Implementación de un algoritmo evolutivo para apoyar la detección de riesgos cardiovasculares a partir del análisis de datos clínicos

Miguel Ángel Askar Rodríguez

Universidad del Valle
Escuela de Ingeniería en Sistemas y Computación
Maestría en Ingeniería Énfasis en Sistemas y Computación
Santiago de Cali
Enero – Mayo de 2020

Implementación de un algoritmo evolutivo para apoyar la detección de riesgos cardiovasculares a partir del análisis de datos clínicos

Miguel Ángel Askar Rodríguez
Código 201803263
miguel.askar@correounivalle.edu.co

Directores
Victor Manuel Vargas Forero
Jaime Velasco Medina

Universidad del Valle
Escuela de Ingeniería en Sistemas y Computación
Maestría en Ingeniería Énfasis en Sistemas y Computación
Santiago de Cali
Enero – Mayo de 2020

Tesis presentada por
Miguel Ángel Askar Rodríguez
Como requisito parcial para la obtención del título de Magíster en Ingeniería

---

Victor Manuel Vargas Forero
Director

Jaime Velasco Medina
Director

Jurado

Jurado

Agradezco a mis padres y a mi hermano por el apoyo incondicional durante la ejecución de este proyecto. A los profesores Víctor y Jaime, mis directores, de quienes aprendí responsabilidad y capacidad de análisis, también porque sus conocimientos nos llevaron al éxito en este proyecto. A mis amigos, en especial Jefferson y Juan Camilo por hacerse los que entendían lo que hacía, también a Diana por su apoyo y ser un pilar emocional gigantesco. También agradezco al tiempo y las diferentes vivencias que hicieron de esta maestría un tiempo de enseñanza tanto académica como socialmente. Doy un agradecimiento especial al equipo del laboratorio de Bionanoelectrónica, quienes se convirtieron en mis amigos y se sorprendían cuando les contaba lo que hacía. Quiero agradecer inmensamente a Luisa por contactarme con 2 de sus amigos médicos, aprovechando también para agradecer a los médicos Juan Manuel Aricapa, Daniela Andrea Bedoya y Daurin Gabriel Guzmán. Por último, agradezco a la Universidad del Valle por permitirme recibir este conocimiento tan útil durante estos 2 años.

Tabla de Contenido

1. Introducción **2**
1.1. Formulación del problema . . . 2
1.2. Descripción del problema . . . 2
1.3. Objetivos . . . 4
1.3.1. Objetivo general . . . 4
1.3.2. Objetivos específicos . . . 4
1.4. Resultados esperados . . . 5
1.5. Metodología . . . 5
1.6. Información sobre los capítulos . . . 7

2. Justificación y alcance del proyecto **8**
2.1. Justificación . . . 8
2.2. Alcance . . . 9

3. Marco referencial **11**
3.1. Marco Teórico . . . 11
3.1.1. Criterios para clasificación de artículos . . . 11
3.1.2. Trabajos previos basados en Redes Neuronales Artificiales y Computación Evolutiva 13
3.1.3. Trabajos previos en Redes Neuronales Artificiales . . . 13
3.1.4. Trabajos previos en Computación Evolutiva . . . 14
3.1.5. Trabajos previos en PSO . . . 14
3.1.6. Trabajos con algoritmos genéticos . . . 14
3.2. Marco Conceptual . . . 16
3.2.1. Proyecto PIPEC-UV . . . 16
3.2.2. Terminología para el área de la salud . . . 16
3.2.3. Terminología para el área de ciencias de la computación . . . 17

4. Diseño de los algoritmos y el prototipo **18**
4.1. Selección de base de datos y atributos de entrenamiento . . . 18
4.1.1. Opinión de expertos respecto a la selección de los atributos . . . 20
4.2. Criterios de comparación entre las estrategias de inteligencia artificial . . . 20
4.3. Red neuronal artificial . . . 22
4.3.1. Consolidación de la red neuronal artificial . . . 23
4.4. Algoritmo evolutivo . . . 24
4.4.1. Consolidación del algoritmo evolutivo . . . 28
4.5. Prototipo . . . 29

5. Implementación de los algoritmos y el prototipo **31**
5.1. Red neuronal . . . 31
5.1.1. Primera iteración . . . 31
5.1.2. Segunda iteración . . . 35
5.1.3. Tercera iteración . . . 37
5.1.4. Iteración adicional . . . 38
5.1.5. Red Neuronal seleccionada . . . 40
5.2. Algoritmo evolutivo . . . 41
5.3. Prototipo . . . 43
5.4. Características de los datos y la predicción de RCV . . . 45

6. Pruebas y comparaciones **47**
6.1. Precisión . . . 47
6.2. Tiempo de entrenamiento . . . 47
6.3. Eficiencia . . . 48
6.4. Capacidad de reentrenamiento y cambios en la precisión: . . . 49
6.5. Capacidad de reentrenamiento con nuevos datos y cambios en la precisión . . . 49
6.6. Comparación de resultados con otro trabajo en el área . . . 50
6.6.1. Red neuronal . . . 50
6.6.2. Red neuronal y algoritmo evolutivo . . . 51
6.6.3. Comparación general: . . . 51
6.7. Validación cruzada . . . 52
6.7.1. Redes neuronales . . . 52
6.7.2. Algoritmo genético . . . 54
6.7.3. Análisis de resultados . . . 54

7. Conclusiones **57**
7.1. Conclusiones . . . 57

8. Trabajos futuros **60**
8.1. Trabajos futuros . . . 60

9. Bibliografía **62**

Anexos **66**

Lista de Figuras

4.1. Diseño de la interfaz gráfica del prototipo 30

5.1. Error durante entrenamiento de la red 7-14-1 33
5.2. Error durante entrenamiento de la red 7-4-1 34
5.3. Error durante entrenamiento de la red 7-14-11-8-1 34
5.4. Error durante entrenamiento de la red 7-14-1 36
5.5. Error durante entrenamiento de la red 7-14-11-8-1 36
5.6. Error durante entrenamiento de la red 7-14-1 37
5.7. Error durante entrenamiento de la red 7-14-11-8-1 38
5.8. Error durante entrenamiento de la red 7-14-10-8-1 (caso exitoso) 39
5.9. Error durante entrenamiento de la red 7-14-10-8-1 40
5.10. Red neuronal seleccionada (7-14-10-8-1) 40
5.11. Precisión, cantidad de cromosomas y generaciones, entrenamiento A.E. para casos sanos . 41
5.12. Precisión, cantidad de cromosomas y generaciones, entrenamiento A.E. para casos riesgos 42
5.13. Ejemplo del sistema de predicción del AG 43
5.14. Interfaz gráfica del prototipo implementado 44

6.1. Matrices de confusión del AE y la RNA 47
6.2. Tiempo de entrenamiento de la R.N.A. y el A.E. 48
6.3. Resultado de la prueba de eficiencia de la R.N.A. y el A.E. 48
6.4. Matriz de confusión, reentrenamiento del A.E. 50
6.5. Matrices de confusión, Validación cruzada del AG y la RNA 55

Lista de tablas

1.1. Relación entre los objetivos específicos y los resultados esperados 5
1.2. Estructura de capítulos 7

3.1. Clasificación de artículos de acuerdo a la técnica utilizada. 12

4.1. Atributos en las bases de datos y el proyecto PIPEC-UV. 19
4.2. Artículos y criterios de comparación. 21
4.3. Cromosoma del algoritmo evolutivo. 24

5.1. Primera iteración de implementación de la red neuronal. 32
5.2. Segunda iteración de implementación de la red neuronal. 35
5.3. Tercera iteración de implementación de la red neuronal. 37
5.4. Iteración de implementación de la red neuronal. 39
5.5. Tabla de distribución de frecuencias de la edad en la base de datos de entrenamiento . . . 45
5.6. Tabla de distribución de frecuencias de la presión arterial sistólica en la base de datos de entrenamiento 45
5.7. Tabla de distribución de frecuencias de colesterol sérico en la base de datos de entrenamiento 46
5.8. Fuentes de obtención de los rangos de las tablas de distribución de frecuencias para edad, presión arterial sistólica y colesterol serum. 46

6.1. Tabla de registro de la prueba de eficiencia de la R.N.A. y el A.E. 49
6.2. Resultados del reentrenamiento de la red neuronal. 50
6.3. Comparación de precisiones con trabajos anteriores 51
6.4. Configuración de los 4 pliegues de la validación cruzada de la RNA 53
6.5. Registro del entrenamiento de la validación cruzada del AG 54
6.6. Precisión y error del AG y la RNA en la validación cruzada 55

Resumen

En esta tesis se inicia con una investigación sobre antecedentes en la predicción de Riesgos Cardiovasculares (RCV) a través de Redes Neuronales Artificiales (RNA) y Algoritmos Evolutivos (AE), con el objetivo de implementar y verificar un algoritmo evolutivo para apoyar la detección de riesgos cardiovasculares a partir del análisis de datos clínicos. Esta verificación se hace por medio de criterios comparativos definidos a través de la investigación de los antecedentes, definiendo que ambas técnicas de IA serían comparadas con otros trabajos en el área de predicción de RCV y por su precisión, tiempo de entrenamiento, eficiencia y capacidad de reentrenamiento con los mismos y nuevos datos.

Para cumplir los objetivos propuestos en esta tesis, la investigación inicial permite conocer los avances de AE para predecir RCV y las conclusiones dadas por distintos autores sirven para iniciar con un panorama general de investigación donde se aborda el problema a través de una fase de planeación que permite definir los criterios de comparación anteriormente descritos. Posteriormente, una fase de diseño que permite anticipar características técnicas como la función de fitness, reproducción y población para el AE y la posible cantidad de capas y neuronas para la RNA. Luego se pasa a la etapa de implementación, donde se deja por escrito el proceso llevado a cabo para codificar el AE y la RNA a través del lenguaje de programación Java y la asistencia del framework Neuroph. Dejando por último una fase de pruebas, donde se utilizan los criterios de comparación definidos y se establecen conclusiones y trabajos futuros.

La implementación y ejecución de las pruebas permitió establecer que el AE alcanzó una precisión de 91.7 % y la RNA alcanzó 76.7 %. Respecto al tiempo de entrenamiento, la RNA tardó 49 minutos y el AE 4 horas y 59 minutos. En el apartado de eficiencia, la RNA no presenta cambios en su tiempo de ejecución al pasar del análisis de 300 pacientes a 1500, contrastando con el AE que presenta un tiempo de ejecución que crece exponencialmente. También se estableció que la capacidad de reentrenamiento con los mismos datos es difícil de conseguir y requiere un rediseño de los algoritmos para que soporten reentrenamiento, sin embargo, al utilizar nuevos datos fue posible hacer un reentrenamiento de ambos algoritmos, disminuyendo la precisión de la RNA a 68 % y manteniéndose en 91.7 % la del AE.

Por último, la comparación con otros trabajos permite concluir que las investigaciones que usan IA para la predicción de RCV no suelen definir una gran variedad de métricas para su comparación, pues regularmente sólo se presenta la precisión del algoritmo resultante. Por tanto, se sugiere utilizar los criterios de comparación presentados en este trabajo para evaluar futuros trabajos en el área de predicción de RCV. Como prueba final, se ejecutó una validación cruzada de 4 pliegues (4-fold) que establece que el AG es confiable al ser verificado a través de la comparación de su error de predicción con el de la RNA, donde el AG supera a la RNA por 2.66 %.

Palabras clave: Inteligencia artificial, Redes Neuronales, Computación Evolutiva, Algoritmos genéticos, Riesgo Cardiovascular, Análisis de datos clínicos.

Capítulo 1

Introducción

1.1. Formulación del problema

¿Cómo implementar y verificar un algoritmo evolutivo para apoyar la detección de riesgos cardiovasculares a partir del análisis de datos clínicos?

1.2. Descripción del problema

La inteligencia artificial (IA) es una herramienta computacional útil cuando se desea analizar grandes cantidades de datos y deducir conclusiones sobre la información procesada. Las diferentes técnicas de IA permiten estudiar y analizar situaciones que con otros medios no es posible evaluar. En el mundo actual, el análisis de grandes cantidades de datos clínicos se ha convertido en una alternativa para determinar factores de riesgo de enfermedades en los pacientes de manera individual y colectiva.

Hoy en día las bases de datos clínicas almacenan datos que van desde las características morfológicas de los pacientes hasta el registro de exámenes como electrocardiografía y pulsioximetría, por esta razón, estos datos son estudiados para encontrar indicadores que permitan concluir que un paciente se encuentra en riesgo de presentar alguna enfermedad cardiovascular (ECV). Las diversas investigaciones plantean el uso de distintas técnicas de inteligencia artificial para el análisis de los datos clínicos y la búsqueda de factores de riesgo en los mismos, algunas de estas técnicas son: redes neuronales artificiales, algoritmos genéticos y PSO (Particle Swarm Optimization).

Si bien las distintas técnicas usadas son útiles para la predicción de riesgos cardiovasculares (RCV), no se establece claramente un criterio para seleccionar una o otra, es decir, aunque unas permiten un alto grado de eficiencia al analizar los datos [1], otras proponen reglas que determinan cuál información tiene mayor relevancia en el momento de determinar los RCV de los pacientes [2].

Como se muestra más adelante en el marco teórico de este documento, entre las técnicas más utilizadas para la predicción de RCV están las redes neuronales artificiales y la computación evolutiva. Por tanto, el problema a abordar es cómo seleccionar o elegir entre un Algoritmo Evolutivo o una Red Neuronal Artificial para el análisis datos clínicos para la predicción de RCV. Es decir, cuál técnica tendría el mejor comportamiento al analizar los mismos datos con procedimientos similares, a través de unos criterios de evaluación definidos.

Dentro del contexto social, el problema de escoger entre un algoritmo evolutivo o uno de redes neuronales

artificiales para la predicción de RCV se exttiende globalmente, ya que estas enfermedades afectan a alrededor de 50 millones de personas alrededor del mundo [3].

En síntesis, la problemática es cómo implementar un algoritmo evolutivo y una red neuronal artificial, comparándolos a través de unos criterios de evaluación, que permitan definir las bases para elegir entre ambos algoritmos a la hora de aplicarlos al contexto de predicción de RCV. Buscando así que los resultados de esta investigación se puedan utilizar como punto de partida para que los investigadores decidan cuál de estas técnicas implementar para el análisis de sus datos clínicos, incluso de enfermedades diferentes a los RCV.

1.3. Objetivos

1.3.1. Objetivo general

Implementar y verificar un algoritmo evolutivo para apoyar la detección de riesgos cardiovasculares a partir del análisis de datos clínicos.

1.3.2. Objetivos específicos

1. Definir los criterios de evaluación y comparación de dos técnicas de procesamiento de datos clínicos para la detección de riesgos cardiovasculares.

2. Seleccionar una técnica de redes neuronales para apoyar la predicción de riesgos cardiovasculares a través del análisis de datos clínicos.

3. Seleccionar una técnica de computación evolutiva para apoyar la predicción de riesgos cardiovasculares a través del análisis de datos clínicos.

4. Implementar las dos técnicas seleccionadas para el procesamiento de datos clínicos para apoyar la predicción de riesgos cardiovasculares.

5. Evaluar y comparar las dos técnicas seleccionadas para el procesamiento de datos clínicos para apoyar la predicción de riesgos cardiovasculares.

1.4. Resultados esperados

Tabla 1.1: Relación entre los objetivos específicos y los resultados esperados

Objetivo específico	**Resultado esperado**
Definir los criterios de evaluación y comparación de dos técnicas de procesamiento de datos clínicos para la detección de riesgos cardiovasculares.	Documento con la justificación de la selección de los criterios de evaluación y comparación a través de la revisión bibliográfica.
Seleccionar una técnica de redes neuronales para apoyar la predicción de riesgos cardiovasculares a través del análisis de datos clínicos.	Revisión bibliográfica y justificación de la selección o diseño de la técnica de predicción con redes neuronales artificiales.
Seleccionar una técnica de computación evolutiva para apoyar la predicción de riesgos cardiovasculares a través del análisis de datos clínicos.	Revisión bibliográfica y justificación de la selección o diseño de la técnica de predicción con algoritmos evolutivos.
Implementar las dos técnicas seleccionadas para el procesamiento de datos clínicos para apoyar la predicción de riesgos cardiovasculares.	Prototipo capaz de procesar datasets clínicos, de acuerdo a las técnicas de inteligencia artificial implementadas y arrojando como salida el resultado de la predicción de RCVs para cada paciente.
Evaluar y comparar las dos técnicas seleccionadas para el procesamiento de datos clínicos para apoyar la predicción de riesgos cardiovasculares.	Documento donde figura la evaluación y comparación de las estrategias de análisis de datos clínicos para la predicción de riesgos cardiovasculares de acuerdo a los criterios definidos al inicio de la investigación.

1.5. Metodología

En este proyecto se utilizó una metodología inductiva dado que en el área de algoritmos evolutivos y redes neuronales (ejes centrales de este trabajo), se cuenta con investigaciones realizadas en diferentes países, las cuales son tomadas como referencia para obtener conclusiones individuales y hacer comparaciones que permiten, posteriormente, hacer conclusiones generales sobre las investigaciones con A.E. y R.N.A. para la predicción de RCV. Es necesario aclarar que la constante búsqueda de mejora de los algoritmos implementados en distintas investigaciones, resulta en trabajos que son continuación de otros, por tanto; las fuentes no son escasas pero sí pueden resultar similares en cuanto a los algoritmos diseñados para procesar los datos.

Las fuentes de información utilizadas durante el desarrollo del proyecto son primarias y secundarias, ya que se hizo uso de artículos científicos, libros, conferencias y congresos. A partir del análisis de la información recolectada de las fuentes mencionadas anteriormente, se implementaron los algoritmos para la predicción de riesgos cardiovasculares a través de datasets clínicos. Este proyecto no pretende ser una réplica de trabajos ya realizados, por el contrario, se lleva a cabo una investigación cuyos productos fueron

un algoritmo de redes neuronales artificiales y un algoritmo evolutivo, ambos para procesar datos clínicos y detectar presuntos riesgos cardiovasculares, así como la evaluación de dichos algoritmos de acuerdo a unos criterios definidos al inicio de la investigación.

Esta tesis tiene un carácter investigativo más que de desarrollo de software, por lo tanto, no se implementa una metodología de desarrollo específica, aunque el proyecto estuvo dividido en fases de acuerdo a los objetivos específicos del mismo:

Definir los criterios de evaluación y comparación de dos técnicas de procesamiento de datos clínicos para riesgos cardiovasculares:

Análisis y planeación: para esta etapa, con base en la información descubierta a través de la revisión bibliográfica, se definieron los criterios de evaluación para la estrategia evolutiva y la de redes neuronales a la hora de predecir riesgos cardiovasculares al analizar datasets clínicos. Esta etapa comprendió 8 semanas.

Seleccionar una técnica de redes neuronales para la predicción de riesgos cardiovasculares a través del análisis de datos clínicos y *Seleccionar o diseñar una técnica de computación evolutiva para la predicción de riesgos cardiovasculares a través del análisis de datos clínicos:*

Diseño: durante esta etapa se diseñaron las estrategias evolutiva y de redes neuronales a utilizar para la predicción de riesgos cardiovasculares, así como los detalles correspondientes a la técnica específica a utilizar. Esta etapa tuvo una duración de 8 semanas.

Codificar las dos estrategias de procesamiento de datos clínicos para riesgos cardiovasculares seleccionadas o diseñadas:

Codificación: para la codificación se tomaron los modelos establecidos en las etapas anteriores y se procedió a su implementación, esto corresponde al algoritmo genético y a la estrategia con redes neuronales artificiales. Tras la codificación de dichas instrucciones, se construyó un prototipo para ejecutar las pruebas correspondientes sobre la información. La etapa de codificación tuvo una duración de 8 semanas.

Evaluar y comparar la equivalencia de las dos estrategias de predicción riesgos cardiovasculares a través del procesamiento datos clínicos:

Pruebas: para la etapa de pruebas, se evaluaron ambas estrategias de acuerdo a los criterios definidos al inicio de la investigación, conllevando a un análisis comparativo entre las dos técnicas de predicción. Para este escenario se utilizó un dataset clínico con 303 registros (1500 al multiplicar las tuplas). Así mismo, durante esta etapa se creó un documento con las respectivas conclusiones sobre el análisis comparativo de los datos entregados por parte de las propuestas codificadas. Esta etapa tuvo una duración de 4 semanas.

1.6. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento y se explica qué objetivos específicos aborda.

Capítulo	**Descripción**	**Objetivo específico**
Capítulo 2: Alcance de la propuesta	En este capítulo se presenta de manera general lo que se pretende lograr a través de la ejecución este proyecto	Ninguno
Capítulo 3: Marco referencial	Se presentan los conceptos básicos para que el lector alcance una comprensión satisfactoria del proyecto	Ninguno
Capítulo 4: Diseño	Como primera instancia se presenta la justificación de la selección de la base de datos, los atributos y la opinión de expertos al respecto. Posteriormente se establecen los criterios de comparación para evaluar los algoritmos al final de la investigación. Por último se diseñan la red neuronal artificial, el algoritmo evolutivo y el prototipo software del proyecto	1, 2 y 3
Capítulo 5: Implementación	En este capítulo se documentan la fase de implementación y entrenamiento de la red neuronal, el algoritmo evolutivo y la construcción del prototipo software del proyecto	4
Capítulo 6: Pruebas y comparaciones	En este capítulo se documentan las pruebas de la red neuronal y el algoritmo evolutivo, así como la comparación entre estos a través de los criterios definidos en el capítulo 4	5
Capítulo 7: Conclusiones	Se presentan las conclusiones del proyecto	5
Capítulo 8: Trabajos Futuros	Se presentan las distintas posibilidades consideradas por el autor para la expansión de la investigación presentada en este trabajo.	5
Capítulo 9: Bibliografía	Se presentan las referencias y fuentes de información del proyecto	Ninguno

Tabla 1.2: Estructura de capítulos

Capítulo 2

Justificación y alcance del proyecto

2.1. Justificación

El protocolo con el cual son atendidos los pacientes en los centros de servicios médicos y clínicos ha evolucionado a través del tiempo. Actualmente existen dispositivos que registran electrónicamente la información de variables fisiológicas específicas, también existen otros dispositivos que pueden registrar información de diferentes variables fisiológicas conjuntamente, como es el caso de la plataforma PIPEC-UV, un proyecto de investigación y desarrollo de la Universidad del Valle. Debido a estos nuevos dispositivos y sistemas, existen bases de datos cada día más extensas donde se cuenta con grandes cantidades de datos clínicos que pueden ser analizados de tal manera que se puedan predecir riesgos cardiovasculares para los pacientes.

Por otro lado, el impacto social de un algoritmo de predicción de riesgos cardiovasculares es alto, esto debido a que las estadísticas indican que las enfermedades cardiovasculares (ECV) han sido la principal causa de muerte en todo el mundo durante los últimos 15 años. Cada año mueren más personas por ECV que por cualquier otra causa. Se calcula que en 2015 murieron 15 millones de personas a causa de estas enfermedades, lo cual representa un 27 % de todas las muertes registradas en el mundo, y más de tres cuartas partes de las defunciones por ECV se producen en los países de ingresos bajos y medios. Esta proporción es mayor que la suma de muertes a causa de enfermedades infecciosas, algunos tipos de cáncer, VIH y accidentes de tránsito [3] [4].

En el momento de utilizar la información que proporcionan las herramientas del primer párrafo para mitigar el problema presentado en el segundo, los investigadores suelen utilizar técnicas de inteligencia artificial. Pero estas son numerosas y los criterios para seleccionar una u otra suelen basarse en las tendencias actuales y en la experiencia o especialidad de los investigadores. Por esta razón se establece que, la implementación de una técnica de inteligencia artificial como los algoritmos evolutivos y su verificación a través de la comparación con una de redes neuronales artificiales, resulta importante para los estudios en donde se pretenda analizar información clínica que está en constante crecimiento y que dicho análisis puede estar enfocado a mejorar la calidad de vida y a detectar tempranamente la población con riesgos de adquirir un ECV. Es importante mencionar que la relevancia de esta comparación radica en sentar las bases para que los investigadores decidan cuál de las dos técnicas es más conveniente de acuerdo la información que tienen disponible para trabajar.

2.2. Alcance

El desarrollo de esta investigación requiere profundizar en el área de análisis de información en búsqueda de riesgos cardiovasculares por medio de Algoritmos Evolutivos (AE) y Redes Neuronales Artificiales (RNA). Por esto, la parte inicial del documento se centra en la búsqueda de trabajos anteriores ejecutados tanto en Algoritmos Evolutivos como en Redes Neuronales. El compendio de trabajos anteriores dio la capacidad de implementar ambas estrategias de análisis de datos clínicos (AE y RNA), consolidándolas en un prototipo que, a través de una interfaz sencilla, permite el ingreso de información de pacientes y arroja un resultado de acuerdo a los datos registrados en su historia clínica.

No era necesario que los datos clínicos a evaluar fuesen obtenidos de alguna base de datos clínica específica. Por otro lado, la estrategia evolutiva y la de redes neuronales, así como su evaluación de acuerdo a los criterios a definir, requería especial atención y fue uno de los aspectos críticos para el éxito de este proyecto. Por lo tanto, la definición de los criterios de evaluación y el diseño de las técnicas de procesamiento de datos clínicos fueron las tareas más extensas del proyecto. Además, se debe tener en cuenta que los atributos seleccionados son los encontrados en la base de datos clínica de Cleveland [5] y que son comunes con los registrados por el proyecto PIPEC-UV, estos atributos son:

- Edad.
- Género.
- Presión arterial sistólica.
- Nivel de colesterol.
- Azúcar en sangre.
- Evaluación del electrocardiograma.
- Máximo ritmo cardiaco.

El prototipo tiene una apariencia simple, sin mucho detalle en su interfaz, permitiendo el ingreso de información de pacientes para detectar su presunto riesgo cardiovascular. Esto se debe a que este trabajo busca comparar las técnicas de redes neuronales artificiales y algoritmos evolutivos, trabajo que tuvo que ser ejecutado con los datos arrojados por cada algoritmo y que no requerían ser presentados gráficamente para usuarios externos al proyecto. Permitiendo así centrar el desarrollo de esta tesis en la investigación, diseño y codificación de las estrategias de análisis de información con inteligencia artificial, así como también lograr minimizar el uso de recursos computacionales en el apartado gráfico, para así utilizarlos en mayor medida para el procesamiento de variables por parte de los algoritmos.

Respecto a la construcción del AE y la RNA, el marco teórico de esta tesis presenta trabajos de predicción de RCV que suelen usar técnicas en conjunto, es decir que no usan específicamente RNA o AE. Sin embargo, teniendo en cuenta trabajos de AE como [6] y de RNA como la revisión presentada en [7], es posible lograr precisiones superiores al 70 % utilizando estas técnicas de manera exclusiva. De esta manera y buscando comparar un AE y una RNA sin que sean asistidas por otras técnicas, se decide construir ambos en su forma básica y sin un agrupamiento previo de sus entradas, tal y como se describe posteriormente en la sección de diseño de este documento.

Profundizando en cómo se evaluaron el AE y la RNA, los trabajos recopilados en el marco teórico y

sobre todo los de Chitra [8], Kaur [9], Khan [10] y Feshki [11], muestran que el aspecto común a evaluar en este tipo de investigaciones es la precisión del algoritmo. Sin embargo, algunos de los autores mencionados anteriormente, incluyen en sus trabajos el tiempo de entrenamiento de su solución y, teniendo en cuenta que algunos mejoran sus trabajos previos, la capacidad de reentrenamiento del algoritmo que proponen. Debido a esto, se seleccionaron como principales características a comparar del AE y la RNA la precisión, tiempo de entrenamiento y capacidad de reentrenamiento. Sin embargo, se optó por agregar otros criterios que involucrasen profundización en el reentrenamiento y comportamiento ante grandes cantidades de datos, asi que se agregaron los criterios de: capacidad de reentrenamiento con datos nuevos y viejos, y eficiencia de los algoritmos al diagnosticar grandes cantidades de pacientes.

Capítulo 3

Marco referencial

En este capítulo se describen los conceptos y definiciones necesarias para el entendimiento de las estrategias a implementar en el desarrollo del proyecto.

3.1. Marco Teórico

3.1.1. Criterios para clasificación de artículos

Para la clasificación de los artículos que se puede observar en la tabla 3.1, se determinan las siguientes categorías basadas en las estrategias computacionales utilizadas en las distintas publicaciones revisadas para esta investigación, las cuáles se catalogan posteriormente bajo estos criterios.

- Particle Swarm Optimization (PSO): esta técnica es considerada un algoritmo evolutivo en tanto conserva sus características, pero su difusión, utilización y semejanza específica con el comportamiento de enjambres en la naturaleza permite que sea considerada como criterio de clasificación para las estrategias utilizadas en la bibliografía.
- Algoritmos genéticos (AG): a diferencia de la técnica de PSO, en esta se encuentran los artículos que usan cromosomas y reglas evolutivas generales sin asociarlas al comportamiento de partículas o enjambres. Representa un enfoque más general de los algoritmos evolutivos.
- Redes neuronales (RN): aunque no es una estrategia evolutiva, es ampliamente utilizada para la predicción de riesgos cardiovasculares a través de conjuntos de datos clínicos y de reglas obtenidas de los mismos, sin dejar a un lado el hecho de que en ocasiones esta técnica es utilizada en conjunto con estrategias evolutivas. Por tanto, una revisión del trabajo actual en Computación Evolutiva para este campo, requiere una clasificación considerando este criterio.

ARTÍCULO	CLASIFICACIÓN
Using PSO Algorithm for Producing Best Rules in Diagnosis of Heart Disease [2]	PSO
Improving the heart disease diagnosis by evolutionary algorithm of PSO and Feed Forward Neural Network [11]	PSO, RN
Remote heart risk monitoring system based on efficient Neural Network and Evolutionary Algorithm [12]	PSO, RN
An evolutionary algorithm for heart disease prediction [6]	AG
Evolutionary and Neural Computing Based Decision Support System for Disease Diagnosis from Clinical Data Sets in Medical Practice [13]	AG, RN
Comparison of two- criterion evolutionary filtering techniques in cardiovascular predictive modelling [14]	AG
Selecting critical clinical features for heart diseases diagnosis with a real-coded genetic algorithm [15]	AG
Heart-disease diagnosis decision support employing fuzzy systems with genetically optimized accuracy- interpretability trade-off [16]	AG
Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system [17]	AG, RN
Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease [18]	AG
An Evolutionary Computation Approach for Optimizing multilevel Data to Predict Patient Outcomes [19]	AG
A frame work for analysis and optimization of multiclass ECG classifier based on Rough set theory [20]	AG
Autonomous evolutionary algorithm in medical data analysis [21]	AG
Efficient heart disease prediction system using optimization technique [22]	PSO
A highly accurate firefly based algorithm for heart disease prediction [23]	PSO
Heart Disease Prediction System Using Supervised Learning Classifier [8]	RN
Analysis of neural networks based heart disease prediction system [7]	RN
An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier [10]	RN
Prediction of Heart Disease using CNN [24]	RN
Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks[25]	AG, RN

Tabla 3.1: Clasificación de artículos de acuerdo a la técnica utilizada.

En lo que a la prevención de riesgos cardiovasculares concierne, las investigaciones con este objetivo son cada vez más. Con el tiempo, algunas técnicas como redes neuronales se han convertido en tendencia para alcanzar la predicción de los RCV. Por tanto, se presentan a continuación los antecedentes para las investigaciones en algoritmos evolutivos de manera independiente y en conjunto con redes neuronales.

3.1.2. Trabajos previos basados en Redes Neuronales Artificiales y Computación Evolutiva

En los últimos años las redes neuronales se han vuelto tendencia sobre todo cuando se trata de tomar decisiones basándose en grandes cantidades datos. Debido a esto, ha surgido una corriente donde se combinan algoritmos evolutivos y redes neuronales en búsqueda de mejorar la precisión de las predicciones.

En trabajos como el de Feshki, M. [11] o el de Radhimeenakshi, S. [12], se presenta una combinación de algoritmos de PSO y de Redes neuronales, donde los primeros se encargan de seleccionar los mejores datos (se consideran mejores de acuerdo a los estudios de cada investigador) o los que más información les proporcionen con el fin de enviar registros filtrados a las redes neuronales, compensando así la carga de trabajo para las RNA con entradas más pequeñas y evitando la gran complejidad generada al hacer que un algoritmo de PSO genere una respuesta a manera de predicción.

Por otro lado, se tiene el trabajo de Pławiak, Paweł [17], donde se utilizan redes neuronales evolutivas. En este trabajo, se usa un algoritmo que principalmente resuelve el problema con Redes neuronales, pero estás van cambiando los pesos que asignan a cada atributo de manera iterativa. Aunque esto requiere una mayor cantidad de procesamiento, representa una mejora para el entrenamiento de la RNA, ya que la red se entrena con una mejor selección de atributos (de acuerdo a la función de fitness) para poder otorgar una predicción con precisión diferente a la esperada con métodos simples de RNA. Una situación similar se presenta en [25] donde una red neuronal es entrenada usando un algoritmo genético, donde la capa oculta es mejorada evaluando sus salidas y mejorándolas, lo que representa una alternativa al aprendizaje en retrospectiva clásico, puesto que además de la salida total de la red neuronal, ahora se tienen en cuenta las salidas de las capas internas, este trabajo logró una precisión del 97.78 % al evaluar el dataset de prueba.

3.1.3. Trabajos previos en Redes Neuronales Artificiales

Las RNA son utilizadas frecuentemente cuando se trata de analizar información y encontrar patrones dentro de esta para otorgar un análisis o resultado. En el ámbito de la predicción de riesgos cardiovasculares, se suele utilizar técnicas de RNA asistidas por análisis previos u organización de la información de entrada de la red a través de otras técnicas de inteligencia artificial. Sin embargo, las RNA siguen siendo el eje central de estos trabajos y en esta sección se describen algunos de los más representativos.

En el trabajo de Kaur, Amandeep [9] se encuentra una revisión del estado del arte de técnicas para la predicción de enfermedades cardiacas, cuyo análisis permite concluir que las máquinas de soporte vectorial son las menos indicadas para esta tarea y, por el contrario, las RNA y las técnicas de minería de datos son las que más precisión tienen, superando el 86 % de precisión en promedio. A manera de anotación, los algoritmos evolutivos no son tenidos en cuenta en este survey.

Por otro lado, está el trabajo de Rajamhoana, Devi, Umamaheswari, Kiruba, Karunya, Deepika [7] donde se concluye que las investigaciones con RNA sufren una tendencia a ser asistidas por otras técnicas de inteligencia artificial, permitiendo clasificar la información de entrada a las redes y elevar los índices de

precisión de las mismas.

Sin embargo, se pueden encontrar trabajos como el de Chitra [8] donde el trabajo de clasificación es hecho en su totalidad por una RNA y alcanza una precisión del 85 %. También es importante mencionar trabajos como el de Singhal, Kumar, Passricha [24] donde se logra una precisión del 95 % y el de Khan [10] donde se alcanza una precisión del 98.2 %, ambos trabajos utilizan la base de datos de Cleveland [5]. Sin embargo, el trabajo de Khan [10] utiliza una versión alterada del conjunto de datos donde se multiplican los registros, por lo cual no se recomienda como una guía absoluta y, por el contrario, se recomienda como objeto de estudio para comprobar la precisión de la estrategia presentada.

3.1.4. Trabajos previos en Computación Evolutiva

En contraste con la sección anterior, los trabajos que no involucran redes neuronales y que se encuentran en la tabla 2.1, entran en la clasificación de uso de técnicas de computación evolutiva. Dentro de estos se pueden observar los algoritmos PSO y los algoritmos genéticos.

3.1.5. Trabajos previos en PSO

De esta categoría se puede hablar del trabajo de Hassoon [2], donde el algoritmo no está enfocado en la predicción como resultado, sino en encontrar las reglas (combinación de valores de atributos) que determinarán con mayor seguridad si un paciente se encuentra en riesgo de sufrir deficiencias cardiovasculares de acuerdo a su cuadro clínico.

Por otro lado, está el trabajo de Suvarna [22], donde se usa PSO con el objetivo de encontrar la relación o clústeres de datos que al ser analizados por técnicas de minería de datos otorgarán resultados más dicientes de acuerdo a reglas y un sistema de medición de eficiencia. Resulta importante mencionar que este trabajo está enfocado en la predicción de los RCV, aunque el método final para analizar los datos no es el algoritmo PSO.

Por último, se tiene el trabajo de Long, [23], enfocado en el diagnóstico asistido de enfermedades cardiacas. Esto a través de una reducción del problema, encontrando clústeres de acuerdo al uso de PSO, procesando posteriormente los datos a través de técnicas de lógica difusa.

3.1.6. Trabajos con algoritmos genéticos

Siendo el foco de esta investigación, los algoritmos genéticos son los que más se pueden encontrar en la tabla 3.1, aunque de estos es importante destacar el trabajo de Jabbar, M. A. Deekshatulu, B. L. Chandra, Priti, 2012 [6]; debido a que en este se puede encontrar una descripción detallada del algoritmo, justifican el uso de algoritmos genéticos y obtienen buenos resultados. En síntesis, se usan los algoritmos genéticos para encontrar las relaciones entre atributos, con la meta de definir reglas que posteriormente serán aprovechadas en minería de datos para la predicción de RCV en los pacientes que se encuentran en la base de datos.

De manera alterna, otro trabajo que cobra gran relevancia es el de Brester [14], ya que representa la comparación de dos técnicas evolutivas para predecir RCV. Su importancia radica en que simplifican los datos a analizar, determinando la importancia de los mismos y cuáles son relevantes para un posterior análisis. Como aspecto a resaltar, el trabajo de Brester da cabida a profundizar en la predicción de diferentes riesgos cardiovasculares, separando las características importantes de cada uno.

Por último, es importante decir que los otros artículos mencionados en la tabla 3.1 no son analizados en detalle en esta sección, debido a la tendencia de todos a convertirse en un análisis de los mejores atributos o determinación de reglas para luego procesar la información a través de técnicas de minería de datos. Sin embargo, estos resultan importantes para la investigación ya que a nivel de detalle se convierten en aportes significativos para la construcción o mejora de algoritmos predictivos de riesgos cardiovasculares.

3.2. Marco Conceptual

3.2.1. Proyecto PIPEC-UV

PIPEC-UV es un proyecto en ejecución dentro de la Universidad del Valle donde se inició con el diseño de una unidad hardware de medición dirigida a los usuarios finales para realizar la medición y registro de sus signos vitales por medio de la integración de diferentes sensores y dispositivos externos inalámbricos que permiten hacer tomas de electrocardiografía, pulsioximetría, toma de presión arterial, IMC, grasa corporal y peso del paciente.

La unidad hardware de medición cuenta con conectividad Bluetooth, ANT+, WiFi y Ethernet, que otorgan capacidad de escalamiento a la unidad, lo cual da paso al objetivo principal del proyecto: la creación de una plataforma que combine dispositivos de atención, de obtención de datos y con almacenamiento centralizado de la información. Todo esto para prevenir el riesgo cardiovascular en la población objetivo.

3.2.2. Terminología para el área de la salud

- Riesgo cardiovascular [26]: El riesgo cardiovascular es la probabilidad de sufrir una enfermedad cardiovascular aterosclerótica (ECV) dada la presencia de determinados factores de riesgo en un período definido de tiempo [27]. Los factores de riesgo con mayor fuerza de asociación con la enfermedad cardiovascular han sido la presión arterial elevada, la diabetes mellitus, la obesidad, el tabaquismo, la inactividad física y la hipercolesterolemia[28]. En las personas sin enfermedad cardiovascular, se ha recomendado evaluar el riesgo de sufrir un evento cardiovascular con base en estimación derivada de puntajes de riesgo cardiovascular como los presentados en [28]. Los puntajes desarrollados hasta el momento no han sido validados en población latino-americana [27]. Aunque por otro lado, es conocido que los puntajes desarrollados en población caucásica, como por ejemplo el de Framingham, sobreestima el riesgo en otras poblaciones [28].

- Causas de enfermedades cardiovasculares: en Colombia, algunos de los distintos hábitos de la población son catalizadores de riesgo en lo que a enfermedades cardiovasculares concierne. Entre las costumbres que tiene la población colombiana y que les acercan a padecer deficiencias cardiovasculares, se tienen las siguientes [29]:

  Tabaquismo: En el informe SEA 2003 [29] se presenta que el consumo de tabaco incrementa el riesgo cardiovascular y, a su vez, el dejar de fumar implica reducciones del riesgo. Por tanto, considerar si un paciente es fumador o no, puede resultar determinante a la hora de establecer si se encuentra en riesgo de sufrir una enfermedad cardiovascular, y de ser así, su gravedad.

  Sobrepeso y obesidad: es un indicador alarmante del RCV, pues al analizar el Índice de Masa Corporal (IMC) de la población que presenta un aumento a través del tiempo, con predominancia en personas de 20 o más años de edad, implica un mayor índice de mortalidad por enfermedades cardiovasculares asociadas al sobrepeso.

  Otros factores que ponen en riesgo a la población: existen factores que son comunes a la hora de establecer riesgos cardiovasculares en la población, pero estos dependen de las condiciones geográficas y principalmente culturales de donde las personas residan. Teniendo esto en cuenta y que no se puede dar especificidad en tanto cada región del país puede abarcar distintos escenarios, los factores de riesgo son: Síndromes metabólicos, alimentación, índice de sedentarismo, estrategias para PP de los factores de riesgos cardiovascular.

- Segmento ST [30]: en las ondas del electrocardiograma, después del complejo QRS se encuentra la

onda T, la distancia entre la onda T y QRS es denominado "segmento ST". Este segmento refleja la recuperación eléctrica de las células, proceso también conocido como repolarización.

3.2.3. Terminología para el área de ciencias de la computación

- Redes neuronales artificiales [31]: son un modelo computacional basado en la aglomeración de neuronas simples que procesan información de manera análoga por medio de su estructura de conjuntos. Las neuronas artificiales de las redes neuronales están conectadas a través de enlaces y procesan la información por medio pesos, transportando datos en un proceso conocido como sinapsis. Su estructura está definida de 3 capas, la de entrada, la de salida y la capa escondida, siendo esta última donde ocurre el procesamiento de la información.

- Algoritmos evolutivos [32]: En las ciencias de la computación, la computación evolutiva es una subárea de la inteligencia artificial que se encarga de crear algoritmos tomando como base los principios de la teoría de la evolución de Charles Darwin. En esta técnica se toman como base cuatro puntos. Primero una población inicial que suele competir y/o reproducirse, acudiendo de esta manera al segundo y tercer aspecto, competencia y reproducción, respectivamente. Dejando como cuarto punto la variación o mutación, correspondiente a los cambios genéticos entre una generación y otra. En la implementación computacional, se cuenta con una función de desempeño o fitness que ayuda a determinar si se ha llegado a una solución o que tan bueno/apto es un individuo para un problema dado.

- PSO [33]: en inglés Particle Swarm Optimization, en español, optimización por enjambre de partículas. Su nombre lo recibe porque esta técnica usa una metaheurística que evoca el comportamiento de las partículas en la naturaleza. Esta técnica consiste en mover las posibles soluciones (partículas) por un espacio, donde se evalúa la eficacia de cada una y permite la convergencia de las mejores soluciones entre las candidatas originales. Al ser una metaheurística y no basar su funcionamiento en técnicas de optimización comprobadas, no garantiza una solución óptima.

- Neuroph [34]:es un framework ligero para el desarrollo de arquitecturas comunes de redes neuronales artificiales en Java. Contiene una biblioteca Java de código abierto diseñada con clases básicas que corresponden a conceptos básicos de redes neuronales. También tiene un editor gráfico de redes neuronales, el cual permite crear y adicionar rápidamente componentes. Se distribuye como código abierto bajo la licencia Apache 2.0.

- Teorema de Kolmogorov [35]: es un teorema cuya demostración presenta bases para el diseño estructural de una red neuronal, este dice: *Dada cualquier función continua f(0,1)^n en R ^{m}, existe una RNA de 3 capas, de propagación hacia adelante, con n elementos de proceso en la capa de entrada, m en la de salida y (2n+1) en la capa oculta; que implementa dicha función de forma exacta.* Es importante mencionar que este es un teorema de existencia y no hay ninguna técnica que diga cómo obtener la arquitectura de red para un determinado problema, ni mucho menos cuál es la arquitectura de red óptima para el problema.

Capítulo 4

Diseño de los algoritmos y el prototipo

Con el objetivo de definir las características de la RNA y el AE, a continuación se presenta el diseño de los mismos y la descripción del proceso a a seguir para sus correspondientes implementación y pruebas.

4.1. Selección de base de datos y atributos de entrenamiento

La selección de la base de datos fue una tarea dispendiosa en cuestión de tiempo, puesto que se encontraba condicionada con la investigación en PIPEC-UV, un proyecto actualmente en ejecución dentro de la Universidad del Valle. Los factores limitantes son mencionados a continuación:

Electrocardiograma previamente analizado: uno de los atributos utilizados en los trabajos que involucran predicción de riesgos cardiovasculares son las ondas de ECG. Sin embargo, en esta investigación no se tenía el objetivo de procesar dichas ondas, sino que se tenía contemplado utilizar un análisis binario de esta información; es decir, que se tuviera un veredicto de si el paciente tenía un registro de ondas de ECG normal o anormal.

Atributos del proyecto PIPEC-UV: las bases de datos a buscar debían contener atributos que estuviesen dentro de los que el proyecto PIPEC-UV puede capturar de los pacientes. Esta condición era flexible en tanto no se esperaba encontrar una base de datos igual a la utilizada por el proyecto en cuestión, sino una que tuviese los atributos en común necesarios para el entrenamiento de los algoritmos. Sin embargo, este hecho se vio reducido a la base de datos que más atributos de PIPEC-UV pudiese abarcar.

Teniendo en cuenta ambos factores, se encontraron dos bases de datos candidatas: Smart Health for Assessing the Risk of Events via ECG Database (Physionet) [36] y Heart Disease UCI o Cleveland (Kaggle) [5] cuya estructura y los atributos que comparten con PIPEC-UV y son presentados en la tabla 4.1:

En la tabla 4.1 se resaltan en amarillo los atributos compartidos entre PIPEC-UV y la base de datos de Physionet (10 atributos en común), en azul los compartidos entre PIPEC-UV y la base de datos de Kaggle (7 atributos en común) y en verde los atributos comunes entre PIPEC-UV y ambas bases de datos. Es evidente que PIPEC-UV comparte una mayor cantidad de datos con Physionet, sin embargo, la base de datos de Kaggle proporciona datos de electrocardiograma diagnosticado como normales o anormales y establece si el paciente está o no en riesgo cardiovascular, lo cuál es ideal para el entrenamiento de ambas estrategias de IA, la de RNA y la de AE. Por esta razón, para esta investigación se seleccionó la base de datos de Kaggle.

Atributo	Physionet	Kaggle	PIPEC-UV
Género	X	X	X
Edad	X	X	X
Peso	X		X
Altura	X		X
Superficie corporal	X		
Índice de masa corporal	X		X
Paciente fumador	X		X
Presión sistólica	X	X	X
Presión diastólica	X		X
Grosor íntima-media carotídeo	X		
Masa ventricular izquierda	X		
Eyección de fracción	X		
Evento cardiovascular	X		X
Tipo de dolor en el pecho			
Colesterol sérico		X	X
Azúcar en la sangre		X	X
Electrocardiograma (ondas)	X		X
Electrocardiograma (diagnosticado)		X	X
Máximo ritmo cardiaco		X	X
Angina inducida por ejercicio		X	
Depresión del segmento ST después del ejercicio		X	
Pendiente del pico del segmento ST después del ejercicio		X	
Cantidad de vasos sanguíneos mayores		X	
Caso de talasemia		X	
Diagnóstico de riesgo cardiovascular		X	

Tabla 4.1: Atributos en las bases de datos y el proyecto PIPEC-UV.

En adición, teniendo en cuenta la base de datos seleccionada, los atributos que esta compartía con PIPEC-UV y el diagnostico de RCV, fueron seleccionados ocho atributos para el entrenamiento de la RNA y el AE, algunos de estos son binarios y otros enteros, y son listados a continuación:

- Género (binario).
- Edad (entero).
- Presión sistólica (entero).
- Colesterol sérico (entero).
- Azúcar en la sangre, mayor a 120 mg/dL (binario).
- Electrocardiograma diagnosticado como normal o anormal (binario).
- Máximo ritmo cardiaco (entero).
- Diagnóstico de riesgo cardiovascular (binario).

Teniendo en cuenta que la base de datos seleccionada tiene 303 registros, esta se dividirá en la base de datos de entrenamiento con 243 registros y la base de datos de prueba con 60 registros, ya que de manera similar se hizo la división de este conjunto de datos en el trabajo de Khan [10].

4.1.1. Opinión de expertos respecto a la selección de los atributos

Después de consultar a 3 expertos (2 médicos internos y 1 médico cirujano), opinan que los atributos seleccionados para entrenar los algoritmos de predicción son bastante útiles y podrían arrojar buenos resultados, recomiendan seguir con la ejecución del proyecto dadas las limitaciones existentes por las bases de datos y el proyecto PIPEC-UV. Sin embargo, indican que cuando se cuente con más atributos, los datos clínicos se deben clasificar en modificables y no modificables, para posteriormente tener en cuenta información como: perímetro abdominal, raza, índice de masa corporal, relación entre presión sistólica y diástolica, microalbuminuria, creatinina en suero, antecedentes cardiovasculares en la familia, si el paciente es fumador, si hace ejercicio y si lleva una dieta hipercalórica.

4.2. Criterios de comparación entre las estrategias de inteligencia artificial

Con el objetivo de establecer criterios que permitieran comparar las características de la RNA y el AE, se utilizó la bibliografía consultada cuando se formuló esta propuesta de trabajo de grado. Sin embargo, se hizo una investigación adicional con el fin de encontrar más criterios para comparar los algoritmos implementados.

En consecuencia, se hallaron las investigaciones presentadas en la tabla 4.2, que sin ser necesariamente de procesamiento de datos clínicos, ofrecen una perspectiva sobre comparaciones entre AE y RNA.

Resulta evidente que la precisión de los algoritmos es un tema de discusión frecuente en investigaciones que involucran AE y RNA, pues los trabajos mencionados en el marco teórico de esta tesis también evalúan estos aspectos. Incluso, las características evaluadas de los trabajos previamente consultados en

Artículo	Características comparadas
Comparative Evaluation of Genetic Programming and Neural Network as Potential Surrogate Models for Coastal Aquifer Management [37]	Iteraciones y generaciones necesarias para alcanzar la solución. Precisión de los algoritmos.
A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models [38]	Tiempo de entrenamiento. Precisión de los algoritmos.
A comparison of linear genetic programming and neural networks in medical data mining [39]	Tiempo de entrenamiento. Precisión de los algoritmos.

Tabla 4.2: Artículos y criterios de comparación.

dicho marco teórico, están contenidas en las presentadas en la tabla 4.2. También es evidente la evaluación del tiempo de entrenamiento de los algoritmos, ya sea que la mencionen como tiempo de entrenamiento, velocidad de aprendizaje o iteraciones necesarias para llegar a la solución. Debido a esto se decidió utilizar los siguientes criterios de comparación para esta investigación:

Precisión: corresponde a la cantidad de casos cuya predicción es correcta, dividido por la cantidad total de casos analizados.

Tiempo de entrenamiento: corresponde al tiempo que se tardó en entrenarse el algoritmo, desde su primera ejecución hasta que alcanzó la precisión deseada.

Adicionalmente, se proponen los siguientes criterios adicionales para aumentar el espectro del análisis comparativo entre los algoritmos implementados:

Eficiencia: corresponde al comportamiento del algoritmo ante grandes cantidades de datos. Se harán pruebas para 500, 700, 1300 y 1500 registros de pacientes.

Capacidad de reentrenamiento (con los mismos datos): corresponde a la capacidad del algoritmo de continuar con su entrenamiento y aumentar su precisión sin la necesidad de incluir nuevos pacientes a la base de datos de entrenamiento.

Capacidad de reentrenamiento (con datos nuevos): corresponde a la capacidad del algoritmo de continuar su entrenamiento y aumentar su precisión agregando nuevos registros de pacientes la base de datos de entrenamiento.

Mejora de la precisión al ser reentrenados: corresponde a la evaluación de la precisión de los algoritmos al ser reentrenados bajo los dos criterios anteriores. Este aspecto se evaluará en conjunto con la capacidad de entrenamiento tanto con los mismos datos como con los datos nuevos.

Por último, se hará una comparación de la precisión y tiempo de entrenamiento de ambas soluciones frente a un trabajo que ya se encuentre publicado.

4.3. Red neuronal artificial

La red neuronal artificial para este trabajo será un perceptrón multicapa debido a que los atributos de los pacientes y sus salidas presentan relaciones no lineales. El entrenamiento será con propagación hacia atrás con momentum. La red se implementará de manera iterativa, basado en los resultados que proporcione después de ser entrenada bajo cada configuración. Esto implica que sus características respecto a su forma y entrenamiento estarán en constante cambio. Las características a tener en cuenta en el diseño de la RNA son las siguientes:

Capas: La cantidad de neuronas es estática sólo en las capas de entrada y salida, es decir 7 neuronas para la capa de entrada (de acuerdo a los atributos seleccionados en la sección 4.1) y 1 neurona para la capa de salida. Por otro lado, la capa oculta puede variar respecto a cuántas subcapas contiene, donde el tamaño de cada subcapa debe ser inferior al tamaño de la subcapa anterior, a excepción de la primera capa oculta; pues esta última estará conectada directamente a la capa de entrada (de tamaño n) y de acuerdo al teorema de Kolmogorov su tamaño puede ser 2n+1 o, según Hecht-Neilson [40], puede tener tamaño n/2. En consecuencia, las configuraciones de neuronas a probar son (las capas están separadas por guiones).

- 7-14-1.
- 7-4-1.
- 7-14-8-1.
- 7-14-11-8-1.
- 7-14-10-8-1.

Función de activación: la función de activación de la red neuronal artificial será sigmoide, debido a que esta se utiliza con valores entre 0 y 1 y entre los atributos se tienen valores binarios. Por tanto, los valores continuos deberán ser normalizados para que se encuentren en el rango (0,1).

Razón de aprendizaje y momentum: de acuerdo a Swingler en [41] estos dos valores se describen como uno solo debido a que están relacionados respecto a la velocidad en la que disminuye el error durante el entrenamiento de la RNA y también permiten controlar si el error presenta oscilaciones durante el entrenamiento. Esto significa que una razón de aprendizaje baja evita la oscilación del error pero puede acercar la solución a mínimos locales, en cambio, un valor alto del momentum evita que la red se quede en mínimos locales, pero puede provocar oscilaciones. Por tanto, estos valores serán alterados de la siguiente manera durante el proceso iterativo.

- Si el error disminuye lentamente de manera estable, se elevarán la razón de aprendizaje y el momentum.
- Si el error oscila, se reducirán la razón de aprendizaje y el momentum, aunque este último en menor medida para evitar los mínimos locales.
- Si el error se mantiene estable, se considerará que la red neuronal ha llegado a una solución.

Peso y bias: estos valores son definidos automáticamente durante el entrenamiento, por tanto no serán definidos para cada iteración y su rol para el funcionamiento de la red neuronal se tomará como caja negra.

Respecto a la salida de la red neuronal, ya que se utiliza la función de transferencia sigmoidea, esta será continua entre 0 y 1, por tanto, se tomará como diagnóstico de paciente riesgo los valores de salida mayores o iguales a 0,5 y sin riesgo a los valores de salida menores a 0,5.

Por último, el lenguaje a utilizar para la implementación de la RNA será Java a través del Framework Neuroph [34], el cual da acceso a un entorno gráfico para el diseño de la red neuronal y permite la configuración de los atributos listados anteriormente.

4.3.1. Consolidación de la red neuronal artificial

Una vez definidos los aspectos téoricos, a continuación se define en pasos el modelo matemático que será soportado por el framework Neuroph para el entrenamiento de la RNA.

- Paso 1: expresar los atributos de entrada del algoritmo y los pesos correspondientes que se alterarán durante el entrenamiento.

$$A_i = A_1, A_2, A_3, A_4, A_5, A_6, A_7 \tag{4.1}$$

$$W_i = W_1, W_2, W_3, W_4, W_5, W_6, W_7 \tag{4.2}$$

  En lo anterior, A_i representa los 7 atributos de entrada y W_i representa sus pesos correspondientes.

- Paso 2: sumar el producto de los 7 atributos con sus 7 pesos correspondientes, agregando también el valor de la neurona bia (b).

$$F_k = (\sum_{i=0}^{7} A_i W_i) + b \tag{4.3}$$

  F_k almacenará el resultado de la sumatoria presentada en la ecuación 4.3 y posteriormente podrá será aplicada para las k capas de la red neuronal.

- Paso 3: Se definen la función de activación y su influencia sobre las neuronas de cada capa.

$$S = \frac{1}{1 + e^{-F}} \tag{4.4}$$

$$A_k = S(F_k) \tag{4.5}$$

  Donde S es la función de activación sigmoidal y A_k guarda la función de activación aplicada a la sumatoria F del conjunto de neuronas de la capa k.

- Paso 4: se calcula la salida de cada conjunto de neuronas, es decir que se aplica para todas las k capas.

$$O = \sum_{n=0}^{k} A_n \tag{4.6}$$

  Donde O corresponde a la salida de la red neuronal.

- Paso 5: se calcula el error en la última capa de la red neuronal, es decir, el error de la salida O, que está dado por la salida de A_k en la capa k, respecto al resultado esperado para los valores de entrada. Esto está dado por la derivada parcial de la composición de la salida en la capa k.

$$\delta^k = \frac{\delta A_k}{\delta S_k} \cdot \frac{\delta S_k}{\delta F_k} \tag{4.7}$$

- Paso 6: se retroprogaga el error a la capa anterior.

$$\delta^{k-1} = \delta^k \cdot \frac{\delta S_{k-1}}{\delta F_{k-1}} \tag{4.8}$$

- Paso 7: se aplica la corrección de los pesos con el algoritmo de propagación hacia atrás, el cual está dado por la siguiente relación.

$$W_{i_{nuevo}} = \alpha \delta^k W_{i_{antiguo}}(A_i) \tag{4.9}$$

  Donde $W_{i\ nuevo}$ indica el valor corregido del peso de la neurona i, α representa el momentum definido para el entrenamiento, δ^k es el error distribuido en la red previamente calculado, $W_{i\ antiguo}$ es el valor del error para la neurona i de la capa k antes de la retropropagación y A_i es la neurona i de la capa k

A través de los pasos anteriormente descritos se ejecutará el algoritmo con la ayuda de Neuroph, sin embargo, ya que se plantea un proceso iterativo y con varias configuraciones de capas a probar para la RNA, la implementación deberá dejar por escrito todo este proceso para consolidar la construcción de la red neuronal artificial.

4.4. Algoritmo evolutivo

El algoritmo evolutivo para este trabajo es implementado en su forma básica, esto es, obedeciendo a los 4 principios presentados en el teorema evolutivo básico: población, mutación, reproducción y competencia. Sin embargo, se hace necesario incluir un cromosoma que se encargará del almacenar la información correspondiente a cada uno de los atributos. Estos principios son presentados a continuación:

Cromosoma: corresponde a un arreglo con 2 filas y 8 columnas, donde las primeras 7 columnas son los atributos que se utilizarán para la predicción y la última, es el diagnóstico del paciente (0= sin riesgo, 1= con riesgo), lo cuál permitirá el entrenamiento del algoritmo evolutivo. Para la otra fila del arreglo, cada columna corresponderá a la desviación que podrá tener el valor del atributo en la misma columna, pero en la primera fila; este valor de desviación iniciará con el valor de cero en la primera generación. Es importante aclarar que los valores discretos y el diagnóstico del paciente no tendrán desviación. El arreglo se puede apreciar en la tabla 4.3.

Género	Edad	Presión sistólica	Colesterol en sangre	Azúcar en la sangre	Electrocardiograma diagnosticado	Máximo ritmo cardiaco	Diagnostico de RCV
—	Desviación	Desviación	Desviación	—	—	Desviación	—

Tabla 4.3: Cromosoma del algoritmo evolutivo.

Población: corresponde al grupo de cromosomas seleccionados para procesar cada paciente y otorgar un resultado respecto a si está o no en riesgo de sufrir una enfermedad cardiovascular. Esta se ve inicializada en la primera generación y alterada en generaciones posteriores a través del proceso representado en el siguiente pseudocódigo.

```
//Entradas: datos de pacientes o generación actual de cromosomas.
//En la primera iteración se crea la población inicial
para n pacientes hacer
    //Crea el arreglo de dos filas correspondiente al cromosoma.
    crearCromosoma();
fin
//Posteriormente se ejecutan los procesos de reproducción, mutación y competencia, ...
//... luego, se seleccionan los mejores cromosomas que conformarán las siguientes poblaciones
```

Algoritmo 1: Inicialización de la población

Reproducción: corresponde a la creación de hijos a partir de los cromosomas de dos padres. Esta es una reproducción puntuada, debido a que se busca adaptar cada cromosoma para que abarque al menos los pacientes abarcados por los dos cromosomas padre. La cantidad de cromosomas en una generación se tomará como n. En la primera generación, el valor n será igual a la cantidad de registros de pacientes que se tome para la base de datos de entrenamiento. En las generaciones posteriores, la cantidad de cromosomas estará determinada por el 98 % del valor n, sin embargo, para el entrenamiento se añadirán de nuevo los cromosomas originales para que se reproduzcan con la nueva generación; esto provocará que tras cada generación la población aumente con la siguiente razón:

$$PSG = n * 0{,}98 + TGE \tag{4.10}$$

Donde PSG corresponde a la población de la siguiente generación, n es el tamaño de la población actual y TGE es el tamaño de la población-generación original (tamaño de la base de datos de entrenamiento).

La primera restricción al reproducirse es que ambos cromosomas sean para el mismo diagnóstico, de esta manera no se podrá reproducir un cromosoma que diagnostique a un paciente sano con un cromosoma que diagnostique a un paciente con riesgo cardiovascular. Posteriormente, debido a que los cromosomas contienen atributos cuyos valores sólo pueden ser 0 o 1, solamente podrán reproducirse los cromosomas cuyo género, azúcar en sangre y electrocardiograma sean iguales; para esto, se analizará la posibilidad de reproducción de cada cromosoma con los otros n-1 cromosomas restantes de la población.

Cuando dos cromosomas son compatibles, inicia su proceso de reproducción. Para la reproducción de la primera generación se tiene que cada valor y desviación para los atributos no discretos estará dado por:

$$NuevoAtributo = \frac{AtributoP_1 + AtributoP_2}{2} \tag{4.11}$$

$$DesviacionNuevoAtributo = |NuevoAtributo - AtributoP_1| \tag{4.12}$$

Donde NuevoAtributo es el valor del nuevo atributo y está dado por el promedio entre los valores del mismo atributo para los padres (AtributoP1 y AtributoP2). Y, DesviacionNuevoAtributo corresponde a la desviación para el nuevo atributo y está dada por la magnitud de la diferencia entre el nuevo atributo y el atributo de uno de los padres.

El anterior proceso da como resultado un nuevo cromosoma, posteriormente es mutado para generar dos nuevos cromosomas y resultando la reproducción en 3 cromosomas nuevos, este proceso se explicará más adelante en la sección de mutación.

La reproducción es diferente a partir de la segunda generación, pues ya se debe tener en cuenta la desviación que trae el cromosoma para cada atributo, por tanto, involucra:

$$NuevoAtributo = \frac{menorValorDesviado + mayorValorDesviado}{2} \tag{4.13}$$

$$menorValorDesviado = menorEntre(AtributoP_1 - desvAtP_1, AtributoP_2 - desvAtP_2) \tag{4.14}$$

$$mayorValorDesviado = mayorEntre(AtributoP_1 + desvAtP_1, AtributoP_2 + desvAtP_2) \tag{4.15}$$

$$DesviacionNuevoAtributo = |NuevoAtributo - AtributoP_1| \tag{4.16}$$

En la ecuación (4.13), NuevoAtributo es el valor del nuevo atributo y está dado por el promedio entre el menor valor desviado (4.14) y el mayor valor desviado (4.15) del atributo de los cromosomas de los padres, asegurando de esta manera que los valores alcanzados por ambos cromosomas estén dentro del nuevo cromosoma hijo; en las ecuaciones (4.14) y (4.15) desvAtP_n corresponde a la desviación del atributo para el padre n. Por último, DesviacionNuevoAtributo (4.16) corresponde a la desviación para el nuevo atributo y está dada por la magnitud de la diferencia entre el nuevo atributo y el atributo de uno de los padres.

Este proceso de reproducción se representa a través del siguiente pseudocodigo.

```
//Entradas: cromosoma padre 1 (p1) y cromosoma padre 2 (p2).
si Es la primera generación entonces
    si p1.genero==p2.genero AND p1.azucar==p2.azucar AND p1.ecg==p2.ecg entonces
        para n posiciones del cromosoma hacer
            //Las siguientes líneas se calculan con las ecuaciones (4.11) y (4.12).
            nuevoCromosoma1= crearCromosomaHijo();
            nuevoCromosoma2= crearCromosomaHijoMutado();
            nuevoCromosoma3= crearCromosomaHijoMutado();
        fin
    fin
en otro caso
    si p1.genero==p2.genero AND p1.azucar==p2.azucar AND p1.ecg==p2.ecg entonces
        para n posiciones del cromosoma hacer
            //Las siguientes líneas se calculan con las ecuaciones (4.13), (4.14), (4.15) y (4.16).
            nuevoCromosoma1= crearCromosomaHijoVersion2();
            nuevoCromosoma2= crearCromosomaHijoMutadoVersion2();
            nuevoCromosoma3= crearCromosomaHijoMutadoVersion2();
        fin
    fin
fin
```

Algoritmo 2: Reproducción de los cromosomas

El anterior pseudocodigo presenta un método llamado crearCromosomaHijoMutado(), este corresponde a tomar el cromosoma resultante de crearCromosomaHijo() y aplicar el proceso de mutación, el cual se presenta en la siguiente sección.

Mutación: corresponde a la aleatoriedad otorgada a la estrategia evolutiva involucrada en este algoritmo, se aplica como una variación de máximo 5 unidades sobre la desviación de los atributos continuos, es decir:

$$d' = d + rand(5) \tag{4.17}$$

Donde d' es el valor nuevo para la desviación del atributo y estará dado por la desviación actual (d) más un valor aleatorio entre 1 y 5. Esta nueva desviación aleatoria será aplicada dos veces sobre el cromosoma resultante en la reproducción guardando ambos resultados, por lo tanto se tendrá un total de 3 cromosomas después de hacer dicho proceso, el cual se presenta a través del siguiente pseudocodigo.

```
//Entrada: cromosoma para mutar.
para n desde 0 hasta 6 hacer
    //El cromosoma tiene 7 posiciones, desde 0 hasta 6.
    si cromosoma[1][n] no es binario entonces
        cromosoma[1][n]= cromosoma[1][n]+rand(5);
    fin
fin
```

Algoritmo 3: Mutación del cromosoma.

Competencia: corresponde a la estrategia para filtrar los mejores cromosomas para la solución del algoritmo evolutivo, cada cromosoma se evalúa a través de la siguiente función de aptitud:

$$A_c = \frac{D_c}{D_p} \tag{4.18}$$

$$D_c = \sum_{i=0}^{n} diagnosticoPaciente(i) \tag{4.19}$$

$$D_p = \left[\sum_{i=0}^{n} pacienteAnalizable(i)\right] + 1 \tag{4.20}$$

Donde A_c es el valor de aptitud del cromosoma y puede ir desde 0 hasta 1, D_p corresponde a los diagnósticos posibles, que son todos los casos en los cuales el cromosoma permite analizar un paciente y, D_c corresponde a los diagnósticos correctos que es la cantidad de diagnósticos acertados de todos los posibles. De esta manera, un cromosoma que diagnostique bien a todos los pacientes que recibe, tendrá un fitness de 1,0 y, un cromosoma que no diagnostique paciente alguno, tendrá 0.0.

De acuerdo a las ecuaciones 4.19 y 4.20, cada cromosoma del algoritmo evolutivo tiene la capacidad de analizar y diagnosticar un paciente, este proceso se representa a través del siguiente pseudocodigo.

```
//Entrada: cromosoma c1 que hace el análisis y cromosoma c2 que representa los datos del
 paciente.
si c1.genero==c2.genero  c1.azucar==c2.azucar  c1.ecg==c2.ecg entonces
|  //El rango del atributo del cromosoma está dato por [atributo - desviacion, atributo +
|   desviacion]
|  si cada atributo entero se encuentra dentro del rango del mismo atributo dentro del
|   cromosoma entonces
|  |  //El diagnóstico del paciente es posible.
|  |  si c1.DiagnosticoRCV==c2.DiagnosticoRCV entonces
|  |  |  //El diagnóstico del paciente es correcto.
|  |  en otro caso
|  |  |  /El diagnóstico del paciente NO es correcto.
|  |  fin
|  en otro caso
|  |  //El diagnóstico del paciente No es posible.
|  fin
en otro caso
|  //El diagnóstico del paciente No es posible.
fin
```

Algoritmo 4: Análisis y diagnóstico del paciente por parte del cromosoma.

4.4.1. Consolidación del algoritmo evolutivo

De acuerdo a lo presentado a lo largo de la sección 4.4, se tienen los elementos suficientes para diseñar el entrenamiento del algoritmo evolutivo, por tanto, este queda representado de la siguiente manera en pseudocódigo.

```
//Entradas: base datos de pacientes (entrenamiento) y aptitudRequerida.
aptitudPromedio= 0;
mientras aptitudPromedio ≤ aptitudRequerida hacer
    cantidadCromosomas= 0;
    aptitud= 0;
    InicializarPoblacion();
    ReproducirCromosomas();
    para n cromosomas hacer
        aptitud+= AnalizarPacientes(); //De acuerdo a lo descrito en çompetencia".
        cantidadCromosomas+= 1;
    fin
    aptitudPromedio= aptitud/cantidadCromosomas;
fin
```

Algoritmo 5: Pseudocódigo del entrenamiento del algoritmo evolutivo.

4.5. Prototipo

Para probar el funcionamiento de los algoritmos se requerirá el uso de un prototipo que permita el análisis de casos específicos de pacientes y también probar grandes cantidades de datos. Por tanto, en esta sección se define el aspecto visual de dicho prototipo, sus entradas y salidas.

El prototipo tendrá un aspecto simple. Con dos secciones principales, una donde se permita el análisis individual de un paciente, otorgando su diagnóstico mediante los 7 atributos necesarios y procesamiento a través de ambos algoritmos, el de RNA y el evolutivo. Por otro lado, estará la sección de análisis masivo, donde se podrá cargar un archivo en formato CSV, este deberá tener el encabezado y sus atributos en el siguiente orden: edad, género, presión sistólica, colesterol, azúcar en sangre, ECG, máximo ritmo cardiaco. Además, el usuario podrá indicar si hay una columna adicional con el diagnóstico del paciente, aspecto que permitirá generar las matrices de confusión correspondientes para cada algoritmo. Como salida, el prototipo generará un archivo en formato CSV con una columna nueva, indicando el diagnóstico para cada paciente.

Las anteriores características se presentan en la figura 4.1, adicionalmente, para el análisis masivo de pacientes, el prototipo indicará el tiempo de ejecución en milisegundos para cada algoritmo.

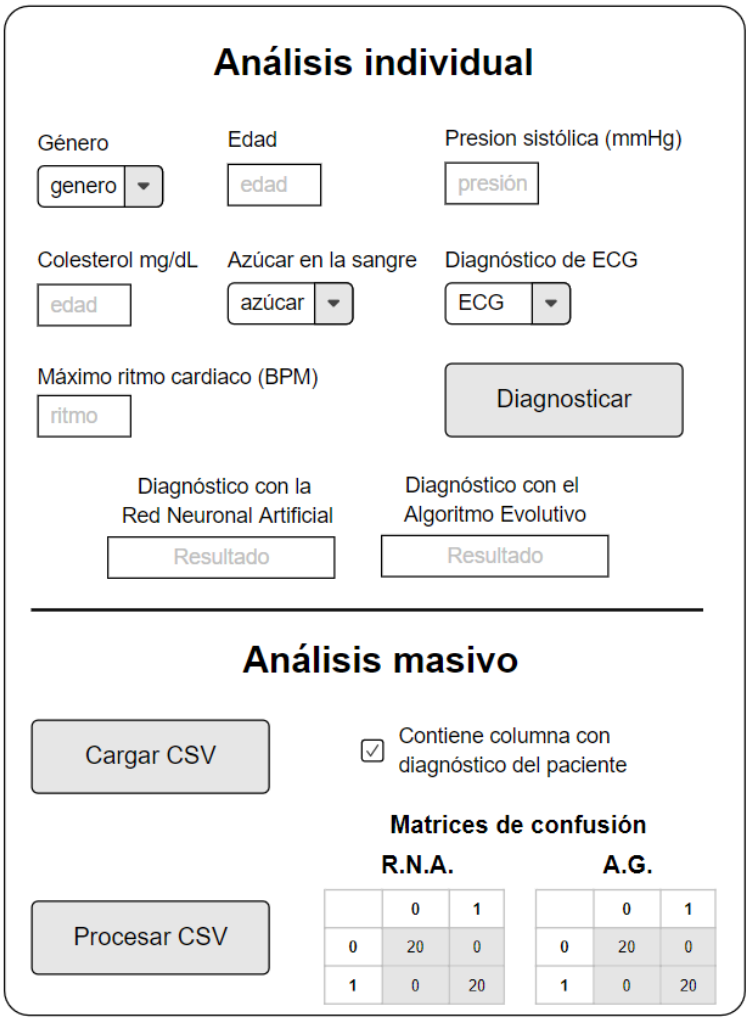

Figura 4.1: Diseño de la interfaz gráfica del prototipo

Capítulo 5

Implementación de los algoritmos y el prototipo

En este capítulo se presenta el registro de la implementación y entrenamiento de la RNA y el AE, es importante mencionar que ambos procesos fueron ejecutados sobre un computador con las siguientes características:

- Procesador: Intel Core i5 4460, 3,2 GHz
- Memoria: 8GB DDR3
- GPU: GTX 960 4GB
- HDD: 2TB, 7200RPM

5.1. Red neuronal

De acuerdo al diseño de la red neuronal artificial, presentado en el capítulo 4, se debía ejecutar pruebas para diferentes configuraciones, evaluando la red neuronal y el comportamiento de su error general para saber cuándo se había llegado a una solución. Teniendo esto en cuenta, se tiene el siguiente registro de ejecución.

5.1.1. Primera iteración

En la tabla 5.1 se tiene un total de 9 ejecuciones para las 5 diferentes configuraciones de capas propuestas para la RNA. La ejecución y registro de esta iteración tomó 20 minutos.

La primera configuración, 7-14-1 tomó 4 ejecuciones para alcanzar el error deseado de 0,01, punto en el que alcanzó una precisión del 71,7 %.

La segunda configuración, con las capas 7-4-1 presenta un error general de la red que desciende muy lentamente, siendo casi estable. Esta situación se presentó en las dos ejecuciones, no se decide hacer más debido a que la estabilidad del error denotaba que se había alcanzado una solución, teniendo esta una precisión del 66,7 %.

La tercera configuración, con las capas 7-14-11-1 alcanzó el error deseado en la primera ejecución al cabo de 9803 iteraciones. Aunque para llegar al error de 0,01 presentó oscilaciones, alcanzó una precisión del 65 %.

N	Iteraciones máximas	Error general requerido	Capas	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones	Precisión
1	10000	0,01	7-14-1	0,2	0,7	El error baja de manera constante, pero lo hace lentamente	0,278	10000	0,62
2	10000	0,01	7-14-1	0,4	0,8	El error sigue bajando de manera constante	0,248	10000	0,7
3	10000	0,01	7-14-1	0,6	0,9	El error sigue bajando. Se procede a entrenarla con 5000 iteraciones más	0,12	10000	0,72
4	5000	0,01	7-14-1	0,6	0,9	El error se alcanzó	0,01	5000	0,72
5	10000	0,01	7-4-1	0,2	0,7	El error se mantiene estable, aunque después de las 5500 iteraciones, empieza a disminuir. Sigue siendo un error muy alto	0,672	10000	0,48
6	10000	0,01	7-4-1	0,4	0,8	El error se mantiene estable, se procede a cambiar la configuración	0,647	10000	0,67
7	10000	0,01	7-14-11-1	0,2	0,7	Tuvo oscilaciones, pero se alcanzó el error	0,01	9803	0,65
8	10000	0,01	7-14-8-1	0,2	0,7	El error se mantuvo estable hacia el final	0,116	10000	0,65
9	10000	0,01	7-14-11-8-1	0,2	0,7	Tuvo muchas oscilaciones, pero alcanzó el error rápidamente	0,01	3824	0,65
10	10000	0,01	7-14-10-8-1	0,2	0,7	Tuvo muchas oscilaciones, pero alcanzó el error rápidamente	0,01	5216	0,55

Tabla 5.1: Primera iteración de implementación de la red neuronal.

La cuarta configuración, con las capas 7-14-8-1 estuvo cerca de alcanzar el error, pero se mantuvo estable hacia el final, consiguió una precisión del 65 %.

La quinta configuración, con las capas 7-14-11-8-1 alcanzó el error deseado en la primera ejecución y fue la más rápida en hacerlo, pues lo consiguió en 3824 iteraciones. Aunque presentó oscilaciones, consiguió una precisión del 65 %.

La sexta configuración, con las capas 7-14-10-8-1 alcanzó el error deseado en la primera ejecución, lo hizo en 5216 iteraciones y, aunque presentó oscilaciones, consiguió una precisión del 55 %.

Después de estas ejecuciones, se analizó cuáles configuraciones alcanzaron mayor precisión y cuáles lo hicieron en el menor tiempo, con el objetivo de continuarlas entrenando en una siguiente iteración. Las seleccionadas están resaltadas en verde en la tabla 5.1 y fueron las siguientes:

- 7-14-1, aunque tomó más tiempo que las otras configuraciones, alcanzó la mayor precisión (71,7 %). En la figura 5.1 se presenta el comportamiento del error en su entrenamiento.
- 7-4-1, es la segunda precisión más alta (66,7 %), presentó un error estable, pero se analizará si se pueden incrementar la razón y el momentum para mejorar el entrenamiento. En la figura 5.2 se presenta el comportamiento del error en su entrenamiento.
- 7-14-11-8-1, alcanzó una precisión del 65 %, aunque no fue la única en llegar este porcentaje, fue la que lo logró más rápido. En la figura 5.3 se presenta el comportamiento del error en su entrenamiento.

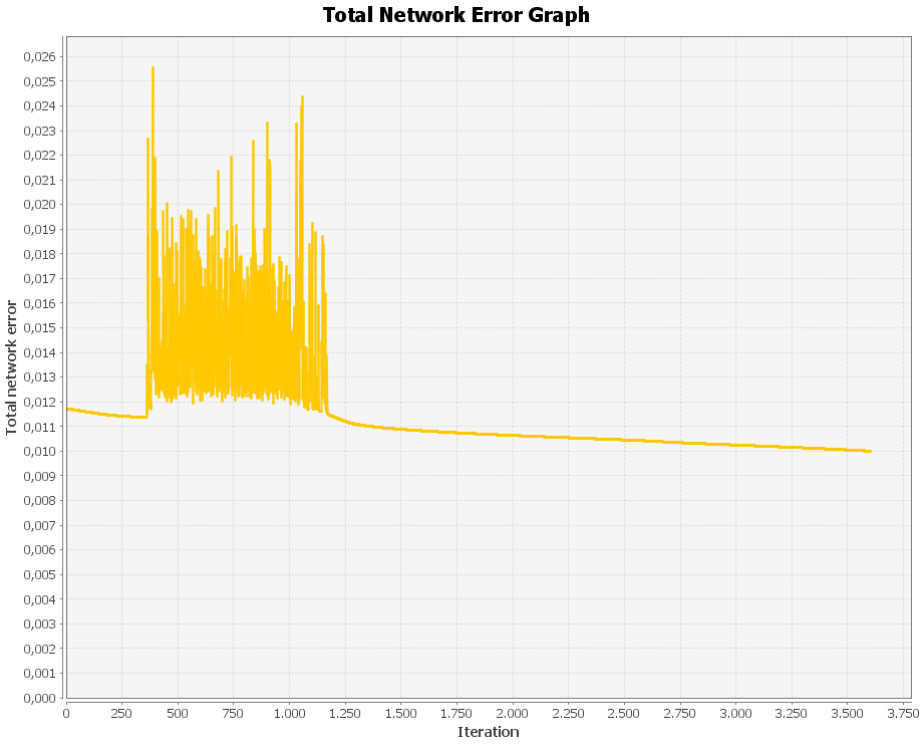

Figura 5.1: Error durante entrenamiento de la red 7-14-1

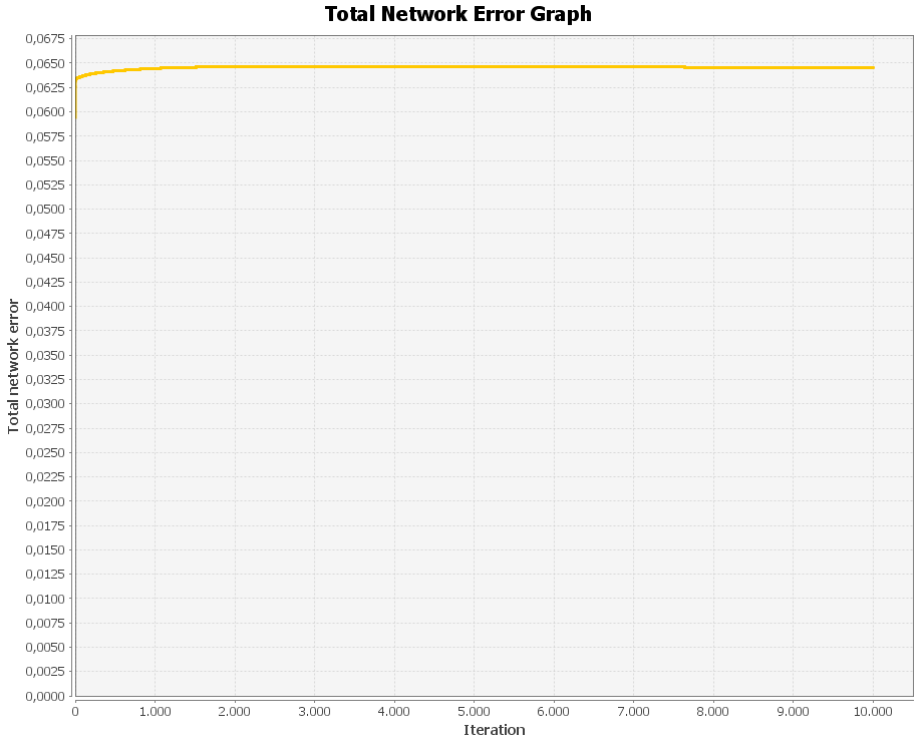

Figura 5.2: Error durante entrenamiento de la red 7-4-1

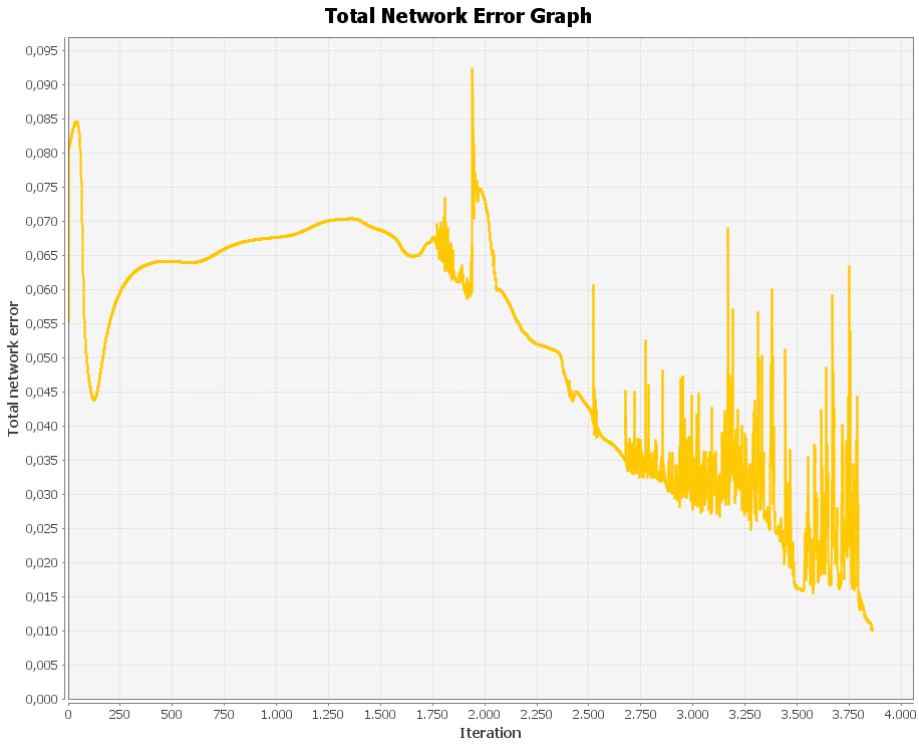

Figura 5.3: Error durante entrenamiento de la red 7-14-11-8-1

5.1.2. Segunda iteración

En la tabla 5.2 se tiene un total de 4 ejecuciones para las 3 diferentes configuraciones de capas propuestas para esta segunda iteración de entrenamiento de la RNA, en esta ocasión se deseaba un error general de 0,007. La ejecución y registro de esta iteración tomó 14 minutos.

N	Iteraciones máximas	Error general requerido	Capas	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones	Precisión
1	7000	0,007	7-4-1	0,4	0,8	El error baja tan lentamente que es casi estable, no hay oscilaciones, se corre aumentando ambos	0,639	7000	0,64
2	7000	0,007	7-4-1	0,7	0,9	El error baja muy lentamente	0,618	7000	0,63
3	7000	0,007	7-14-1	0,4	0,8	El error es similiar a menos raíz de x, luego tiende a estabilizarse	0,0083	7000	0,7
4	7000	0,007	7-14-11-8-1	0,2	0,5	Al bajar el momentum, disminuyen las oscilaciones, alcanza el error deseado	0,007	1178	0,7

Tabla 5.2: Segunda iteración de implementación de la red neuronal.

La primera configuración, con las capas 7-4-1 se reentrenó con los mismos valores para la razón de aprendizaje y momentum, en este caso con un máximo de 7000 iteraciones y mantuvo estabilidad en el error. Sin embargo, se volvió a correr con mayor momentum y mayor razón de aprendizaje, el error general disminuyó lentamente, casi de manera estable, por lo tanto se establece que llegó a una solución con una precisión del 63,3 %.

La segunda configuración, con las capas 7-14-1 se reentrenó reduciendo los valores de la razón de aprendizaje y el momentum con la finalidad de eliminar las oscilaciones del error, se consigue estabilidad y alcanza un error cercano al deseado con una precisión del 70 %.

La tercera configuración, con las capas 7-14-11-8-1 se reentrenó reduciendo el momentum para evitar las oscilaciones, en efecto disminuyeron. Se alcanzó el error deseado en 1178 iteraciones con una precisión del 70 %.

Después de estas ejecuciones, se analizó cuáles configuraciones alcanzaron mayor precisión y si su error aún podía disminuirse con entrenamientos posteriores en una siguiente iteración. Las seleccionadas están resaltadas en verde en la tabla 5.2 fueron las siguientes:

- 7-14-1, aunque su precisión se vio disminuida, el comportamiento del error indica que aún puede bajar más. En la figura 5.4 se presenta el comportamiento del error en su entrenamiento.

- 7-14-11-8-1, su precisión aumentó y el error indica que aún puede disminuirse. En la figura 5.5 se presenta el comportamiento del error en su entrenamiento.

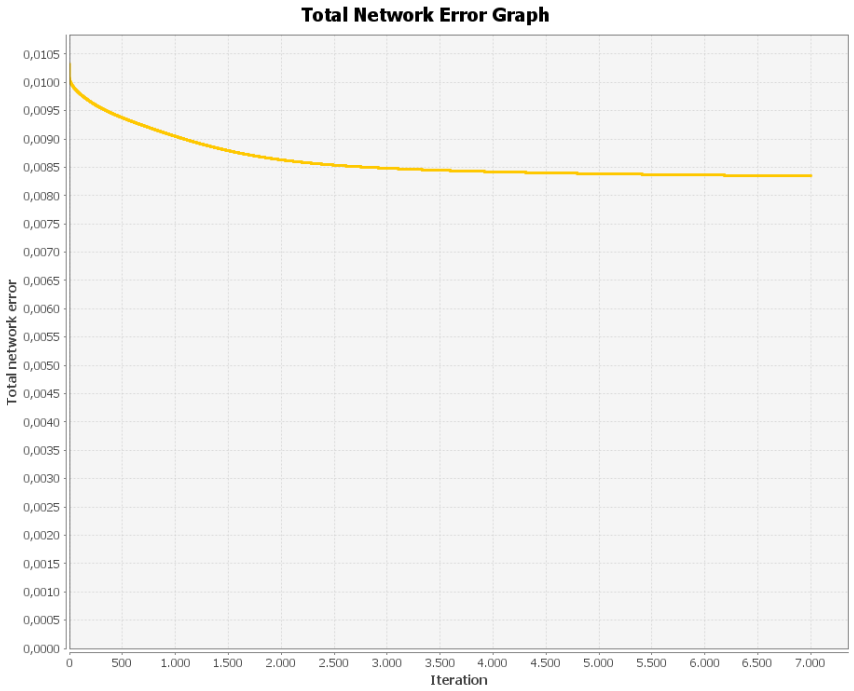

Figura 5.4: Error durante entrenamiento de la red 7-14-1

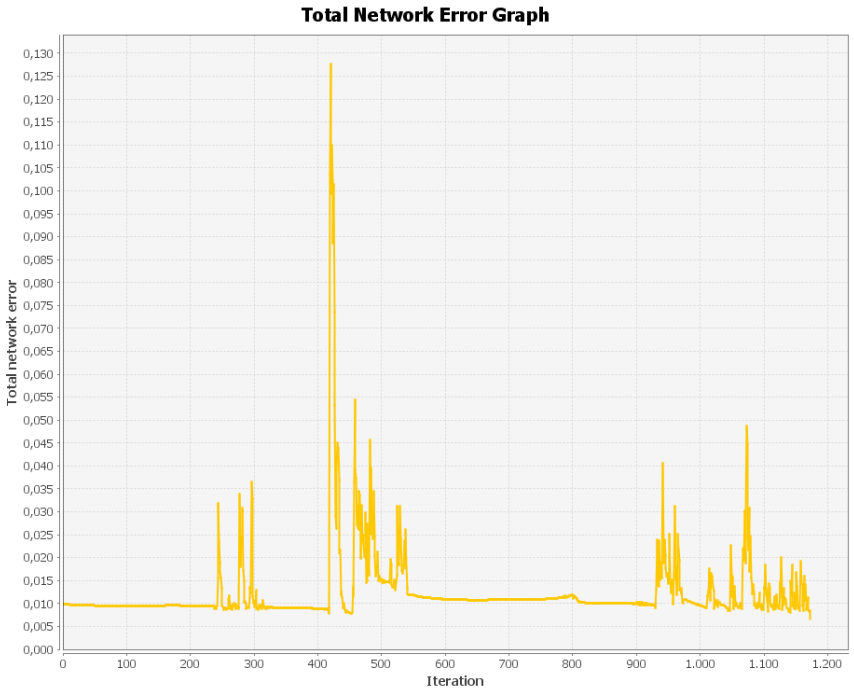

Figura 5.5: Error durante entrenamiento de la red 7-14-11-8-1

5.1.3. Tercera iteración

En la tabla 5.3 se tiene un total de 2 ejecuciones, una para cada una de las 2 diferentes configuraciones de capas propuestas para esta tercera iteración de entrenamiento de la RNA, en esta ocasión se deseaba un error general de 0,006. La ejecución y registro de esta iteración tomó 9 minutos.

N	Iteraciones máximas	Error general requerido	Capas	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones	Precisión
1	5000	0,006	7-14-1	0,6	0,9	Error baja muy lentamente	0,00828	5000	0,7
2	5000	0,006	7-14-11-8-1	0,1	0,5	No se perciben oscilaciones, alcanza el error	0,006	281	0,68

Tabla 5.3: Tercera iteración de implementación de la red neuronal.

La primera configuración, con las capas 7-14-1 se reentrenó elevando tanto el momentum como la razón de aprendizaje y el error seguía bajando lentamente, de manera casi estable y no alcanzó el valor deseado. Por tanto, se determina que se llegó a una solución y no se puede mejorar. En la figura 5.6 se presenta el comportamiento del error en su entrenamiento.

La segunda configuración, con las capas 7-14-11-8-1 no presentó oscilaciones y el error llegó al valor deseado. Sin embargo, la precisión disminuyó a 68,3 %. En la figura 5.7 se presenta el comportamiento del error en su entrenamiento.

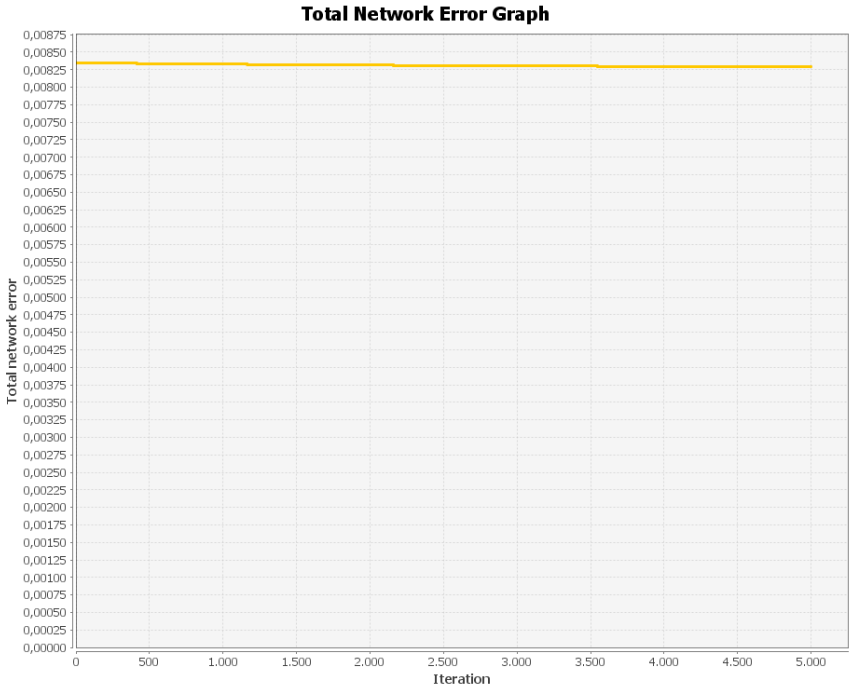

Figura 5.6: Error durante entrenamiento de la red 7-14-1

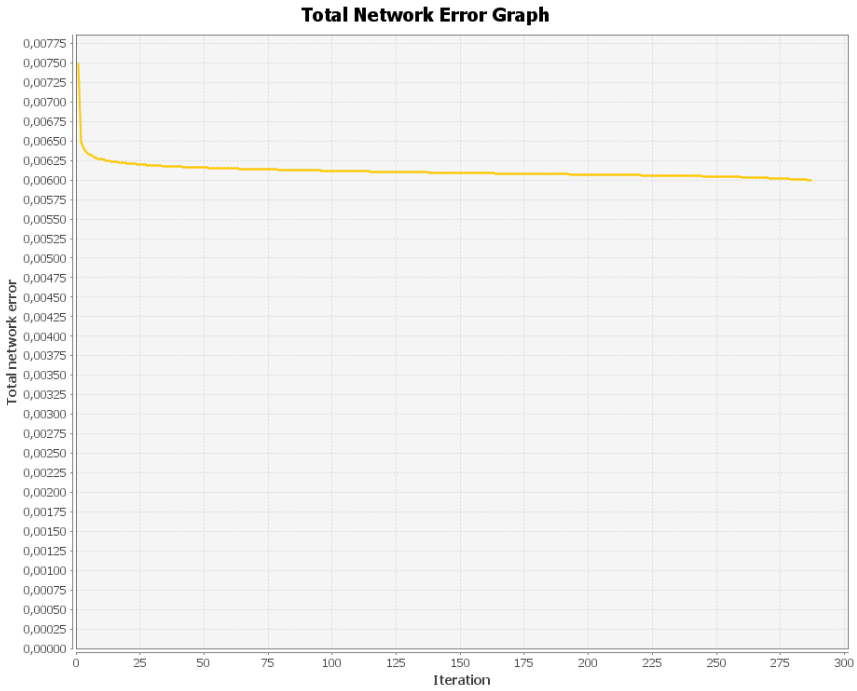

Figura 5.7: Error durante entrenamiento de la red 7-14-11-8-1

Al analizar los resultados de esta iteración se percibe una disminución en la precisión del algoritmo y parece que la configuración seleccionada no fue la adecuada. Sin embargo, en la primera iteración había una configuración que también alcanzó el error deseado en la primera ejecución, esta es la de las capas 7-14-10-8-1, la cual tuvo una precisión del 55 % pero su error bajaba constantemente, así que se hace una iteración adicional para continuar el entrenamiento de esta RNA.

5.1.4. Iteración adicional

En la tabla 5.4 se tiene un total de 3 ejecuciones para una configuración de la RNA: 7-14-10-8-1. en esta ocasión se deseaba un error general de 0,003. La ejecución y registro de esta iteración tomó 6 minutos.

En la primera ejecución, el error deseado era de 0,007 y lo alcanzó sin oscilaciones después de reducir el momentum. Además, logró una notable mejora en su precisión, pues ahora su valor es de 76,7 %. Debido a que el error parece poder bajar más, se hace un reentrenamiento con los mismos valores para la razón de aprendizaje y el momentum.

En la segunda ejecución, el error deseado era de 0,005 y lo alcanzó sin oscilaciones. Sin embargo, en este caso no mejoró su precisión, pues se mantuvo en el mismo valor (76,7 %). Teniendo en cuenta que el error parece poder seguir bajando, se continúa el entrenamiento con los mismos valores para la razón de aprendizaje y el momentum. En la figura 5.8 se presenta el comportamiento del error en su entrenamiento.

En la tercera ejecución, el error deseado era de 0,003, lo alcanzó con oscilaciones al inicio y hacia el final se estabilizó, lo cual indica que se llegó a una solución. Sin embargo, la precisión se redujo al 60 %, por tanto no se continúa el entrenamiento y la RNA tomada como mejor solución es la de la segunda ejecución. En la figura 5.9 se presenta el comportamiento del error en su entrenamiento.

N	Iteraciones máximas	Error general requerido	Capas	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones	Precisión
1	7000	0,007	7-14-10-8-1	0,2	0,5	No se perciben oscilaciones, alcanza el error	0,007	286	0,77
2	5000	0,005	7-14-10-8-1	0,2	0,5	No se perciben oscilaciones, alcanza el error, aparenta poder seguir disminuyendo (figura 5.8	0,005	492	0,77
3	4000	0,003	7-14-10-8-1	0,2	0,5	Presenta oscilaciones al inicio, se alcanza el error con estabilidad(figura 5.9	0,003	747	0,6

Tabla 5.4: Iteración de implementación de la red neuronal.

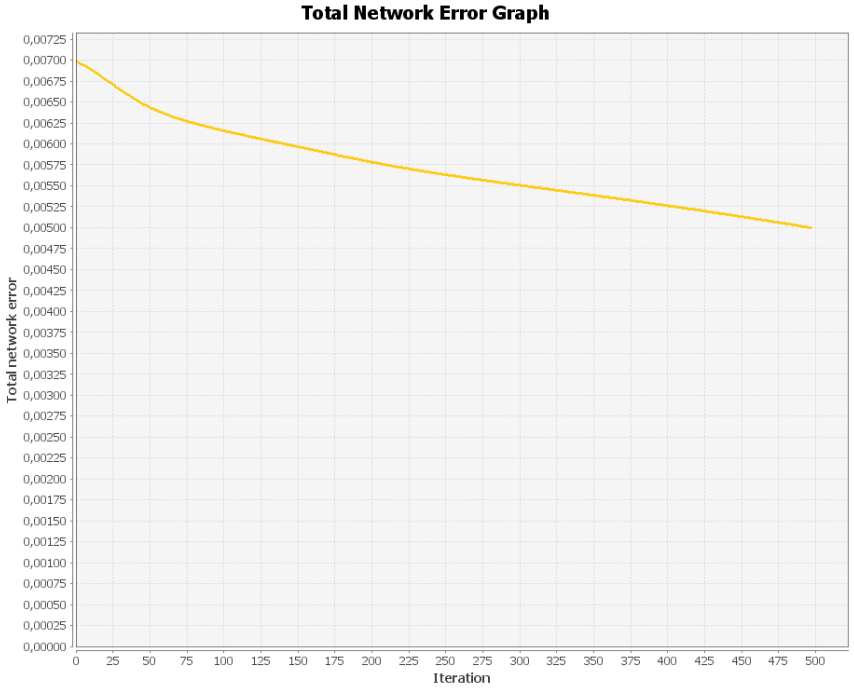

Figura 5.8: Error durante entrenamiento de la red 7-14-10-8-1 (caso exitoso)

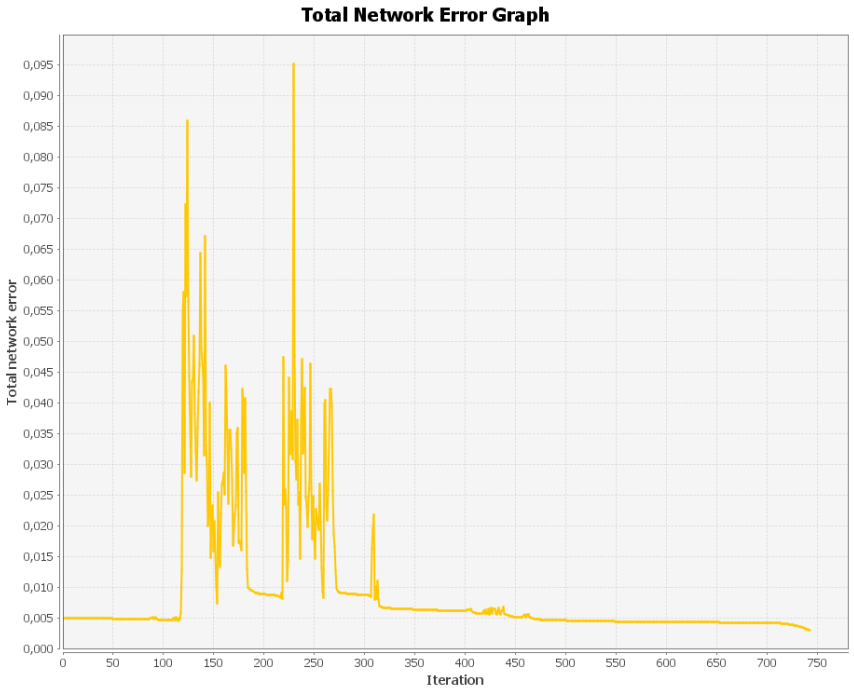

Figura 5.9: Error durante entrenamiento de la red 7-14-10-8-1

5.1.5. Red Neuronal seleccionada

De acuerdo al proceso iterativo para encontrar una configuración adecuada para la RNA para la predicción de riesgos cardiovasculares, se tiene que la seleccionada es la presentada en la figura 5.10, tiene una configuración de 7-14-10-8-1 y está totalmente conectada. Adicionalmente, cada subcapa de la capa oculta contiene una neurona bias asignada para que en el entrenamiento se tuviese más control de la activación de cada neurona dentro de la capa oculta..

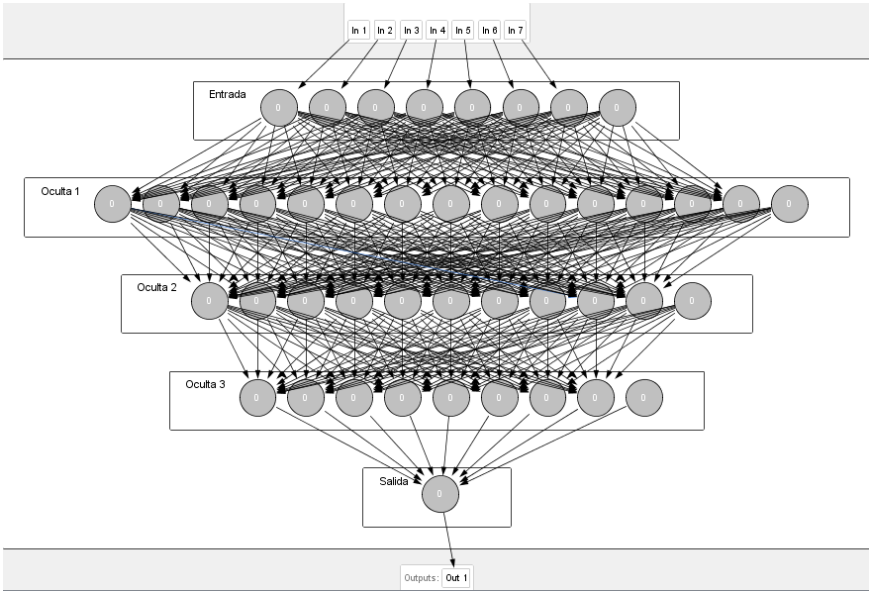

Figura 5.10: Red neuronal seleccionada (7-14-10-8-1)

5.2. Algoritmo evolutivo

De acuerdo al diseño del algoritmo evolutivo, presentado en el capítulo 4, se procedió a implementar el cromosoma como un vector de 3 dimensiones en lugar de 2, esto con el objetivo de dar lugar a trabajos futuros con mejoras que involucren alguna variación adicional para los atributos. Por otro lado, la población, mutación, reproducción y competencia fueron implementados como se diseñaron originalmente.

El algoritmo se implementó en el lenguaje de programación Java. Las distintas generaciones fueron almacenadas en archivos planos en formato txt, incluyendo la generación final (solución), la cuál es cargada desde dicho archivo cuando se requiere hacer un diagnóstico.

El entrenamiento del algoritmo evolutivo inició con la base de datos de entrenamiento sin modificaciones, sin embargo, después de la tercera iteración se hacia notable la tendencia del algoritmo de dejar a un lado los cromosomas de pacientes cuyo diagnóstico fuese "sin riesgo cardiovascular", esto provocaba que el algoritmo se quedase sin la información necesaria para ser probado y, por tanto, el entrenamiento no alcanzaba la precisión deseada.

En vista de esto, se decidió separar la base de datos en dos, una para pacientes con riesgo y otra para pacientes sin riesgo. De esta manera se corrió el entrenamiento de dos algoritmos evolutivos similares, diferenciándose sólo de el tipo de diagnóstico que otorgaban.

El entrenamiento del algoritmo para los pacientes sin riesgo tomó 4 horas y 44 minutos y dio paso a 10 generaciones hasta alcanzar en la última una precisión de 68.04 % sobre el conjunto de datos de prueba, con 1000 cromosomas como se evidencia en la figura 5.11.

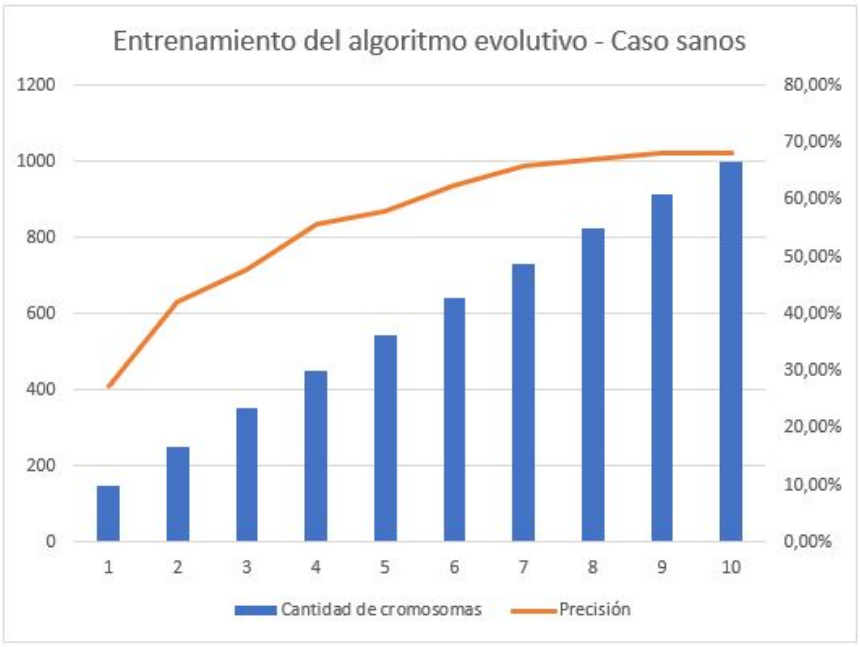

Figura 5.11: Precisión, cantidad de cromosomas y generaciones, entrenamiento A.E. para casos sanos

Por otro lado, el entrenamiento del algoritmo para los pacientes con riesgo tomó 4 horas y 59 minutos y dio paso a 9 generaciones hasta alcanzar en la última una precisión de 72.4 % sobre el conjunto de datos

de prueba, con 1111 cromosomas como se evidencia en la figura 5.12.

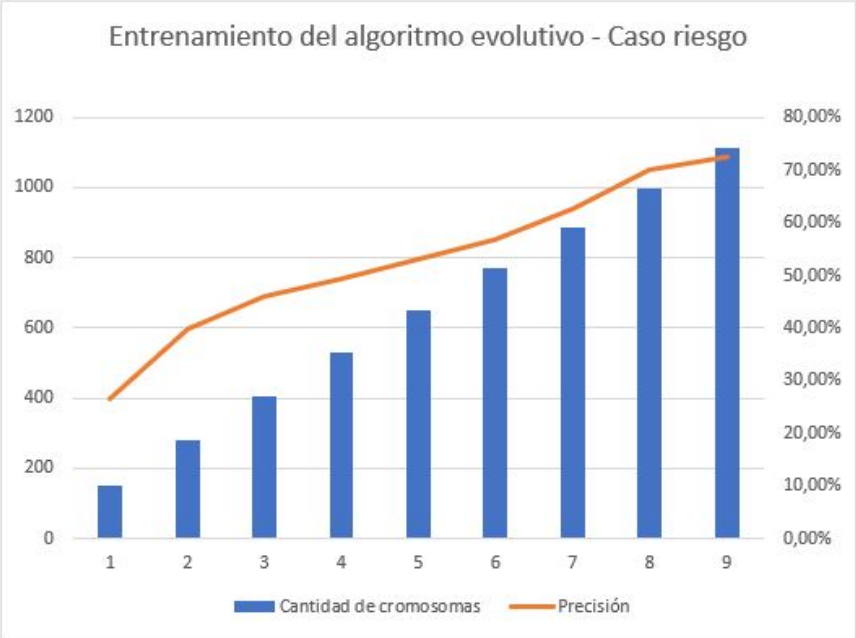

Figura 5.12: Precisión, cantidad de cromosomas y generaciones, entrenamiento A.E. para casos riesgos

El entrenamiento del algoritmo no se detuvo de acuerdo a un tiempo de ejecución o por alcanzar una precisión deseada, sino que al monitorearlo era evidente que la ejecución tomaría demasiado tiempo debido a la complejidad temporal asumida por la reproducción de la creciente población de cromosomas para el diagnóstico de RCV.

Posteriormente, se unieron ambos algoritmos evolutivos haciendo que cada uno otorgue un resultado de cuántos cromosomas de la solución establecen que el paciente tiene un diagnóstico de acuerdo a dicha solución. Para ilustrar esto, se presenta un ejemplo en la figura 5.13, en dicha figura se tienen los datos de un paciente y 4 cromosomas, 2 para diagnóstico sano y 2 para diagnóstico con riesgo. Después de comparar los datos, los del paciente concuerdan (resaltados en verde) con los dos cromosomas de diagnóstico sano y sólo con uno de diagnóstico con riesgo, debido a que la presión sistólica, colesterol y máximo ritmo cardiaco del segundo cromosoma de diagnóstico con riesgo no alcanzan a abarcar los datos del paciente. Es decir, a manera de ejemplo, el genoma tiene una presión de 145 con 10 de desviación, es decir que su rango es [135-155] y el paciente tiene 130 para este atributo, por tanto no se abarca. Teniendo esto en cuenta, se diagnostica el paciente como sano debido a que la mayoría de cromosomas dieron este diagnóstico.

Paciente						
Género	Edad	Presión sistólica	Colesterol	Azúcar en la sangre	Electrocar diograma	Máximo ritmo cardiaco
0	41	130	204	0	0	172

Cromosomas para diagnóstico sano								
N°	Dato	Género	Edad	Presión sistólica	Colesterol	Azúcar en la sangre	Electrocar diograma	Máximo ritmo cardiaco
1	Valor	0	38	130	200	0	0	175
	Desviación	-	3	5	10	-	-	5
2	Valor	0	40	127	203	0	0	180
	Desviación	-	4	4	9	-	-	8

Cromosomas para diagnóstico con riesgo								
N°	Dato	Género	Edad	Presión sistólica	Colesterol	Azúcar en la sangre	Electrocar diograma	Máximo ritmo cardiaco
1	Valor	0	39	130	204	0	0	177
	Desviación	-	3	6	5	-	-	6
2	Valor	0	50	145	250	0	0	190
	Desviación	-	10	10	15	-	-	8

Figura 5.13: Ejemplo del sistema de predicción del AG

Otro aspecto a tener en cuenta es que este valor se normaliza de acuerdo a la cantidad de cromosomas dentro de cada solución y, así, se pueden comparar los resultados; esto quiere decir que se tiene en cuenta que hay 1000 genomas para casos sanos y 1111 para casos con riesgo, así que se comparan los análisis proporcionalmente y no directamente. Por último, si el valor normalizado arrojado por el algoritmo evolutivo que detecta los riesgos es mayor que el del que detecta los sanos, se establece que el paciente está en riesgo cardiovascular, de lo contrario se dice que el paciente no se encuentra en riesgo.

La unión de los dos algoritmos evolutivos bajo la estrategia descrita en el párrafo anterior alcanzó una precisión del 91,7 % para el conjunto de datos de prueba, a pesar de que las soluciones para pacientes con riesgo y sin riesgo tuviesen una precisión del 73,3 % y 76,6 % respectivamente de manera individual.

5.3. Prototipo

El prototipo para ejecutar y probar los algoritmos implementados no cambió mucho desde su diseño inicial como se puede apreciar en la figura 5.14. El único aspecto agregado fue un indicador del archivo cargado, presentando su nombre una vez ha sido seleccionado.

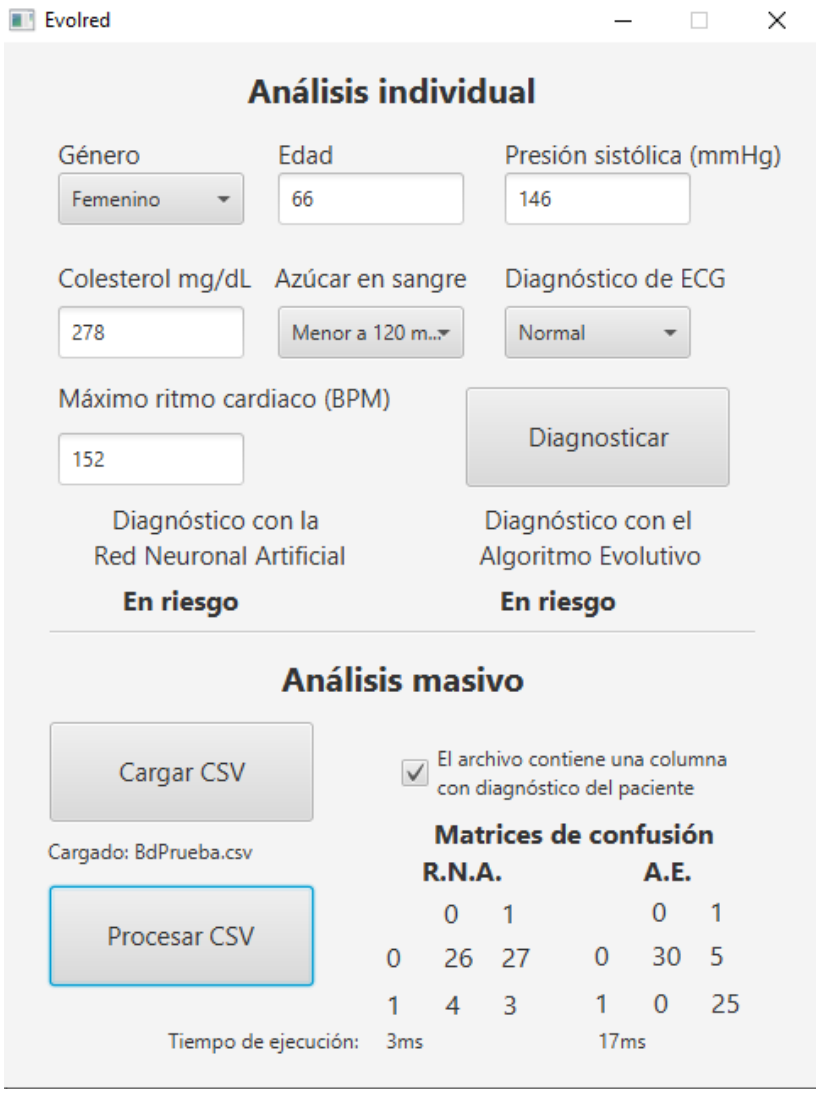

Figura 5.14: Interfaz gráfica del prototipo implementado

En la sección de análisis individual se pueden ingresar los datos continuos con decimales con punto, por otro lado, los datos discretos pueden ser seleccionados a través de listas desplegables cuyas opciones se encuentran en lenguaje natural.

En la sección de análisis masivo, el usuario debe cargar un archivo en formato CSV separado por comas, con la primera línea como encabezado y con los atributos en el siguiente orden: edad, género, presión sistólica, colesterol, azúcar en sangre, ECG, máximo ritmo cardiaco. En caso de haber una columna adicional con el diagnóstico del paciente, se debe activar esta opción a través la casilla de verificación correspondiente. Activar esta opción permitirá que el prototipo presente las matrices de confusión correspondientes para la Red Neuronal Artificial y el Algoritmo Evolutivo. Por último, ya sea que el usuario active o no active la casilla de verificación para la columna de diagnóstico, el prototipo generará un nuevo archivo CSV en la ruta del original, agregando a su nombre la palabra “Diagnosticado”; este archivo tendrá dos columnas adicionales que corresponden al resultado de la predicción de riesgo cardiovascular para cada paciente.

5.4. Características de los datos y la predicción de RCV

Con el objetivo de analizar el dominio del AG y la RNA implementados, se hizo una análisis de la base de datos [5] seleccionada para el entrenamiento (165 registros), para así determinar los rangos y frecuencias de los atributos. Además, se comparó la información obtenida con distintas fuentes para establecer si la base de datos corresponde a una población objetivo para analizar RCV. Teniendo en cuenta las características de los datos, se analizaron los 4 atributos que no son binarios, obteniendo así las tablas de frecuencias de la edad (tabla 5.5), Presión arterial sistólica (tabla 5.6), Colesterol sérico (tabla 5.7); sin embargo, no se obtuvo la tabla de frecuencias del máximo ritmo cardiaco porque su valor ideal varía de acuerdo a la edad del paciente, pero es importante mencionar que en la base de datos de entrenamiento su menor valor es de 71BPM y su mayor valor es de 202BPM. Los rangos presentados en las tablas de distribución de frecuencias están determinados por el estudio de los atributos presentado en las fuentes relacionadas en la tabla 5.8.

X_i	Edades	Frecuencia absoluta (f_i)	Frecuencia acumulada (F_i)	Frecuencia relativa (h_i)	Frecuencia relativa acumulada (H_i)
1	0-35	5	5	0,0303	0,0303
2	36-40	8	13	0,04848	0,07879
3	41-45	35	48	0,21212	0,29091
4	46-50	18	66	0,10909	0,4
5	51-55	37	103	0,22424	0,62424
6	56-60	27	130	0,16364	0,78788
7	61-65	18	148	0,10909	0,89697
8	66-70	12	160	0,07273	0,9697
9	71-75	4	164	0,02424	0,99394
10	76-80	1	165	0,00606	1
	Total	165	165	1	1

Tabla 5.5: Tabla de distribución de frecuencias de la edad en la base de datos de entrenamiento

X_i	Presiones	Frecuencia absoluta (f_i)	Frecuencia acumulada (F_i)	Frecuencia relativa (h_i)	Frecuencia relativa acumulada (H_i)
1	0-90	0	0	0	0
2	91-119	37	37	0,22424	0,22424
3	120-129	40	77	0,24242	0,46667
4	130-139	44	121	0,26667	0,73333
5	140-179	43	164	0,26061	0,99394
6	180-200	1	165	0,00606	1
	Total	165	165	1	1

Tabla 5.6: Tabla de distribución de frecuencias de la presión arterial sistólica en la base de datos de entrenamiento

X_i	Colesterol	Frecuencia absoluta (f_i)	Frecuencia acumulada (F_i)	Frecuencia relativa (h_i)	Frecuencia relativa acumulada (H_i)
1	0-124	0	0	0	0
2	125-200	30	30	0,18182	0,18182
3	201-409	133	163	0,80606	0,98788
4	410-564	2	165	0,01212	1
	Total	165	165	1	1

Tabla 5.7: Tabla de distribución de frecuencias de colesterol sérico en la base de datos de entrenamiento

Atributo	Nombre de la fuente	fuente
Edad	Medigraphic, RCV global	[42]
Presión arterial sistólica	Cómo leer una tabla de presión arterial para determinar RCV	[43]
Colesterol sérico	What is serum cholesterol?	[44]

Tabla 5.8: Fuentes de obtención de los rangos de las tablas de distribución de frecuencias para edad, presión arterial sistólica y colesterol serum.

Teniendo en cuenta las tablas de distribución de frecuencias presentadas, se puede establecer que la base de datos de entrenamiento contiene información de pacientes que pueden pertenecer a una población con RCV, sirviendo como soporte para la utilización de la base de datos para el entrenamiento de los algoritmos de IA propuestos en esta tesis. Además, este estudio permite afirmar que tanto el AG como el como la RNA están en capacidad de analizar pacientes que se encuentren en los siguientes rangos.

- Edades entre los 35 y 80 años.
- Presión arterial sistólica entre 91mmHg y 200mmHg.
- Colesterol sérico entre 125mg/dL y 564mg/dL.
- Máximo ritmo cardiáco entre 71BPM y 202BPM.

Capítulo 6

Pruebas y comparaciones

De acuerdo a los criterios definidos en el capítulo 4, se presenta a continuación la evaluación y comparación del algoritmo evolutivo y la red neuronal.

6.1. Precisión

Para el entrenamiento de ambos algoritmos se utilizó la misma base de datos de entrenamiento, por tanto, se probaron con la misma base de datos de prueba. Esta contenía 60 registros de pacientes y el resultado de cada algoritmo después de analizarla se puede apreciar en las matrices de confusión en la figura 6.1.

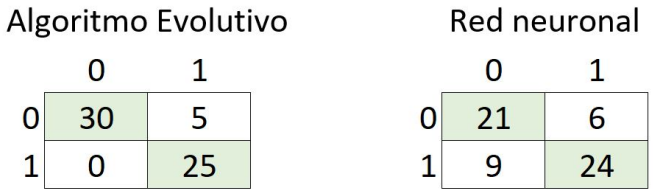

Figura 6.1: Matrices de confusión del AE y la RNA

De acuerdo a estas matrices de confusión, la precisión de los algoritmos es de 76,7 % para el de redes neuronales y de 91,7 % para el algoritmo evolutivo

6.2. Tiempo de entrenamiento

Como se presentó anteriormente en el capítulo 5 y teniendo en cuenta que se corrieron dos versiones del algoritmo evolutivo y hubo una que tardó más que la otra, el tiempo de entrenamiento de este algoritmo fue de 4 horas y 59 minutos. En cambio, el entrenamiento de la red neuronal se ejecutó en distintas iteraciones controladas de manera manual, por lo cual este proceso tardó un total de 49 minutos para hacer la evaluación de las configuraciones de capas propuestas. Por tanto, como se observa en la figura 6.2, se hace evidente que el tiempo de entrenamiento de la red neuronal para esta solución fue mucho menor que el tiempo que tomó entrenar la solución por medio del algoritmo evolutivo.

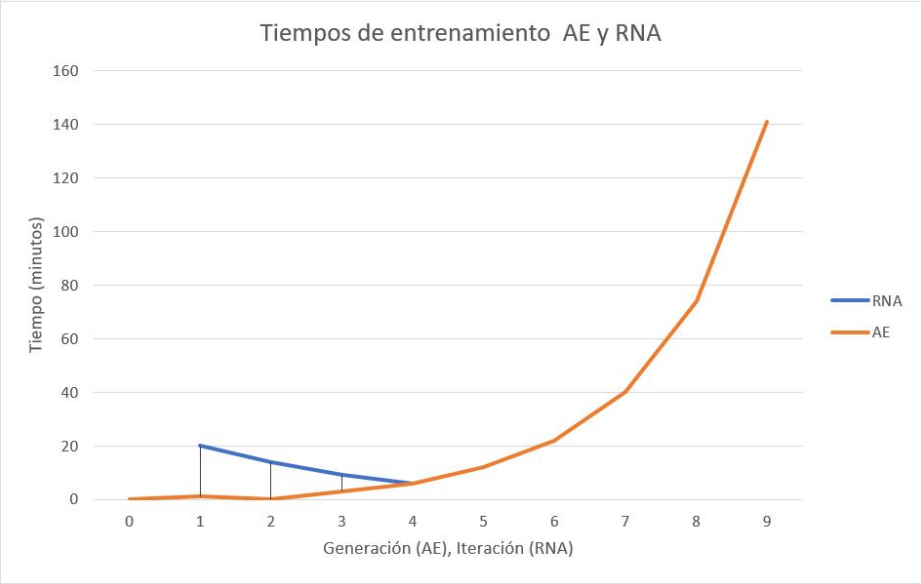

Figura 6.2: Tiempo de entrenamiento de la R.N.A. y el A.E.

6.3. Eficiencia

Con el objetivo de medir el comportamiento ante grandes cantidades de datos, se multiplicarán los registros de las bases de datos, ya que el prototipo no distinguirá que hay casos repetidos y permitirá analizar el tiempo de ejecución de cada algoritmo para distintas cantidades de pacientes. La cantidad de pacientes y el tiempo de ejecución se pueden apreciar en la figura 6.3 y la tabla 6.1.

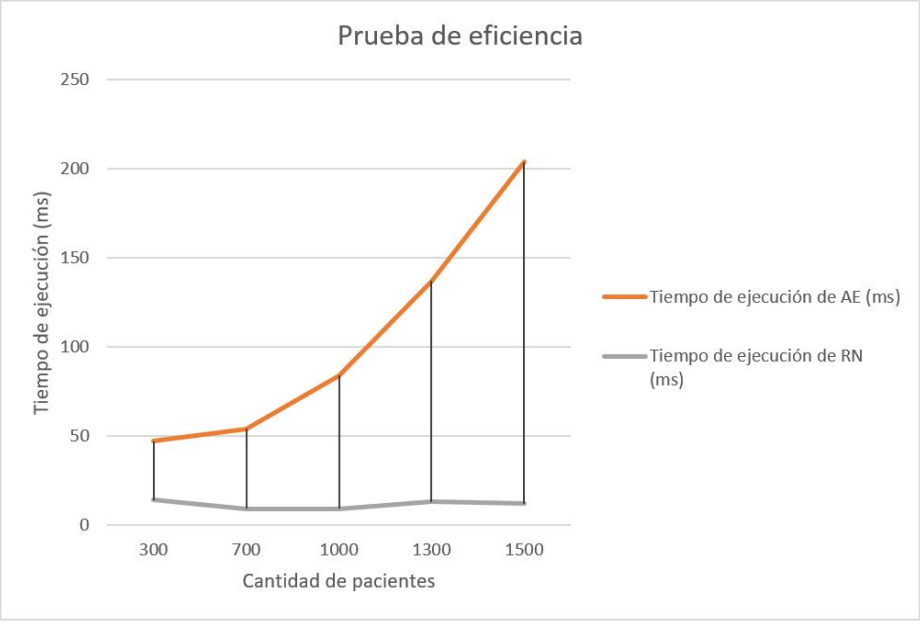

Figura 6.3: Resultado de la prueba de eficiencia de la R.N.A. y el A.E.

Aunque las pruebas fueron ejecutadas sobre un mismo computador (descrito al inicio del capítulo 5), es

Cantidad de pacientes	Tiempo de ejecución de AE (ms)	Tiempo de ejecución de RN (ms)
300	47	14
700	54	9
1000	84	9
1300	137	13
1500	204	12

Tabla 6.1: Tabla de registro de la prueba de eficiencia de la R.N.A. y el A.E.

evidente que el tiempo de ejecución del algoritmo de redes neuronales se mantiene estable a pesar de subir la cantidad de pacientes en el análisis, incluso desciende en algunas ocasiones y no parece verse afectado de manera importante por las grandes cantidades de datos. Por otro lado, el algoritmo evolutivo presenta un tiempo de ejecución cuyo crecimiento aparenta ser exponencial, viéndose directamente afectado por el incremento en la cantidad de pacientes a analizar.

6.4. Capacidad de reentrenamiento y cambios en la precisión:

Como se puede apreciar en la sección 5.1.5, la configuración seleccionada de la red neuronal es su penúltima versión, pues se hizo un reentrenamiento sobre esta y aunque el error general de la red disminuyó, también lo hizo la precisión de esta. Por tanto, aunque es reentrenable, la red neuronal no frece ventajas al hacerlo debido a que ya había llegado a una solución.

Para el caso del algoritmo evolutivo, este da como resultado un listado de cromosomas con las desviaciones correspondientes a cada atributo de los 7 seleccionados. Este listado es diferente a la entrada original, el cuál es un listado de atributos de pacientes sin desviaciones. Por tanto, reentrenar esta solución con los mismos datos y pretender continuar donde se había dejado, requeriría una reestructuración del algoritmo y dejaría de ser la misma solución. En conclusión, el algoritmo evolutivo no se puede reentrenar con los mismos datos.

6.5. Capacidad de reentrenamiento con nuevos datos y cambios en la precisión

Para el reentrenamiento de los nuevos algoritmos con nuevos datos, se seleccionar 20 registros de la base de datos de prueba de manera aleatoria y se agregaron a la base de datos de entrenamiento.

El algoritmo de redes neuronales se reentrenó como se muestra en la tabla 6.2, retomando con los últimos valores utilizados de razón de aprendizaje y momentum para la configuración de capas 7-14-10-8-1. En la primera ejecución se presentó una gran cantidad de oscilaciones y el error empezó en un valor muy alto, debido a esto no se alcanzó el error deseado dentro de las 4000 iteraciones propuestas, además la precisión de la red neuronal disminuyó al 68.0 %.

Se corrió un nuevo entrenamiento para la red neuronal, esta vez con 7000 iteraciones como máximo y se disminuyó el momentum para reducir las oscilaciones del error. Sin embargo, las oscilaciones no se redujeron como se esperaba pero se legó al error deseado, aunque la precisión de la red neuronal se mantuvo igual.

En lo que respecta al algoritmo evolutivo, como se estableció en la sección anterior, no es reentrena-

N	Iteraciones máximas	Error general requerido	Capas	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones	Precisión
1	4000	0,003	7-14-10-8-1	0,2	0,5	Se presentan demasiadas oscilaciones, se deduce que el error se toma como si no hubiese entrenamiento	0,145	4000	0,68
2	7000	0,007	7-14-10-8-1	0,2	0,4	Alcanza el error, aunque hay demasiadas oscilaciones	0,007	3472	0,68

Tabla 6.2: Resultados del reentrenamiento de la red neuronal.

ble, sin embargo, para añadir nuevos datos se podía entrenar una versión nueva de este algoritmo y unirla a la actual y, de esta manera, incluir el entrenamiento para los datos nuevos. Como se puede apreciar en la gráfica, se corrió el entrenamiento para el algoritmo entrenado bajo los casos de riesgo y los casos sanos de manera separada, tomando menos de un minuto en completar su entrenamiento, llegando a la precisión individual deseada del 85 %. A pesar de esto, después de unir la solución nueva a la actual, la precisión no aumentó y esto se evidencia en la matriz de confusión presentada en la figura 6.4.

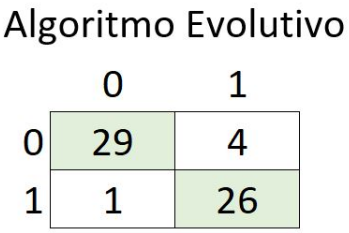

Figura 6.4: Matriz de confusión, reentrenamiento del A.E.

6.6. Comparación de resultados con otro trabajo en el área

Los trabajos que involucran redes neuronales artificiales o algoritmos evolutivos sin la necesidad de combinar estas técnicas de inteligencia artificial o utilizar otras para mejorar los resultados, es algo poco frecuente. Debido a esto, se comparará inicialmente la red neuronal de manera independiente con otro trabajo donde se usa esta técnica de IA y, posteriormente, tanto la RNA como el AE con un trabajo que utiliza RNA y una variación de un AE en forma de PSO con la misma base de datos seleccionada para esta tesis.

6.6.1. Red neuronal

En el trabajo de Chitra [8] se presenta el entrenamiento de una red neuronal artificial para la predicción de enfermedades cardiacas. Se logra una precisión del 85 % con la base de datos de prueba, la cual es la misma que la utilizada en esta tesis. La diferencia principal radica en que en este caso, el

autor utilizó todos los atributos presentes en la base de datos, para un total de 13; esto contrasta con los 7 atributos seleccionados para este trabajo, siendo este el punto débil y la causa de una precisión menor.

A pesar de no contar con más criterios de comparación entre esta tesis y el trabajo en [8], resulta importante resaltar la influencia de la cantidad de atributos para mejorar la precisión de la red neuronal. Por otro lado, las técnicas de IA como las RNA tienes numerosos aspectos para ser evaluadas y los trabajos de investigación suelen centrarse en evaluar sólo la precisión del algoritmo.

6.6.2. Red neuronal y algoritmo evolutivo

En el trabajo de Feshki y Shijani [11] se presenta el uso de un algoritmo de PSO y una red neuronal de realimentación para el diagnóstico de enfermedades cardiacas. En esta propuesta se utilizan todos los 13 atributos de la base de datos de Cleveland [5], la misma utilizada en esta tesis, aunque para esta última sólo se utilizaron 7 atributos. La diferencia principal con este trabajo, radica en que los autores asignaron pesos a los atributos de la base de datos según le asignaban importancia y disponibilidad a cada uno.

Como es usual en este tipo de trabajos, se utiliza la precisión del algoritmo como criterio de su evaluación. De esta manera presentan el algoritmo de RNA con realimentación con una precisión del 89.56 %, superando por 12.89 puntos porcentuales al algoritmo de RNA presentado en esta tesis.

Por último, el algoritmo de PSO no es presentado con su precisión y en su lugar, es combinado con la RNA para lograr una precisión del 91.94 %, superando sólo por 0.27 puntos porcentuales al algoritmo evolutivo presentado en esta tesis.

6.6.3. Comparación general:

Con el objetivo de comparar la precisión con otros algoritmos propuestos en trabajos anteriores, se hizo una investigación adicional de artículos sobre predicción de RCV que usasen la misma base de datos que esta tesis. La tabla 6.3 presenta la precisión de dichos trabajos junto con la de los algoritmos de esta tesis.

Autor	Técnicas	Cantidad de atributos	Precisión
Feshki y Shijani [11]	PSO + RNA	8	91.9 %
Panday y Godara [45]	RNA	13	82.8 %
subbalakshmi,et al [46]	RB	15	81.0 %
Kumari and Godara [47]	SVM	13	84.1 %
El-Rashidy., et al [48]	FC	13	63.2 %
Abdullah y Rajalaxmi [49]	DT	13	63.3 %
Soni [50]	AR	13	81.5 %
Khemphila [51], Boonjiing	RNA	8	89.6 %
Khan [10]	Cluster + RNA	7	93.3 %
Esta tesis	RNA	7	76.7 %
Esta tesis	AE	7	91.7 %

Tabla 6.3: Comparación de precisiones con trabajos anteriores

La tabla 6.3 presenta técnicas no mencionadas anteriormente en esta tesis, debido a que se había centrado la búsqueda en trabajos de PSO, AE y RNA. Sin embargo, para comparar la precisión con otros

trabajos que usasen el mismo dataset, se agregaron las siguientes categorías.

- RB: Redes Bayesianas.
- SVM: Máquinas de soporte vectorial.
- FC: Agrupamiento difuso.
- DT: Árboles de decisión.
- AR: Reglas de asociación.

Es importante resaltar que la red neuronal de esta tesis alcanzó una precisión baja en comparación con la mayoría de trabajos (76.7 %), sin embargo, esta RNA es uno de los trabajos que menos atributos de la base de datos utiliza (7 en total). Por último, el algoritmo evolutivo de esta tesis logró un 91.7 % de precisión, siendo uno de los más altos y el único arriba de 90 %, que sólo utiliza 7 atributos y que no utiliza otra técnica de inteligencia artificial para preprocesar la información y/o mejorar sus resultados.

6.7. Validación cruzada

Con el objetivo de hacer la validación del algoritmo genético a través de la red neuronal seleccionada se ejecutó una validación cruzada con 4 pliegues (4-fold cross validation) diseñando los conjuntos de los pliegues con la siguiente distribución: 3 conjuntos de 76 registros y uno de 75 registros, siendo este último el seleccionado como conjunto de datos de prueba y dando un total de 303 registros.

La selección de registros para los conjuntos de datos se hizo de manera aleatoria, pero teniendo en cuenta que se debía lograr una distribución balanceada entre los registros que diagnostican pacientes en riesgo y los que diagnostican pacientes sin riesgo.

6.7.1. Redes neuronales

Para la validación cruzada del algoritmo se utilizó el framework Neuroph. Sin embargo, se detectó que este framework arroja un error de asignación de variable (*NullPointerException*) en la validación cruzada para el diseño de 4 pliegues, además, era clave que los conjuntos de datos fuesen los mismos tanto para el entrenamiento de la RNA como del AG, por tanto, se decidió entrenar 4 RNA con los 4 pliegues diseñados y así obtener el error de cada red después del entrenamiento.

El entrenamiento de las 4 RNA tomó alrededor de una hora, este tiempo reducido se debe a que se respetó la secuencia de entrenamiento de la red original 7-14-10-8-1, tal como se puede ver en la tabla 6.4. Esta secuencia se siguió a excepción de los casos donde el error general se elevase o se mantuviese estable, puesto que esto implicaría que la RNA entrenada se alejaría de las características de la red original, considerando en estos casos que la configuración RNA-dataset había alcanzado su mejor solución.

Conf.	Iteraciones máximas	Error general requerido	Razón de aprendizaje	Momentum	Anotaciones	Error alcanzado	Iteraciones
Pliegue 1	10000	0,01	0,2	0,7	El error presenta comportamiento estable y en descenso hacia las últimas iteraciones	0,01	7900
Pliegue 1	7000	0,007	0,2	0,5	No se alcanza el error deseado, igualmente se continúa el entrenamiento	0,016	7000
Pliegue 1	5000	0,005	0,2	0,5	El error presenta estabilidad en la última mitad de las iteraciones, además, se elevó su valor	0,02	5000
Pliegue 1	4000	0,003	0,2	0,5	-	-	-
Pliegue 2	10000	0,01	0,2	0,7	El error se presenta inestable, pero durante todo el entrenamiento estuvo en descenso	0,01	5480
Pliegue 2	7000	0,007	0,2	0,5	El error presenta estabilidad durante la mayoría del entrenamiento, además, no se alcanza el error deseado	0,0088	7000
Pliegue 2	5000	0,005	0,2	0,5	-	-	-
Pliegue 2	4000	0,003	0,2	0,5	-	-	-
Pliegue 3	10000	0,01	0,2	0,7	El error se presenta inestable, aunque su comportamiento aparenta que el valor del error puede disminuir	0,01	9320
Pliegue 3	7000	0,007	0,2	0,5	El error presenta estabilidad durante la mayoría del entrenamiento, además, no se alcanza el error deseado	0,0087	7000
Pliegue 3	5000	0,005	0,2	0,5	-	-	-
Pliegue 3	4000	0,003	0,2	0,5	-	-	-
Pliegue 4	10000	0,01	0,2	0,7	El error se presenta inestable, pero se alcanza el valor deseado rápidamente	0,01	3794
Pliegue 4	7000	0,007	0,2	0,5	El error se alcanzó rápidamente y parece que puede disminuir más	0,007	197
Pliegue 4	5000	0,005	0,2	0,5	El error se alcanzó rápidamente y parece que puede disminuir más	0,005	170
Pliegue 4	4000	0,003	0,2	0,5	El error se alcanzó rápidamente, también se presenta alta inestabilidad hacia el final del entrenamiento	0,003	335

Tabla 6.4: Configuración de los 4 pliegues de la validación cruzada de la RNA

Como se puede ver en la tabla 6.4, el pliegue 4 fue la única configuración que consiguió terminar el entrenamiento que siguió la RNA original. Esto afectará negativamente la precisión de las RNA entrenadas en esta validación cruzada. Por último, la tabla 6.4 también presenta los errores alcanzados por cada configuración, dando como promedio un error del 0,010125 o 1,01 %; alejándose considerablemente del error alcanzado por la RNA original: 0.005 o 0,5 %.

6.7.2. Algoritmo genético

Para la validación cruzada del algoritmo genético se utilizó el mismo tipo de entrenamiento que para el AG original, utilizando las 4 configuraciones de datos determinadas para la validación de la RNA. Este entrenamiento para la validación cruzada se corrió sobre el lenguaje de programación Java y, al igual que la solución original, no hace uso de framework alguno. Debido a que la codificación se hace sin la asistencia de un framework, no se tiene acceso al error general de entrenamiento, por tanto, se decidió comparar la validación cruzada del AG con la de la RNA a través de su error en las predicciones.

El entrenamiento de los 4 AG debía hacerse igual que el original, es decir que debían ejecutarse 2 entrenamientos por cada conjunto de datos: un entrenamiento para detectar pacientes sanos y otro para detectar pacientes en riesgo, para así unir posteriormente las soluciones. Este compendio de ejecuciones tomó poco menos de 3 días, notándose la gran complejidad temporal del algoritmo de entrenamiento. Sin embargo, algunas ejecuciones fueron detenidas antes de alcanzar la precisión deseada, puesto que se estimaba que el paso a una siguiente generación podría tomar más de dos días para cada AG. En la tabla 6.5 se puede observar la relación de la configuración AG-dataset, cantidad de generaciones alcanzadas, cantidad de genomas en la última generación y su precisión.

Configuración	Generaciones	Genomas	Precisión
Pliegue 1, pacientes sanos	13	1135	58,7 %
Pliegue 1, pacientes en riesgo	11	1283	73,13 %
Pliegue 2, pacientes sanos	13	1127	68,86 %
Pliegue 2, pacientes en riesgo	10	1188	73,88 %
Pliegue 3, pacientes sanos	3	330	90,32 %
Pliegue 3, pacientes en riesgo	10	1188	75,37 %
Pliegue 4, pacientes sanos	10	904	86,96 %
Pliegue 4, pacientes en riesgo	11	1292	79,85 %

Tabla 6.5: Registro del entrenamiento de la validación cruzada del AG

Como se puede ver en la tabla 6.5, el pliegue 3 para pacientes sanos fue la configuración que consiguió una mayor precisión, seguida del pliegue 4 para pacientes sanos, presentando así una diferencia gradual con los otros pliegues. Esto afectará negativamente a los AG entrenados en esta validación cruzada, puesto que disminuirá la precisión final de cada uno de las 4 soluciones producto de este proceso.

6.7.3. Análisis de resultados

Después de haber ejecutado el entrenamiento del AG y la red neuronal con los conjuntos de datos para la validación cruzada, se obtuvieron sus matrices de confusión como se puede ver en la figura 6.5 y partir de estas se calculó su precisión y error como se puede ver en la tabla 6.6.

RNA-1

	0	1
0	28	15
1	13	19

AG-1

	0	1
0	32	18
1	9	16

RNA-2

	0	1
0	28	11
1	14	22

AG-2

	0	1
0	35	19
1	7	14

RNA-3

	0	1
0	23	11
1	19	22

AG-3

	0	1
0	31	18
1	11	15

RNA-4

	0	1
0	11	27
1	2	35

AG-4

	0	1
0	10	19
1	3	43

Figura 6.5: Matrices de confusión, Validación cruzada del AG y la RNA

IA	Precisión (%)	Error (%)
RNA-1	62,67	37,33
RNA-2	66,67	33,33
RNA-3	60	40
RNA-4	61,33	38,67
Promedio RNA	62,67	37,33
AG-1	64	36
AG-2	65,33	34,67
AG-3	61,33	38,67
AG-4	70,67	29,33
Promedio AG	65,33	34,67

Tabla 6.6: Precisión y error del AG y la RNA en la validación cruzada

De acuerdo a esto, se obtiene que el error promedio de la RNA es del 37,33 % y el error promedio del AG es del 34,67 %; suponiendo una diferencia del 2,66 %. Teniendo en cuenta que el error general del entrenamiento de la RNA es del 1,01 %, el AG presenta unos resultados de entrenamiento confiables.

De acuerdo a esto, el valor de error presentado por el AG en su validación cruzada es satisfactorio para

asegurar su confiabilidad, puesto que la RNA presenta un error de predicción similar y su error de entrenamiento la hace confiable, verificando así por comparación el AG. Teniendo en cuenta este comportamiento del AG y los trabajos consultados en el marco teórico de esta tesis, se concluye que los AG pueden conseguir soluciones de alta precisión aunque con una mayor necesidad de recursos computacionales durante el entrenamiento en comparación con las RNA.

Capítulo 7

Conclusiones

7.1. Conclusiones

Después de la investigación respecto al uso de los algoritmos evolutivos para la predicción de riesgos cardiovasculares, se puede inferir que existe una tendencia a usarlos como técnicas de optimización o preprocesamiento de la información para después analizarla usando redes neuronales. Esto, aunque trae buenos resultados, minimiza en cierta medida el potencial de los algoritmos evolutivos pues, como se presentó en este trabajo, por sí mismo un A.E. puede generar buenos resultados respecto a la precisión para predecir RCV.

Otro aspecto que se detecta en las investigaciones de manera general, es la tendencia a evaluar las técnicas de predicción sólo respecto a su precisión, dejando de lado aspectos como eficiencia, tiempo de entrenamiento y capacidad de reentrenamiento. Es de resaltar que una adecuada documentación de los aspectos analizados en los criterios de comparación de este trabajo, podría traer nuevos temas de discusión entre expertos en el área de algoritmos para predicción de RCV como la capacidad de reentrenamiento y tiempo de entrenamiento de sus soluciones, además esto podría llevar a vislumbrar detalles que podrían encaminar al perfeccionamiento de estos algoritmos, incluso para la predicción enfermedades distintas a RCV.

Un aspecto no muy mencionado pero que se pudo percibir durante la revisión bibliográfica es la tendencia de los investigadores a usar las redes neuronales como un sistema de caja negra, pues establecen un diseño con una cantidad de capas y un modelo matemático que terminan ignorando al dejar todo el trabajo a los frameworks. Así que se presenta la recomendación de un diseño iterativo de las RNA como el presentado en esta tesis, el cual permite justificar el número de capas y cantidad neuronas de cada capa para un perceptrón, además, si el diseño iterativo se acompaña con una correcta documentación y justificación, se podría facilitar identificar el camino correcto para una solución o entender, dado el caso, por qué el entrenamiento no funciona como se espera.

Respecto a los algoritmos evolutivos, la revisión bibliográfica permitió encontrar tendencias por parte de los autores a utilizar técnicas desarrolladas en otras investigaciones. Esto, aunque es uno de los principales pilares de la investigación, deja de lado las técnicas básicas evolutivas, como la diseñada a partir de la teoría evolutiva de Darwin. Uno de los casos más frecuentes son los algoritmos de PSO, que son utilizados en varias investigaciones y con buenos resultados, pero se pierde el potencial y alto grado de control que otorga el diseño estructural de una solución evolutiva más simple como el diseño basado en: población, reproducción, mutación y competencia.

Debido a lo descrito en el párrafo anterior, se optó por la implementación de un algoritmo evolutivo desde sus principios básicos y esto resultó en un código corto y de caja blanca. Por otro lado, un aspecto destacable es cómo se entrenó el algoritmo, puesto que se dividió el problema en dos partes (sanos y con riesgo), permitiendo dos entrenamientos con ejecuciones al tiempo que después de ser unidos, alcanzaron una precisión alta en comparación con trabajos similares que hacen uso de un mayor número de atributos para la predicción de RCV.

Al momento de comparar los resultados sobre los conjuntos de datos de prueba, resultó evidente cómo el algoritmo evolutivo presentaba una mayor precisión, pues alcanzó 91.7 % en comparación el 76.7 % alcanzado por la red neuronal. Sin embargo, el AE también presentaba un rendimiento menor en cuestiones de eficiencia y tiempo de entrenamiento pues el tiempo de entrenamiento del A.E. fue de 4 horas y 59 minutos, muy elevado en comparación con la R.N.A. que se entrenó en 49 minutos; así mismo, a la hora de diagnosticar grandes cantidades de pacientes, el tiempo de ejecución del A.E. presenta un crecimiento similar al exponencial a medida que aumenta la cantidad de registros por analizar, contrastando totalmente con la R.N.A. cuya eficiencia no se vio afectada significativamente al aumentar la cantidad de pacientes desde 300 a 1500. Esto puede ser un aspecto de bastante peso a la hora de seleccionar entre usar un algoritmo evolutivo o una red neuronal cuando se trate de un problema que requiera el procesamiento de grandes cantidades de datos.

Respecto a la capacidad de reentrenamiento de los algoritmos es notorio el hecho de que aunque se pueden continuar los intentos de reentrenar la red neuronal de esta tesis, esta puede disminuir su error general y sin embargo, no aumentar o incluso disminuir su precisión. Este aspecto también se presentó cuando se utilizaron nuevos datos para el reentrenamiento, lo que sugiere que reentrenar una red neuronal que ha llegado a su solución podría no ser la mejor solución para incrementar la precisión de esta, esto se convierte en un tema para discusiones en trabajos futuros.

Respecto al algoritmo evolutivo, su reentrenamiento con los datos originales es posible en tanto sea diseñado para ser reentrenado. Esto se afirma ya que el reentrenamiento fue posible cuando se introdujeron nuevos datos, estableciendo que podía hacerse un entrenamiento para estos registros nuevos y agregar el resultado al conjunto de cromosomas solución producto del entrenamiento inicial; esta acción mantuvo estable la precisión, lo cual indica que los nuevos datos fueron recibidos adecuadamente dentro del conjunto de cromosomas y ayudaron a actualizar los casos detectables por el algoritmo, pues de ser el caso contrario, la precisión se hubiese visto afectada negativamente.

Al momento de comparar los resultados de esta tesis con otro trabajo del área, el enfoque fue principalmente sobre los aspectos de precisión de los algoritmos y permitió establecer que el algoritmo evolutivo alcanzó un porcentaje de precisión alto y es comparable con trabajos previos respecto a la predicción de riesgos cardiovasculares. Sin embargo, esta comparación remarcó el hecho de que la predicción de enfermedades suele hacerse a través de la combinación de técnicas de inteligencia artificial, que aunque no era el objetivo de este trabajo, podría ser el indicio de trabajos futuros con A.E. y R.N.A. comparables en distintos criterios con otras investigaciones.

También es relevante mencionar que la propuesta inicial de verificar el algoritmo evolutivo a través de la comparación con una red neuronal artificial resultó satisfactoria, puesto que más que conseguir una mayor precisión, permitió establecer criterios de comparación útiles entre estas dos técnicas. Además, una revisión adicional de trabajos con técnicas de I.A. diferentes a las utilizadas en este trabajo, permitió identificar que el A.E. alcanzó una precisión elevada al estar arriba del 90 %, lo cual es poco usual ya que

los algoritmos propuestos para la misma base de datos y para la predicción de RCV suelen alcanzar este porcentaje sólo cuando son asistidos con otras técnicas de I.A.

Por último, la validación cruzada permite la comparación de las técnicas de IA al presentar un error promedio similar (diferencia de 2,66 %) entre ambos algoritmos, aunque este error calculado corresponde a la precisión de los algoritmos y no al error de entrenamiento; por lo cual y con base tanto en la precisión del AG como de la RNA, se establece que el AG es tan confiable como la RNA. Sin embargo, resulta importante mencionar que el costo computacional de entrenar el AG es mucho mayor que el de entrenar la RNA, puesto que esta última requirió 49 minutos para su entrenamiento y el AG requirió 4 horas y 59 minutos.

Capítulo 8

Trabajos futuros

8.1. Trabajos futuros

Durante la ejecución de esta investigación hubo ciertos aspectos que se consideraron como posibles mejoras, pero que a su vez salían de los objetivos propuestos, por tanto, estos se presentan a continuación como trabajos futuros:

1. **Reentrenamiento de la red neuronal:** este fue un aspecto que no logró alcanzarse completamente dentro de la tesis, esto debido a que se usó la misma red neuronal para reentrenarse agregando nuevos registros de pacientes. Por tanto, se sugiere como trabajo posterior, el diseño de un sistema de predicción de riesgos cardiovasculares a través de múltiples redes neuronales, donde se utilicé la predicción de estas como conjunto para otorgar un análisis del riesgo cardiovascular de un paciente. Este trabajo permitiría crear una red neuronal por cada base de datos de entrenamiento nueva, sin necesidad de afectar a las RNA que ya han sido entrenadas y sin correr el riesgo de disminuir su precisión.

2. **Reentrenamiento del algoritmo evolutivo:** aunque en esta investigación este aspecto dio buenos resultados al unir una nueva solución de los nuevos datos a la solución antigua, se propone profundizar en este tipo de soluciones para definir si es adecuado el reentrenamiento del algoritmo evolutivo por medio de esta estrategia. Puesto que en caso de ser una opción viable para los trabajos con computación evolutiva en general, representaría la disminución de costos temporales a la hora de crear soluciones de este tipo para la predicción de RCV en bases de datos clínica que estén en constante crecimiento.

3. **Aumento de la precisión de la predicción de RCV por unión de técnicas:** la precisión de la predicción suele verse incrementada con la combinación de técnicas de inteligencia artificial, tal y como se evidencia en los trabajos consultados y consignados en el marco teórico de esta tesis. Por tanto, se sugiere como trabajo futuro tomar las dos técnicas implementadas en esta investigación y buscar la forma de unirlas para que se complementen a la hora de otorgar una predicción, de forma que un algoritmo cubra los casos que el otro no puede. Es importante aclarar que no se está sugiriendo dividir el proceso de análisis de la información en partes como preprocesado y procesado, sino que se habla de la posibilidad de unir ambas soluciones en la misma sección, la cual es todo el proceso de análisis de la información clínica para predecir RCV.

4. **Aumento de la precisión de la predicción de RCV por adición de atributos:** como se presenta en el marco teórico, existen trabajos de RNA que alcanzan precisiones que superan el 90 % utilizando el mismo conjunto de datos utilizado en esta tesis, sin embargo, estas hacen uso de todos

los atributos proporcionados por la base de datos. Por esta razón y de acuerdo a la opinión de los expertos presentada en la sección 4.1.1, se propone como trabajo futuro aumentar la precisión del AE y la RNA a través la inclusión del procesamiento de los siguientes atributos modificables y no modificables:

- Perímetro abdominal.
- Raza.
- Índice de masa corporal (IMC).
- Relación entre la presión sistólica y diastólica.
- Prueba de microalbuminuria.
- Creatinina en suero.
- Antecedentes cardiovasculares familiares.
- Condición de fumador.
- Condición de dieta hipercalórica.
- Ejercicio semanal.

5. **Disminución del tiempo de entrenamiento del A.E:** una vez entrenado el algoritmo evolutivo, fue notorio cómo su tiempo de entrenamiento fue mucho mayor al de la red neuronal artificial. Debido a esto se sugiere el uso de una o más técnicas de las siguientes para disminuir la complejidad temporal de la ejecución del entrenamiento de este A.E.

   - Dividir la base de datos de entrenamiento en pequeños conjuntos de información, luego, entrenar un A.E. por cada uno de estos conjuntos y posteriormente unir y evaluar las soluciones en una sola.
   - Llevar el entrenamiento del algoritmo evolutivo a un sistema distribuido que permita ejecutar el entrenamiento del algoritmo en diferentes equipos y posteriormente unir la solución.
   - Después de dividir la base de datos de entrenamiento en pequeños conjuntos de información, ejecutar el entrenamiento de los A.E. de manera paralela usando sistemas hardware como FPGA para conseguir una ejecución más rápida y hallar la solución de manera más eficiente.

6. **Analizar otra base de datos:** el AE y la RNA presentadas en esta tesis fueron entrenados con un solo conjunto de datos, por tanto, se convierte en un trabajo futuro ejecutar estos mismos algoritmos sobre otra base de datos. Esto significa que se debe conseguir una base de datos con los mismos atributos con los que se entrenaron los algoritmos o construir una a través del proyecto PIPEC-UV, teniendo en cuenta que los algoritmos fueron diseñados para integrarse con dicho proyecto de investigación.

Capítulo 9

Bibliografía

[1] M. S. R. Nalluri, K. Kannan, M. Manisha, and D. S. Roy, "Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization," *Journal of Healthcare Engineering*, vol. 2017, 2017.

[2] M. Hassoon, M. S. Kouhi, M. Zomorodi-Moghadam, and M. Abdar, "Using PSO Algorithm for Producing Best Rules in Diagnosis of Heart Disease," *2017 International Conference on Computer and Applications, ICCA 2017*, pp. 306–311, 2017.

[3] O. mundial de la salud OMS, "Nota descriptiva, centro de prensa: enfermedades cardiovasculares. organización mundial de la salud oms," 2015.

[4] O. mundial de la salud OMS, "Nota descriptiva, centro de prensa: Las 10 principales causas de defunción.," 2017.

[5] Z. S. W. S. M. U. H. B. S. M. P. M. V. M. C. L. B. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D. University Hospital and P. Cleveland Clinic Foundation: Robert Detrano, M.D., "Heart disease uci," 2018.

[6] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, "An evolutionary algorithm for heart disease prediction," *Communications in Computer and Information Science*, vol. 292 CCIS, pp. 378–389, 2012.

[7] S. Rajamhoana, C. A. Devi, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, "Analysis of neural networks based heart disease prediction system," *International Conference on Human System Interaction (HSI)*, vol. 11, 2018.

[8] C. R, "Heart Disease Prediction System Using Supervised Learning Classifier," *Bonfring International Journal of Software Engineering and Soft Computing*, vol. 3, no. 1, pp. 01–07, 2013.

[9] A. Kaur, "Heart Disease Prediction Using Data Mining Techniques: a Survey," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, pp. 569–572, 2018.

[10] M. A. Khan, "An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier," *IEEE Access*, vol. 8, pp. 34717–34727, 2020.

[11] M. G. Feshki and O. S. Shijani, "Improving the heart disease diagnosis by evolutionary algorithm of PSO and Feed Forward Neural Network," *2016 Artificial Intelligence and Robotics, IRANOPEN 2016*, pp. 48–53, 2016.

[12] S. Radhimeenakshi and G. M. Nasira, "Remote Heart Risk Monitoring System based on Efficient Neural Network and Evolutionary Algorithm," *Indian Journal of Science and Technology*, vol. 8, no. 14, 2015.

[13] M. Sudha, "Evolutionary and Neural Computing Based Decision Support System for Disease Diagnosis from Clinical Data Sets in Medical Practice," *Journal of Medical Systems*, vol. 41, no. 11, 2017.

[14] C. Brester, J. Kauhanen, T.-P. Tuomainen, E. Semenkin, and M. Kolehmainen, "Comparison of two-criterion evolutionary filtering techniques in cardiovascular predictive modelling.," in *ICINCO (1)*, pp. 140–145, 2016.

[15] H. Yan, J. Zheng, Y. Jiang, C. Peng, and S. Xiao, "Selecting critical clinical features for heart diseases diagnosis with a real-coded genetic algorithm," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 1105–1111, 2008.

[16] O. A.-i. Trade-off, M. B. Gorzałczany, and F. Rudzi, "Heart-Disease Diagnosis Decision Support Employing Fuzzy Systems with Genetically," 2017.

[17] P. Pławiak, "Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system," *Expert Systems with Applications*, vol. 92, pp. 334–349, 2018.

[18] T. Vivekanandan and N. C. Sriman Narayana Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Computers in Biology and Medicine*, vol. 90, no. April, pp. 125–136, 2017.

[19] S. Barnes, S. Saria, and S. Levin, "An Evolutionary Computation Approach for Optimizing Multilevel Data to Predict Patient Outcomes," *Journal of Healthcare Engineering*, vol. 2018, pp. 1–10, 2018.

[20] A. Ratnaparkhi and R. Ghongade, "A frame work for analysis and optimization of multiclass ECG classifier based on Rough set theory," *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, pp. 2740–2744, 2014.

[21] M. Šprogar, M. Šprogar, and M. Colnarič, "Autonomous evolutionary algorithm in medical data analysis," *Computer Methods and Programs in Biomedicine*, vol. 80, no. SUPPL. 1, 2005.

[22] C. Suvarna, A. Sali, and S. Salmani, "Efficient heart disease prediction system using optimization technique," *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 374–379, 2017.

[23] N. C. Long, P. Meesad, and H. Unger, "A highly accurate firefly based algorithm for heart disease prediction," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8221–8231, 2015.

[24] S. Singhal, H. Kumar, V. Passricha, and C. Science, "Prediction of Heart Disease using CNN," pp. 257–261, 2018.

[25] K. Uyar and A. Ilhan, "Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks," *Procedia Computer Science*, vol. 120, pp. 588–593, 2017.

[26] C. Salud, C. multimedia, Q. cardiovascular, and Q. cardiovascular, "Qué es el riesgo cardiovascular," 2019.

[27] Cubillos, "Evaluación de tecnologías en salud: aplicaciones y recomendaciones en el sistema de seguridad social en salud colombiano. programa de apoyo a la reforma de salud," 2006.

[28] A. A. B. C. Piepoli MF, Hoes AW, *1º European Guidelines on cardiovascular disease prevention in clinical practice: The Sixth Joint Task Force of the European Society of Cardiology and Other Societies on Cardiovascular Disease Prevention in Clinical Practice (constituted by representatives of 10 societies and by invited experts): Developed with the special contribution of the European Association for Cardiovascular Prevention Rehabilitation (EACPR).* 2016.

[29] C. Guijarro, H. Coordinador, J. Luis, and M. González, *1º Conferencia de Prevención y Promoción de la Salud en la Práctica Clínica en España.* 2007.

[30] ecocardio, "Segmento st y onda t," 2020.

[31] M. Van Gerven and S. Bohte, "Artificial neural networks as models of neural information processing," *Frontiers in Computational Neuroscience*, vol. 11, p. 114, 2017.

[32] T. Back, *Handbook of evolutionary computation.* Inst. of Physics Publ. [u.a.], 2002.

[33] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[34] Sourceforge, "Neuroph 2.96," 2020.

[35] R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem," in *Proceedings of the international conference on Neural Networks*, vol. 3, pp. 11–14, IEEE Press New York, 1987.

[36] P. Melillo, R. Izzo, A. Orrico, P. Scala, M. Attanasio, M. Mirra, N. De Luca, and L. Pecchia, "Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis," *PLOS ONE*, vol. 10, pp. 1–14, 03 2015.

[37] J. Sreekanth and B. Datta, "Comparative Evaluation of Genetic Programming and Neural Network as Potential Surrogate Models for Coastal Aquifer Management," *Water Resources Management*, vol. 25, no. 13, pp. 3201–3218, 2011.

[38] B. Can and C. Heavey, "A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models," *Computers and Operations Research*, vol. 39, no. 2, pp. 424–436, 2012.

[39] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 17–26, 2001.

[40] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251 – 257, 1991.

[41] K. Swingler, *Applying Neural Networks: A Practical Guide.* Academic Press, 1996.

[42] C. Batista, G. Izquierdo, E. Ledesma, R. Pérez, and R. Garcí, "Epidemiología de los factores de riesgo cardiovascular y riesgo cardiovascular global en personas de 40 a 79 años en atención primaria," 2020.

[43] A. McDermott, "Tabla de hipertensión: Cómo comprender tu presión arterial," 2018.

[44] J. Huizen, "Serum cholesterol: What to know and how to manage levels," 2018.

[45] S. P. P. N. Godara, "Decision support system for cardiovascular heart disease diagnosis using improved multilayer perceptron," *International Journal of Computer Applications*, vol. 975, p. 8887.

[46] G. Subbalakshmi, K. Ramesh, and M. C. Rao, "Decision support in heart disease prediction system using naive bayes," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 2, pp. 170–176, 2011.

[47] M. Kumari and S. Godara, "Comparative study of data mining classification methods in cardiovascular disease prediction 1," 2011.

[48] M. A. El-Rashidy, T. E. Taha, N. M. Ayad, and H. S. Sroor, "An intelligent model for automated heart disease diagnosis," *Journal of Computer Science and Engineering*, vol. 4, no. 2, pp. 1–10, 2010.

[49] A. S. Abdullah and R. Rajalaxmi, "A data mining model for predicting the coronary heart disease using random forest classifier," in *International Conference in Recent Trends in Computational Methods, Communication and Controls*, pp. 22–25, 2012.

[50] J. Soni, U. Ansari, D. Sharma, and S. Soni, "Intelligent and effective heart disease prediction system using weighted associative classifiers," *International Journal on Computer Science and Engineering*, vol. 3, no. 6, pp. 2385–2392, 2011.

[51] A. Khemphila and V. Boonjing, "Heart disease classification using neural network and feature selection," in *2011 21st International Conference on Systems Engineering*, pp. 406–409, IEEE, 2011.

Anexos

Debido a la extensión de los anexos, estos se encuentran de manera digital en el repositorio del proyecto, ubicado en: `https://github.com/migmaxor/TesisMaestria`

A continuación se hace una descripción de lo que se encuentra en los anexos:

- Ejecutable (No recomendado): Contiene los archivos compilados del prototipo que permite cargar CSV para la predicción de Riesgos Cardiovasculares (RCV), no se recomienda ya que la versión actual de java bloquea la aplicación por no tener un certificado emitido por una autoridad de confianza.
- Entrenamiento AG:
    1. AgCeros: Contiene el código del algoritmo genético utilizado para entrenarse con los casos de pacientes sanos, así mismo, contiene los resultados del entrenamiento.
    2. AgUnos: Contiene el código del algoritmo genético utilizado para entrenarse con los casos de pacientes con RCV, así mismo, contiene los resultados del entrenamiento.
- PrototipoEvolRed: Contiene el proyecto de Netbeans con la GUI hecha en JavaFx para cargar y ejecutar el prototipo para la predicción de RCV con el AG y la RNA. Dentro del SRC del proyecto, se encuentra una carpeta llamada "datasets", donde podrá encontrar los datasets de prueba y entrenamiento de los algoritmos para poder cargarlos en el prototipo.
- Red Neuronal:
    1. Ejecución sin GUI: contiene un proyecto de Netbeans que ejecuta la red neuronal con el dataset de prueba sin necesidad de una GUI.
    2. Errores generales del entrenamiento: Contiene todas las imágenes de los errores de entrenamiento de las diferentes RNA candidatas.
    3. Proyecto de Neuroph: Contiene el proyecto de Neuroph con la RNA seleccionada.
    4. Red iteración adicional: Contiene la red neuronal producto de la iteración adicional.
    5. Redes primera iteración: Contiene las redes neuronales producto de la primera iteración.
    6. Redes segunda iteración: Contiene las redes neuronales producto de la segunda iteración.
    7. Redes tercera iteración: Contiene las redes neuronales producto de la tercera iteración.
- Validación Cruzada:
    1. Código fuente AG: contiene el código fuente del AG para las 8 ejecuciones.

2. Error general RNA Entrenamiento: contiene los errores al entrenar la RNA en las 4 configuraciones de dataset determinadas. Están nombrado: It¡iteración¿-¡ConfiguraciónDataset¿
3. Resultados entrenamiento: contiene los datos de entrenamiento de los 4 datasets, las 4 soluciones del AG resultantes y las 4 RNA resultantes.
4. MatricesConfusionCruzada.JPG: una imagen de las matrices de confusión resultantes de la validación cruzada.

Además, se incluye un archivo comprimido llamado heart-disease-uci.zip, el cual contiene el dataset utilizado para entrenar y probar el AG y la RNA, el cual es tomado de [5].