

Una hélice alfa [61] es que es una estructura que caracteriza a una SAA en estado nativo, es por eso que la sección del árbol de decisión que se muestra en la figura 26 hace referencia sólo al momento del plegamiento cuando se dan las condiciones en la configuración de la vecindad para que la variable clase, que en este caso la celda 25, permanezca en contacto.

Como primera medida, se tiene que Villin Headpiece [62] al alcanzar su estructura nativa tendría 3 hélices alfa como las ilustradas en la figura 25, luego al sobreponer el fenotipo de la vecindad encontrada para el modelo de AC que se está evaluando sobre uno de los mapas de contacto perteneciente a una de las últimas secciones de trayectoria, debería evidenciarse que la regla, como se ilustrada en la figura 27, es decir, deben observarse las hélices alfa en términos de configuraciones de estados en el mapa de contacto .

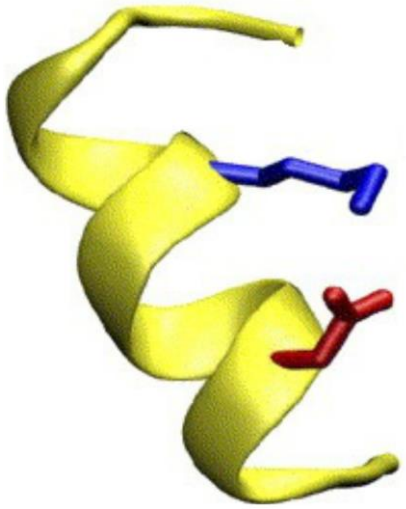

Figura 25: Hélices Alfa en Villin Headpiece

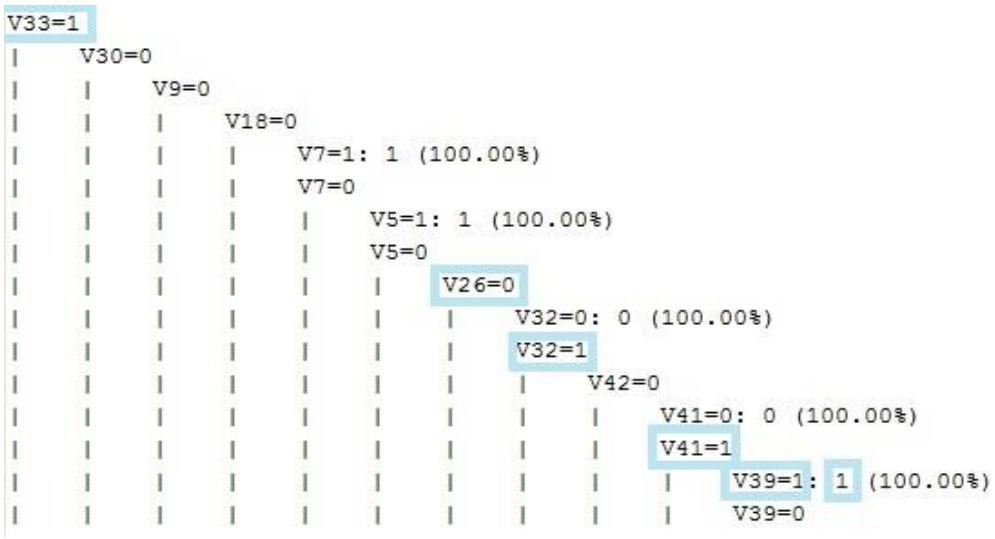

Figura 26: Árbol de Decisión Mejor modelo de AC construido

De acuerdo con lo descrito anteriormente, la sección del árbol de decisión considerada en la figura 24 es importante ya que sirve para demostrar que el modelo de autómata celular captura una de las características básicas de la conformación tridimensional estable de una SAA, una hélice alfa, evidenciado en la figura 25, donde se observa que de acuerdo con las reglas encontradas para la vecindad, se encuentran las 3 hélices alfa.

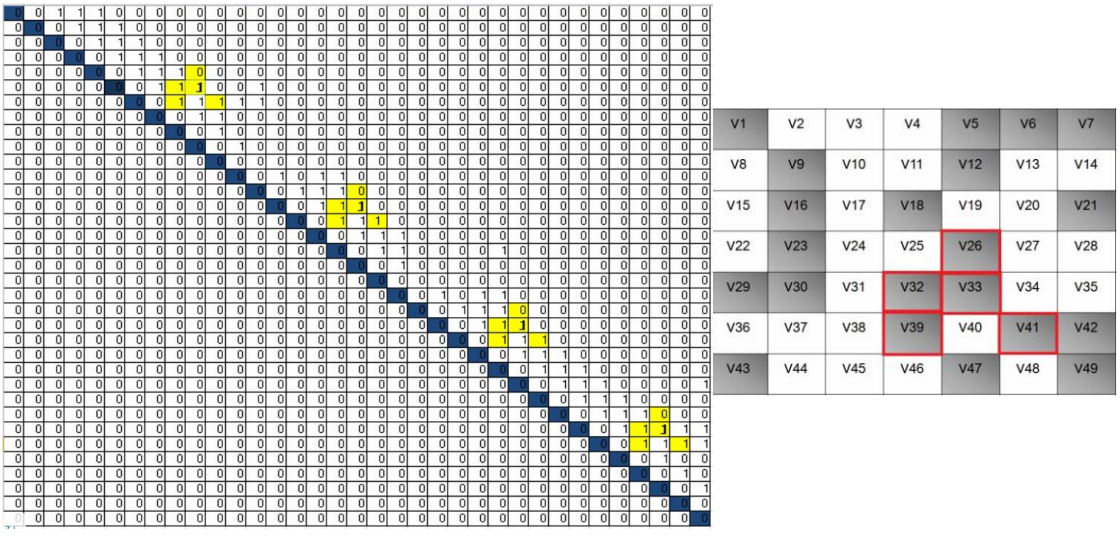

Figura 27: Hélices Alfa de Villin Headpiece Vistas en un Mapa de Contacto

3.5 FASE 5: EVALUACIÓN

En esta fase de la metodología CRISP – DM se evalúa el grado con el que el modelo alcanza los objetivos de negocio, y se determina si existe alguna razón que demuestre, desde la perspectiva del negocio, que el modelo construido es deficiente.

3.5.1 Evaluación de Resultados

La Estrategia co – evolutiva implementada, es capaz de identificar modelos de AC en trayectorias de plegamiento de proteínas, con una precisión superior al 97%, por lo tanto se puede concluir que la forma como fue planteada, es adecuada para alcanzar los objetivos propuestos.

Con respecto a las medidas de calidad establecidas en el diseño de la prueba, se obtuvieron los siguientes resultados:

- La precisión que ofrece este individuo es de 0.974.
- Especificidad = 0,982.
- Sensitividad = 0.876.
- MCC = 0.821
- Distancia con el predictor perfecto en el espacio ROC = 0.124.

Debido a que la Especificidad es mayor que la Sensitividad, es posible afirmar que, aunque el modelo construido predice contactos y no contactos con una precisión de 0,97, es mejor prediciendo no contactos, que prediciendo contactos. Adicionalmente, a pesar que la precisión es alta, existe un 3% de incertidumbre que corresponde a las características del fenómeno que no se están consideradas en el modelo de autómata celular construido.

Por otra parte, aunque la vecindad encontrada incluye 22 celdas, lo que la hace más grande que la del modelo de AC de referencia, permite que el modelo de AC al que está asociada, prediga contactos y no contactos con una precisión que supera el umbral definido en los objetivos.

Otro aspecto importante del modelo de AC encontrado es que, al analizar la tabla de reglas asociada a él, es posible observar que el modelo captura aspectos importantes de la dinámica subyacente en el fenómeno biológico [63].

3.5.2 Revisión del Proceso

Para confirmar que los parámetros encontrados tras el ajuste manual eran los más adecuados, se realizaron más de 14 ejecuciones adicionales, obteniendo en todas estas resultados considerablemente buenos que aunque superan los umbrales de las medidas de calidad establecidas en el objetivo, no tienen un mejor desempeño que el del mejor modelo analizado en la evaluación del modelo.

3.6 FASE 6: DESPLIEGUE

3.6.1 Reporte Final

En esta parte del documento se describe la experiencia adquirida y los resultados obtenidos a lo largo del proceso de minería de datos. Para empezar, durante la fase de entendimiento del negocio, se seleccionaron documentos que incluían información que contribuyó a la contextualización y comprensión del fenómeno de estudio. Gracias a la ejecución de esta fase de la metodología, fue posible identificar falencias en las investigaciones realizadas anteriormente, permitiendo definir un enfoque adecuado para la estrategia.

En la segunda y tercera fase, para el entendimiento, la selección y preparación de los datos, fue de mucha utilidad la información de proyectos de investigaciones relacionadas, ya que gracias a los resultados obtenidos en ellos fue posible agilizar la selección y procesamiento de los datos.

Se pudo confirmar el acierto en las decisiones tomadas en las primeras tres etapas, durante las fases de modelado y evaluación, ya que habían buenas bases para la construcción de la estrategia, que al final permitió la obtención de modelos de AC que reproducen una trayectoria de plegamiento de proteína con alta precisión, superior al 95%

3.6.2 Despliegue de resultados

Los resultados del proceso de minería de datos serán desplegados mediante una Herramienta software que apoyada en el framework CAIF – PFT, implementa una Estrategia Co - Evolutiva que permite identificar modelos de AC para trayectorias de plegamiento de proteínas.

4 ESTRATEGIA CO – EVOLUTIVA

4.1 DESCRIPCIÓN DE LA TÉCNICA A IMPLEMENTAR

Se implementó una estrategia co – evolutiva, soportada en el patrón arquitectónico del framework CAIF-PFT [2], para la identificación de modelos de AC que extraen reglas de evolución a partir de la información proporcionada por una trayectoria de plegamiento de proteína obtenida mediante dinámica molecular.

En la Figura 28 puede apreciarse un esquema conceptual que explica a grandes rasgos cómo se definieron las entradas, el procesamiento y la salida esperada del algoritmo co – evolutivo (ACE) definido en la sección 3.4 de la fase de Modelado.

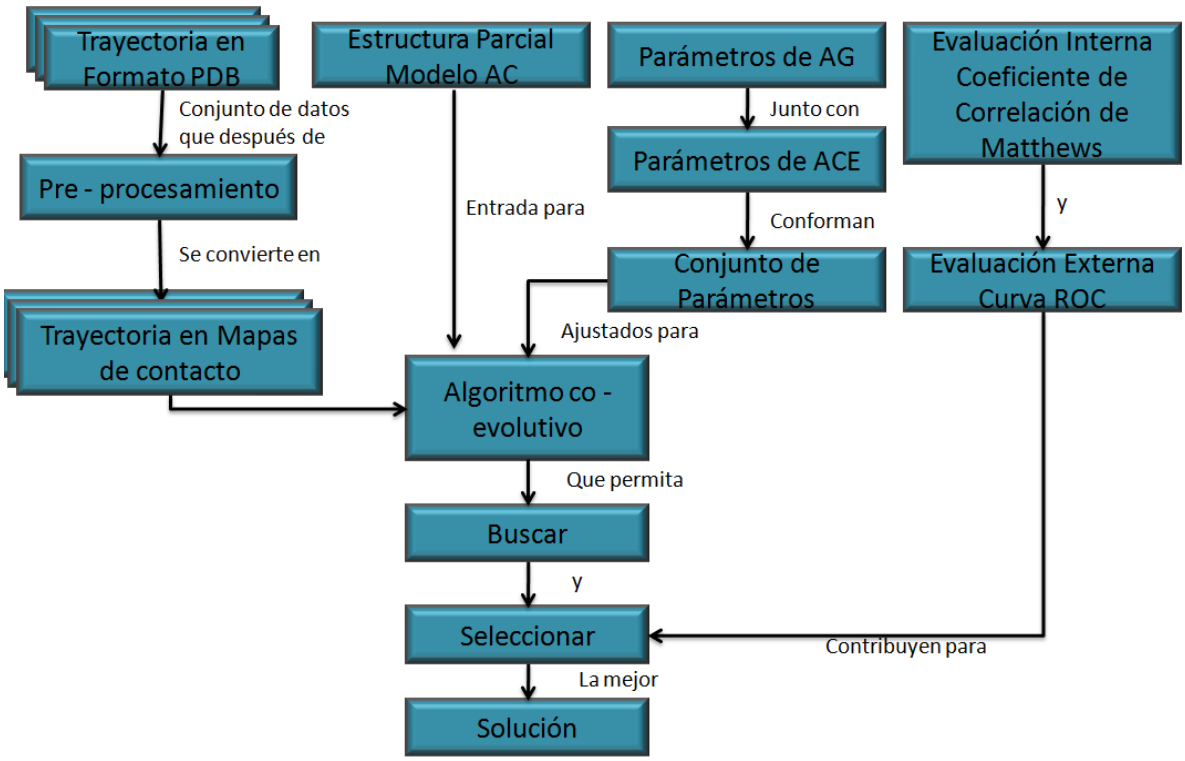

Figura 28: Esquema Conceptual: Definición ACE

4.2 CASOS DE USO.

Uno de los artefactos más importante de la metodología XP son las historias de usuario, ya que estas junto con las pruebas, son la materia prima para el desarrollo. Respecto a las historias de usuario, en investigaciones previas como [64][65] se demostró que en algunas aplicaciones de la metodología XP es más conveniente utilizar casos de uso que historias de usuario, ya que los casos de uso son buenos para mostrar los caminos alternativos de una característica específica y para lograrlo con historias de usuario deberían escribirse los caminos excepcionales en historias de usuario diferentes lo cual implicaría una adaptación de la técnica en lugar de un plan de uso, lo que contrasta con los casos de uso.

Adicionalmente, de acuerdo a lo definido en [2], los casos de uso complementan las historias de usuario, y son necesarios para instanciar el Framework dado que en su diseño se presenta una relación casi 1:1 entre casos de uso e historias de usuario.

Partiendo de antecedentes aplicados a las condiciones propias del proyecto y de las instrucciones de uso del framework CAIF – PFT se tomó la decisión de sustituir las historias de usuario por casos de uso, cada uno de los cuales se relacionó posteriormente a un caso de prueba.

El diagrama de casos de uso general de la aplicación se muestra en la figura 29, el cual está compuesto por cinco casos de uso, unos ejecutados directamente por el usuario, y otros por el sistema.

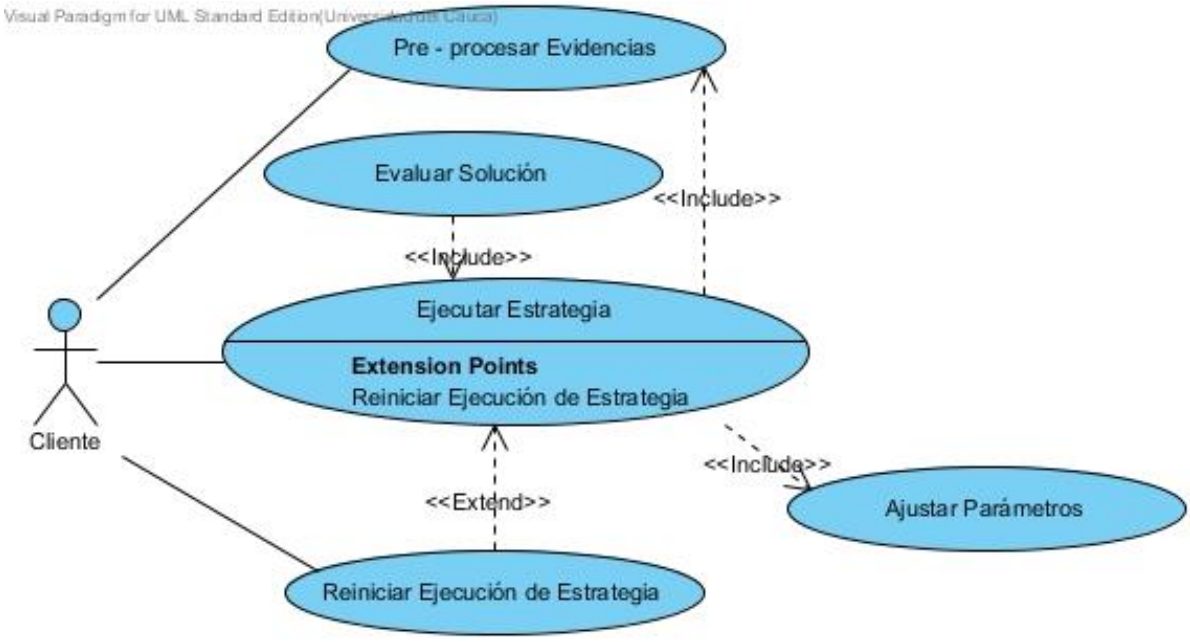

Figura 29: Diagrama Casos de Uso

La implementación de la aplicación esta soportada en el framework CAIF – PFT [2], cuya utilización en la implementación de aplicaciones para la identificación de modelos de AC está dirigida por recetas. A continuación se muestra junto con las plantillas de cada caso de uso, las recetas con las que está relacionado según la metodología de adaptación del framework y las clases que intervienen

4.2.1 Plantillas de los Casos de Uso:

CU001	**PRE- PROCESAR EVIDENCIAS**
DESCRIPCIÓN	La aplicación deberá permitir al usuario indicar la ubicación de una trayectoria de plegamiento de proteínas en formato PDB, para efectuar sobre ellas un pre – procesamiento, según se describe en el siguiente caso de uso.
PRE-CONDICIÓN	Se tiene disponible una trayectoria de simulación de Plegamiento de proteína en formato PDB en un medio de almacenamiento secundario con permisos de lectura y escritura. La trayectoria puede ser proporcionada en un único archivo o en múltiples archivos ubicados en un directorio raíz común.
SECUENCIA NORMAL	1. El usuario proporciona la ubicación de la trayectoria de simulación de plegamiento en formato PDB, y selecciona la opción de pre – procesar las evidencias. 2. La aplicación realiza el pre-procesamiento de la trayectoria proporcionada y genera los mapas de contacto correspondientes.
POST-CONDICIÓN	En la ruta proporcionada se crea un directorio que contiene un conjunto de mapas de contacto correspondiente a cada modelo PDB presente en la trayectoria de plegamiento original.
EXCEPCIONES	La ubicación de la trayectoria proporcionada no es correcta. El programa informa el error al usuario y finalizaría la aplicación.
FRECUENCIA	Este caso de uso se espera que se lleve a cabo mínimo una vez.

El Framework CAIF – PFT proporciona una funcionalidad para pre – procesar evidencias; para hacer uso de ella, se deben seguir los pasos estipulados en las recetas de la documentación del Framework.

Las recetas relacionadas son la receta 2: Pre - procesamiento de trayectorias de plegamiento, receta 1: Definir estructura para manejar evidencias y la receta 2.1: Pre - procesar trayectoria en formato PDB.

La estructura del punto de variación resultante al seguir las recetas para implementar el caso de uso se ilustra en la Figura 30. Donde se muestran las clases del framework CAIF – PFT involucradas, y la relación que se establece con las clases de la aplicación que implementada.

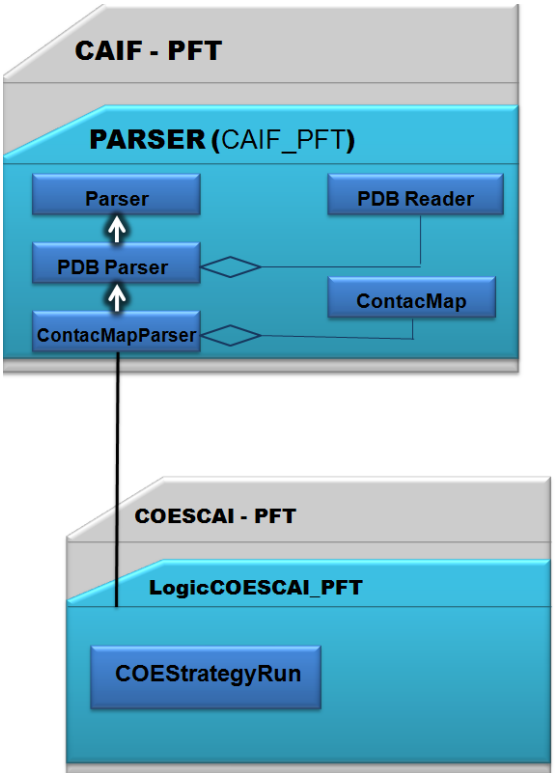

Figura 30: Estructura punto de variación- caso de uso PRE- PROCESAR EVIDENCIAS.

CU002	**EVALUAR SOLUCIÓN**
DESCRIPCIÓN	La aplicación deberá evaluar la mejor solución encontrada tras la ejecución de la estrategia co - evolutiva, según se describe en el siguiente caso de uso.
PRE-CONDICIONES	1. La estrategia debió ejecutarse mínimo una vez. 2. El archivo que contiene la solución a ser evaluada se encuentra disponible en un medio de almacenamiento secundario con acceso de lectura. 3. La trayectoria en formato de mapas de contactos que debería producir la simulación del modelo de AC correspondiente a la solución.
	1. La aplicación valida que sea posible cargar la trayectoria de

SECUENCIA NORMAL	mapas de contacto disponibles para efectuar la evaluación y la solución a evaluar. 2. Se inicia la simulación del modelo de AC correspondiente a la solución indicada por el usuario, y se arrojan los valores correspondientes la evaluación de la solución.
POST-CONDICIÓN	Las medidas de calidad especificadas definidas en la aplicación calculadas sobre una simulación del modelo de AC correspondiente a la solución indicada, más un archivo de texto con el árbol de decisión de las reglas del modelo.
EXCEPCIONES	No es posible cargar la trayectoria en mapas de contacto o la solución a evaluar. Se informa el error y la aplicación termina.
FRECUENCIA	Este caso de uso se espera que se lleve a cabo mínimo una vez.

Las recetas relacionadas con la implementación de este caso de uso son la receta 6: Implementar evaluador de solución, receta 4: Definir una estrategia concreta y la receta 7: Definir ajustes de la estrategia concreta.

La estructura del punto de variación, en el que se relacionan las clases del framework CAIF – PFT y las clases de la aplicación, se muestra en la Figura 31.

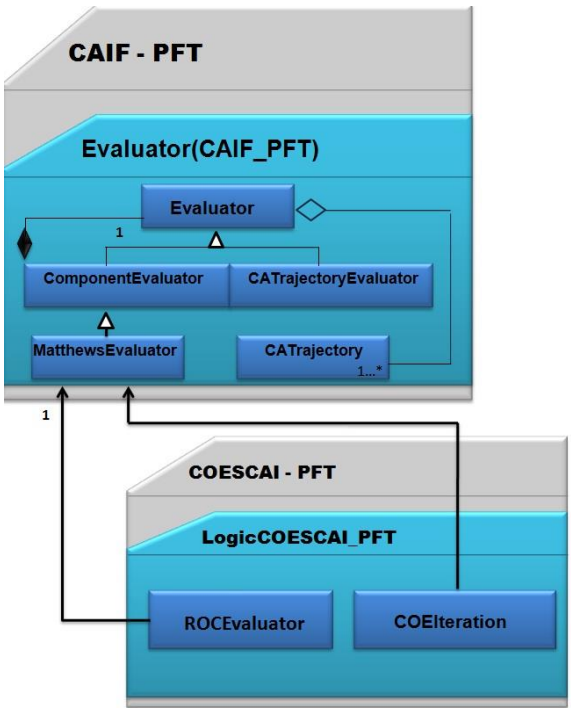

Figura 31: Estructura punto de variación - caso de uso EVALUAR SOLUCIÓN

CU003	**EJECUTAR ESTRATEGIA**
DESCRIPCIÓN	La aplicación deberá permitir a el usuario iniciar la ejecución de la estrategia co - evolutiva definida, según se describe en el siguiente caso de uso:
PRE-CONDICIÓN	Se encuentra operativo el ambiente de ejecución distribuido para la aplicación.
SECUENCIA NORMAL	1. Si el usuario desea reutilizar los parámetros que ha definido en una ejecución previa, y en dicha ejecución se ha pre - procesado una trayectoria de plegamiento de proteínas, puede seleccionar la opción cargar parámetros. Si no, inclusión al caso de uso ajustar parámetros. 2. Si el usuario cuenta con una trayectoria de plegamiento de proteína pre-formateada a mapas de contacto almacenada en memoria secundaria, en una ubicación con permisos de acceso para lectura, con los nombres de archivo que permitan que se ordenen sin alterar la consecutivita correspondiente al paso de simulación que representa, puede proporcionar su ubicación. Si no, inclusión al caso de uso pre – procesar evidencias. 3. La aplicación valida que sea posible cargar la trayectoria de mapas de contacto y que los parámetros proporcionados sean correctos. 4. Se inicia la ejecución de la estrategia co - evolutiva hasta que se cumpla la condición de parada: Número máximo de iteraciones; ó se presenta decrecimiento en el proceso de búsqueda de mejores soluciones medida por el promedio acumulado de las mejores soluciones en cada iteración versus la media móvil de las últimas 40 generaciones.
POST-CONDICIONES	1. Un conjunto de las mejores soluciones, correspondiente a los individuos de la estrategia co - evolutiva evaluados durante cada iteración. 2. Un conjunto de poblaciones para una nueva ejecución, generadas a partir de la última generación evaluada durante la ejecución de la estrategia co - evolutiva. 3. Un reporte de las estadísticas de ejecución de la estrategia co - evolutiva.
EXCEPCIONES	Si no es posible cargar la trayectoria de mapas de contacto o los parámetros proporcionados no son correctos, se informa el error y la aplicación termina.
FRECUENCIA	Este caso de uso se espera que se lleve a cabo mínimo una vez.

Para implementar este caso de uso, se siguieron las recetas 4: Definir estrategia concreta, 3: Definir contexto para la ejecución de la técnica de identificación de modelos de AC a partir de trayectorias de plegamiento de proteína, receta 1: Definir estructura para manejar evidencias, 2: Pre procesar trayectoria, 4: Definir una estrategia concreta, y la receta 5: Definir representación de solución. La estructura del punto de variación se muestra en la Figura 32.

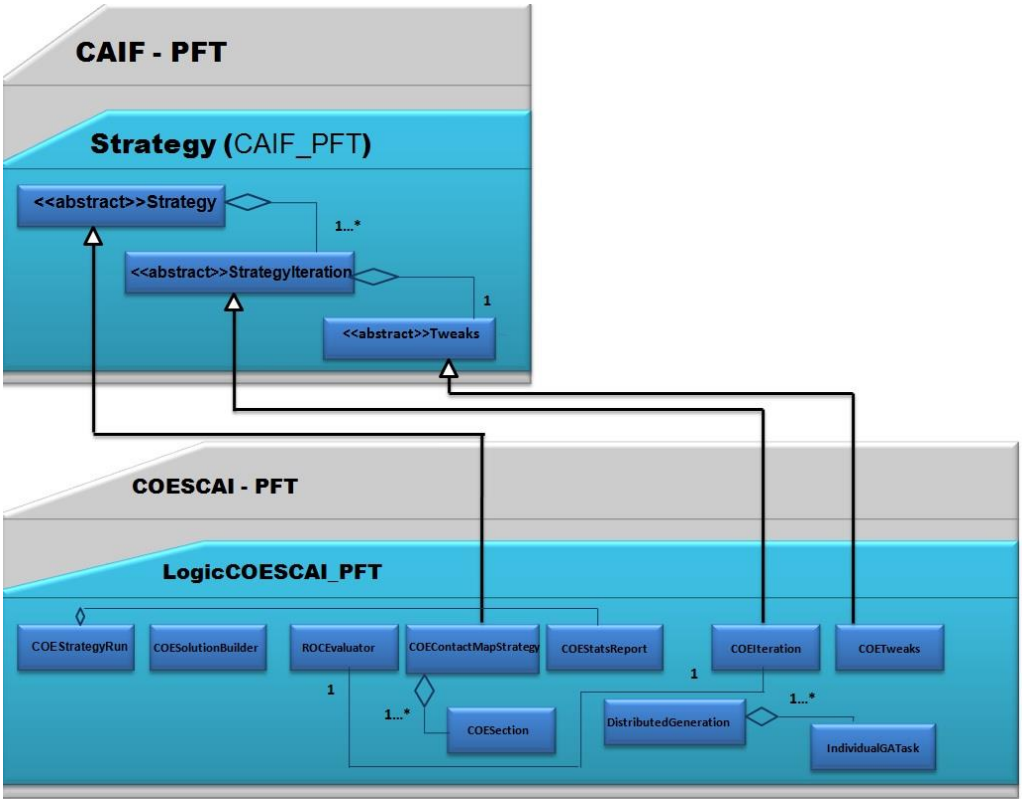

Figura 32: Estructura punto de variación del caso de uso EJECUTAR ESTRATEGIA.

CU004	**REINICIAR EJECUCION DE ESTRATEGIA**
DESCRIPCIÓN	La aplicación deberá permitir al usuario recuperar una ejecución en caso de fallos eléctricos o de conexión, o en caso de interrumpir la ejecución, según se describe en el siguiente caso de uso.
PRE-CONDICIONES	1. La trayectoria de plegamiento de proteína pre-formateada a mapas de contacto y el archivo de configuración de parámetros se encuentran almacenados en memoria secundaria, en una ubicación con permisos de acceso para lectura. 2. Está disponible el conjunto de poblaciones correspondiente a

	una ejecución anterior del algoritmo, en un medio de almacenamiento secundario con permisos de lectura. 3. Está disponible el reporte de la anterior ejecución anterior la estrategia co - evolutiva en un medio de almacenamiento secundario con permisos de lectura. 4. El entorno distribuido se encuentra operativo.
SECUENCIA NORMAL	1. El usuario selecciona la opción reiniciar estrategia. 2. La aplicación carga como población inicial el conjunto de poblaciones generadas en la última iteración llevada a cabo en la anterior ejecución del algoritmo, además de la iteración a la cual corresponden esas poblaciones. También el reporte de las estadísticas de la ejecución anterior del algoritmo. 3. A partir de este conjunto de poblaciones y de los parámetros que se fijaron durante la última ejecución, se reiniciará la ejecución del algoritmo hasta que se cumpla la condición de parada: Número máximo de iteraciones; ó se presenta decrecimiento en el proceso de búsqueda de mejores soluciones medida por el promedio acumulado de las mejores soluciones en cada iteración versus la media móvil de las últimas 40 generaciones.
POST-CONDICIONES	1. Un conjunto de las mejores soluciones, correspondiente a los individuos de la estrategia co - evolutiva evaluados durante cada iteración. 2. Un conjunto de poblaciones para una nueva ejecución, generadas a partir de la última generación evaluada durante la ejecución de la estrategia co - evolutiva. 3. Un reporte de las estadísticas de ejecución de la estrategia co - evolutiva.
EXCEPCIONES	1. Si no es posible cargar la trayectoria en mapas de contacto, el archivo de estadísticas, o los parámetros requeridos para la ejecución, se informa el error y la aplicación termina. 2. Sí no está disponible el reporte de la anterior ejecución anterior la estrategia co - evolutiva, la ejecución se hará como una nueva y no como reinicio de una anterior.

CU005	**AJUSTAR PARÁMETROS**
DESCRIPCIÓN	La aplicación deberá permitir al usuario indicar los parámetros necesarios para la ejecución de la estrategia co – evolutiva, según se describe en el siguiente caso de uso.
PRE-CONDICIÓN	Ninguna

SECUENCIA NORMAL	1. El usuario define los siguientes parámetros: ● *Número máximo de generaciones*. El usuario debe proporcionar un número natural entre 1 y 2000 con el cual indica el número de iteraciones. ● *Número de secciones*. El usuario debe proporcionar un número natural no primo que además de indicar el número de secciones en el que se dividirá el Dataset, indica también el número de poblaciones que se considerarán en la ejecución de la estrategia co – evolutiva. Adicionalmente, el usuario define cómo se distribuirán las n secciones en el espacio co - evolutivo, para ello debe definir: ● Número de columnas ● Número de filas Ambos valores proporcionados deben ser números naturales divisores del número que se estableció como la cantidad de secciones. ● *Parámetros de un algoritmo genético*. ● 1. El usuario define el *tamaño de la población* y su distribución, para esto debe proporcionar un número natural no primo que indicará el número de individuos que tendrá cada una de las poblaciones que se evaluarán con la estrategia co – evolutiva. Adicionalmente, el usuario define cómo se distribuirán los n individuos en cada una de las secciones del espacio co - evolutivo, para ello debe definir: ● Número de columnas ● Número de filas Ambos valores proporcionados deben ser números naturales divisores del número que se estableció como el tamaño de la población. ● 2. El usuario define la *probabilidad de selección,* para esto debe proporcionar un número flotante entre 0 y 1 que indica la probabilidad para elegir los mejores individuos. ● 3. El usuario define la *probabilidad de cruce*. El usuario debe proporcionar un número flotante entre 0 y 1 que indica indicar la frecuencia con la que se producen cruces entre los mejores individuos. Si la probabilidad definida es 0, los individuos de la siguiente generación serán

	copias de los individuos seleccionados para la reproducción. Si por el contrario la probabilidad de cruce es 1 (ó del 100%), el "hijo" será creado totalmente por cruce y no por partes. ● 4. El usuario define la *probabilidad de mutación*, para esto debe proporcionar un número flotante entre 0 y 1 para indicar la frecuencia con la que los genes del cromosoma de un individuo son modificados. Si la probabilidad de mutación es 0, no habrán cambios, pero si por el contrario es 1 (ó 100%) la totalidad del cromosoma se cambia. 2. La aplicación valida si los valores proporcionados por el usuario son correctos y los almacena en un archivo de texto.
POST-CONDICIÓN	En un archivo de texto Config.txt se almacenan los valores dados a los parámetros.
EXCEPCIONES	Los parámetros proporcionados no son correctos, la aplicación informa del error y permite modificar los valores.
FRECUENCIA	Este caso de uso se espera que se lleve a cabo mínimo una vez.

4.3 DIAGRAMA DE PAQUETES

El diagrama de las paquetes que se utilizó en la implementación de la estrategia Co – Evolutiva se muestra en la sección 2.1 de los Anexos. Allí es posible observar el paquete COESCAIF – PFT, que contiene las clases de la aplicación implementada, además se observa el paquete del framework CAIF – PFT que es el que proporciona el soporte a la aplicación. Además se observa el soporte al esquema distribuido que proporcionan las clases de PYRO.

5 ANÁLISIS Y RESULTADOS

5.1 PRUEBAS ALGORITMO CO – EVOLUTIVO

En los procesos de minería de datos, la prueba que generalmente se utiliza es Holdout Validation, la cual consiste en dividir el dataset en dos, una parte para la obtención de modelos (entrenamiento), y la otra parte para validar los modelos obtenidos (pruebas). La división del dataset utilizado en este proyecto de investigación, se realizó como se explica en la sección “Manejo del Volumen de los Datos”, donde se estableció que el 67.5% de los datos se utilizarían para entrenamiento, y el 32.5% restante para pruebas.

Las pruebas que se documentan en esta sección hacen referencia a las ejecuciones realizadas con el objetivo de confirmar que los parámetros encontrados durante la afinación sí arrojan buenos resultados en términos de convergencia y desempeño de los modelos de AC encontrados, dichos parámetros se muestran en la tabla 6

Tabla 6. Parámetros para Pruebas

Parámetros	Valores
Número Máximo de iteraciones	1000
Número de secciones	9
Número de columnas secciones	3
Número de filas secciones	3
Tamaño de la población	25
Número de columnas Población	5
Número de filas Población	5
Probabilidad de selección	0,07
Probabilidad de cruce	0,9
Probabilidad de mutación	0,03

Los resultados obtenidos en las ejecuciones fueron evaluados desde dos perspectivas:

- La obtención de modelos o entrenamiento, se evalúa con base en los reportes estadísticos que medían el desempeño del algoritmo en cada iteración.
- La validación de los modelos obtenidos se evaluaron de acuerdo a las medidas definidas en la sección “Diseño de la Prueba.”

5.1.1 PRUEBA N° 1.

5.1.1.1 Obtención de Modelos - Prueba No. 1

En la Figura 33 se ilustra el fitness interno y externo de cada uno de los mejores individuos obtenidos. Se puede observar que el fitness interno del mejor individuo durante todas las iteraciones fue inferior a 0.87. Por otra parte, con respecto al fitness externo se puede apreciar que no fue superior a 0.055, lo que indica que los individuos encontrados estaban a una distancia relativamente pequeña del predictor perfecto en el espacio ROC.

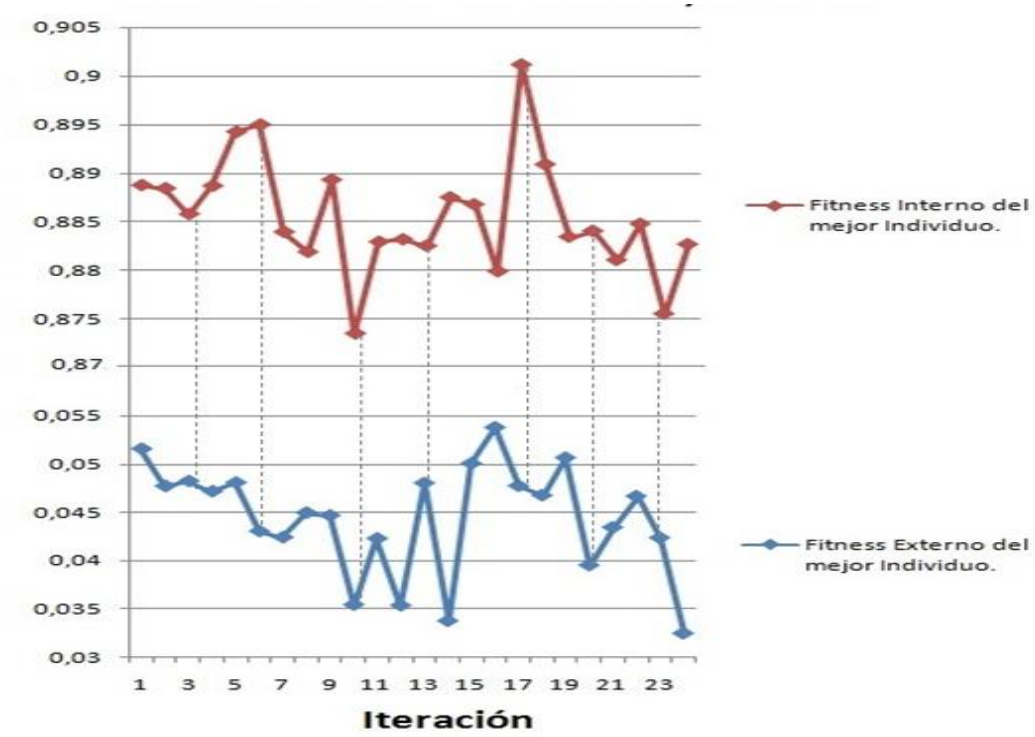

Figura 33: Fitness Interno vs Fitness Externo del Mejor Individuo - Prueba N° 1.

Al observar la Figura 34 donde se ilustran la diversidad de población y la media acumulada del fitness interno, es posible apreciar que la diversidad de población se mantiene por encima del 80% en todas las iteraciones y que la media acumulada del fitness interno tiende a estabilizarse.

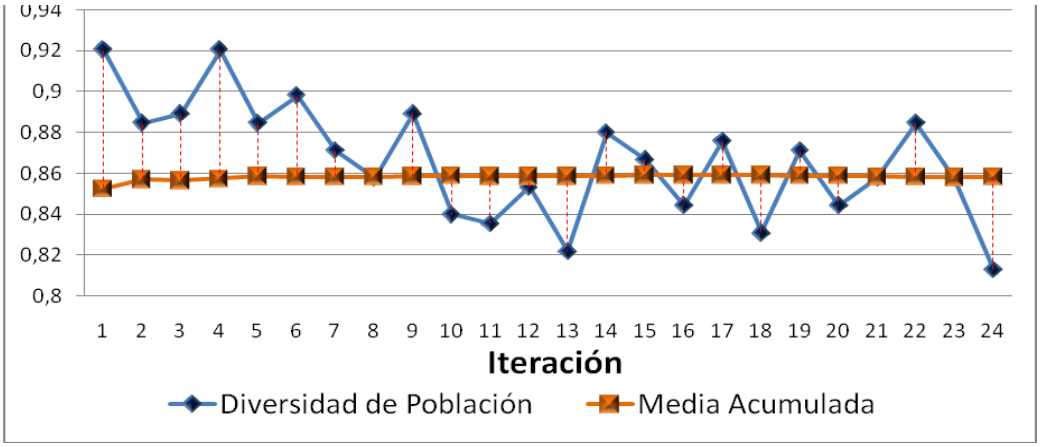

Figura 34: Media Acumulada y Diversidad de Población, Prueba N° 1

5.1.1.2 Validación de Modelos – Prueba No. 1

Tras realizar las pruebas del mejor modelo encontrado en la primera ejecución de prueba, se calcularon las medidas de calidad establecidas en el Diseño de la prueba, obteniendo para este caso, los valores mostrados en la tabla 7.

Tabla 7. Medidas de Calidad de Prueba No. 1

Medida Calculada	Valor
Precisión	0.962194823027
Especificidad	0.965923753297
Sensitividad	0.914506767482
MCC	0.768265615165
Distancia con el predictor perfecto en el espacio ROC	0.0920341425545

En la tabla 7 se puede observar, de acuerdo a los valores de las medidas, que proporción de contactos correctamente actualizados por el modelo de AC es de 0.914506767482 y que la probabilidad de predecir correctamente un no contacto es de más del 96%. Sin embargo, el MCC que es una medida que no es sensible a sesgos de clase, es apenas del 76.8%, lo que indica que el modelo, aunque es bueno, no clasifica de manera

satisfactoria los verdaderos positivos y los verdaderos negativos a diferencia del modelo obtenido durante el ajuste de parámetros.

Adicionalmente, si se observa la vecindad del la figura 35, es claro que para representar el individuo de este modelo, se necesita casi el 50% de las celdas de la vecindad de Moore de radio 3, definida inicialmente como vecindad límite, y que la celda central se incluye.

10100100011100000100011011010011100011010111100111

V1	V2	*V3*	V4	V5	*V6*	V7
V8	V9	*V10*	*V11*	*V12*	V13	V14
V15	V16	V17	*V18*	V19	V20	V21
V22	*V23*	V24	*V25*	*V26*	V27	*V28*
V29	V30	*V31*	*V32*	*V33*	V34	V35
V36	*V37*	*V38*	V39	*V40*	V41	*V42*
V43	*V44*	V45	V46	*V47*	*V48*	*V49*

Figura 35: Genotipo y Fenotipo del Mejor Modelo de AC de la Prueba No. 1

Las implicaciones de una representación tan grande, consisten en que, si bien, el modelo de AC asociado simula la trayectoria de plegamiento de proteínas con gran precisión, el costo computacional sería considerablemente grande. Adicionalmente, la información que se puede extraer del árbol de decisión generado a partir de una representación de esta magnitud, puede no suministrar información clara del fenómeno que contribuya en su explicación. Un ejemplo de esta situación es el segmento del árbol de decisión que representa las reglas del mejor modelo de AC de la primera ejecución ilustrado en la figura 36.

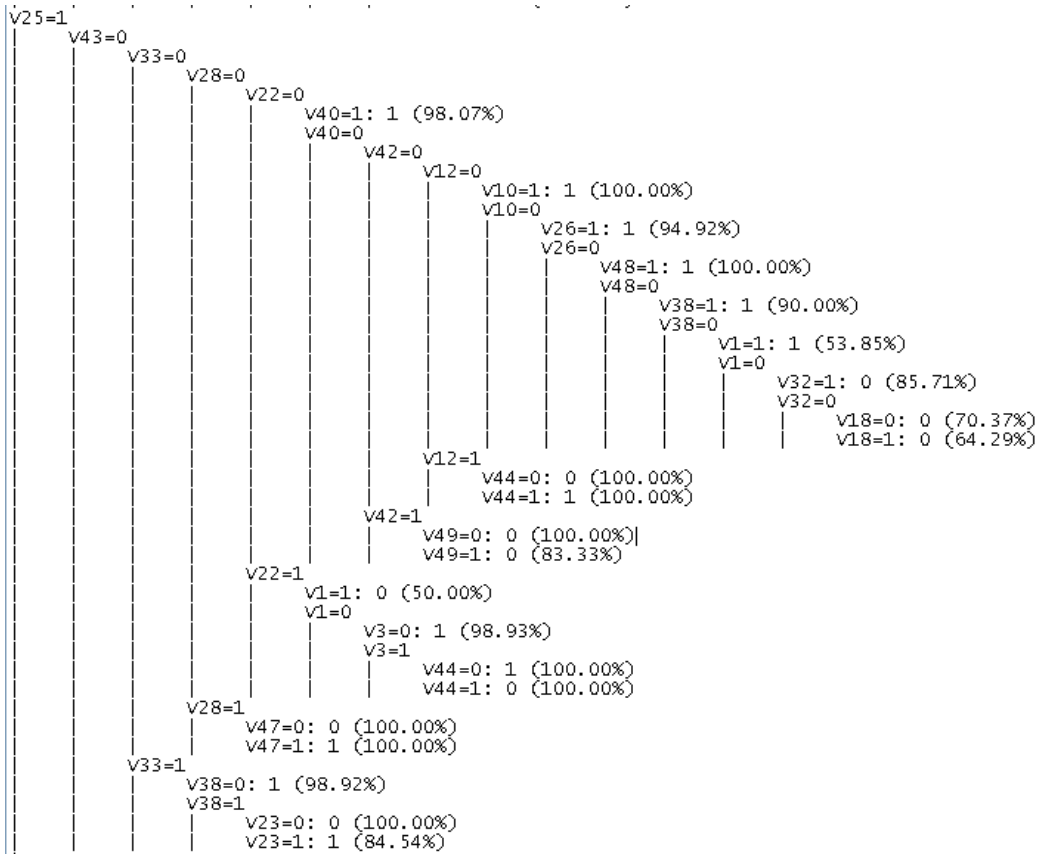

Figura 36: Árbol de Decisión del Mejor Modelo de AC de la Prueba No. 1

5.1.2 PRUEBA N° 2.

5.1.2.1 Obtención de Modelos – Prueba No. 2

Durante esta prueba se pudo observar que el fitness interno del mejor individuo durante todas las iteraciones fue superior a 0.84 (ver Figura 37). Con respecto al fitness externo, el valor más alto que se obtuvo fue de 0.08, lo que es conveniente ya que entre más bajo sea su valor, mayor será la probabilidad de que el modelo de AC obtenido por algoritmo muestre un mejor comportamiento.

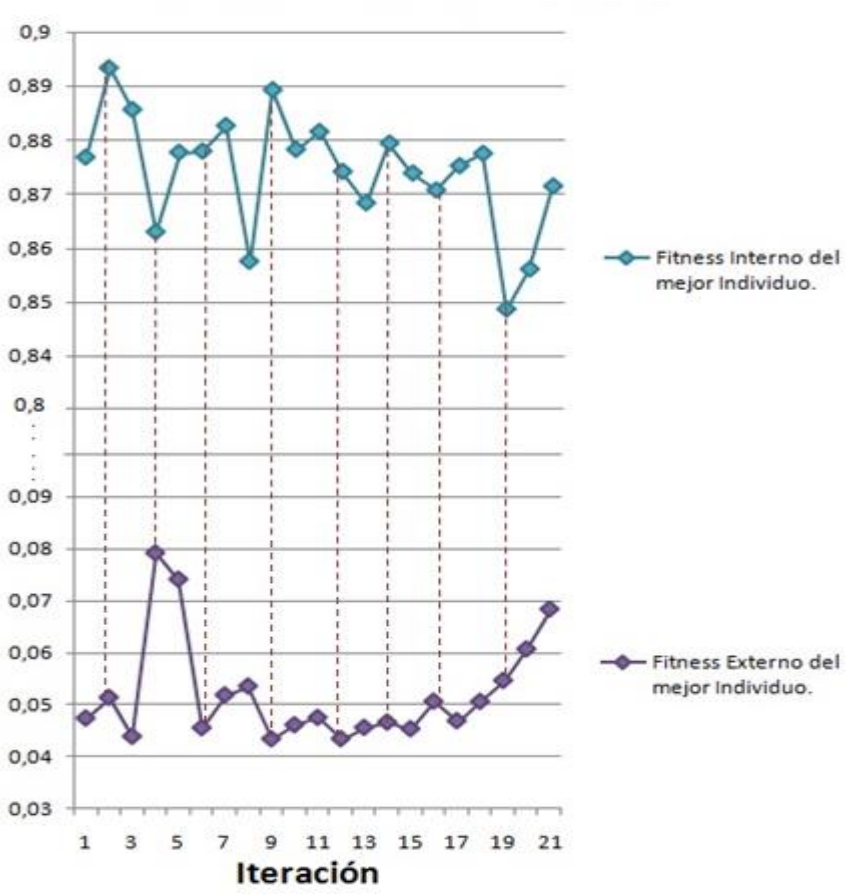

Figura 37: Fitness Interno vs Fitness Externo del Mejor Individuo, Prueba N° 2.

Por otra parte, en la Figura 38 es posible observar que la diversidad de población se mantiene por encima del 81% y la media acumulada del fitness interno tiende a estabilizarse a partir de la octava iteración manteniéndose sobre el 0.81, demostrando así que el desempeño del algoritmo es bueno en términos generales, pues encuentra individuos con un MCC penalizado cercano a 1 indicando que los verdaderos contactos y no contactos son clasificados de manera satisfactoria.

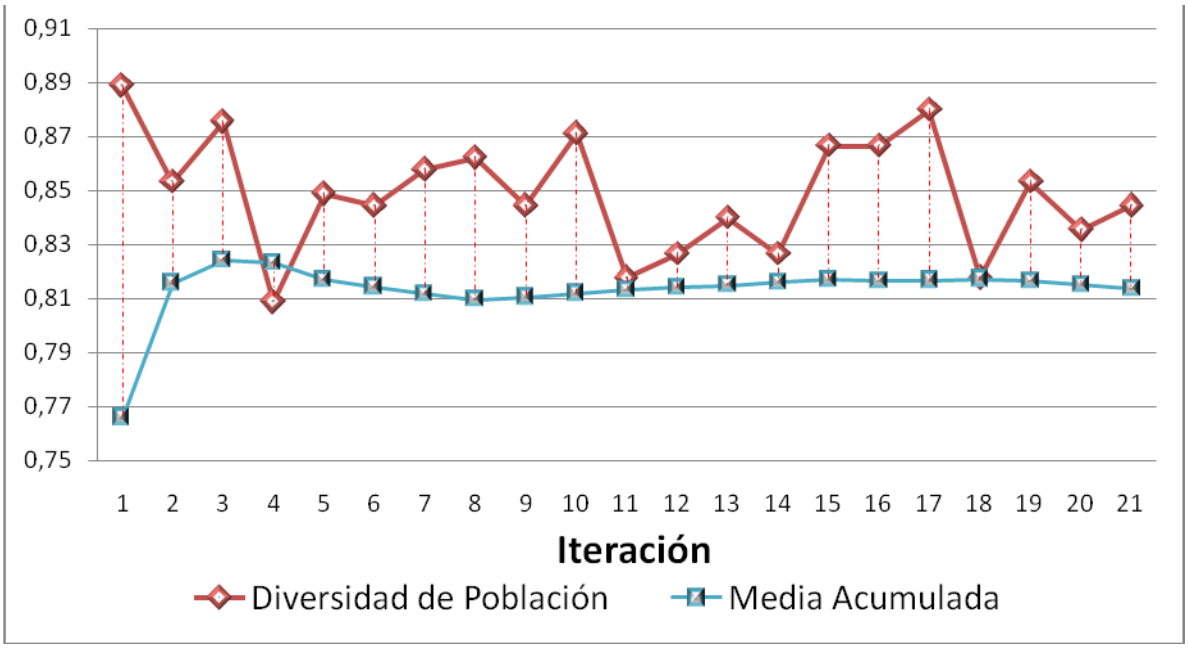

Figura 38: Media Acumulada y Diversidad de Población, Prueba N° 2

5.1.2.2 Validación de Modelos – Prueba No. 2

Tras realizar las pruebas del mejor modelo encontrado en la segunda prueba, se calcularon las medidas de calidad, obteniendo para este caso los valores mostrados en la tabla 8.

Tabla 8. Medidas de Calidad de Prueba No. 2

Medida Calculada	Valor
Precisión	0.993518474056
Especificidad	0.995995803123
Sensitividad	0.961836732221
MCC	0.952131235677
Distancia con el predictor perfecto en el espacio ROC	0.0383727585692

Según la tabla 8 se puede observar, que el mejor modelo de AC encontrado en esta ejecución, es capaz de clasificar verdaderos positivos y los verdaderos negativos de manera satisfactoria, ya que el MCC de este modelo es de 95.21. Adicionalmente, la precisión de modelo supera el 99%, y la probabilidad de que el modelo prediga verdaderos contactos es de 0.96183673222, de la misma manera, el modelo es capaz de predecir verdaderos no contactos con un acierto en el 99.5% de los datos, lo que indica que el modelo está muy cercano a no predecir ningún falso contacto.

El fenotipo y el genotipo del mejor modelo encontrado en la segunda ejecución ilustrado en la figura 39 muestran una vecindad de 28 celdas, equivalente a más del 57% de las celdas de la vecindad de Moore de radio 3, definida inicialmente como vecindad límite, lo cual podría ser una explicación del el evidente buen desempeño del modelo de AC asociado a la vecindad.

Figura 39: Genotipo y Fenotipo del Mejor Modelo de AC de la Prueba No. 2

Las implicaciones negativas de una representación tan grande radican en el alto costo computacional para llevar a cabo la simulación con el modelo de AC. Adicionalmente, la información que se puede extraer del árbol de decisión que representa las reglas del mejor modelo de AC de esta segunda ejecución, ilustrado en la figura 40, no explican claramente el fenómeno de plegamiento de proteínas, inconveniente que se puede deber a que se están considerando atributos que no son representativos que en lugar de aportar información, entorpecen la lectura del árbol de decisión.

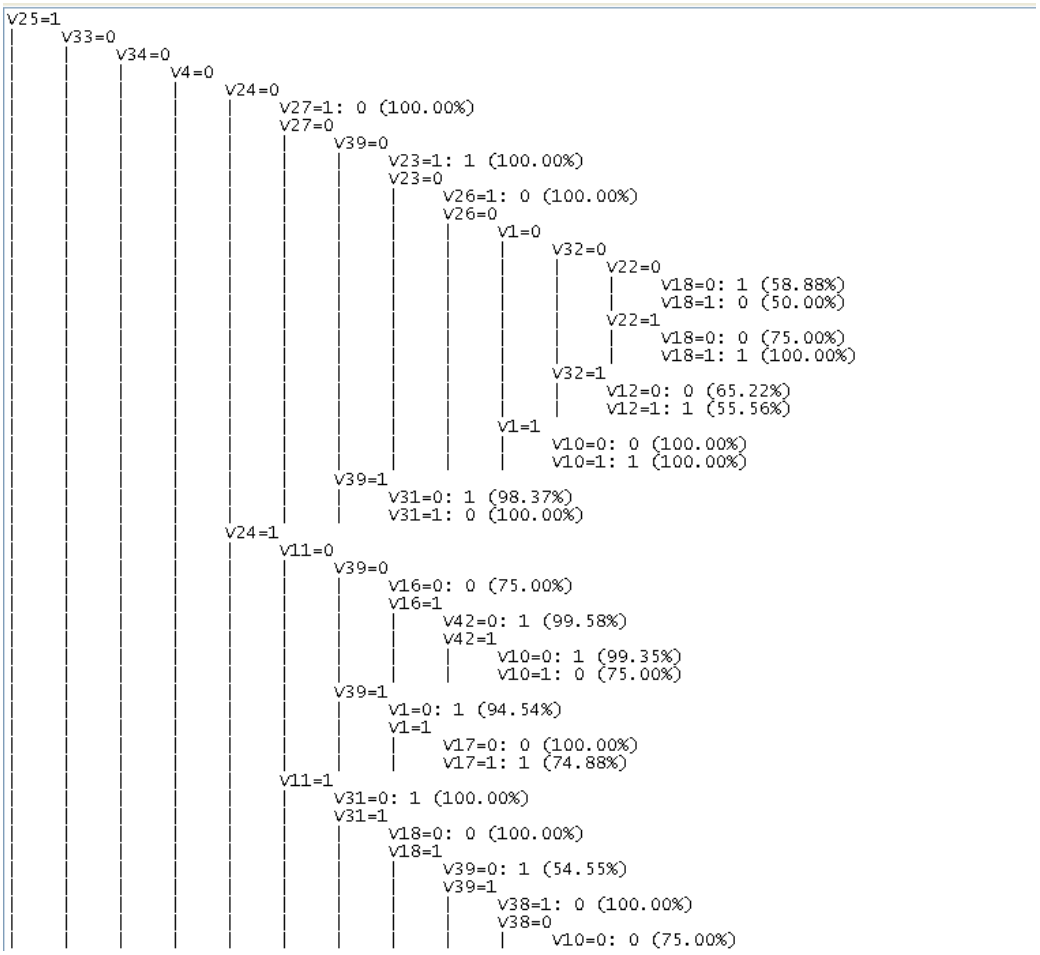

Figura 40: Árbol de Decisión del Mejor Modelo de AC de la Prueba No. 2

5.1.3 PRUEBA N° 3.

5.1.3.1 Obtención de Modelos – Prueba No. 3

En la Figura 41 se ilustra el fitness interno y externo de cada uno de los mejores individuos encontrados en la tercera ejecución. Se puede observar que el fitness interno del mejor individuo durante todas las iteraciones fue superior a 0.84, y que el fitness externo se mantuvo por debajo de 0,06, indicando que los modelos encontrados realizaban buenas predicciones de contactos y no contactos.

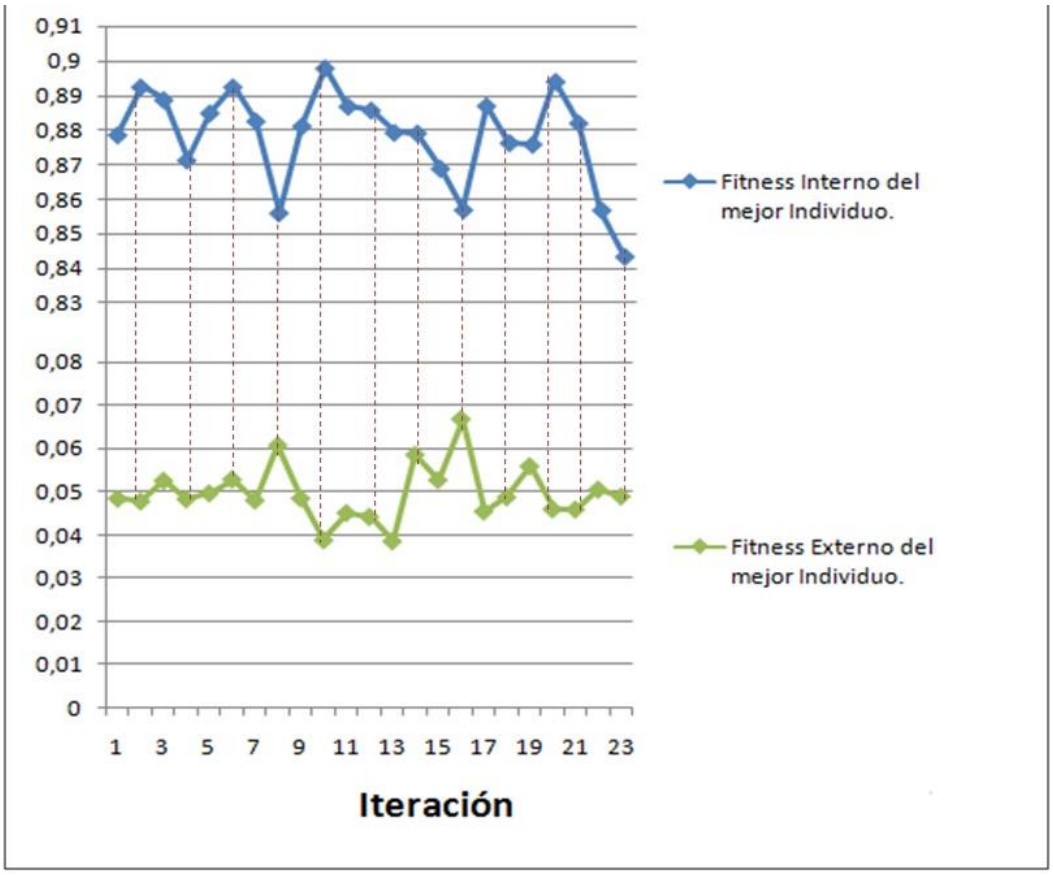

Figura 41: Fitness Interno vs Fitness Externo del Mejor Individuo, Prueba N° 3.

En la Figura 42 se puede observar que la diversidad de población se mantiene por encima del 80% durante todas las iteraciones y que la media acumulada del fitness interno tiende a estabilizarse después de la novena iteración, indicando que al algoritmo avanza rápidamente en la búsqueda de buenas soluciones.

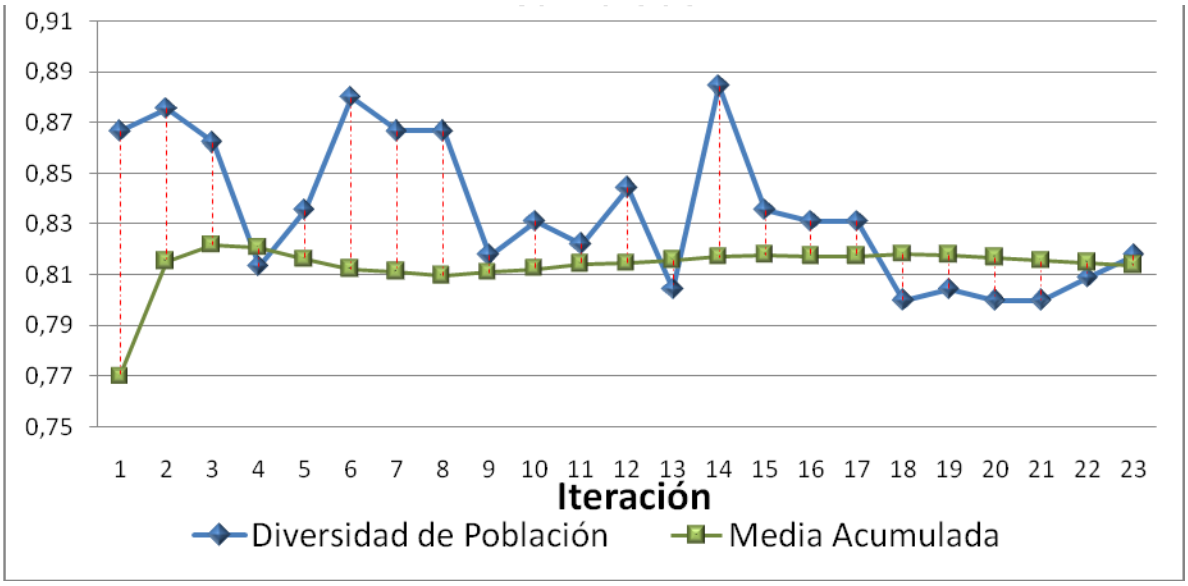

Figura 42: Media Acumulada y Diversidad de Población, Prueba N° 3

5.1.3.2 Validación de Modelos - Prueba No. 3

Tras realizar las pruebas del mejor modelo encontrado en la tercera prueba, se calcularon las medidas de calidad, obteniendo los valores mostrados en la tabla 9.

Tabla 9. Medidas de Calidad de Prueba No. 3

Medida Calculada	Valor
Precisión	0.993518474056
Especificidad	0.995995803123
Sensitividad	0.961836732221
MCC	0.952131235677
Distancia con el predictor perfecto en el espacio ROC	0.0383727585692

Un MCC superior a 95.21 indica que el mejor modelo de AC encontrado en esta ejecución clasifica satisfactoriamente los verdaderos positivos y los verdaderos negativos. Esto se

suma al hecho de que la precisión de modelo supera el 99% lo que indica que los contactos y no contactos son actualizados de manera casi exacta respecto a la trayectoria original. Por otra parte, la probabilidad de que el modelo prediga verdaderos contactos es de más del 96.1%, de la misma manera, el modelo es capaz de predecir verdaderos no contactos con una probabilidad superior al 99%, lo que indica que el modelo está muy cercano a no predecir ningún falso positivo.

El fenotipo y el genotipo del mejor modelo, ilustrados en la figura 43 muestran una vecindad de 26 celdas, equivalente a más del 53% de las celdas de la vecindad de Moore de radio 3, definida inicialmente como vecindad límite. Las implicaciones negativas de una representación tan grande radican en el alto costo computacional para llevar a cabo la simulación con el modelo de AC.

Figura 43: Genotipo y Fenotipo del Mejor Modelo de AC de la Prueba No. 3

La información que se puede extraer del árbol de decisión que representa las reglas del mejor modelo de AC de esta tercera ejecución, ilustrado en la figura 44, no explican claramente el fenómeno de plegamiento de proteínas, puesto que incluye vecinos demasiado lejanos a la celda que es considerada como clase.

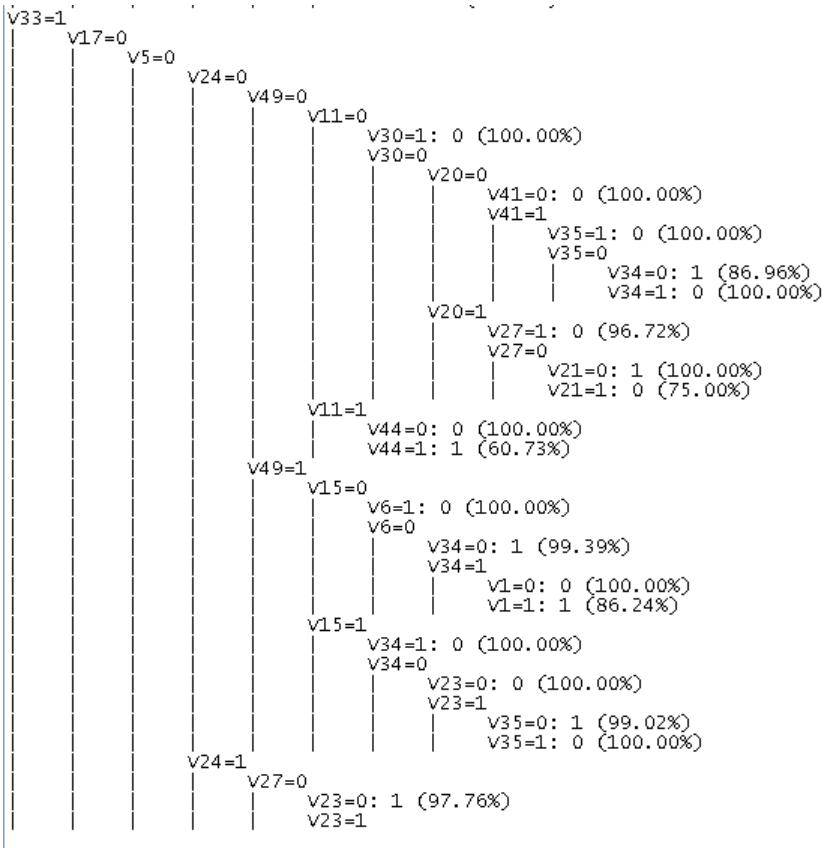

Figura 44: Árbol de Decisión del Mejor Modelo de AC de la Prueba No. 3

5.2 ANÁLISIS GENERAL DE RESULTADOS

5.2.1 Obtención de Modelos

Los resultados obtenidos en la etapa de entrenamiento o también conocida como etapa de obtención de modelos, se muestran en la tabla 10.

Tabla 10. Resultados Etapa de Entrenamiento

Número de Prueba	Prueba N° 1	Prueba N°2	Prueba N°3	Promedio Total
Promedio fitness interno mejor individuo	0,886	0,874	0,878	0.879
Max fitness Interno mejor individuo	0,901	0,893	0,898	0.897
Min fitness Interno mejor individuo	0,873	0,849	0,843	0.855
Promedio fitness Externo mejor individuo	0,044	0,052	0,050	0.048
Max fitness Externo mejor individuo	0,054	0,079	0,067	0.066
Min fitness Externo mejor individuo	0,032	0,043	0,039	0,038
Media Acumulada	0,858	0,813	0,813	0,828
Max Media Acumulada	0,859	0,824	0,822	0,835
Min Media Acumulada	0,853	0,766	0,770	0,796
Diversidad de población	0,866	0,849	0,835	0,850
Max Diversidad de población	0,920	0,889	0,884	0,898
Min Diversidad de población	0,813	0,809	0,800	0,807

De acuerdo con los resultados tabulados y a los apreciados en los reportes estadísticos de cada ejecución, es posible realizar las siguientes observaciones:

- La diversidad de población en todas las ejecuciones se mantuvo sobre el 80%, lo que significa que el algoritmo no llegó a encapsularse en máximos locales, confirmando una vez más que las estrategias co – evolutivas tienden a explorar de manera intensiva el espacio de búsqueda.
- La media acumulada del fitness interno tendió a estabilizarse, en todas las ejecuciones, entre las segunda y novena iteración, lo que demuestra que el algoritmo converge rápidamente en la búsqueda de buenas soluciones.
- La constante variación de las medidas de desempeño interno y externo, es el resultado de la división del dataset en secciones y estratos ILAS. Este comportamiento del algoritmo, aunque parece indeseable y da la sensación de inestabilidad, permite encontrar modelos de AC que no se sobre ajustan a los datos de entrenamiento.
- El fitness externo, que medía la distancia con el predictor perfecto, fue en promedio inferior a 0.04, lo que indica que la tasa de falsos positivos era pequeña, comparada con la tasa de verdaderos positivos, que era cercana a 1.

Considerando que el algoritmo que implementa la estrategia Co - Evolutiva tenía 3 criterios de parada, uno relacionado con la diversidad de población, otro relacionado con el desempeño de los individuos y el último con el número máximo de iteraciones.

Respecto a los dos primeros criterios de parada se puede llegar a las siguientes conclusiones:

- De acuerdo a los reportes generados, la diversidad de población nunca fue inferior al 20% durante 5 iteraciones, por lo cual esta condición de parada, nunca se llego a cumplir. Durante las ejecuciones que se realizaron, la diversidad de población siempre estuvo por encima del 75%. Por lo tanto es posible afirmar que la estrategia diseñada exploraba de manera intensiva el espacio de búsqueda pues no se quedaba estancada en máximos locales.

- Tomando como referente los reportes que se generaban tras cada ejecución, era posible observar, que el desempeño de los individuos, una vez afinados los parámetros y alcanzado cierto número de iteraciones, era superior al 85% y se mantenía estable, por lo que la probabilidad de satisfacer el criterio de parada en el que se estipula que la media acumulada sea menor a la media móvil (calculada cada 40 iteraciones), era muy alta; llegando a la conclusión que el algoritmo convergía rápidamente.

5.2.2 Validación de Modelos

Las siguientes apreciaciones están basadas en los resultados obtenidos durante la etapa de validación de modelos consolidados en la Tabla 11.

Tabla 11. Resultados Etapa de Validación de Modelos

No. Prueba	1	2	3	Promedio
MCC	0.768	0.952	0.952	0.890
Distancia en ROC	0.092	0.038	0.038	0.056
Sensitividad	0.914	0.961	0.961	0.945
Especificidad	0.965	0.995	0.995	0.983
Precisión	0.962	0.993	0.993	0.982
Cantidad de Celdas Incluidas en la Representación	24	28	26	26

- En la validación de los modelos encontrados en las ejecuciones, se pudo constatar que el MCC fue siempre superior al 0.8, lo que indica que los modelos que se obtenían con el algoritmo realizaban buenas clasificaciones. Adicionalmente, los valores obtenidos por los modelos en cada una de las medidas de calidad planteadas en el diseño de la prueba, fueron notablemente superiores a los umbrales definidos en los criterios de éxito del proyecto

- Aunque los modelos obtenidos alcanzan reglas para cierta sección de la trayectoria utilizada en la etapa de entrenamiento, muestran un muy buen desempeño en la sección de la trayectoria destinada para las validaciones de los modelos, lo que indica un sobreajuste a los datos de entrenamiento casi nulo

- En general, los modelos obtenidos arrojan buenas medidas de calidad, las cuales deben ser analizadas en conjunto con la representación del individuo con el fin de establecer las ventajas y desventajas de cada modelo. En estos términos, la principal desventaja de los modelos obtenidos con la estrategia co - evolutiva es la cantidad de celdas incluidas en las vecindades, las cual determina la complejidad de las reglas de AC encontradas.

- Al comparar los modelos obtenidos a la luz del desempeño en cuanto a tiempo de computo, los reportes estadísticos, las medidas de calidad, la cantidad de celdas incluidas en la vecindad, y la claridad de las reglas del modelo de AC, se llega a la conclusión de que, aunque en todas las ejecuciones de pruebas se obtuvieron buenos modelos que satisfacen los criterios de éxito del proyecto, el mejor modelo es el encontrado en la ejecución de ajuste de parámetros ya que satisface los objetivo del proyecto utilizando una representación que incluye sólo 21 celdas en la vecindad, lo cual permite obtener un árbol de clasificación derivado de las reglas legible, que explica características propias del fenómeno de plegamiento de proteínas.

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

El framework CAIF –PFT, es una herramienta que puede ser utilizada en trabajos futuros que busquen soluciones al problema de plegamiento de proteínas, utilizando un enfoque similar al planteado en este proyecto de investigación, puesto que proporciona una arquitectura orientada a objetos que ajustable, y una serie de funcionalidades que facilitan y agilizan el desarrollo y evaluación de soluciones. Adicionalmente, el framework CAIF – PFT es tan flexible en el manejo de las evidencias, que posibilita implementar un enfoque como el definido en el esquema ILAS, permitiendo diseñar una estrategia co - evolutiva que encuentra modelos de AC que no se sobre ajustan a los datos.

Debido al volumen y a la complejidad de los datos de los que se disponía, fue necesario planear un método que además de facilitar el manejo eficiente de los recursos computacionales, aportara en el diseño de una buena estrategia Co – Evolutiva. Por esta razón los datos se dividieron en diferentes conjuntos, que se procesaron de manera distribuida utilizando las funcionalidades de la librería Pyro. El manejo de los datos también se ve influenciado por una serie de factores relacionados con la disponibilidad de hardware y con las implicaciones que tiene trabajar en un entorno distribuido como el tiempo requerido para distribuir las tareas, retornar los resultados y sincronizar el entorno.

El volumen de los datos es un aspecto importante que debe considerarse en la definición de la técnica de modelado ya que la afecta de manera directa, como por ejemplo a la hora de elegir el número de poblaciones a utilizar. Con respecto a esto, el enfoque utilizado que consistió básicamente en tener múltiples poblaciones, cada una trabajando con un conjunto de datos diferente, además de incrementar la probabilidad de encontrar un modelo que no se sobre ajuste a los datos, contribuyó a que los individuos tuvieran mayor posibilidad de explorar el espacio de búsqueda de manera intensiva [56], de tal manera que les fue posible escapar de óptimos locales [57].

Uno de los principales inconvenientes de los Algoritmos Genéticos es la llamada convergencia prematura, que se refiere a que el algoritmo arroja como resultado óptimo un óptimo local, razón por la que no intenta buscar el óptimo global. Este problema, que es causado la mayoría de veces por el uso de una baja probabilidad de cruce, una probabilidad de mutación pequeña o un tamaño de población muy reducida no fue exhibido por la estrategia implementada, ya que con el ajuste manual de parámetros se garantizó que se seleccionaran unos adecuados, esto se pudo constatar con los reportes estadísticos que se obtuvieron en las ejecuciones de prueba en las cuales se pudo observar que el fitness externo, que medía la distancia con el predictor perfecto, fue en promedio inferior a 0.04, lo que indica que la tasa de falsos positivos era pequeña, comparada con la tasa de verdaderos positivos, que era cercana a 1. Adicionalmente, La diversidad de población en todas las ejecuciones se mantuvo sobre el 80%, lo que significa que el algoritmo no llegó a encapsularse en máximos locales, confirmando una vez más que las estrategias co – evolutivas tienden a explorar de manera intensiva el

espacio de búsqueda. De acuerdo con dichos reportes, la media acumulada del fitness interno tendió a estabilizarse, en todas las ejecuciones, entre las segunda y novena iteración, lo que demuestra que el algoritmo converge rápidamente en la búsqueda de buenas soluciones.

La estrategia Co – Evolutiva desarrollada, después del ajuste de parámetros, demostró que es capaz de encontrar modelos de AC en trayectorias de plegamiento de proteína, que satisfacen las medidas de calidad establecidas en los objetivos del proyecto. Para encontrar el mejor modelo requirió explorar una cantidad de individuos, equivalente al 5%, aproximadamente, de la totalidad de los individuos explorados en la estrategia evolutiva que se tomó como referente. Además, la Estrategia co – evolutiva definida e implementada, es capaz de identificar modelos de AC en trayectorias de plegamiento de proteínas, con una precisión superior al 97%, los cuales tienen una tabla de reglas asociada que permite extraer aspectos importantes de la dinámica subyacente en el fenómeno biológico [63].

Respecto a la representación seleccionada, se presentarían inconvenientes con proteínas que contengan menos de 7 AA ya que a cualquier celda diferente a la celda central tendría que aplicársele condiciones de frontera, pero se podría afirmar que tal inconveniente no tendría lugar ya que la proteína más corta que se conoce hasta la fecha consta de 20 AA, lo que favorece la utilización del modelo de AC encontrado en trayectorias de plegamiento de proteínas más largas o más cortas que la utilizada en las etapas de entrenamiento y validación, en las que fue posible notar que aunque los modelos obtenidos alcanzaron reglas para cierta sección de la trayectoria utilizada en la etapa de entrenamiento, muestran un muy buen desempeño en la sección de la trayectoria destinada para las validaciones de los modelos, lo que indica un sobreajuste a los datos de entrenamiento casi nulo.

6.2 TRABAJOS FUTUROS

Para trabajos futuros podría considerarse la opción de utilizar otras trayectorias, y ejecutar la estrategia en un entorno distribuido más amplio con el fin de explorar de manera más intensiva el espacio de búsqueda; con esto se podría contribuir en el desarrollo de un predictor de mapas de contacto.

A raíz de los inconvenientes presentados en la etapa de afinamiento de parámetros, surge la idea de la construcción de un método que afine los parámetros del algoritmo, de manera automática, y que controle las ejecuciones en un entorno distribuido, esto con el objetivo de contribuir a la generalización de la técnica.

REFERENCIAS Y BIBLIOGRAFIA

[1] T. Schlick. Molecular Modeling and Simulation: An Interdisciplinary Guide. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2002.

[2] N. Díaz. ". Framework Computacional para el Diseño Inverso de Modelos de Autómata Celular de Secuencias Cortas de Aminoácidos Soportado en un Proceso de Minería de Datos". *Encuentro Nacional de Investigación en Postgrados – ENIP* (2008) 2: pp. 53-70.

[3] M. Marques-Pita & L.M. Rocha. "Conceptual Structure in Cellular Automata: The Density Classification Task". *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI). MIT Press, Cambridge* (2008) : pp. 1-8.

[4] L. Pagie & M. Mitchell. "A Comparison of Evolutionary and Coevolutionary Search". *International Journal of Computational Intelligence and Applications* (2002) 20: pp. 53-70.

[5] T. de Camino Beck & C.R. Cartago. Un Lenguaje para la Especificación de Autómatas Celulares con Aplicaciones en Biología. Instituto Tecnológico de Costa Rica Departamento de Computación Programa de Maestría, 2000.

[6] M. Sipper. "Co-evolving Non-uniform Cellular Automata to Perform Computations". *Physica D: Nonlinear Phenomena* (1996) 92: pp. 193-208.

[7] L.B. Kier, C. Cheng et al. "A Cellular Automata Model of Ligand Passage Over a Protein Hydrodynamic Landscape". *Journal of Theoretical Biology* (2002) 215: pp. 415-426.

[8] P. Chopra & A. Bender. "Evolved Cellular Automata for Protein Secondary Structure Prediction Imitate the Determinants for Folding Observed in Nature". *In Silico Biol* (2007) 7: pp. 87-93.

[9] Y. Bar-Yam. "Polymer Simulation Using Cellular Automata: 2-D Melts, Gel-Electrophoresis And Polymer Collapse". *Some New Directions in Science on Computers* (1997) 1: pp. 73-73.

[10] R. Zwanzig, A. Szabo et al. "Levinthal's paradox". *Proc Natl Acad Sci U S A* (1992) 89: pp. 20-22.

[11] C.B. Anfinsen. "Principles that Govern the Folding of Protein Chains". *Science* (1973) 181: pp. 223-230.

[12] I. Instituto de Ciencia de Materiales de Aragon. "Hurgando en la Estructura de las Moléculas". *http://www.unizar.es/icma/divulgacion/pdf/pdfdifraccionrayos.pdf* () : p. Accesado: 2010.

[13] J.F. del Río Portilla. "Determinación de la Estructura de Proteínas por Resonancia Magnética Nuclear". *Mensaje Bioquímico* (2003) 27: pp. 65-83.

[14] Y.M. Rhee & V.S. Pande. "Multiplexed-replica exchange molecular dynamics method for protein folding simulation". *Biophysical Journal* (2003) 84: pp. 775-786.

[15] Z. Li & H.A. Scheraga. "Monte Carlo-minimization approach to the multiple-minima problem in protein folding". *Proceedings of the National Academy of Sciences of the United States of America* (1987) 84: pp. 6611-6611.

[16] K. Beck. Extreme programming explained: Embrace change. Reading, Massachusetts: Addison-Wesley, 2000.

[17] P. Baldi, S. Brunak et al. "Assessing the accuracy of prediction algorithms for classification: an overview". *Bioinformatics* (2000) 16: pp. 412-412.

[18] K.S. Tang & S. Kwong. Genetic algorithms: concepts and designs. Springer Verlag, 1999.

[19] G. Ceci, A. Mucherino et al. Computational Methods for Protein Fold Prediction: an Ab-Initio Topological Approach, Data Mining in Biomedicine. Springer Optimization and Its Applications, Panos Pardalos et al (Eds.), 2007.

[20] H.A. Scheraga, M. Khalili et al. "Protein-Folding Dynamics: Overview of Molecular Simulation Techniques". *ANNUAL REVIEW OF PHYSICAL CHEMISTRY* (2007) 58: pp. 57-57.

[21] B. Al-Lazikani, E.E. Hill et al. "Protein structure prediction". *Methods Mol Biol* (2008) 453: pp. 33-85.

[22] H. Berman, K. Henrick et al. "The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data". *Nucleic acids research* (2006) : .

[23] S. Luke & Z. Edition. "Essentials of Metaheuristics". *Disponible en http://www.cs.put.poznan.pl/mkomosinski/materialy/optymalizacja/_Essentials.pdf* (2009) Online Version 0.5: pp. 1-229.

[24] T. Bossomaier, T. Cranny et al. "A New Paradigm for Evolving Cellular Automata Rules". *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (1999) 1: pp. 169-176.

[25] F. Seredynski & P. Bouvry. "Multiprocessor Scheduling Algorithms Based on Cellular Automata Training". *University of Luxembourgo* (2008) : pp. 1-6.

[26] T. Back, R. Breukelaar et al. "Inverse Design of Cellular Automata by Genetic Algorithms: An Unconventional Programming Paradigm". *Unconventional Programming Paradigms* (2005) : pp. 161-172.

[27] Ken-Ichi Maeda & Chiaki Sakama. "Identifying Cellular Automata Rules". *Journal of Cellular Automata* (2006) 2: p. 1:20.

[28] N. Krasnogor, D.H. Marcos et al. "Protein Structure Prediction as a Complex Adaptive System". *Frontiers in Evolutionary Algorithms (FEA98)* (1998) : p. 441–447.

[29] G. Terrazas, P. Siepmann et al. "An Evolutionary Methodology for the Automated Design of Cellular Automaton-based Complex Systems". *Journal of Cellular Automata* (2006) 2: pp. 77-102.

[30] E. Sapin, O. Bailleux et al. "A New Universal Cellular Automaton Discovered by Evolutionary Algorithms". *Lecture Notes In Computer Science* (2004) : pp. 175-187.

[31] N. Krasnogor, G. Terrazas et al. "A Critical View of the Evolutionary Design of Self-assembling Systems". *Artificial Evolution* (2006) : pp. 179-188.

[32] J. Bacardit, M. Stout et al. "Automated Alphabet Reduction Method with Evolutionary Algorithms for Protein Structure Prediction". *Proceedings of the 9th annual conference on Genetic and evolutionary computation* (2007) : pp. 346-353.

[33] R.L. Wainwright, T. Outline et al. "Introduction to Genetic Algorithms Theory and Applications". *The Seventh Oklahoma Symposium on Artificial Intelligence* (1993) : pp. 1-49.

[34] H. Juille & J.B. Pollack. "Coevolving the ideal trainer: Application to the discovery of cellular automata rules". *Genetic Programming* (1998) : pp. 519-527.

[35] S.G. Ficici. "Multiobjective Optimization and Coevolution". *Multiobjective Problem Solving from Nature* (2008) : pp. 31-52.

[36] T. Bach. "The SIMP/STEP Manual A Python Environment for Cellular and Lattice-Gas Automata". (2005) : .

[37] R. Wirth & J. Hipp. "CRISP-DM: Towards a Standard Process Model for Data Mining". *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (2000) : pp. 29-39.

[38] P. Chapman, J. Clinton et al. "CRISP-DM 1.0: Step-by-step data mining guide". *CRISP-DM consortium* (2000) : pp. 1-78.

[39] T.M. Mitchell. Machine Learning. McGraw-Hill Science/Engineering/Math, 1997.

[40] M.A. Sherman. "SimTK and its users". *SimTK Engr. J* (2005) 5: pp. 1-10.

[41] J. Demšar, B. Zupan et al. "Orange: From experimental machine learning to interactive data mining". *Knowledge discovery in databases: PKDD 2004* (2004) : pp. 537-539.

[42] Python-based distributed programming with trickle. . 2007.

[43] D.L. Ensign, P.M. Kasson et al. "Heterogeneity Even at the Speed Limit of Folding: Large-scale Molecular Dynamics Study of a Fast-folding Variant of the Villin Headpiece". *Journal of Molecular Biology* (2007) 374: pp. 806-816.

[44] J. Werfel, M. Mitchell et al. "Resource sharing and coevolution in evolving cellular automata". *IEEE Transactions on Evolutionary Computation* (2000) 4: pp. 388-393.

[45] P. James, M. Mitchell et al. "The evolutionary design of collective computation in cellular automata". *Machine Learning* (1998) : .

[46] W. Humphrey, A. Dalke et al. "VMD: visual molecular dynamics". *Journal of Molecular Graphics* (1996) 14: pp. 33-38.

[47] M. Vendruscolo & E. Domany. "Protein folding using contact maps". *Vitamins and Hormones* (2000) 58: pp. 172-213.

[48] M. Vendruscolo, E. Kussell et al. "Recovery of protein structure from contact maps". *Folding and Design* (1997) 2: pp. 295-306.

[49] J.J. Moré & Z. Wu. "Distance geometry optimization for protein structures". *Journal of Global Optimization* (1999) 15: pp. 219-234.

[50] N. Chr & J. Smith. "Coevolution in ecosystems: Red Queen evolution or stasis?". *Evolution* (1984) : pp. 870-880.

[51] M. Mitchell, M.D. Thomure et al. "The role of space in the success of coevolutionary learning". *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems* (2006) : .

[52] R. Naiouf, A. De Giusti et al. "Procesamiento paralelo y distribuido. Fundamentos y aplicaciones". () : .

[53] P. Jogalekar & M. Woodside. "Evaluating the scalability of distributed systems". *Parallel and Distributed Systems, IEEE Transactions on* (2000) 11: pp. 589-603.

[54] F. Seredynski. "Evolutionary Paradigms". *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies. New York: Springer* (2006) : pp. 111-111.

[55] R. Breukelaar & T. Back. "Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior". *Proceedings of the 2005 conference on Genetic and evolutionary computation* (2005) : pp. 107-114.

[56] Z. Michalewicz. Genetic algorithms+ data structures. Springer, 1996.

[57] C. Ergun & K. Hacioglu. "Multiuser detection using a genetic algorithm in CDMA communications systems". *Communications, IEEE Transactions on* (2000) 48: pp. 1374-1383.

[58] A. Eiben & S. Smit. "Evolutionary Algorithm Parameters and Methods to Tune them". *Autonomous Search* (2012) : pp. 15-15.

[59] S. Smit & A. Eiben. "Parameter tuning of evolutionary algorithms: Generalist vs. specialist". *Applications of Evolutionary Computation* (2010) : pp. 542-551.

[60] J.A. Capote, O.V. Abreu et al. "Los Modelos de Simulación Computacional de Incendios: Ciclo de Vida". *Revista Montajes e Instalaciones* (2004) : pp. 121-124.

[61] D. Eisenberg. "The discovery of the α-helix and β-sheet, the principal structural features of proteins". *Proceedings of the National Academy of Sciences* (2003) 100: pp. 11207-11207.

[62] J.C. McKnight, D.S. Doering et al. "A thermostable 35-residue subdomain within villin headpiece". *Journal of molecular biology* (1996) 260: pp. 126-134.

[63] Mining Cellular Automata models on protein folding trajectories. . 2011.

[64] R. Davies. "The power of stories". *WWW Retrieved* (2001) : pp. 1-06.

[65] Are Use Cases Beneficial for Developers Using Agile Requirements?. . 2007.