

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA

Facultad de Ingeniería

Escuela Académico Profesional de Ingeniería en Informática y Sistemas

**“DESARROLLO DE UN SOFTWARE UTILIZANDO REDES NEURONALES
ARTIFICIALES PARA LA SIMULACIÓN DEL CICLO HIDROLÓGICO
DE LA LAGUNA ARICOTA.”**

TESIS

Presentada por:

Bach. Walter Steven Yanarico Espinoza

**Para optar el Título Profesional de
INGENIERO EN INFORMÁTICA Y SISTEMAS**

**TACNA – PERÚ
2012**

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA

FACULTAD DE INGENIERÍA

JURADO CALIFICADOR Y CALIFICACIÓN DE LA SUSTENTCIÓN DE TESIS

TESIS N° _____

TITULO PROFESIONAL DE:

Ingeniero en Informática y sistemas

La Secretaría Académico de la Facultad de Ingeniería, por resolución de Facultad N°0848-2012-FAIN/UNJBG, designó Jurado para la sustentación oral de la Tesis titulada: "DESARROLLO DE UN SOFTWARE UTILIZANDO REDES NEURONALES ARTIFICIALES PARA LA SIMULACIÓN DEL CICLO HIDROLÓGICO DE LA LAGUNA ARICOTA"

El mismo que está conformado por

Presidente: MSc. Juan Ubaldo Jiménez Castilla

Secretario: MSc. Iván Vladimir Pino Tellería


Vocal: MSc. Edgar Aurelio Taya Acosta

Para calificar la sustentación de la Tesis en acto público el día 20 de Setiembre del 2012.

Presentado por el Bachiller Walter Steven Yanarico Espinoza, de la Escuela Académico Profesional de Ingeniería en Informática y Sistemas.

El Jurado Calificador en forma secreta e individual emitió su opinión sobre el tema de la tesis expuesta y procedió a obtener el promedio que arrojó el calificativo de aprobado con la nota de Quince (15).

Para ratificar lo detallado firman:



MSc. Juan Ubaldo Jiménez Castilla
Presidente



MSc. Iván Vladimir Pino Tellería
Secretario



MSc. Edgar Aurelio Taya Acosta
Vocal

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA


FACULTAD DE INGENIERÍA

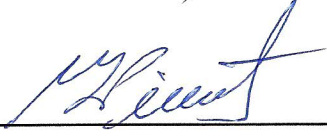
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS


“DESARROLLO DE UN SOFTWARE UTILIZANDO REDES NEURONALES
ARTIFICIALES PARA LA SIMULACIÓN DEL CICLO HIDROLOGICO DE LA LAGUNA
ARICOTA”

TESIS SUSTENTADA Y APROBADA EL 20 DE SETIEMBRE DEL 2012

ESTANDO EL JURADO CALIFICADOR INTEGRADO POR:

Presidente : 
MSc. Juan Ubaldo Jiménez Castilla
Presidente

Secretario : 
MSc. Iván Vladimir Pino Tellería
Secretario

Vocal : 
MSc. Edgar Aurelio Taya Acosta
Vocal

Asesor : 
MSc. Edwin Hinojosa Ramos

DEDICATORIA

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres, Luisa Elena Espinoza Rojas y Segundo Walter Yanarico López.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por los ejemplos de perseverancia y constancia que los caracterizan y que me han infundado siempre, por el valor mostrado para salir adelante por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

Al amor de mi vida, Alexia Saribeth Rodríguez Contreras.

Por haber aparecido en mi vida, brindándome no solo su amor, sino también por el apoyo incondicional y la confianza que me da para alcanzar cada uno de mis objetivos planteados en esta vida, y por estar siempre a mi lado en todo momento.

A mis amigos.

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora, seguimos siendo los mejores amigos: Wilberth Tesillo, Abdóm Fora.

Finalmente a los maestros, aquellos que marcaron cada etapa de nuestro camino universitario, y que me ayudaron en asesorías y dudas presentadas en la elaboración de la tesis.

CONTENIDO

RESUMEN	xv
INTRODUCCIÓN	xvi
I. PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1. DESCRIPCIÓN DEL PROBLEMA	1
1.1.1. ANTECEDENTES DEL PROBLEMA.	1
1.1.2. PROBLEMÁTICA DE LA INVESTIGACIÓN.	2
1.2. FORMULACIÓN DEL PROBLEMA	3
1.2.1. PROBLEMA GENERAL	3
1.2.2. PROBLEMAS ESPECÍFICOS	3
1.3. JUSTIFICACIÓN	3
1.4. ALCANCES Y LIMITACIONES	4
1.5. OBJETIVOS	5
1.5.1. OBJETIVO GENERAL.	5
1.5.2. OBJETIVOS ESPECÍFICOS	5
1.6. HIPÓTESIS	6
1.6.1. HIPÓTESIS GLOBAL.	6
1.6.2. SUB – HIPÓTESIS.	6

1.7. VARIABLES	7
1.7.1. IDENTIFICACIÓN DE VARIABLES.	7
1.7.2. DEFINICIÓN DE VARIABLES.	7
1.7.3. OPERACIONALIZACIÓN DE LAS VARIABLES.	8
1.8. DISEÑO DE LA INVESTIGACIÓN	9
1.8.1. DISEÑO EXPERIMENTAL.	9
1.8.2. POBLACIÓN.	10
1.8.3. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCIÓN DE DATOS.	10
1.8.4. ANÁLISIS DE DATOS Y SELECCIÓN DE PRUEBAS ESTADÍSTICAS.	10
II. MARCO TEÓRICO	11
2.1. MARCO REFERENCIAL	11
2.1.1. HIDROLOGÍA.	11
2.1.2. METODOLOGÍA SISTÉMICA.	11
2.2. BASES TEÓRICAS RESPECTO AL PROBLEMA.	12
2.2.1. CICLO HIDROLÓGICO.	12
2.2.2. SIMULACIÓN DIGITAL.	14
2.2.3. MODELO.	14
2.2.4. DISTRIBUCIONES DE PROBABILIDAD.	15
2.2.5. DETERMINACIÓN DEL TIPO DE DISTRIBUCIÓN.	16
2.2.5.1. Pruebas de Bondad y Ajuste X^2 .	17
2.2.6. REDES NEURONALES ARTIFICIALES (RNA).	18

2.2.6.1.	Modelo de neurona artificial.	18
2.2.6.2.	Aprendizaje o Entrenamiento de una Red Neuronal.	20
2.2.6.3.	Perceptrón multicapa.	22
2.2.6.4.	El algoritmo Backpropagation.	22
III.	DESARROLLO	24
3.1.	METODOLOGÍA.	24
3.2.	ANÁLISIS PREVIO DEL SISTEMA A MODELAR.	26
3.2.1.	IDENTIFICACIÓN DE VARIABLES HIDROLÓGICAS.	26
3.2.2.	DIAGRAMA DE FLUJO.	29
3.2.2.1.	Diagrama de Flujo General.	29
3.2.2.2.	Análisis de Variables Hidrológicas.	31
3.2.2.3.	Entrenamiento de Redes Neuronales Artificiales.	32
3.2.2.4.	Simulación de Variables Hidrológicas.	33
3.2.3.	MODELAMIENTO DEL SISTEMA.	34
3.2.3.1.	Diagrama de Casos de Uso.	34
3.2.4.	ANALIZAR EL COMPORTAMIENTO DE VARIABLES HIDROLÓGICAS.	46
3.3.	DISEÑO DEL SOFTWARE DE SIMULACIÓN	49
3.3.1.	DISEÑO DE LA BASE DE DATOS.	49
3.3.2.	DISEÑO DE LA RNA	50
3.3.3.	ENTRENAMIENTO DE LA RNA	50
3.3.3.1.	Calcular Incrementos del Volumen y Nivel de Espejo.	50

3.3.3.2.	Selección de Parámetros para el Entrenamiento de la RNA	51
3.3.3.3.	Comenzando Entrenamiento.	55
3.3.4.	PASOS PARA LA SIMULACIÓN.	55
3.3.4.1.	Seleccionar Mes y Año.	55
3.3.4.2.	Carga de RNA previamente entrenada.	55
3.3.4.3.	Generación de Variables Aleatorias.	55
3.3.4.4.	Calcular los Incrementos usando la RNA.	56
3.3.4.5.	Calcular el Volumen y Nivel de Espejo.	56
IV.	RESULTADOS	57
4.1.	RESULTADOS OBTENIDOS DEL ANÁLISIS DE LAS VARIABLES HIDROLÓGICAS.	57
4.2.	RESULTADOS OBTENIDOS DEL ENTRENAMIENTO DE LA RNA	59
4.3.	RESULTADOS OBTENIDOS DE VALIDAR EL SOFTWARE DE SIMULACIÓN.	62
4.3.1.	VARIABLES HIDROLÓGICAS GENERADAS ALEATORIAMENTE.	62
4.3.2.	VARIABLES DE SALIDA OBTENIDAS CON LA RNA	66
4.4.	RESULTADOS OBTENIDOS AL REALIZAR LA PROYECCIÓN.	69
V.	DISCUSIONES	71
5.1.	VALIDACIÓN DE OBJETIVOS E HIPÓTESIS.	71
5.1.1.	SUB-HIPÓTESIS Y OBJETIVOS ESPECÍFICOS.	71
5.1.2.	HIPÓTESIS GLOBAL Y OBJETIVO GENERAL.	75

VI. CONCLUSIONES Y RECOMENDACIONES	76
6.1. CONCLUSIONES	76
6.2. RECOMENDACIONES.	77
VII.REFERENCIA BIBLIOGRÁFICA	78
ANEXOS	83
ANEXO 1. INSTRUMENTOS	83
ANEXO 1.1. MEDICIÓN DE LOS INDICADORES.	83
ANEXO 2. MEDICIONES METEOROLÓGICAS	85
ANEXO 2.1. MEDICIÓN DE LA PRECIPITACIÓN TOTAL MENSUAL.	85
ANEXO 2.2. MEDICIÓN DE LA EVAPORACIÓN TOTAL MENSUAL.	86
ANEXO 2.3. MEDICIÓN DEL CAUDAL BOMBEADO.	87
ANEXO 2.4. MEDICIÓN DE LAS DESCARGAS MEDIAS MENSUALES DEL RIO CALLAZAS.	88
ANEXO 2.5. MEDICIÓN DE LAS DESCARGAS MEDIAS MENSUALES DEL RÍO SALADO.	89
ANEXO 2.6. MEDICIÓN DE LAS DESCARGAS MENSUALES DE LOS RÍOS CALLAZAS Y SALADO.	90
ANEXO 2.7. MEDICIÓN DE LAS FILTRACIONES EN CONFLUENCIA ESQUINA - PUQUINA.	92
ANEXO 2.8. MEDICIÓN DE LAS FILTRACIONES EN CONFLUENCIA CON LA CC.HH. ARICOTA N° 2.	92

ANEXO 2.9. MEDICIÓN DEL VOLUMEN ÚTIL DE AGUA.	93
ANEXO 2.10. MEDICIÓN DEL NIVEL DEL ESPEJO DE AGUA.	94
ANEXO 3. CÓDIGO FUENTE	96
ANEXO 3.1. FUNCIONES GENERALES.	96
ANEXO 3.2. CLASES PARA EL ANÁLISIS DE LAS VARIABLES HIDROLÓGICAS.	97
ANEXO 3.3. COMPONENTE RNA	121
ANEXO 3.4. CLASES PARA EL ENTRENAMIENTO DE LA RNA	129
ANEXO 3.5. CÓDIGO FUENTE: DEL ENTRENAMIENTO DE RNA	131
ANEXO 3.6. CÓDIGO FUENTE: DE LA SIMULACIÓN.	131

FIGURAS

Figura 01. El Ciclo Hidrológico	13
Figura 02. Series de Tiempo	16
Figura 03. Neuronas Artificiales	19
Figura 04. Esquema Hidroeléctrico de la Laguna Aricota	25
Figura 05. Diagrama de Flujo: Esquema del Software de Simulación	30
Figura 06. Diagrama de Flujo: Análisis de Variables Hidrológicas	31
Figura 07. Diagrama de Flujo: Entrenamiento de RNA	32
Figura 08. Diagrama de Flujo: Simulación de Variables Hidrológicas	33
Figura 09. Casos de Uso: Funciones del Sistema	34
Figura 10. Caso de Uso: Generar Reportes.	35
Figura 11. Caso de Uso: Ingresar Mediciones.	36
Figura 12. Caso de Uso: Realizar Análisis de Variables Hidrológicas	37

Figura 13. Diagrama de Secuencia: Realizar Prueba de Bondad y Ajuste.	39
Figura 14. Caso de Uso: Realizar Entrenamiento de RNA	40
Figura 15. Diagrama de Secuencia: Realizar Entrenamiento de RNA	42
Figura 16. Caso de Uso: Realizar Simulaciones de Variables Hidrológicas.	43
Figura 17. Diagrama de Secuencia: Realizar Simulación de Variables Hidrológicas.	45
Figura 18. Comportamiento de las Precipitaciones en el Mes de Enero.	47
Figura 19. Diagrama Entidad/Relación.	49

TABLAS

Tabla 01. Especificación de Caso de Uso: Generar Reportes	35
Tabla 02. Especificación de Caso de Uso: Ingresar Mediciones	36
Tabla 03. Especificación de Caso de Uso: Realizar Análisis de Variables Hidrológicas	38
Tabla 04. Especificación de Caso de Uso: Realizar Entrenamiento de RNA	41
Tabla 05. Especificación de Caso de Uso: Realizar Simulaciones de las Variables Hidrológicas	44
Tabla 06. Prueba de Bondad y Ajuste: Precipitación del Mes de Enero	48
Tabla 07. Resultados: Variables Hidrológica / Tipo de Distribución	57
Tabla 08. Resultados: Entrenamiento de la RNA	60
Tabla 09. Resultados: Margen de Error del Entrenamiento	61
Tabla 10. Resultados: Variable Aleatoria – Precipitación (mm)	62
Tabla 11. Resultados: Variable Aleatoria – Evaporación (mm)	63

Tabla 12. Resultados: Variable Aleatoria – Caudal Bombeado (m ³ /s)	63
Tabla 13. Resultados: Variable Aleatoria – Descargas del Río Callazas (m ³ /s)	64
Tabla 14. Resultados: Variable Aleatoria – Descargas del Río Salado (m ³ /s)	64
Tabla 15. Resultados: Variable Aleatoria – Descargas del Río Callazas y Salado (m ³ /s)	65
Tabla 16. Resultados: Variable Aleatoria – Filtraciones Esquina – Puquina (m ³ /s)	65
Tabla 17. Resultados: Variables Aleatoria – Filtraciones CC.HH. Aricota N° 2 (m ³ /s)	66
Tabla 18. Resultados: Mediante Entrenamiento – Incremento de Volumen (MMC)	67
Tabla 19. Resultados: Mediante Entrenamiento – Volumen (MMC)	67
Tabla 20. Resultados: Mediante Entrenamiento – Incremento del Nivel de Espejo (msnm)	68
Tabla 21. Resultados: Mediante Entrenamiento – Nivel de Espejo (msnm)	68
Tabla 22. Resultados: Proyección – Volumen (MMC)	69

Tabla 23. Resultados: Proyección – Nivel de Espejo (msnm)	70
Tabla 24. Validación de Sub-Hipótesis 01: Precipitación	72

RESUMEN

El presente trabajo de Investigación, tiene como objetivo desarrollar un Software que facilite el control de la información obtenida por las distintas estaciones Meteorológicas, ubicadas alrededor de la laguna Aricota, las cuales recolectan datos como son de: Precipitación, Evaporación, Descargas, Caudal Bombeado, Filtraciones. Así mismo, el Software al contar con dicha Información almacenada en una Base de Datos, podrá ser visualizada y manipulada de manera sencilla, permitiendo así realizar un estudio mas profundo.

Dicho Software cuenta también con módulos que permiten realizar: en análisis de cada Variable Hidrológica, asimismo realizar el entrenamiento de una Red Neuronal Artificial (para obtener valores de incremento del volumen y nivel de agua de la Laguna Aricota), y la Simulación del Ciclo Hidrológico de la Laguna Aricota, buscando así **predecir** *“la cantidad de Agua es con la que contamos”*. Pudiendo así estimar el Recurso Hídrico para un Futuro, que en el Dpto. de Tacna es relativamente escaso, y buscar la manera de no desperdiciar dicho recurso.

INTRODUCCIÓN

El trabajo aquí expuesto tiene por objeto ser presentado ante la Universidad Nacional Jorge Basadre Grohmann, para optar el Título Profesional de Ingeniero en Informática y Sistemas.

Situada en la provincia de Candarave, a 154 km de la ciudad de Tacna, la laguna de Aricota es el lugar preciso para practicar deportes acuáticos como la pesca. Así como es también una impresionante fuente de agua, la cual tiene una superficie de 1 456 km² y alberga a las famosas truchas arco iris, ideales para la pesca lacustre.

También es hábitat de aves como patillos y guallatas. En la laguna se ubica la central hidroeléctrica de Aricota, importante para la región por el abastecimiento de luz eléctrica que brinda a las ciudades de Tacna, Moquegua e Ilo. Así como de innumerables leyendas que existen acerca de la Laguna Aricota.

En el presente trabajo de investigación, se busca analizar los parámetros que intervienen en el aumento o disminución del Volumen del agua de la Laguna Aricota, para así mediante cálculos hacer una estimación del volumen de agua con la que se contará en un futuro cercano.

Para el inicio de la Investigación, fue necesario analizar como el comportamiento de cada Variable Hidrológica (Precipitación, Evaporación, Descargas, Bombeos, Filtraciones), las cuales poseen una característica y complejidad propia de cada una, para lo cual se realizó pruebas de bondad de ajuste, y así ver que tipo de distribución es la que sigue cada una de las variables hidrológicas.

Para realizar un cálculo, tanto del volumen como del nivel de agua de la Laguna Aricota, es necesario realizarlo mediante operaciones empíricas, por lo que se optó con hacer uso de Redes Neuronales Artificiales (R.N.A.), entrenadas previamente, y así lograr obtener una simulación lo más cercana a la realidad, y con un margen de Error lo mas pequeño posible.

Para la aplicación de una Red neuronal Artificial, se utilizó el algoritmo de Backpropagation para un Perceptrón Multicapa, el cual es el más conocido y se adecua muy bien al modelo propuesto, para la realización los cálculos necesarios y así obtener valores de volumen y nivel de espejo más confiables.

Al momento de realizar el entrenamiento de la R.N.A., se utilizó ocho variables de entradas: Precipitación, Evaporación, Caudal Bombeado, Descargas medias del Rio Salado, Descargas medias del Rio Callazas, Descargas medias de los Ríos Callazas y Salado, Filtraciones en Confluencia Esquina – Puquina, Filtraciones en Confluencia con la Central Hidroeléctrica N° 2; y se tomó dos variables de salida: Volumen total de agua, Nivel de Espejo del agua de la Laguna Aricota.

Una vez entrenada la R.N.A., se realizó la generación de variables hidrológicas aleatorias de cada una de las variables de entradas, tomándolas como variables de entradas para la R.N.A., previamente entrenada, obteniendo tanto el volumen como el nivel de espejo de la Laguna Aricota, variables simuladas las cuales poseen un margen de error del 5%.

Con los resultados obtenidos, se podrán realizar estudios de refinanciamiento hídrico de la Laguna Aricota, o darle un mejor uso controlado de sus aguas.

I. PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. Descripción del Problema

1.1.1. Antecedentes del Problema.

La laguna Aricota se formó por deslizamientos en el cauce del río Curibaya, presenta un área de cuenca de 1 456 km², para un volumen de 840 millones de m³ (MMC), y la profundidad máxima de la laguna llega hasta los 130 metros, para un nivel de espejo de agua con ancho promedio de 1.2 km y 8 km de longitud.

El esquema hidroeléctrico de Aricota está constituido por dos centrales hidroeléctricas: la C.H. Aricota I y luego la C.H. Aricota II.

Durante los años de operación de los requerimientos tanto de las centrales como los otros usuarios aguas abajo, se produjo la disminución del volumen almacenado en la laguna Aricota, habiendo alcanzado el volumen mínimo en 1996, de 90 MMC y una profundidad de 25 m.

Ante el inminente descenso de los recursos de la Laguna, el Proyecto Especial Tacna (PET) ejecutó obras de afianzamiento hídrico de la laguna Aricota, que sumado a los periodos de lluvias favorables de los últimos años, han permitido que la laguna empiece a incrementar su nivel y volumen de agua almacenada, habiéndose logrado un volumen actual de 154 MMC.

1.1.2. Problemática de la Investigación.

El agua es la sustancia más abundante en la Tierra, es el principal constituyente de todos los seres vivos, y es una fuerza importante que constantemente está cambiando la superficie terrestre. También es un factor clave en la climatización de nuestro planeta para la existencia humana y a la vez tiene influencia en el progreso de la civilización.

El Departamento de Tacna, se caracteriza, entre otros aspectos por una extremada escasez de recursos hídricos, por lo cual existen diversos proyectos para su debida conservación, realizando estudios y evaluación sobre el uso adecuado de dicho recurso.

Esto origina la necesidad de contar con información hidrológica y climática, que permita evaluar las diferentes fuentes de recursos hídricos en cuanto a cantidad y calidad, de manera que se pueda contar con información histórica que ha sido acumulada en diversas investigaciones precedentes a esta, para poder actualizar los estudios hidrológicos de los proyectos ya existentes, y poder así optimizar su aprovechamiento, sin llegar al abuso de este recurso y explotación innecesaria, manteniendo así un adecuado control y uso responsable.

Por lo que se busca manipular dicha información de una manera más eficaz, para lo cual ha de requerirse de Modelos Hidrológicos que brinden mediciones simuladas en distintos instantes de tiempo, y así poder

“predecir” el Volumen del Agua en la laguna Aricota, y poder tomar decisiones adecuadas sobre su uso y aprovechamiento.

1.2. Formulación del Problema

1.2.1. Problema General

¿El uso de Redes Neuronales Artificiales optimiza la simulación del Ciclo Hidrológico de la Laguna Aricota?

1.2.2. Problemas Específicos

- ¿Las Variables Hidrológicas generadas aleatoriamente, son confiables?
- ¿Qué tan confiable es el uso de Redes Neuronales Artificiales, para la simulación del Ciclo Hidrológico de la Laguna Aricota?
- ¿Qué se demuestra con la simulación al año 2020, respecto al Volumen de agua de la Laguna Aricota?

1.3. Justificación

El presente Trabajo de Investigación (Tesis), busca brindar una herramienta que permita realizar simulaciones del ciclo hidrológico de la Laguna Aricota en distintos instantes de tiempo, haciendo uso de las Redes Neuronales Artificiales, las cuales permitirán realizar cálculos complejos (cálculos empíricos), y así obtener valores de simulaciones del volumen de agua con la que se contará en un futuro. Siguiendo así con los estudios de afianzamiento hídrico, y lograr reponer el volumen de

agua de la Laguna Aricota en un nivel que permita su uso, tanto para las hidroeléctricas, como para los usuarios aguas abajo.

1.4. Alcances y Limitaciones

Tomando en cuenta que existen Instituciones que se encargan de realizar proyectos, y realizar la medición de los datos de interés, los cuales usaremos para el siguiente estudio, se ha de seguir los siguientes pasos para realizar el análisis de las variables, así como la simulación del ciclo hidrológico de la Laguna Aricota:

- Se ha de realizar un estudio (Análisis del Tipo de Distribución: **Prueba de Bondad de Ajuste**) de las Variables Hidrológicas existentes en el Ciclo Hidrológico de la laguna Aricota, recolectando información cuantitativa, la cual será almacenada en una Base de Datos. Dicha información meteorológica recolectada servirá para conocer la mecánica del sistema a estudiar y el comportamiento de cada Variable, y la función de distribución a la cual se ajusta cada Variables Hidrológica.
- Se ha de generar las Variables Aleatorias de cada Variable Hidrológica que influye en el Ciclo Hidrológico de la Laguna Aricota, lo cual se hará siguiendo el comportamiento de cada variable, y en distintos instantes de tiempo; para una simulación más exacta se generaran variables aleatorias por cada mes, es decir, una Variable Hidrológica podrá generar hasta 12 variables para un año (una por cada mes), puesto que en cada mes el comportamiento de estas variables es distinto.

- Se realizará el entrenamiento de una Red Neuronal Artificial (RNA), tomando como Variables de Entrada las Variables Hidrológicas, y como Variables de Salida los incrementos del Volumen de agua y el Nivel de Espejo.
- Finalmente al realizar la generación de las Variables Hidrológicas, por mes/año, se usará la RNA, previamente entrenada, para calcular así los incrementos del Volumen y del Nivel, y acumularlos a los anteriores. Para así obtener el Volumen y Nivel definitivo simulado a una fecha dada.

1.5. Objetivos

1.5.1. Objetivo General.

O: Simular el Ciclo Hidrológico de la Laguna Aricota, utilizando Redes Neuronales Artificiales.

1.5.2. Objetivos Específicos

O₁: Analizar el comportamiento y el tipo de Distribución que sigue cada variable hidrológica, que interviene en el Ciclo Hidrológico de la Laguna Aricota, y así obtener el tipo de distribución a la cual se ajusta cada una de ellas.

O₂: Generar las variables hidrológicas aleatorias, según el tipo de distribución a la que pertenecen; y así obtener el Volumen y Nivel de Espejo simulados, utilizando Redes Neuronales Artificiales.

O₃: Obtener una proyección, mediante el uso del software de simulación, el valor del Volumen y del Nivel de Espejo de agua de la Laguna Aricota para el año 2020.

1.6. Hipótesis

1.6.1. Hipótesis Global.

H₀: Los resultados obtenidos, por medio del software desarrollado para la simulación del ciclo hidrológico de la laguna Aricota, no poseen un alto grado de confiabilidad.

H₁: Los resultados obtenidos, por medio del software desarrollado para la simulación del ciclo hidrológico de la laguna Aricota, poseen un alto grado de confiabilidad.

1.6.2. Sub – Hipótesis.

H₁: Las Variables Hidrológicas generadas aleatoriamente, influyen favorablemente a la simulación, ya que poseen un algo grado de confiabilidad.

H₂: La aplicación de Redes Neuronales Artificiales, favorece el proceso de la Simulación, permitiendo calcular tanto el Volumen como el Nivel de Espejo de Agua de la Laguna Aricota, con un alto grado de confiabilidad.

H₃: Los valores del volumen y nivel de espejo de agua, obtenidos mediante la simulación al año 2020, demuestran que se necesitarán

realizar proyectos para un afianciamento hídrico de la Laguna Aricota.

1.7. Variables

1.7.1. Identificación de Variables.

a. Variables Independientes.

- Software utilizando Redes Neuronales Artificiales.

b. Variables Dependientes.

- Simulación del Ciclo Hidrológico de la laguna Aricota.

1.7.2. Definición de Variables.

a. Simulación del Ciclo Hidrológico de la laguna Aricota.

- **Por la Función que Cumple en la Hipótesis.**

Variable Dependiente.

- **Por su Naturaleza.**

Activa.

- **Por el Método de Estudio.**

Cuantitativa.

- **Por la posesión de la Característica.**

Continuo.

- **Por los valores que adquiere.**

Politómica.

b. Software utilizando Redes Neuronales Artificiales.

- **Por la Función que Cumple en la Hipótesis.**

Variable Independiente.

- **Por su Naturaleza.**

Desactiva.

- **Por el Método de Estudio.**

Cualitativo.

- **Por la posesión de la Característica.**

Discreta.

- **Por los valores que adquiere.**

Politómica.

1.7.3. Operacionalización de las Variables.

a. Variable Dependiente: Simulación del Ciclo Hidrológico de la laguna Aricota.

- **Definición Conceptual:** Conjunto de procesos que diseñan y modela el comportamiento de las variables hidrológicas que influyen en las variaciones del Volumen de la Laguna Aricota.
- **Definición Operacional:** Variable que verifica que tan fiable son los resultados obtenidos mediante el uso del software de simulación, comparándolo con valores reales (datos históricos).
- **Indicadores:** Confiabilidad.
- **Valores:** Porcentaje.

b. Variable Independiente: Software utilizando Redes Neuronales Artificiales.

- **Definición Conceptual:** Software para el estudio y simulación del Ciclo Hidrológico de la laguna Aricota y de cada una de las variables hidrológicas que influyen en el sistema.
- **Definición Operacional:** Contribuye en el análisis y la simulación del ciclo hidrológico de la Laguna Aricota, minimizando los procesos, operaciones y cálculos necesarios.
- **Indicadores:** Confiabilidad, Porcentaje de Error de RNA.
- **Valores:** Porcentaje.

Ver Anexo 1.1.

1.8. Diseño de la Investigación

1.8.1. Diseño Experimental.

Diseño Experimental con Pre-Prueba y Pos-Prueba y Grupo de Control.

RG_1	O_1	X	O_2
RG_2	O_3	-	O_4

Puesto que se ha de realizar diversas pruebas a las variables aleatorias, y comprobar si su comportamiento corresponde o no al de las variables hidrológicas (datos históricos).

1.8.2. Población.

Se ha de tomar las mediciones realizadas desde los años 1967 hasta la fecha (en algunos casos, ya que no todas las mediciones se empezaron a realizar partir de ese año), estos datos serán de: las precipitaciones, las descargas, la evaporación, filtraciones, y caudal bombeado, lo cual buscará obtener una Simulación más precisa.

1.8.3. Técnicas e Instrumentos para la recolección de Datos.

Datos recopilados de las Instituciones que intervinieron hasta ahora: Mediciones que se han realizado desde el año 1967 hasta la fecha (Mediciones Hidrológicas), Estudios realizados, Proyectos, etc.

1.8.4. Análisis de datos y selección de Pruebas Estadísticas.

Se realizará la comprobación de los resultados obtenidos, realizando Pruebas de Homogeneidad de Datos, tales como: la Prueba de T de Student (Prueba para la "No Homogeneidad" en la media).

Verificando así, que los resultados obtenidos mediante el uso del software de simulación, y las redes neuronales artificiales, son confiables, y que estos siguen la distribución obtenida mediante el análisis de las variables hidrológicas.

II. MARCO TEÓRICO

2.1. Marco Referencial

2.1.1. Hidrología.

(Ven Te, 1994) En la Tierra, el agua existe en un espacio llamado *hidrósfera*, que se extiende desde unos quince kilómetros arriba en la atmósfera hasta un kilómetro por debajo de la litosfera o corteza terrestre. El agua circula en la hidrósfera a través de un laberinto de caminos que constituyen el ciclo hidrológico.

(Mijares, 1989) Hidrología es la ciencia natural que estudia al agua, su ocurrencia, circulación y distribución en la superficie terrestre, sus propiedades químicas y físicas, y su relación con el medio ambiente, incluyendo a los seres vivos.

(Vera Hernández, 2002) Es la ciencia que estudia el agua, y sus manifestaciones en la atmósfera, sobre y debajo de la superficie terrestre; sus propiedades y sus interrelaciones con la naturaleza.

2.1.2. Metodología Sistémica.

(Aracil) Para el estudio de los sistemas en general se ha desarrollado lo que se conoce como metodología sistémica, o conjunto de métodos mediante los cuales abordar los problemas en los que la presencia de sistemas es dominante.

En realidad, la metodología sistémica pretende aportar instrumentos con los que estudiar aquellos problemas que resultan de las interacciones que se producen en el seno de un sistema, y no de disfunciones de las partes consideradas aisladamente.

2.2. Bases Teóricas respecto al Problema.

2.2.1. Ciclo Hidrológico.

(Ven Te, 1994) El ciclo hidrológico es el foco central de la hidrología. El ciclo no tiene principio ni fin y sus diversos procesos ocurren en forma continua. En la figura 01, se muestra en forma esquemática como el agua se evapora desde los océanos y desde la superficie terrestre para volverse parte de la atmósfera; el vapor de agua se transporta y se eleva en la atmósfera hasta que se condensa y precipita sobre la superficie terrestre o los océanos; el agua precipitada puede ser interceptada por la vegetación, convertirse en flujo superficial y descargar en los ríos como escorrentía superficial. La mayor parte del agua interceptada y de escorrentía superficial regresa a la atmósfera mediante la evaporación. El agua infiltrada puede percollar profundamente hasta ríos para descargar el agua subterránea de donde emerge en manantiales o se desliza hacia ríos para formar la escorrentía superficial, y finalmente fluye hacia el mar o se evapora en la atmósfera a medida que el ciclo hidrológico continúa.

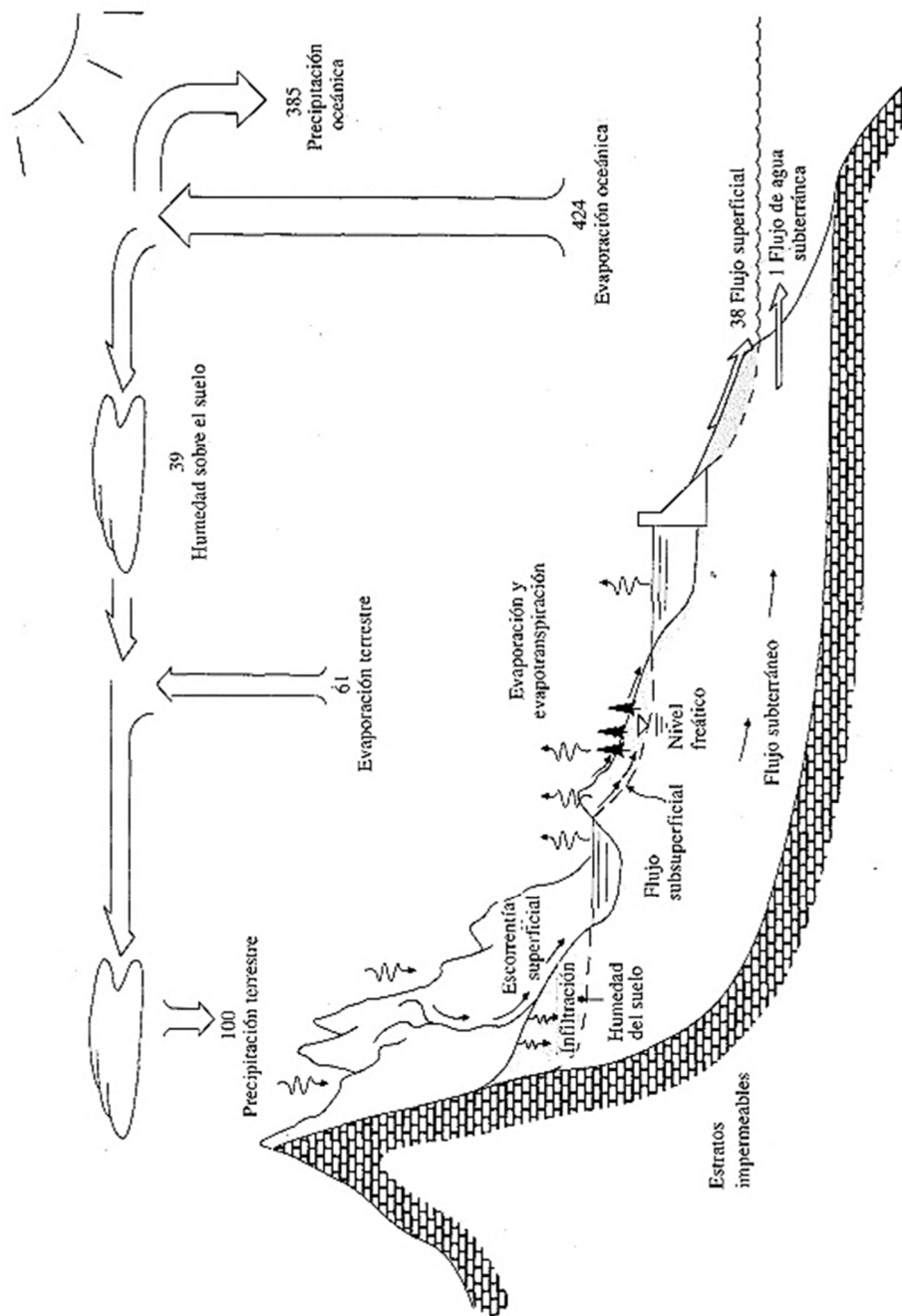


Figura 01. El Ciclo Hidrológico

Fuente: Ven Te, Ch. Hidrología Aplicada. Pág. 3

2.2.2. Simulación Digital.

(Domínguez Valente, López Octaviano, Martínez Martínez, Reyes Martínez, & Vázquez Hernández, 2008) La simulación digital es una técnica que permite imitar (o simular) en un ordenador el comportamiento de un sistema real o hipotético según ciertas condiciones particulares de operación.

Aunque la simulación digital es una técnica relativamente reciente y en constante evolución, el uso de la simulación como metodología de trabajo es una actividad muy antigua, y podría decirse que inherente al proceso de aprendizaje del ser humano.

Para poder comprender la realidad y toda la complejidad que un sistema puede conllevar, ha sido necesario construir artificialmente objetos y experimentar con ellos dinámicamente antes de interactuar con el sistema real. La simulación digital puede verse como el equivalente computarizado a este tipo de experimentación.

Para ello es necesario construir objetos (modelos) que representan la realidad, de tal modo que pueden ser interpretados por un ordenador.

2.2.3. Modelo.

(Azarang & García Dunna, 1998) Un modelo es una representación de la realidad desarrollado con el propósito de estudiarla. En la mayoría de los análisis no es necesario considerar todos los detalles de la

realidad, entonces, el modelo no sólo es un sustituto de la realidad sino también una simplificación de ella.

2.2.4. Distribuciones de Probabilidad.

(Azarang & García Dunna, 1998) Al modelar un sistema, se debe diferenciar entre dos tipos de datos: los primeros permaneces sin cambio a través del tiempo y se conocen como “parámetros”; los segundos presentan cambios a través del tiempo y se conocen como “variables”. Por ejemplo, el modelado de un sistema mediante simulación es útil cuando la información del sistema tiene carácter dinámico y probabilístico, debido principalmente a que la interacción de esa información es, por lo general, difícil de analizar.

La variabilidad que presenta el segundo tipo de datos debe modelarse de acuerdo con ciertas ecuaciones matemáticas que sean capaces de reproducirla; en la mayoría de los casos dicha variabilidad puede clasificarse dentro de alguna distribución de probabilidad. Así pues, uno de los pasos más importantes de todo el proceso de modelado estocástico es la búsqueda de información y su análisis estadístico posterior basado principalmente en la clasificación de cada serie de datos dentro de alguna distribución de probabilidad.

2.2.5. Determinación del Tipo de Distribución.

(Azarang & García Dunna, 1998) En la mayor parte de los sistemas, al analizar la información, esta se encuentra disponible en forma de series a través del tiempo, como se muestra en la Figura 02.

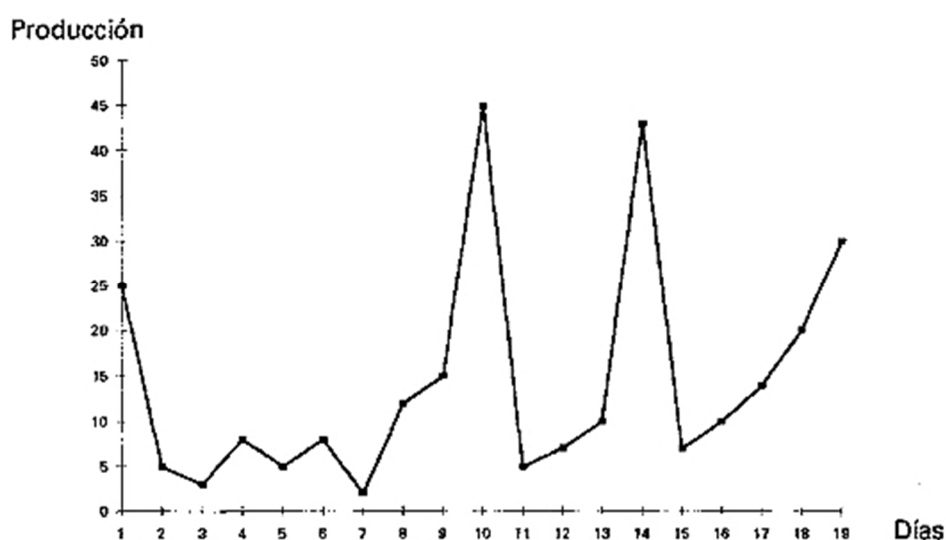


Figura 02. Series de Tiempo

Fuente: Azarang, M; García Dunna, E.; Simulación y Análisis de Modelos Estocásticos. Pág. 11.

Esta información, tabulada en dicho formato no es de utilidad cuando se trata de obtener un comportamiento basado en variabilidad con cierto comportamiento probabilístico. Así pues, si el analista desea conocer el comportamiento, es necesario modificar la forma de presentación de datos y presentarla como tablas de frecuencia, con la finalidad de realizar cualquiera de las pruebas de bondad y ajuste.

2.2.5.1. Pruebas de Bondad y Ajuste X^2 .

(Azarang & García Dunna, 1998) Esta prueba se utiliza para encontrar la distribución de probabilidad de una serie de datos. La metodología de la prueba X^2 es la siguiente:

1. Se colocan los n datos históricos en una tabla de frecuencias de $m = 1 + 3.32 * \text{Log}10(n)$ intervalos. Se obtiene la frecuencia observada en cada intervalo i (FO_i). Se calcula la media y la varianza de los datos.
2. Se propone una distribución de probabilidad de acuerdo con la forma de la tabla de frecuencias obtenida en el paso 1.
3. Con la distribución propuesta, se calcula la frecuencia esperada para cada uno de los intervalos (FE_i) mediante la integración de las distribuciones propuesta y su posterior multiplicación por el número total de datos.
4. Se calcula el estimador: $C = \sum_{i=1}^m [(FE_i - FO_i)^2 / FE_i]$
5. Si el estimado C es menor o igual al valor correspondiente X^2 con $m-k-1$ grados de libertad ($k =$ número de parámetros estimados de la distribución) y a un nivel de confiabilidad de $1 - \alpha$, entonces no se puede rechazar la hipótesis de que la información histórica sigue la distribución propuesta en el punto 2.

2.2.6. Redes Neuronales Artificiales (RNA).

(Boza Condorena, 2005) Las Redes Neuronales Artificiales permiten la solución de problemas complejos, utilizando procedimientos que no se sustentan en una secuencia de pasos, sino que inspirados en el cerebro humano, utilizan una combinación de elementos simples de proceso (neuronas) interconectados, que operan en forma paralela para resolver los problemas. El conocimiento de una red neuronal no se almacena en instrucciones, el poder de la red está en su topología y en los valores de las conexiones (pesos) entre neuronas.

(Bertona, 2005) Las redes neuronales artificiales (RNA) son modelos matemáticos que intentan reproducir el funcionamiento del sistema nervioso. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada.

2.2.6.1. Modelo de neurona artificial.

(Bertona, 2005) La neurona artificial es un elemento de procesamiento simple que a partir de un vector de entradas produce una única salida. En general podemos encontrar tres tipos de neuronas artificiales, donde cada una de las cuales tiene su contraparte en el sistema nervioso:

1. Las que reciben información directamente desde el exterior, a las cuales se las denomina *neuronas de entrada*.

2. Las que reciben información desde otras neuronas artificiales, a las cuales se las denomina *neuronas ocultas*. Es en estas neuronas, en particular en sus sinapsis, es donde se realiza la representación de la información almacenada.
3. Las que reciben la información procesada y las devuelven al exterior. A estas neuronas se las denomina *neuronas de salida*.

La figura 03, muestra los elementos que componen una neurona artificial:

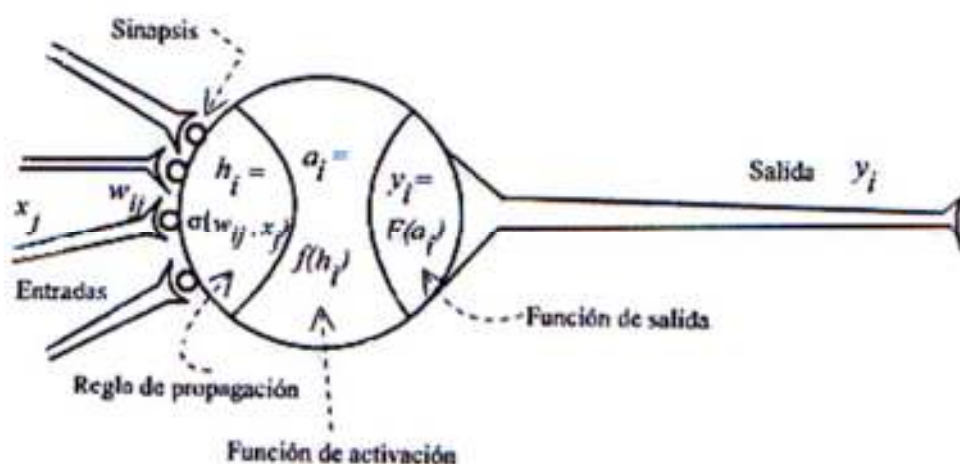


Figura 03. Neurona Artificial

Fuente: Bertona, L.F; Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos. Pág. 7.

- **Conjunto de entradas, $\mathbf{x}_j(t)$.** Estas pueden ser provenientes del exterior o de otras neuronas artificiales.
- **Peso sinápticos, w_{ij} .** Representan el grado de comunicación entre la neurona artificial j y la neurona artificial i. Pueden ser excitadores o inhibidores.
- **Regla de propagación, $\sigma_i(w_{ij}, \mathbf{x}_j(t))$.** Integra la información proveniente de las distintas neuronas artificiales y proporciona el valor del potencial postsináptico de la neurona i.
- **Función de activación, $f_i(a_i(t-1), h_i(t))$.** Provee el estado de activación actual de la neurona i.
- **Función de salida, $F_i(a_i(t))$.** Representa la salida actual de la neurona i. De esta forma, la salida producida por una neurona i, para un determinado instante de tiempo t, puede ser escrita en forma general de la siguiente manera: $y_i(t) = F_i\left(f_i\left[a_i(t-1), \sigma_i\left(w_{ij}, x_j(t)\right)\right]\right)$

2.2.6.2. Aprendizaje o Entrenamiento de una Red Neuronal.

(Morales González, 2010) El proceso de aprendizaje o entrenamiento en una red neuronal puede ser visto como un problema de actualización de los pesos sinápticos de las neuronas de manera tal que la red neuronal mejore la forma en que realiza cierta tarea específica. Estas mejoras en el desempeño de una red neuronal son logradas a lo largo de este proceso de entrenamiento y mediante la

actualización iterativa de los pesos sinápticos, los cuales están representados por un vector o matriz denominado vector o matriz de pesos W . El aprendizaje de las RNA es un aprendizaje basado en ejemplos o patrones, estos ejemplos deben tener dos características fundamentales, primero, ser significativo, debe haber un número suficiente de ejemplos para que la neurona sea capaz de adaptar sus pesos en forma eficaz y ser representativo, es decir los ejemplos deberán ser diversos y reflejar todas los posibles estados de la red. Algunos autores aconsejan usar muestras aleatorias como ejemplos, sin embargo, en la mayoría de las aplicaciones no se encontrarán muestras aleatorias sino muestras de tamaño muy grande.

En general, el proceso de aprendizaje se alcanza de dos maneras. En ocasiones, las redes neuronales utilizarán información previamente establecida, y basándose en dicha información se lleva a cabo la actualización de los pesos. En otras ocasiones las redes neuronales deben aprender a establecer los pesos sin esta información. A estos tipos de aprendizaje se les denomina Supervisado y No Supervisado o Auto Organizado, respectivamente (en general, cualquier técnica de clasificación y predicción puede encontrarse en uno de estos dos grupos).

2.2.6.3. Perceptrón multicapa.

(Bertona, 2005) El perceptrón multicapa es una extensión del perceptrón simple. La topología de un perceptrón multicapa está definida por un conjunto de capas ocultas, una capa de entrada y una de salida. No existen restricciones sobre la función de activación aunque en general se suelen utilizar funciones sigmoideas.

2.2.6.4. El algoritmo Backpropagation.

(Bertona, 2005) Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

El sistema de entrenamiento mediante back-propagation consiste en:

- Empezar con unos pesos sinápticos cualquiera (generalmente elegidos al azar).

- Introducir unos datos de entrada (en la capa de entradas) elegidos al azar entre los datos de entrada que se van a usar para el entrenamiento.
- Dejar que la red genere un vector de datos de salida (propagación hacia delante).
- Comparar la salida generada por la red con la salida deseada.
- La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas.
- El error se propaga hacia atrás (back-propagation), hacia la capa de neuronas anterior, y se usa para ajustar los pesos sinápticos en esta capa.
- Se continua propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas.

Este proceso se repetirá con los diferentes datos de entrenamiento.

III. DESARROLLO

3.1. Metodología.

Para el desarrollo de la siguiente aplicación de simulación, se optó por seguir la Metodología eXtreme Programming (XP). Por ser una metodología de desarrollo ágil, ya que lo que se busca es desarrollar un software que funcione, más que conseguir una buena documentación; además de ser necesario que el usuario final (cliente) participe no sólo al inicio, sino en todo el ciclo de vida del desarrollo del software (realizando preguntas sobre si se esta realizando el proceso adecuado para el desarrollo, tanto del análisis como del modelo de simulación). Así como también responder rápidamente a las sugerencias o correcciones de posibles errores que pudieran surgir, al momento de estar desarrollando el software de simulación.

A continuación en la Figura 4, el esquema de la Laguna Aricota, donde se observa conjuntamente las Hidroeléctricas Aricota I y Aricota II, así como también se observan las filtraciones, y el estado de los túneles que se utilizan para realizar el bombeo que van hacia la Hidroeléctrica Aricota I.

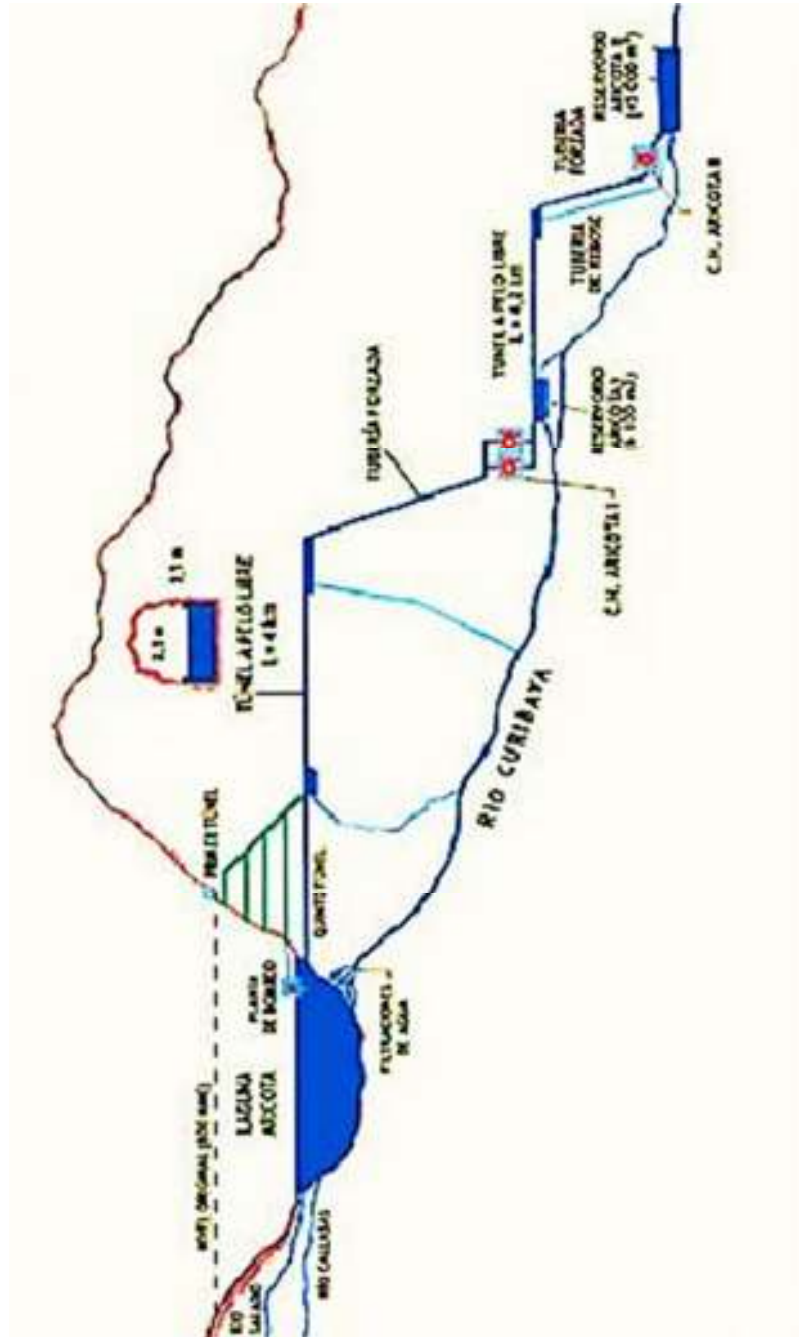


Figura 04. Esquema Hidroeléctrico de la Laguna Aricota

Fuente: Proyecto Especial Tacna

3.2. Análisis previo del Sistema a Modelar.

3.2.1. Identificación de Variables Hidrológicas.

a. Precipitación Total Mensual.

Medición de las lluvias, para la realización de las mediciones este se hace uso mediante la aplicación de pluviómetros, y se miden mediante la escala de medida de milímetros (mm).

- Código de Estación : 1910110B.
- Nombre de Estación : Aricota.
- Categoría de Estación : CO.

Contando con mediciones desde diciembre del año 1993 a diciembre del 2010. *Ver Anexo 2.1.*

b. Evaporación Total Mensual.

Medición de la evaporación de las aguas de la Laguna, y se encuentra en una escala de medida de milímetros (mm).

- Código de Estación : 1910114B.
- Nombre de Estación : Aricota.
- Categoría de Estación : CO.

Contando con mediciones desde diciembre del año 1993 a diciembre del 2006 y de enero del 2009 a diciembre del 2010. *Ver Anexo 2.2.*

c. Caudal bombeado, hacia las hidroeléctricas: Aricota I y Aricota II.

Variable hidrológica en la cual actúa la mano del hombre, se mide en metros cúbicos por segundo (m^3/s).

- Código de Estación : 1910111R.
- Nombre de Estación : Bombeo Aricota.
- Categoría de Estación : Limnigráfica (Quinto Túnel).

Contando con mediciones desde febrero del año 1967 a diciembre del 2010. *Ver Anexo 2.3.*

d. Descargas de los Ríos:

Ingreso de los Ríos hacia la Laguna Aricota, cuya unidad de medida es en metros cúbicos por segundo (m^3/s).

○ **Descargas Medias Mensuales del Rio Callazas.**

Control de las mediciones del caudal del Rio Callazas.

- Código de Estación : 1910111R.
- Nombre de Estación : Candarave (Pallata)
- Categoría de Estación : Limnimétrica.

Contando con mediciones desde octubre del año 1963 a noviembre de 1973, de enero de 1991 a diciembre del 2004, y de enero del 2006 a diciembre del 2010. *Ver Anexo 2.4.*

○ **Descargas Medias Mensuales del Rio Salado.**

Control de las mediciones del caudal del Rio Salado.

- Código de Estación : 1910111B.
- Nombre de Estación : Aricota – La Yesera.
- Categoría de Estación : Limnimétrica.

Y se cuentan datos desde octubre del año 1963 a noviembre de 1973, de enero de 1991 a diciembre del 2004, y de enero del 2006 a diciembre del 2010. *Ver Anexo 2.5.*

○ **Descargas Mensuales de los Ríos Callazas y Salado.**

Control de las descargas de los Ríos Callazas y Salado, el ingreso a la Laguna Aricota.

- Código de Estación : 1910111Q
- Nombre de Estación : Aricota.
- Categoría de Estación : Limnimétrica.

Y se cuentan datos desde octubre del año 1963 a diciembre del 2004, y de enero del 2006 a diciembre del 2010. *Ver Anexo 2.6.*

e. Filtración.

Salida o pérdida de la Laguna Aricota, al no contarse con información mensual de las mediciones de las filtraciones, se realizó un Análisis Total de la variable, las mediciones son desde enero del

año 1992 a junio del 2011, cuya unidad de medida es en metros cúbicos por segundo (m^3/s).

- Filtraciones en Confluencia Esquina - Puquina. *Ver Anexo 2.7.*
- Filtraciones en Confluencia con la Central Hidroeléctrica Aricota N° 2. *Ver Anexo 2.8.*

3.2.2. Diagrama de Flujo.

Para un desarrollo adecuado del software, fue necesario hacer los diagramas de flujo correspondientes a cada módulo, ya que dichos diagramas facilitarán la implementación del software de simulación.

Teniendo en cuenta esto, se realizaron los siguientes diagramas de flujo:

3.2.2.1. Diagrama de Flujo General.

Al momento de realizar el análisis, de como es que se implementará el sistema, se realizó primeramente un Diagrama de Flujo General, en el cual uno puede observar el funcionamiento General del Sistema de Simulación, así también de como es que interactúan cada uno de los subsistemas.

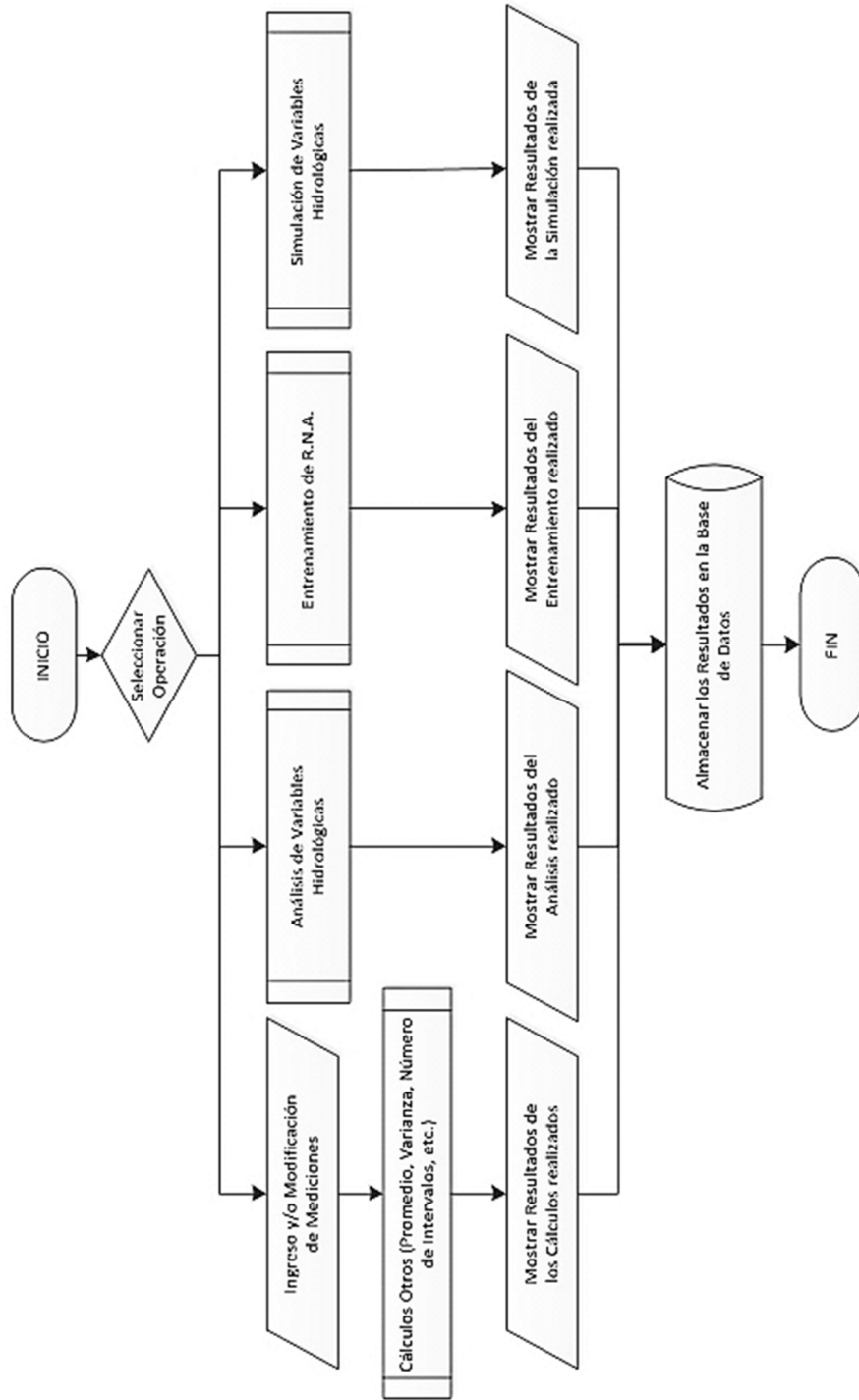


Figura 05. Diagrama de Flujo: Esquema del Software de Simulación

Fuente: Propia

3.2.2.2. Análisis de Variables Hidrológicas.

Se realizó un análisis previo, sobre el proceso a seguir para realizar el análisis de las variables hidrológicas.

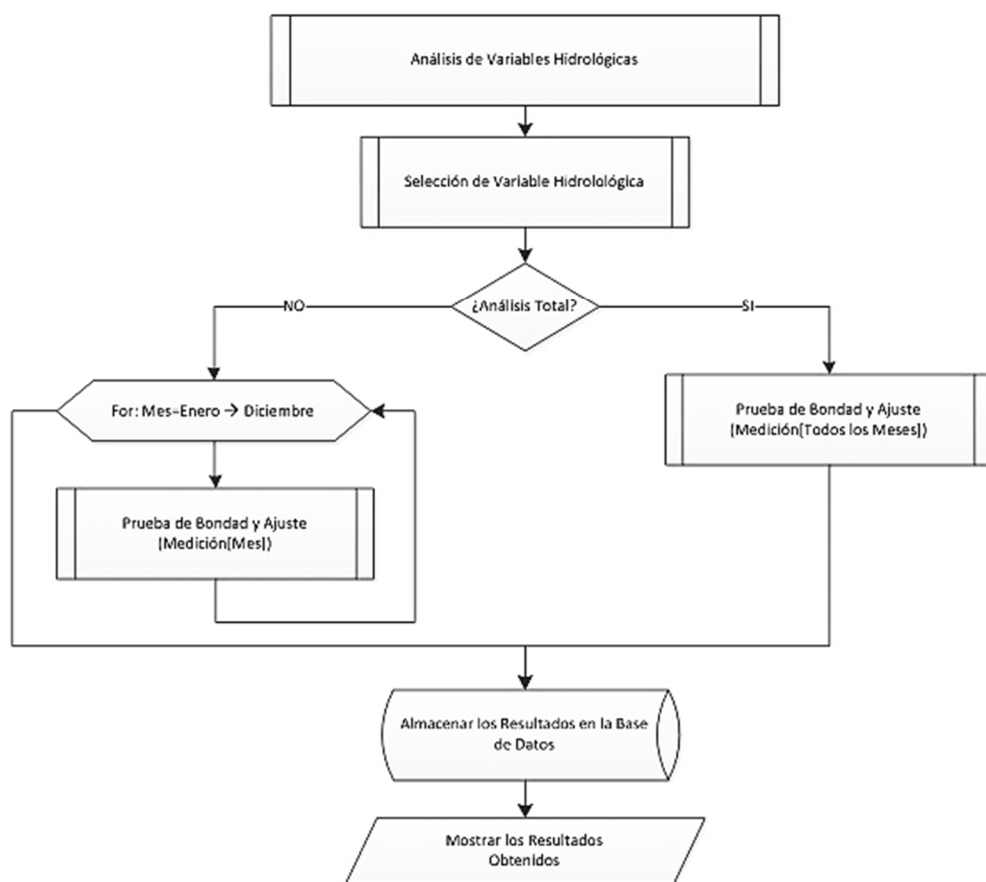


Figura 06. Diagrama de Flujo: Análisis de Variables Hidrológicas

Fuente: Propia

3.2.2.3. Entrenamiento de Redes Neuronales Artificiales.

El proceso para realizar el entrenamiento de la RNA viene a ser el siguiente, teniendo en cuenta que previamente se implementó el componente **TRed_Neural**:

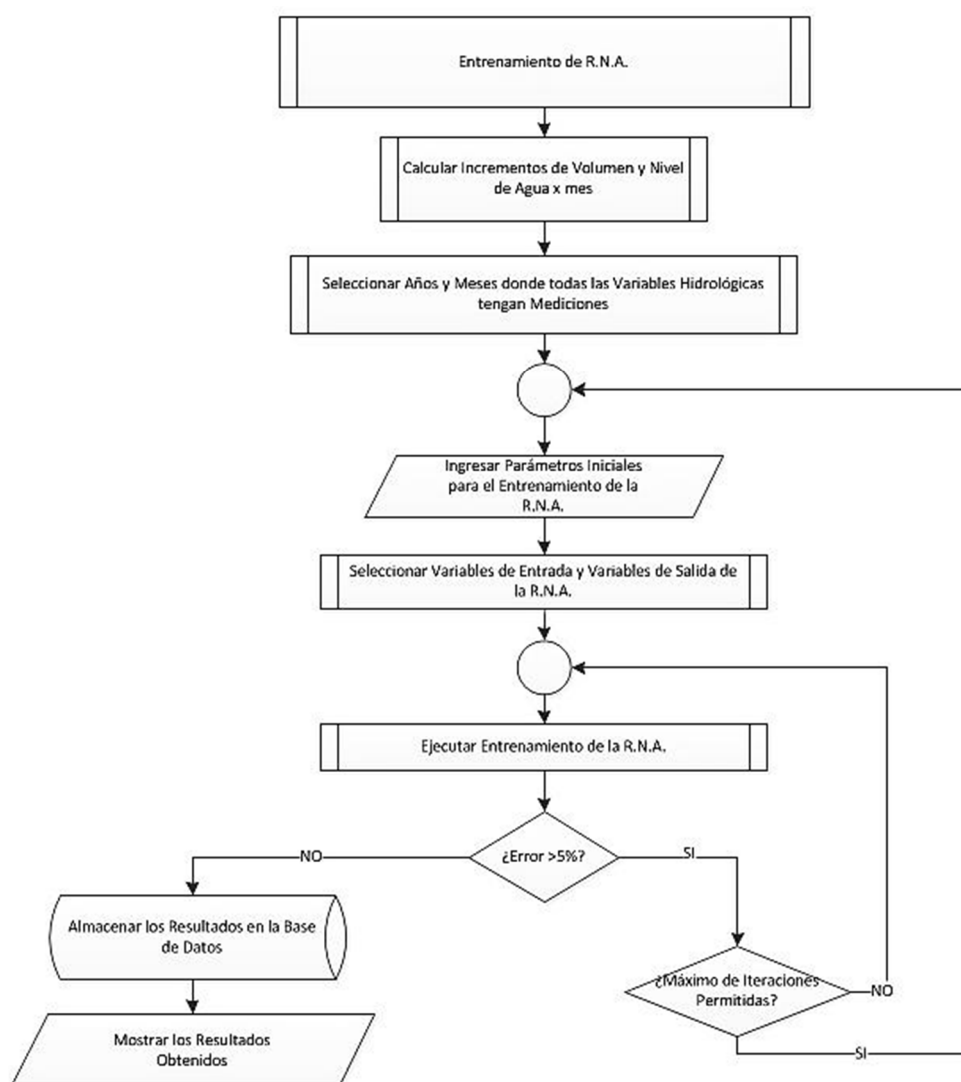


Figura 07. Diagrama de Flujo: Entrenamiento de RNA

Fuente: Propia

3.2.2.4. Simulación de Variables Hidrológicas.

El algoritmo para realizar la simulación, es el siguiente:

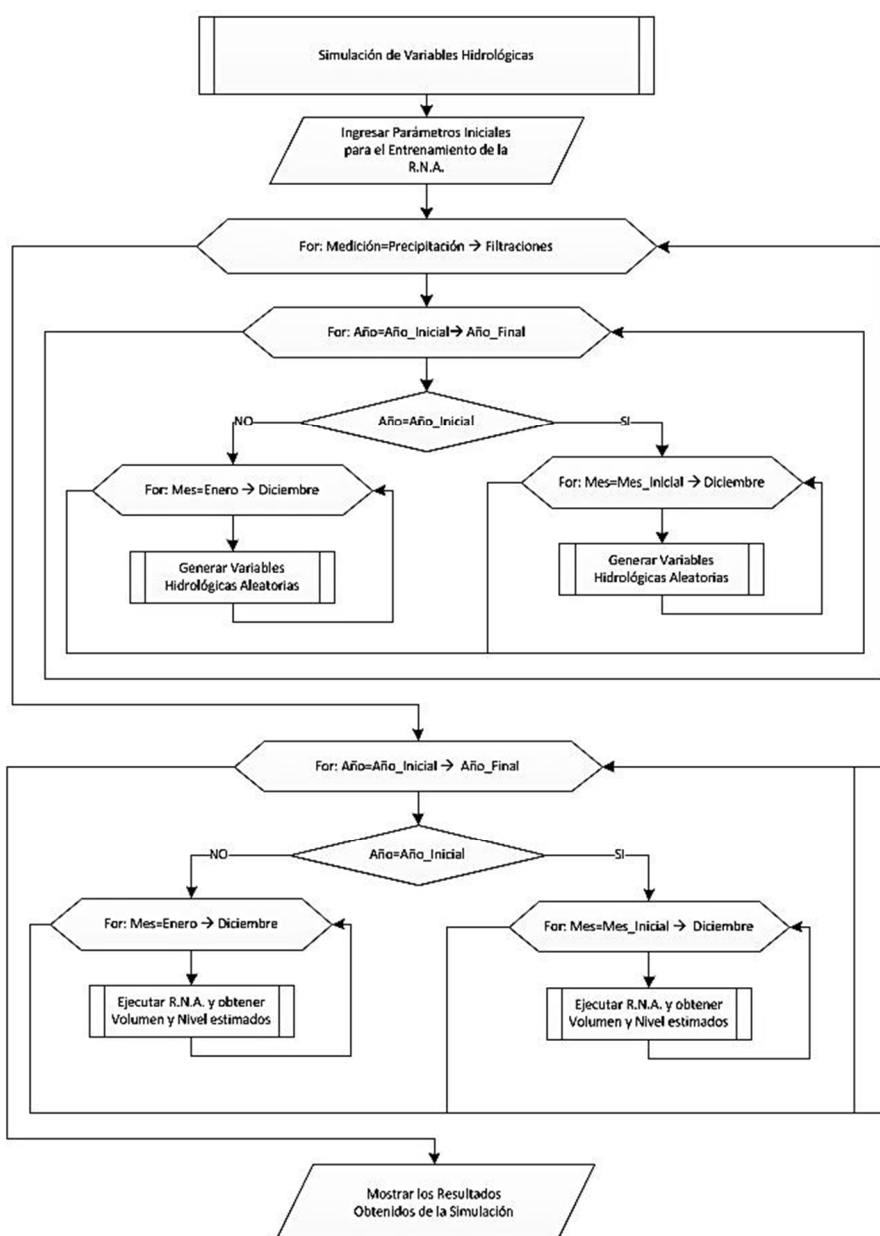


Figura 08. Diagrama de Flujo: Simulación de Variables Hidrológicas

Fuente: Propia

3.2.3. Modelamiento del Sistema.

3.2.3.1. Diagrama de Casos de Uso.

Los diagramas de caso de uso, nos dan una idea de como es que se va a diseñar el software, acerca de sus funciones, procesos, etc. El Diagrama de Caso de Uso General, sirve como base para el inicio del desarrollo del software de simulación.

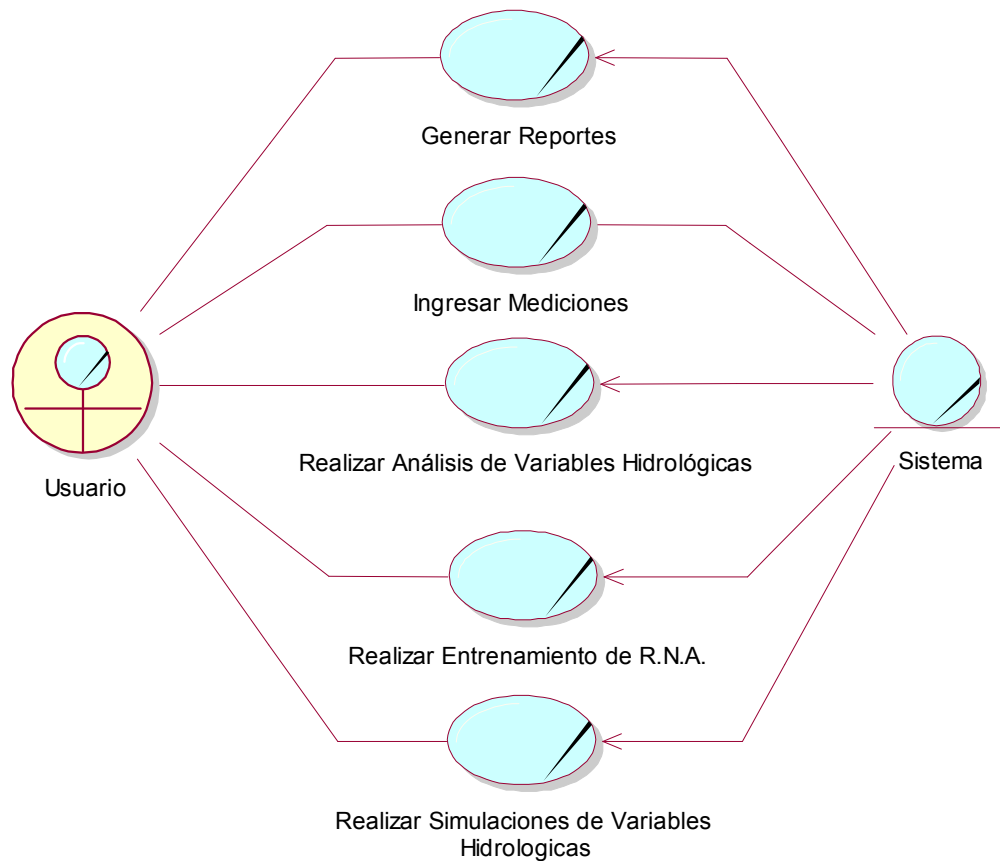


Figura 09. Casos de Uso: Funciones del Sistema

Fuente: Propia

(a) **Generar Reportes.** Se realiza el siguiente procedimiento al momento de generar los reportes que son necesarios.

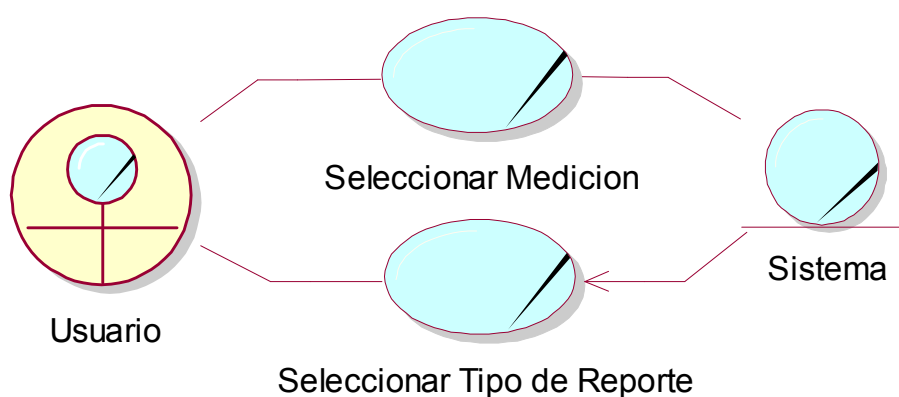


Figura 10. Caso de Uso: Generar Reportes.

Fuente: Propia

Tabla 01. Especificación de Caso de Uso: Generar Reportes

ÍTEM	DESCRIPCIÓN
Nombre CU_02:	Generar Reportes
Precondición:	Petición del Usuario para realizar reportes.
Postcondición:	Impresión de reportes al usuario.
Secuencia Normal:	<ol style="list-style-type: none"> 1 El usuario solicita reportes de las mediciones. 2 El usuario realiza las consultas necesarias. 3 El sistema carga los datos obtenidos luego de las consultas. 4 El sistema reporta las mediciones realizadas.
Excepciones:	No Existen Excepciones.

Fuente: Propia

(b) Ingresar Mediciones.

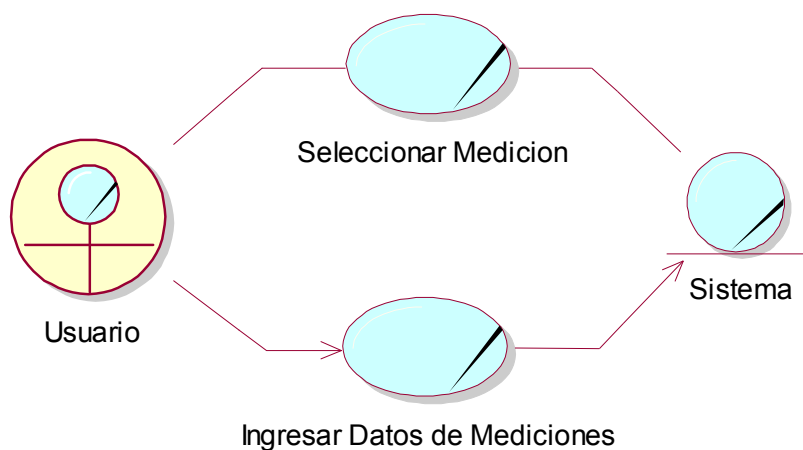


Figura 11. Caso de Uso: Ingresar Mediciones.

Fuente: Propia

Tabla 02. Especificación de Caso de Uso: Ingresar Mediciones

ÍTEM	DESCRIPCIÓN
Nombre CU_03:	Ingresar Mediciones.
Precondición:	Petición de Ingreso de Nuevas Mediciones
Postcondición:	El sistema de información estará actualizado.
Secuencia Normal:	<ol style="list-style-type: none">1 El Usuario solicita el ingreso de nuevas mediciones.2 El Usuario selecciona el tipo de medición que se ingresará.3 El Usuario ingresa las nuevas mediciones.4 CU_04.
Excepciones:	No Existen Excepciones.

Fuente: Propia

(c) Realizar Análisis de Variables Hidrológicas.

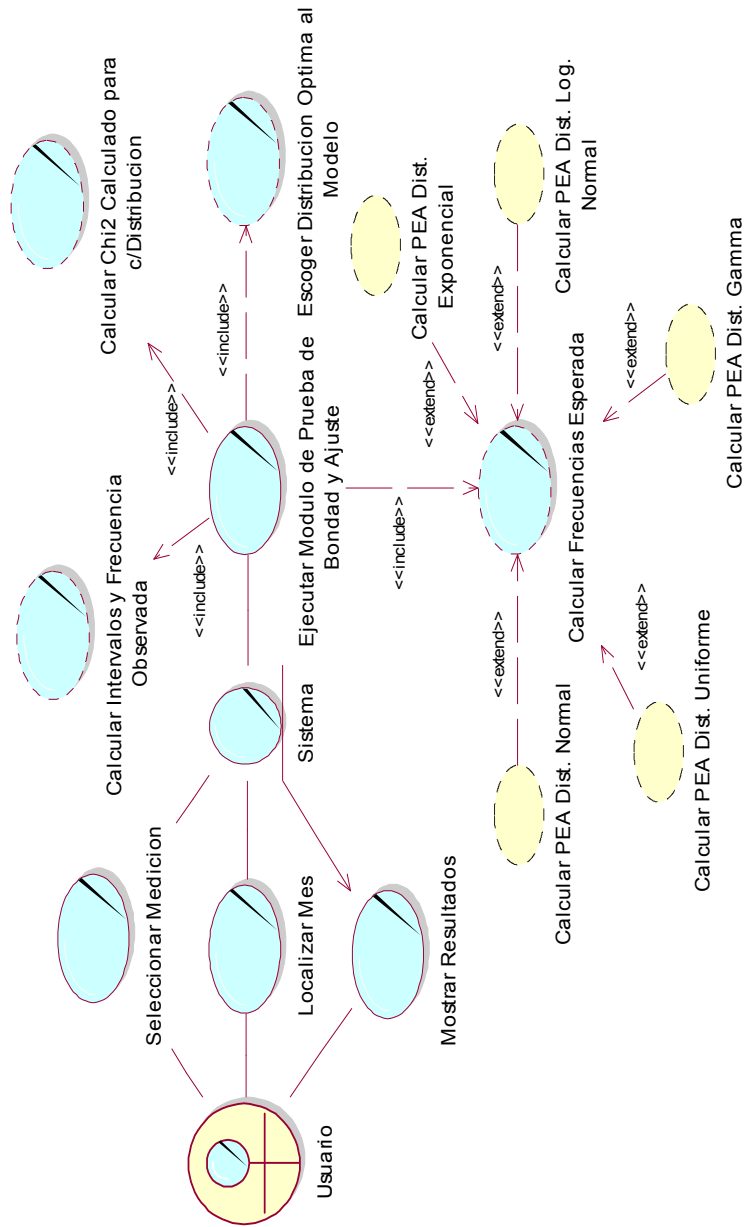


Figura 12. Caso de Uso: Realizar Análisis de Variables Hidrológicas

Fuente: Propia

Tabla 03. Especificación de Caso de Uso: Realizar Análisis de Variables Hidrológicas

ÍTEM	DESCRIPCIÓN
Nombre CU_04:	Realizar Análisis de Variables Hidrológicas.
Precondición:	Petición del Usuario para realizar el análisis de las variables hidrológicas.
Postcondición:	El sistema obtiene el Tipo de Distribución que siguen las variables hidrológicas.
Secuencia Normal:	
1	El usuario solicita actualizar el Análisis de las Variables Hidrológicas.
2	Selección del tipo de Análisis: Mensual o General.
3	El Sistema calcula el tipo de distribución que siguen las variables hidrológicas, haciendo uso de pruebas de Bondad de Ajuste.
4	El sistema reportas los resultados obtenidos, de las pruebas realizadas.
5	CU_05.
Excepciones:	
1	En caso que no haya datos, o todos tomen valores igual a 0, entonces se deduce que no sigue ningún tipo de distribución, o que no se realizará simulación de aquella variable hidrológica.

Fuente: Propia

- Diagrama de Secuencias

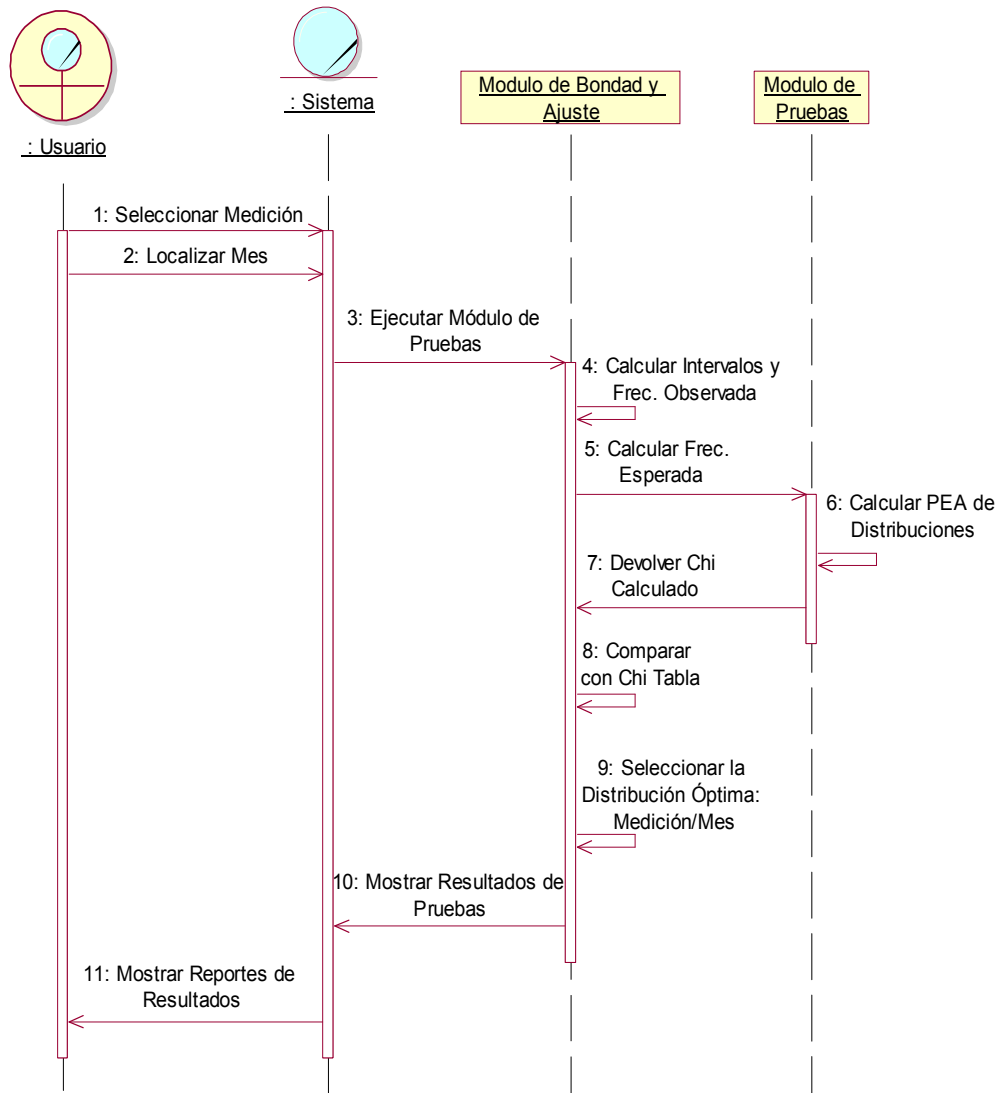


Figura 13. Diagrama de Secuencia: Realizar Prueba de Bondad y Ajuste.

Fuente: Propia

(d) Realizar Entrenamiento de RNA

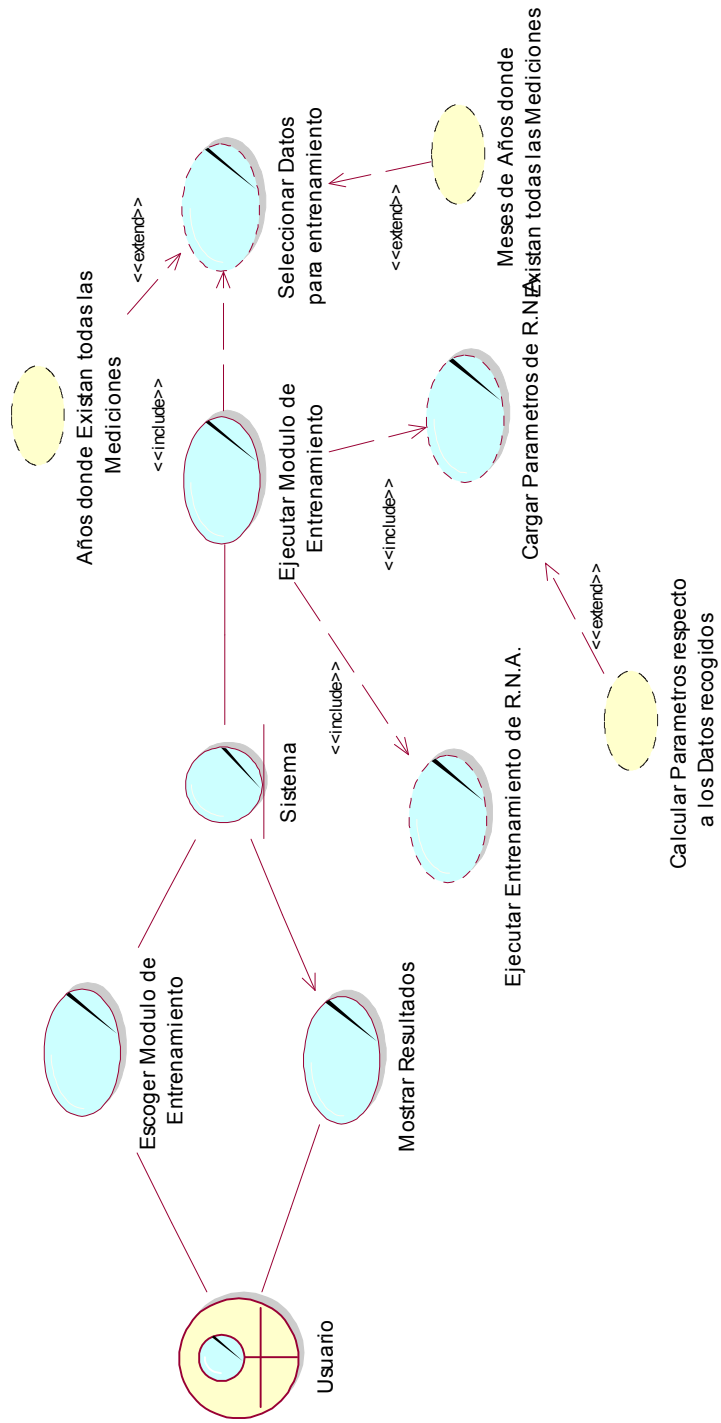


Figura 14. Caso de Uso: Realizar Entrenamiento de RNA

Fuente: Propia

Tabla 04. Especificación de Caso de Uso: Realizar Entrenamiento de RNA

ÍTEM	DESCRIPCIÓN
Nombre CU_05:	Realizar entrenamiento de RNA
Precondición:	Petición del Usuario solicita realizar el entrenamiento de la RNA, para así poder realizar la simulación respectiva.
Postcondición:	El Sistema obtiene una nueva RNA entrenada, lista para realizar la simulación de las variables hidrológicas.
Secuencia Normal:	<ol style="list-style-type: none"> 1 El Usuario solicita realizar el entrenamiento de la RNA 2 El Usuario selecciona los parámetros necesarios para el entrenamiento, y guarda los parámetros. 3 El Usuario ejecuta el entrenamiento, si los resultados no tienen un alto grado de confiabilidad entonces se vuelve a realizar el entrenamiento. 4 Si no se obtienen resultados favorables a pesar del entrenamiento entonces se regresa al Paso 2. 5 Se guarda la RNA entrenada, para su uso en la Simulación. 6 CU_06.
Excepciones:	<ol style="list-style-type: none"> 1 Si no se obtiene una solución aceptable, se realiza un nuevo entrenamiento, o se seleccionan nuevos parámetros para el entrenamiento de la RNA

Fuente: Propia

- **Diagrama de Secuencias**

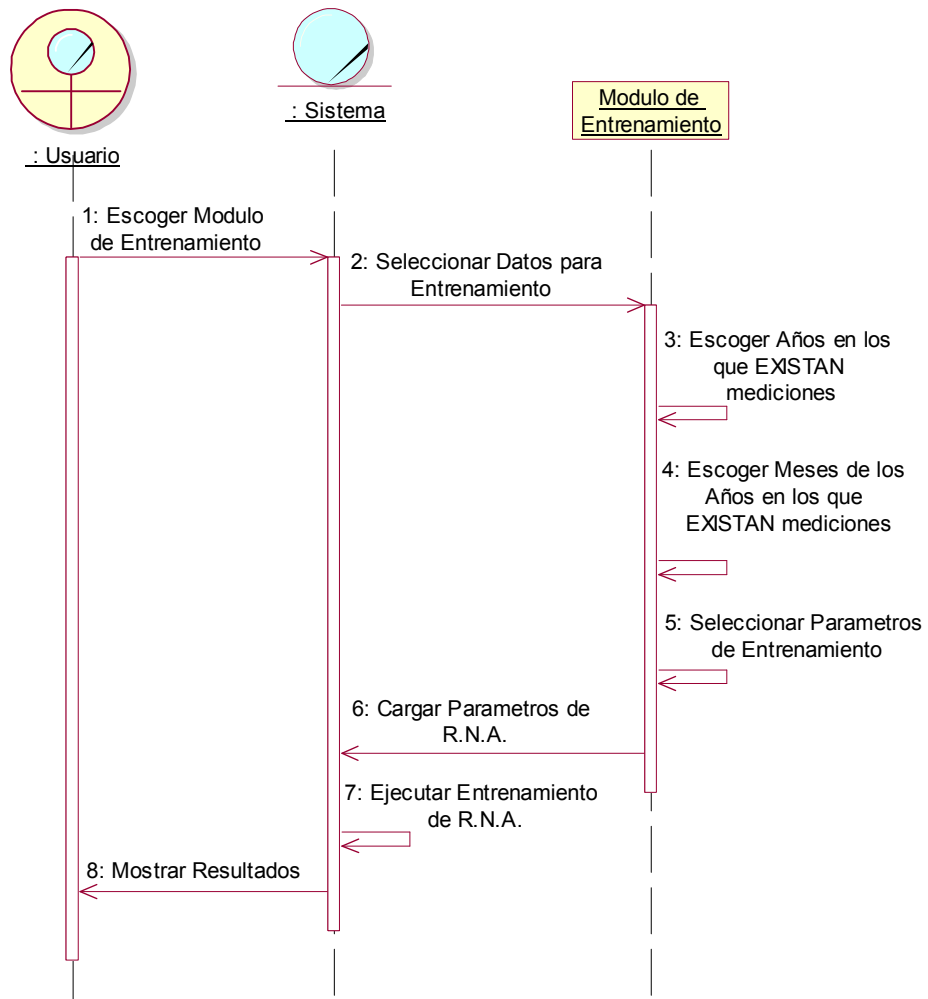


Figura 15. Diagrama de Secuencia: Realizar Entrenamiento de RNA

Fuente: Propia

(e) Realizar Simulaciones de Variables Hidrológicas.

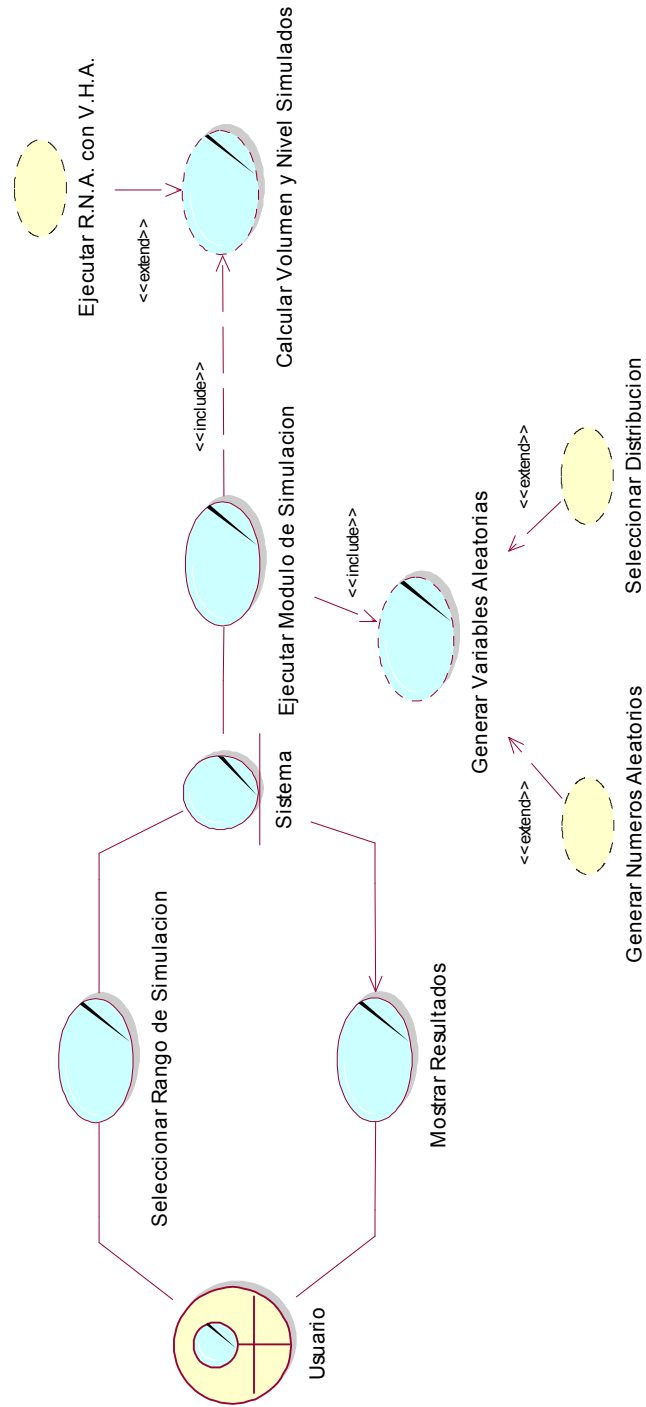


Figura 16. Caso de Uso: Realizar Simulaciones de Variables Hidrológicas.

Fuente: Propia

Tabla 05. Especificación de Caso de Uso: Realizar Simulaciones de las Variables Hidrológicas

ÍTEM	DESCRIPCIÓN
Nombre CU_06:	Realizar Simulación de las Variables Hidrológicas.
Precondición:	Petición del Usuario para realizar la Simulación de las Variables Hidrológicas.
Postcondición:	Valores obtenidos por medio de la Simulación.
Secuencia Normal:	<ol style="list-style-type: none"> 1 El Usuario solicita realizar una simulación de las variables hidrológicas. 2 El Usuario escoge el periodo de tiempo de la simulación. 3 El Usuario selecciona una de las RNA entrenadas previamente. 4 El Usuario ejecuta la Simulación de las Variables Hidrológicas. 5 El Sistema realiza todos los cálculos necesarios para la Simulación. 6 Se muestran los resultados obtenidos.
Excepciones:	No Existen excepciones.

Fuente: Propia

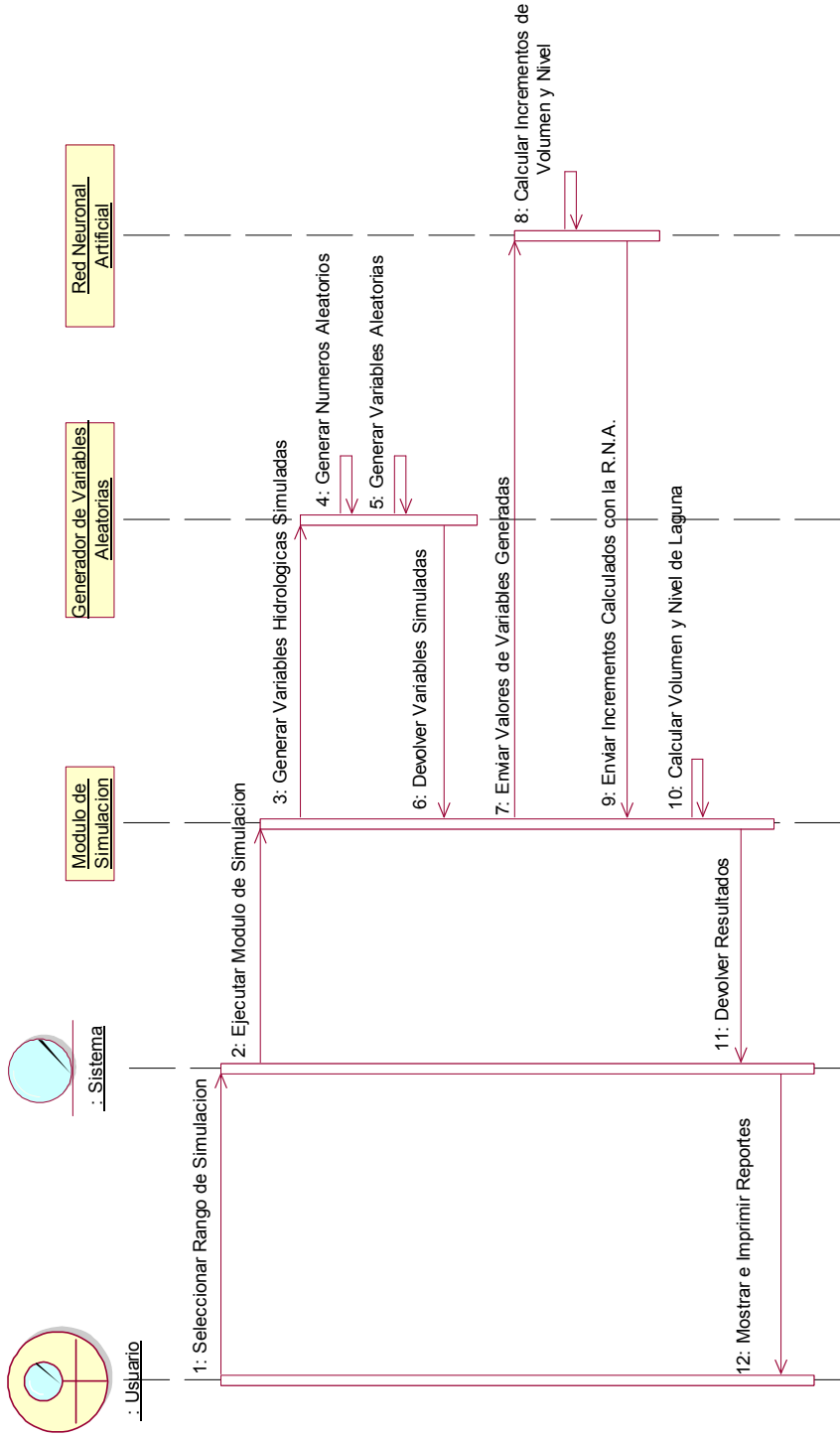


Figura 17. Diagrama de Secuencia: Realizar Simulación de Variables Hidrológicas.

Fuente: Propia

3.2.4. Analizar el Comportamiento de Variables Hidrológicas.

Se desarrolló un análisis del comportamiento de cada variable, dando como resultado el tipo de Distribución a la que pertenece, y a la cual se ajusta su *comportamiento*, usando para esto las pruebas de Bondad de Ajuste – Chi Cuadrado.

Tomándose como base las Distribuciones: Normal, Exponencial, Logarítmica Normal, Gamma y Uniforme; que son las que mejor se ajustan cuando se trata de datos hidrológicos.

Obteniendo el *óptimo* de las Distribuciones con menor Chi calculado y menor valor obtenido al comparar con la Tabla Chi Cuadrado, y se afirma que es el más cercano a lo esperado. Se observó casos en los cuales era necesario ajustar el Nivel de Significancia, para así obtener nuevos resultados que se ajusten a la Distribución de Probabilidad.

Se observó que cada variable hidrológica que interviene en el ciclo hidrológico de la laguna Aricota, posee un comportamiento distinto en el tiempo (meses), por lo que se realizó un análisis más detallado, no sólo por tipo de medición, sino también por mes.

Para la realización del Análisis de las Variables Hidrológicas se implementaron varias clases. *Ver Anexo 3.*

Para cada medición de las variables hidrológicas se siguieron los mismos pasos que se muestra en el ejemplo siguiente:

- **Pasos a seguir para realizar el análisis de las Mediciones de cada una de las Variables Hidrológicas.**

- **1er Paso.** Seleccionar la Variable Hidrológica a estudiar, y el mes en estudio.

Mediciones del mes de Enero de las Precipitaciones.

- **2do Paso.** Realizar un gráfico de barras, usando los datos de las mediciones del mes de enero.

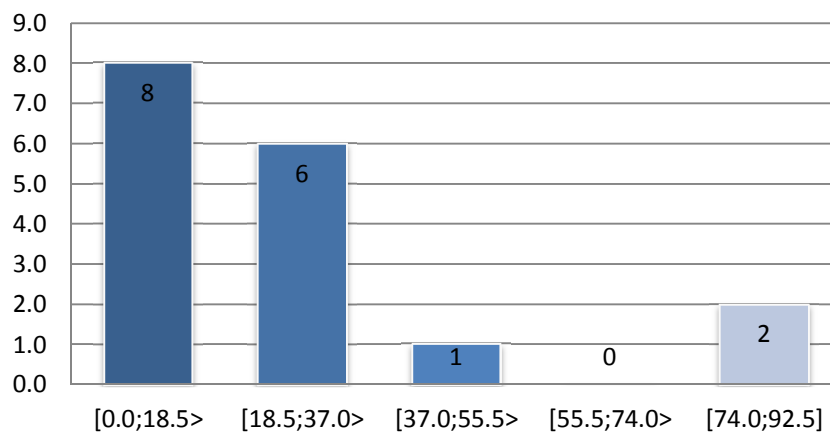


Figura 18. Comportamiento de las Precipitaciones en el Mes de Enero.

Fuente: Propia

- **3er Paso.** Plantear la Hipótesis Nula, de acuerdo al comportamiento que se observa en la gráfica anterior.

H₀: Los datos siguen una Distribución Exponencial.

- **4to Paso.** Realizar la Prueba de Bondad de Ajuste X^2 . Para los datos tomados anteriormente, se obtuvo el siguiente resultado, siguiendo el comportamiento visto previamente en la Gráfica 15:

Tabla 06. Prueba Bondad y Ajuste: Precipitación del Mes de Enero

INTERVALOS	F.O.	P.E.A	P.E.	F.E.	X^2
[0,0;18,5>	8	0,503	0,503	8,551	0,036
[18,5;37,0>	6	0,753	0,25	4,25	0,721
[37,0;55,5>	1	0,877	0,124	2,108	0,582
[55,5;74,0>	0	0,939	0,062	1,054	1,054
[74,0;92,5]	2	0,969	0,03	0,51	4,353
TOTAL	17		0,969	16	6,746

Fuente: Propia

Obteniendo un valor de Chi Calculado = 6,746.

- **5to Paso.** Calcular el valor del Chi según la Tabla. En este caso con un nivel de significancia de 0,05 y 4 grados de libertad.

$$Chi\ Tabla_{0,05;4} = 9,488$$

- **6to Paso.** Comparar el valor del Chi Tabla con el Chi Calculado, y analizar resultado.

$$Chi\ Tabla_{0,05,4} = 9,488 > 6,746$$

Al ser el Chi Tabla mayor que el Chi Calculado, se acepta la hipótesis nula, aceptando que en el mes de Enero las lluvias siguen una Distribución Exponencial.

3.3. Diseño del Software de Simulación

3.3.1. Diseño de la Base de Datos.

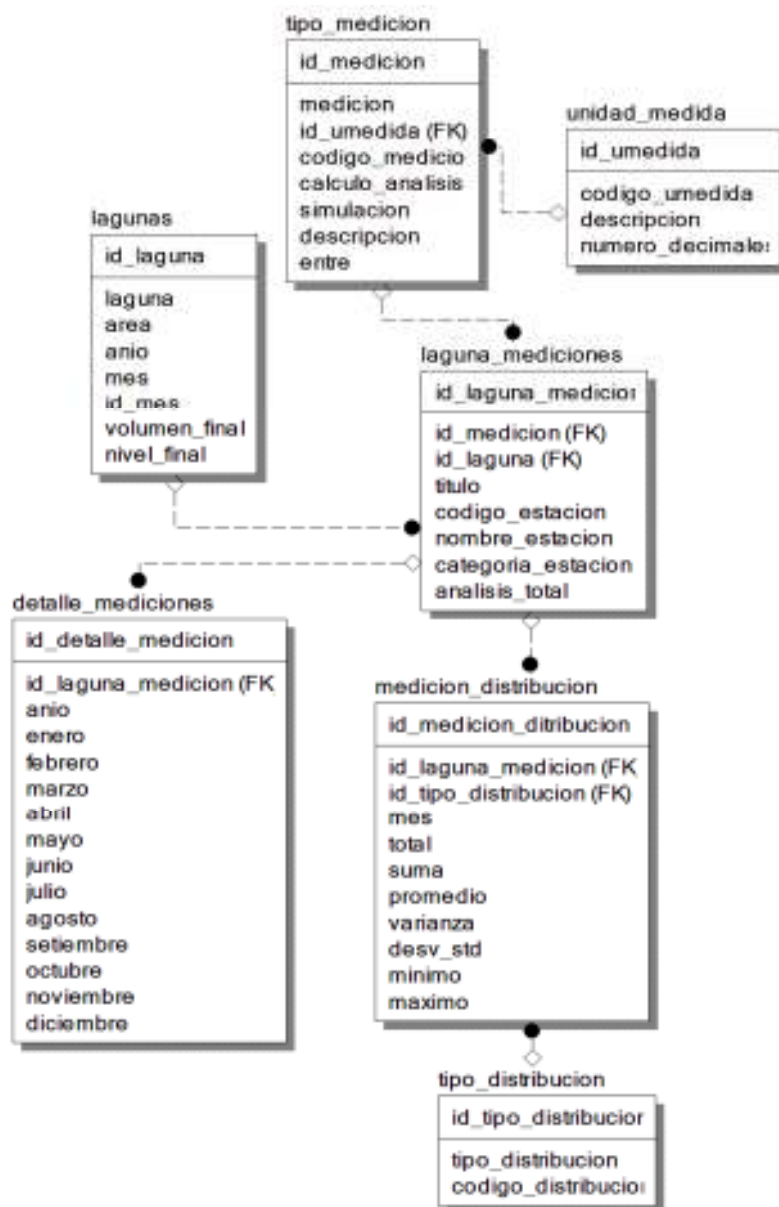


Figura 19. Diagrama Entidad/Relación.

Fuente: Propia

3.3.2. Diseño de la RNA

Se desarrolló un componente de una RNA, para lo cual se siguió la topología del *perceptrón multicapa*, haciendo uso del algoritmo de *Backpropagation*, y usando como función de activación la *función sigmoideal*.

La implementación del Componente *TRed_Neural*, se ve en el *Anexo 3.3*.

3.3.3. Entrenamiento de la RNA

Una vez desarrollado el entrenamiento de la RNA:

3.3.3.1. Calcular Incrementos del Volumen y Nivel de Espejo.

Antes de empezar con el entrenamiento de la RNA, es necesario calcular el incremento tanto del Volumen como del Nivel de Espejo, el cual empezará desde el mes siguiente:

- Si se desea calcular el incremento del mes de Enero, entonces tenemos que: $\Delta Vol_{i,j} = Vol_{i,j} - Vol_{i-1,12}$
- Si se desea calcular el incremento del volumen de un mes distinto a Enero, entonces tenemos que: $\Delta Vol_{i,j} = Vol_{i,j} - Vol_{i,j-1}$

Donde:

$i = \text{Año Inicial} \rightarrow \text{Año de Simulación}; y$

$j = \text{Recorrido de los Meses}(\text{Enero} \rightarrow \text{Diciembre})$

De la misma forma para el Incremento del Nivel de Espejo. Y almacenar dichos valores en la tabla “incremento_mediciones”.

Los valores de los incrementos del volumen y el nivel de espejo, son las salidas necesarias para realizar el entrenamiento de la RNA, por lo que estos valores se calculan al inicio y antes del ciclo de entrenamiento de la RNA.

3.3.3.2. Selección de Parámetros para el Entrenamiento de la RNA

Los parámetros necesarios para el entrenamiento de la RNA son los siguientes:

1. Valores de Entrada.

- **Número de Entradas.** En el análisis de las Variables Hidrológicas, se observó que son un total de 8 entradas para realizar el entrenamiento de la RNA, siendo estas las siguientes.
 - Mediciones de la Precipitación.
 - Mediciones de la Evaporación.
 - Mediciones del Caudal Bombeado.
 - Mediciones de las Descargas Rio Salado.
 - Mediciones de las Descargas Rio Callazas.
 - Mediciones de las Descargas Callazas y Salado.
 - Mediciones de las Filtraciones en Confluencia Esquina – Puquina.

- Mediciones de las Filtraciones en Confluencia CC.HH. Aricota N° 2.
- **Valor de Entrada Máximo.** Las mediciones de evaporación generan un pico demasiado alto, que al compararlo con los valores de las otras mediciones generan *ERRORES* en el entrenamiento de la RNA, por lo que se optó que en el caso de la evaporación se ha de dividir cada valor de las mediciones entre *100*, para que todas se encuentren en un mismo nivel, y no genere problemas con el entrenamiento, ni en la simulación. Resuelto esto, se realizó la elección de tomar como Valor Máximo de Entrada, el valor máximo de las mediciones, siendo indistinto si este pertenece a una medición de las precipitaciones, evaporación, caudal bombeado, descargas, filtraciones.
- **Valor de Entrada Mínimo.** Ya que no puede existir una medición (valor de entrada) con valor negativo, sin embargo existe la probabilidad de que este sea cero, se tomó como valor mínimo el mismo *cero*.

2. Valores de Salida.

- **Número de Salidas.** En el estudio de las Variables Hidrológicas, se observó que pueden tomarse un total de 2 salidas para el entrenamiento de la RNA, siendo estas las siguientes.

- Incremento del Volumen de Agua de la Laguna Aricota.
- Incremento del Nivel de Espejo de la Laguna Aricota.

Se tomó los incrementos de las mediciones de volumen y nivel de espejo, ya que para la simulación del Ciclo Hidrológico de la Laguna Aricota, tanto el volumen como el nivel se irán incrementando de forma positiva o negativamente (se realiza un incremento mensual).

- **Valor de Salida Máximo.** Habiéndose tomado únicamente los incrementos, y al verse que estos valores del volumen y nivel de espejo se encuentran en una escala similar, se tomó el valor máximo de estas.
- **Valor de Salida Mínimo.** Habiéndose tomado únicamente los incrementos, y al verse que estos valores del volumen y nivel de espejo se encuentran en una escala similar, se tomó el valor mínimo de estas, sin importar si es negativo, ya que el incremento puede ser tanto positivo como negativo.

3. Factores de Aprendizaje.

- **Momento.** Siendo una variable que sirve para controlar las oscilaciones, controlando los cambios en los pesos en un determinado tiempo, y logrando así la convergencia de la red en

un menor número de iteraciones, se tomó como valor de **momento = 0.9**.

- **Razón de Aprendizaje.** Para la razón de aprendizaje se seleccionó, a través de un proceso de ensayo y error, un valor de 0,1 (entre valores sugeridos que oscilan entre 0,05 y 0,75)
- **Margen de Error.** Ya que todo el proceso que se desarrolló para encontrar los mejores valores para el *Momento* y la *Razón de Aprendizaje*, se tomó aquellos cuyo *Error* sea el menor posible, buscando que este sea menor a un 5%.

4. Valores de Entrenamiento.

Antes de empezar el entrenamiento de la RNA es necesario seleccionar los datos que servirán para dicho entrenamiento, para esto se siguió los siguientes pasos de selección (intersección de años y mes con valores de medición):

- **Seleccionar Años.** Comenzar seleccionando los años en los cuales todas las Variables Hidrológicas tienen mediciones; ya que no en todos los años se han realizado mediciones por cada una de las Variables Hidrológicas.
- **Seleccionar Meses.** Hacer un recorrido de los años seleccionados anteriormente, para buscar en que meses de dicho año existen mediciones de las Variables Hidrológicas; ya que no

en todo el año se han realizado mediciones de las Variables Hidrológicas.

- Almacenar las Mediciones de los Años y Meses seleccionados.

3.3.3.3. Comenzando Entrenamiento.

Comenzar el entrenamiento de la RNA, y seguir con las iteraciones hasta que los valores obtenidos de salida sean lo más cercano a los ingresados (incrementos de volumen y nivel de agua), o un error menor al 5%. *Ver Anexo 3.4 y 3.5.*

3.3.4. Pasos para la Simulación.

Una vez desarrollado el entrenamiento de la RNA, *Ver Anexo 3.6:*

3.3.4.1. Seleccionar Mes y Año.

Al realizar una simulación, es necesario escoger el año y el mes al cual se quiere *predecir* el volumen y el nivel del espejo de la Laguna Aricota.

3.3.4.2. Carga de RNA previamente entrenada.

Se carga la RNA entrenada previamente, la cual ha debido ser guardada en un archivo con extensión “.rna”.

3.3.4.3. Generación de Variables Aleatorias.

Al haberse desarrollado el análisis de las variables hidrológicas, se obtuvo el tipo de distribución que sigue cada variable hidrológica, y

usando los parámetros correspondiente para cada tipo de distribución. Haciendo uso de la clase implementada. *Ver Anexo 3.2.*

3.3.4.4. Calcular los Incrementos usando la RNA.

Utilizando las variables hidrológicas generadas aleatoriamente, las cuales se han de utilizar como las variables de entrada para la RNA, la cual nos permitirá calcular los incrementos, tanto del volumen como del nivel de agua de la Laguna Aricota.

3.3.4.5. Calcular el Volumen y Nivel de Espejo.

Teniendo los incrementos del volumen y nivel de espejo del agua de la Laguna Aricota, desde un Año y Mes Inicial (ultima medición existente), hasta un Año y Mes de simulación. Se procede a calcular los volúmenes y niveles de agua, por cada mes, de la siguiente manera:

- Si se desea calcular el Volumen Simulado del mes de Enero, entonces tenemos que: $VolSimulado_{i,j} = Vol_{i-1,12} + \Delta Vol_{i,j}$
- Si se desea calcular el Volumen Simulado de un mes distinto a Enero, entonces tenemos que: $VolSimulado_{i,j} = Vol_{i,j-1} + \Delta Vol_{i,j}$

Donde:

$i = \text{Año Inicial} \rightarrow \text{Año de Simulación}; y$

$j = \text{Recorrido de los Meses}(\text{Enero} \rightarrow \text{Diciembre})$

De la misma forma para los cálculos del Nivel de Espejo que ha de ser simulado.

IV. RESULTADOS

4.1. Resultados obtenidos del Análisis de las Variables Hidrológicas.

Haciendo uso de la Prueba de Bondad y Ajuste, se realizó el análisis de las Variables Hidrológicas, y se obtuvo los siguientes resultados:

Tabla 07. Resultados: Variables Hidrológica / Tipo de Distribución

Variable Hidrológica	Mes	Distribución
Precipitación Total Mensual	Enero	Dist. Exponencial
	Febrero	Dist. Normal
	Marzo	Dist. Normal
	Abril	Dist. Exponencial
	Mayo	-
	Junio	-
	Julio	-
	Agosto	-
	Setiembre	-
	Octubre	-
	Noviembre	Dist. Gamma
	Diciembre	Dist. Exponencial
Descargas Medias Mensuales (Río Salado)	Enero	Dist. Normal
	Febrero	Dist. Exponencial
	Marzo	Dist. Exponencial
	Abril	Dist. Gamma
	Mayo	Dist. Normal
	Junio	Dist. Uniforme
	Julio	Dist. Normal
	Agosto	Dist. Normal
	Setiembre	Dist. Normal
	Octubre	Dist. Normal
	Noviembre	Dist. Normal
	Diciembre	Dist. Normal

(continúa)

Tabla 07. Resultados: Variables Hidrológica / Tipo de Distribución
(Continuación)

Variable Hidrológica	Mes	Distribución
Descargas Medias Mensuales (Río Callazas)	Enero	Dist. Exponencial
	Febrero	Dist. Exponencial
	Marzo	Dist. Exponencial
	Abril	Dist. Exponencial
	Mayo	Dist. Normal
	Junio	Dist. Normal
	Julio	Dist. Normal
	Agosto	Dist. Normal
	Setiembre	Dist. Normal
	Octubre	Dist. Normal
	Noviembre	Dist. Normal
	Diciembre	Dist. Normal
Descargas Medias Mensuales (Ríos Salado y Callazas)	Enero	Dist. Gamma
	Febrero	Dist. Exponencial
	Marzo	Dist. Exponencial
	Abril	Dist. Exponencial
	Mayo	Dist. Gamma
	Junio	Dist. Normal
	Julio	Dist. Normal
	Agosto	Dist. Normal
	Setiembre	Dist. Uniforme
	Octubre	Dist. Normal
	Noviembre	Dist. Uniforme
	Diciembre	Dist. Normal
Evaporación Total Mensual	Enero	Dist. Normal
	Febrero	Dist. Normal
	Marzo	Dist. Normal
	Abril	Dist. Normal
	Mayo	Dist. Normal
	Junio	Dist. Normal
	Julio	Dist. Normal
	Agosto	Dist. Uniforme
	Setiembre	Dist. Normal

(continúa)

**Tabla 07. Resultados: Variables Hidrológica / Tipo de Distribución
(Continuación)**

Variable Hidrológica	Mes	Distribución
Evaporación Total Mensual	Octubre	Dist. Normal
	Noviembre	Dist. Normal
	Diciembre	Dist. Normal
Caudal Bombeado de la Laguna Aricota	Enero	Dist. Normal
	Febrero	Dist. Normal
	Marzo	Dist. Uniforme
	Abril	Dist. Normal
	Mayo	Dist. Normal
	Junio	Dist. Normal
	Julio	Dist. Normal
	Agosto	Dist. Normal
	Setiembre	Dist. Normal
	Octubre	Dist. Normal
	Noviembre	Dist. Normal
	Diciembre	Dist. Normal
Filtraciones Laguna Aricota (Esquina - Puquina)	Total	Dist. Normal
Filtraciones Laguna Aricota (CC.HH. N° 2)	Total	Dist. Normal

Fuente: Propia

4.2. Resultados Obtenidos del Entrenamiento de la RNA

Para realizar el entrenamiento de la RNA, se escogió como valores de entrada las 8 variables hidrológicas (precipitación, evaporación, descargas, caudal bombeado, filtraciones), y 2 variables que representan la salida de la RNA (incremento del volumen y el nivel de espejo).

Se seleccionaron los datos minuciosamente, ya que no en todos los años y meses existen mediciones de todas las variables hidrológicas, por lo que se siguió un algoritmo para la selección de los siguientes datos:

Tabla 08. Resultados: Entrenamiento de la RNA

ENTRADAS								SALIDAS	
PRE	DES 1	DES 2	DES 3	EVA	CAU	FIL	FIL	NIV	VOL
0,0	0,697	0,587	1,305	1,716	1,568	0,439	0,300	-0,35	-1,96
0,0	0,809	0,763	1,728	1,984	1,215	1,145	0,300	-0,26	-1,75
0,0	1,673	1,426	3,249	1,573	1,043	1,268	0,408	0,13	0,91
0,0	1,187	1,432	2,869	1,300	1,072	1,240	0,400	-0,17	-1,19
0,0	1,226	1,736	3,562	1,554	1,293	1,313	0,360	-0,04	-0,28
14,8	0,593	0,750	1,703	1,675	1,331	0,962	0,410	-0,56	-3,81
20,9	1,707	1,008	2,938	1,292	1,348	1,108	0,400	-0,10	-0,68
0,0	0,526	0,915	1,691	1,553	1,361	1,030	0,400	-0,45	-3,04
0,0	0,636	0,806	1,693	1,274	1,389	0,933	0,345	-0,46	-3,07
0,0	0,483	0,801	1,445	1,987	1,433	0,810	0,285	-0,55	-3,63
0,0	0,383	0,609	1,083	2,168	1,635	0,795	0,280	-0,64	-4,17
5,2	1,420	0,523	2,140	1,519	1,160	0,638	0,250	-0,26	-1,70
0,0	0,673	0,710	1,536	1,610	1,170	0,611	0,250	-0,26	-1,70
0,0	0,468	0,850	1,483	1,422	1,171	0,655	0,250	-0,29	-1,89
0,0	0,631	0,816	1,617	1,199	1,170	0,719	0,250	-0,30	-1,95
0,0	0,749	0,891	1,832	1,262	1,484	0,641	0,250	-0,29	-1,88
0,0	0,902	0,877	1,966	1,588	1,477	0,705	0,250	-0,33	-2,13
0,0	0,852	0,556	1,591	2,009	1,492	0,678	0,250	-0,41	-2,63
0,0	0,459	0,636	1,280	2,231	1,458	0,664	0,250	-0,60	-3,83
0,0	0,481	0,577	1,224	2,125	1,483	0,798	0,250	-0,61	-3,88
0,0	0,377	0,596	1,123	2,212	1,483	0,752	0,250	-0,67	-4,23
52,8	6,380	3,677	10,227	1,625	0,291	0,653	0,250	3,44	21,66
0,0	1,237	0,785	2,191	1,305	1,031	0,935	0,250	-0,13	-0,86
0,0	1,000	0,810	1,894	1,398	1,490	0,640	0,190	-0,32	-2,04
17,0	1,542	1,780	4,042	1,635	1,420	0,625	0,190	-0,09	-0,56
0,0	0,655	0,728	1,814	1,565	1,440	0,640	0,190	-0,46	-2,81
0,0	0,628	0,711	1,779	1,746	1,400	0,631	0,150	-0,39	-2,37
0,0	0,647	0,690	1,687	1,787	1,410	0,582	0,150	-0,40	-2,41

Fuente: Propia

Luego de haber realizado un total de 237 iteraciones se obtuvo el siguiente resultado:

Tabla 09. Resultados: Margen de Error del entrenamiento

DESEADO		RESULTADO		ERROR	
NIV (a)	VOL (b)	NIV (c)	VOL (d)	a - c	b - d
-0,35	-1,96	-0,33	-1,96	0,02	-0.01
-0,26	-1,75	-0,22	-1,77	0,04	-0.01
0,13	0,91	0,11	0,95	0,02	-0.04
-0,17	-1,19	-0,14	-1,14	0,03	0.05
-0,04	-0,28	-0,04	-0,32	0,00	-0.04
-0,56	-3,81	-0,57	-3,80	-0,01	0.02
-0,10	-0,68	-0,07	-0,70	0,03	-0.03
-0,45	-3,04	-0,46	-3,13	-0,01	-0.09
-0,46	-3,07	-0,46	-3,00	0,00	0.07
-0,55	-3,63	-0,53	-3,48	0,03	0.15
-0,64	-4,17	-0,73	-4,39	-0,09	-0.22
-0,26	-1,70	-0,27	-1,68	-0,01	0.02
-0,26	-1,70	-0,27	-1,74	-0,01	-0.04
-0,29	-1,89	-0,31	-2,00	-0,02	-0.12
-0,30	-1,95	-0,28	-1,81	0,02	0.13
-0,29	-1,88	-0,32	-2,00	-0,03	-0.12
-0,33	-2,13	-0,33	-2,20	0,00	-0.07
-0,41	-2,63	-0,42	-2,72	-0,01	-0.09
-0,60	-3,83	-0,57	-3,63	0,03	0.21
-0,61	-3,88	-0,61	-3,88	0,00	0.00
-0,67	-4,23	-0,66	-4,08	0,02	0.15
3,44	21,66	3,44	21,67	0,00	-0.01
-0,13	-0,86	-0,15	-0,94	-0,02	-0.09
-0,32	-2,04	-0,30	-1,90	0,02	0.15
-0,09	-0,56	-0,10	-0,56	-0,01	0.00
-0,46	-2,81	-0,39	-2,48	0,07	0.33
-0,39	-2,37	-0,41	-2,59	-0,02	-0.22
-0,40	-2,41	-0,40	-2,54	0,00	-0.13
		Máximo Error		0,066	0,328
		Mínimo Error		-0,087	-0,223

Fuente: Propia

Como se observa, tanto en el incremento del nivel como del volumen, la diferencia entre los valores deseados y los obtenidos como resultados es mínima, por lo que se afirma que el entrenamiento de la RNA ha sido un éxito.

4.3. Resultados Obtenidos de Validar el Software de Simulación.

Se realizó una Simulación de Prueba, siendo esta desde el año 1970 hasta el año 1980, para verificar la diferencia de lo simulado con lo real, teniendo los siguientes resultados:

4.3.1. Variables Hidrológicas Generadas Aleatoriamente.

Tabla 10. Resultados: Variable Aleatoria – Precipitación (mm)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	7,6	80,9	30,9	1,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	15,0
1971	13,7	56,7	6,2	2,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	10,2
1972	95,5	81,4	59,5	0,3	0,0	0,0	0,0	0,0	0,0	0,0	2,2	3,0
1973	48,1	24,9	24,4	0,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,1
1974	2,5	1,7	29,4	2,4	0,0	0,0	0,0	0,0	0,0	0,0	0,2	4,6
1975	29,4	35,9	22,2	2,4	0,0	0,0	0,0	0,0	0,0	0,0	0,2	15,7
1976	51,8	61,9	13,5	1,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	23,7
1977	16,1	2,0	35,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	3,9
1978	11,8	56,6	18,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0	1,3	8,9
1979	26,7	27,7	11,3	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	6,9
1980	30,3	36,3	43,0	0,8	0,0	0,0	0,0	0,0	0,0	0,0	0,2	1,1

Fuente: Propia

Tabla 11. Resultados: Variable Aleatoria – Evaporación (mm)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	252,2	84,8	179,3	162,0	181,0	135,2	154,8	178,7	166,6	218,2	271,2	210,0
1971	181,7	151,9	167,2	165,0	167,2	157,2	137,9	150,7	224,7	236,7	214,5	165,0
1972	133,2	134,0	192,8	185,8	160,4	125,6	139,7	174,0	186,2	243,6	189,8	154,5
1973	126,5	157,0	166,2	142,8	168,6	150,6	126,6	156,1	224,8	205,0	207,5	164,4
1974	177,8	109,3	146,9	175,3	148,7	143,2	149,2	166,5	203,2	202,8	169,4	199,2
1975	127,0	144,9	160,1	144,8	157,3	113,1	149,5	156,7	178,3	195,7	205,4	181,6
1976	118,8	89,5	137,7	127,7	144,4	112,8	149,6	170,2	164,9	212,3	141,2	172,7
1977	101,3	84,9	157,1	157,9	128,3	133,8	146,5	178,8	217,3	224,4	140,4	144,6
1978	119,1	145,6	109,7	154,7	162,9	121,2	165,2	176,5	223,7	223,4	185,5	203,8
1979	170,7	75,3	117,0	157,6	146,1	147,5	158,4	163,3	203,9	209,5	182,1	189,4
1980	91,7	93,7	150,4	149,2	144,3	143,7	114,3	158,0	178,5	228,2	180,7	243,4

Fuente: Propia

Tabla 12. Resultados: Variable Aleatoria – Caudal Bombeado (m3/s)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	0,89	0,50	0,76	1,61	0,51	3,65	1,10	0,66	1,80	0,44	3,22	0,90
1971	2,10	1,07	1,54	1,91	1,59	0,99	0,98	2,38	1,41	1,66	2,05	2,49
1972	3,70	0,98	0,16	0,82	0,63	2,15	1,25	3,13	1,53	1,58	1,68	1,13
1973	2,25	1,90	1,13	2,36	0,67	1,48	1,25	0,48	1,63	0,06	3,60	1,95
1974	0,25	0,40	1,37	2,18	0,56	2,24	2,17	1,84	2,71	1,06	1,85	2,55
1975	0,89	1,94	0,50	1,41	1,32	2,48	0,09	2,13	2,27	1,89	0,02	1,17
1976	1,87	2,31	0,76	1,33	0,77	1,42	1,81	0,76	1,14	1,33	1,28	1,72
1977	1,36	2,30	2,83	1,49	2,22	2,55	0,45	2,45	0,68	1,43	2,42	2,33
1978	1,76	0,42	0,49	0,66	0,85	2,32	1,44	3,08	1,32	1,58	2,45	1,04
1979	2,75	1,21	0,34	1,98	1,15	0,53	1,69	1,60	1,54	2,47	0,31	1,94
1980	1,83	1,20	2,24	0,93	0,65	1,80	0,10	0,88	2,57	0,13	1,62	2,07

Fuente: Propia

**Tabla 13. Resultados: Variable Aleatoria – Descargas del Río
Callazas (m3/s)**

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	0,64	2,71	1,22	0,39	0,73	0,67	1,09	0,11	0,58	0,51	0,42	0,75
1971	3,20	5,40	0,40	0,44	0,64	0,50	1,41	0,21	0,47	0,48	0,45	0,62
1972	4,09	0,64	1,81	2,73	0,73	0,86	0,27	0,64	0,38	0,60	0,54	0,67
1973	0,59	2,41	4,49	0,31	0,53	0,71	1,25	0,73	0,66	0,76	0,26	0,54
1974	2,34	4,13	1,08	2,18	1,14	0,56	1,13	0,15	0,77	0,35	0,63	0,65
1975	0,71	0,65	3,83	0,14	0,64	0,65	0,29	0,96	0,32	0,55	0,53	0,36
1976	2,65	0,37	1,80	1,20	0,70	0,63	0,71	0,80	0,53	0,38	0,58	0,65
1977	4,39	1,17	0,75	0,27	0,90	0,85	0,07	0,71	0,79	0,50	0,63	0,97
1978	1,03	6,01	0,91	0,58	0,53	0,73	0,26	0,54	0,67	0,53	0,61	0,42
1979	0,55	2,06	2,82	0,13	1,24	0,81	1,05	1,02	0,81	0,53	0,68	0,43
1980	0,13	0,01	2,77	0,73	0,56	0,72	0,59	0,83	0,56	0,66	0,50	0,45

Fuente: Propia

**Tabla 14. Resultados: Variable Aleatoria – Descargas del Río
Salado (m3/s)**

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	2,37	6,47	0,04	1,98	0,97	0,57	0,68	1,14	0,66	0,53	0,44	1,08
1971	3,57	6,22	3,22	0,09	0,89	0,62	0,69	0,07	0,44	0,48	0,33	0,31
1972	0,66	0,23	0,57	0,01	0,70	1,05	0,34	0,99	0,81	0,64	0,35	0,68
1973	2,11	0,43	0,49	0,10	0,90	1,14	0,10	0,63	0,78	0,84	0,19	0,94
1974	0,76	3,11	0,05	0,28	1,27	1,00	0,36	0,99	0,44	0,51	0,59	0,79
1975	0,09	0,54	1,66	1,75	0,96	0,67	0,08	0,89	0,09	0,62	0,64	0,47
1976	0,50	1,03	4,68	1,55	0,84	0,31	0,33	0,94	0,56	0,40	0,37	0,24
1977	1,98	2,41	6,86	1,11	0,83	0,53	0,98	0,52	0,84	0,70	0,57	0,42
1978	2,31	5,46	1,18	1,01	0,80	0,54	1,16	0,84	0,63	0,62	0,56	0,53
1979	1,13	0,87	3,01	0,90	0,98	0,29	0,91	0,86	0,73	0,35	0,61	0,74
1980	3,32	2,30	0,88	0,21	0,94	0,67	0,93	0,50	0,59	0,30	0,57	0,97

Fuente: Propia

Tabla 15. Resultados: Variable Aleatoria – Descargas del Río Callazas y Salado (m3/s)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	2,64	17,11	0,24	0,75	0,12	2,05	1,85	1,57	1,18	0,73	1,14	1,55
1971	1,71	9,40	7,15	0,37	0,04	1,97	1,11	1,37	1,32	1,73	1,11	1,96
1972	1,73	5,17	7,59	3,77	0,07	1,30	2,50	1,73	1,20	1,75	1,34	2,20
1973	6,47	1,80	0,20	2,87	0,01	1,28	0,64	1,49	1,46	1,08	1,40	2,24
1974	3,55	4,85	0,78	0,38	0,01	1,27	1,35	1,69	1,40	1,48	1,44	1,27
1975	1,34	0,60	7,22	0,83	0,04	1,33	1,03	1,39	1,33	0,85	0,75	1,13
1976	2,26	1,26	6,72	1,10	0,05	0,66	1,50	0,79	1,59	0,75	1,09	1,39
1977	2,83	8,43	0,37	4,43	0,05	1,58	1,60	1,08	0,99	1,22	0,82	1,85
1978	3,86	5,74	4,15	2,30	0,04	0,67	2,42	0,93	1,19	0,59	0,72	2,28
1979	1,92	0,89	1,97	1,41	0,03	1,46	0,38	1,61	1,01	0,94	1,25	0,54
1980	1,79	28,60	0,74	14,50	0,04	1,97	1,23	1,33	0,90	1,13	0,91	0,73

Fuente: Propia

Tabla 16. Resultados: Variable Aleatoria – Filtraciones Esquina – Puquina (m3/s)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	0,65	0,88	0,40	0,70	0,55	0,70	0,65	0,74	0,53	1,16	0,80	0,87
1971	0,66	0,65	0,88	0,83	0,94	1,08	0,38	0,85	1,00	0,51	0,70	0,67
1972	1,16	1,15	1,00	0,96	0,52	0,82	0,60	0,61	0,95	0,73	0,87	0,57
1973	0,61	0,58	0,53	0,80	0,73	0,17	0,95	0,83	0,53	1,75	0,65	0,65
1974	1,21	0,72	0,86	0,78	0,87	0,89	0,82	1,09	0,91	0,34	0,74	1,25
1975	1,35	0,60	0,78	0,53	0,74	0,83	0,85	1,09	0,40	0,54	1,26	0,24
1976	0,84	0,76	0,61	0,34	0,60	0,86	0,97	0,74	0,66	0,63	0,96	1,12
1977	1,00	0,65	0,82	0,46	0,59	0,62	0,89	1,06	0,83	0,68	0,85	0,32
1978	0,89	0,81	0,61	1,04	0,92	0,68	0,84	0,83	0,78	0,57	0,49	0,54
1979	0,63	0,68	0,41	1,13	0,60	0,53	0,22	1,25	0,94	0,85	0,84	0,11
1980	1,10	0,68	0,91	0,98	1,06	0,65	0,52	1,17	0,68	1,18	0,46	0,83

Fuente: Propia

Tabla 17. Resultados: Variables Aleatoria – Filtraciones CC.HH.

Aricota N° 2 (m3/s)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	0,30	0,24	0,15	0,19	0,33	0,32	0,26	0,19	0,25	0,17	0,17	0,28
1971	0,38	0,37	0,34	0,26	0,27	0,25	0,31	0,30	0,23	0,26	0,22	0,24
1972	0,28	0,15	0,18	0,24	0,45	0,28	0,30	0,23	0,39	0,25	0,26	0,28
1973	0,40	0,28	0,32	0,34	0,26	0,37	0,29	0,27	0,44	0,20	0,29	0,18
1974	0,33	0,13	0,28	0,16	0,26	0,36	0,25	0,20	0,33	0,20	0,33	0,30
1975	0,30	0,19	0,17	0,26	0,19	0,21	0,31	0,18	0,32	0,20	0,37	0,25
1976	0,24	0,15	0,23	0,19	0,32	0,21	0,29	0,38	0,40	0,23	0,25	0,19
1977	0,24	0,14	0,41	0,15	0,38	0,22	0,23	0,32	0,16	0,20	0,15	0,42
1978	0,30	0,18	0,18	0,30	0,36	0,26	0,26	0,40	0,21	0,30	0,19	0,24
1979	0,29	0,26	0,15	0,14	0,27	0,31	0,16	0,12	0,26	0,29	0,02	0,24
1980	0,13	0,19	0,23	0,46	0,31	0,33	0,30	0,28	0,14	0,26	0,29	0,30

Fuente: Propia

4.3.2. Variables de Salida obtenidas con la RNA

Al haber obtenido los resultados de las variables hidrológicas generadas aleatoriamente, y siguiendo cada una su propia distribución de probabilidad (obtenida mediante las pruebas de bondad de ajuste realizada a cada una y por mes), se utilizaron para el calculo de los incrementos del volumen y del nivel de espejo de agua, utilizando la RNA previamente entrenada.

Obteniendo los siguientes resultados, con un nivel de confiabilidad del 95%.

Tabla 18. Resultados: Mediante Entrenamiento – Incremento de Volumen (MMC)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	2,48	23,99	7,55	-2,59	1,26	-5,81	-1,31	0,17	-3,17	-2,07	-6,14	-2,84
1971	-1,84	19,47	12,79	-5,47	-4,50	-1,48	4,30	-5,49	-3,60	-3,33	-5,01	-5,69
1972	-4,06	-2,09	4,84	-1,66	2,21	-3,97	-1,50	-5,17	-3,51	-3,02	-4,75	-1,86
1973	0,73	-0,06	14,44	-4,21	-1,44	4,33	-3,74	-0,60	-2,78	3,06	-6,28	-3,55
1974	1,25	-3,85	1,60	-4,69	-0,34	-4,49	-4,78	-3,32	-5,60	-1,12	-4,02	-5,76
1975	1,30	1,88	7,62	-1,58	-2,84	-4,93	-0,93	-4,71	-4,29	-3,98	-0,12	0,76
1976	-2,11	-2,67	4,51	3,94	0,88	-3,79	-4,48	-1,01	-1,89	-3,79	-3,85	-1,91
1977	0,70	1,97	-3,23	2,16	-1,66	-4,50	0,36	-5,54	-1,76	-3,41	-5,36	-3,41
1978	-1,03	11,36	3,71	0,77	-2,96	-4,10	-0,82	-5,77	-3,39	-3,82	-4,79	-3,08
1979	-2,80	4,58	7,81	-4,48	1,13	0,70	4,76	-3,57	-4,01	-5,66	-1,01	-0,46
1980	-1,54	24,39	-0,66	22,78	-2,68	-2,98	4,55	-2,65	-5,36	-0,28	-2,79	-5,01

Fuente: Propia

Tabla 19. Resultados: Mediante Entrenamiento – Volumen (MMC)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	680,53	704,52	712,07	709,48	710,74	704,93	703,62	703,79	700,62	698,55	692,41	689,57
1971	687,73	707,2	719,99	714,52	710,02	708,54	712,84	707,35	703,75	700,42	695,41	689,72
1972	685,66	683,57	688,41	686,75	688,96	684,99	683,49	678,32	674,81	671,79	667,04	665,18
1973	665,91	665,85	680,29	676,08	674,64	678,97	675,23	674,63	671,85	674,91	668,63	665,08
1974	666,33	662,48	664,08	659,39	659,05	654,56	649,78	646,46	640,86	639,74	635,72	629,96
1975	631,26	633,14	640,76	639,18	636,34	631,41	630,48	625,77	621,48	617,5	617,38	618,14
1976	616,03	613,36	617,87	621,81	622,69	618,9	614,42	613,41	611,52	607,73	603,88	601,97
1977	602,67	604,64	601,41	603,57	601,91	597,41	597,77	592,23	590,47	587,06	581,7	578,29
1978	577,26	588,62	592,33	593,1	590,14	586,04	585,22	579,45	576,06	572,24	567,45	564,37
1979	561,57	566,15	573,96	569,48	570,61	571,31	576,07	572,5	568,49	562,83	561,82	561,36
1980	559,82	584,21	583,55	606,33	603,65	600,67	605,22	602,57	597,21	596,93	594,14	589,13

Fuente: Propia

Tabla 20. Resultados: Mediante Entrenamiento – Incremento del Nivel de Espejo (msnm)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	0,29	6,08	0,39	-0,36	0,16	-1,26	-0,19	-0,04	-0,52	-0,24	-1,36	-0,32
1971	-0,10	1,94	0,92	-1,05	-0,71	-0,24	0,43	-1,13	-0,56	-0,55	-0,92	-1,19
1972	-2,69	-0,72	1,09	-0,17	0,24	-0,64	-0,33	-0,99	-0,55	-0,46	-0,84	-0,31
1973	0,14	-0,29	0,90	-0,81	-0,17	0,35	-0,55	-0,09	-0,42	0,42	-1,50	-0,61
1974	0,21	-0,49	-0,08	-0,70	0,04	-0,77	-0,82	-0,54	-1,10	-0,23	-0,67	-1,13
1975	0,14	-0,25	0,77	-0,26	-0,39	-0,91	-0,17	-0,78	-0,81	-0,66	0,04	0,03
1976	-1,35	-1,42	0,34	0,40	0,10	-0,61	-0,77	-0,11	-0,31	-0,61	-0,62	-0,22
1977	0,08	-0,16	-0,48	0,03	-0,25	-0,81	-0,03	-1,06	-0,20	-0,53	-1,02	-0,59
1978	-0,19	0,22	0,33	0,05	-0,40	-0,70	-0,19	-1,18	-0,50	-0,60	-0,86	-0,50
1979	-0,52	-0,01	0,90	-0,78	0,16	0,03	0,44	-0,50	-0,60	-1,13	-0,12	-0,13
1980	-0,21	4,18	-0,99	2,29	-0,33	-0,49	0,42	-0,35	-1,03	0,00	-0,46	-0,87

Fuente: Propia

Tabla 21. Resultados: Mediante Entrenamiento – Nivel de Espejo (msnm)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1970	2 831,3	2 837,4	2 837,8	2 837,4	2 837,6	2 836,3	2 836,1	2 836,1	2 835,6	2 835,3	2 834,0	2 833,6
1971	2 833,5	2 835,5	2 836,4	2 835,3	2 834,6	2 834,4	2 834,8	2 833,7	2 833,1	2 832,6	2 831,7	2 830,5
1972	2 827,8	2 827,1	2 828,2	2 828,0	2 828,2	2 827,6	2 827,3	2 826,3	2 825,7	2 825,3	2 824,4	2 824,1
1973	2 824,2	2 824,0	2 824,9	2 824,0	2 823,9	2 824,2	2 823,7	2 823,6	2 823,2	2 823,6	2 822,1	2 821,5
1974	2 821,7	2 821,2	2 821,1	2 820,4	2 820,5	2 819,7	2 818,9	2 818,3	2 817,2	2 817,0	2 816,3	2 815,2
1975	2 815,3	2 815,1	2 815,9	2 815,6	2 815,2	2 814,3	2 814,1	2 813,3	2 812,5	2 811,9	2 811,9	2 811,9
1976	2 810,6	2 809,2	2 809,5	2 809,9	2 810,0	2 809,4	2 808,6	2 808,5	2 808,2	2 807,6	2 807,0	2 806,8
1977	2 806,8	2 806,7	2 806,2	2 806,2	2 806,0	2 805,2	2 805,1	2 804,1	2 803,9	2 803,4	2 802,3	2 801,7
1978	2 801,6	2 801,8	2 802,1	2 802,2	2 801,8	2 801,1	2 800,9	2 799,7	2 799,2	2 798,6	2 797,7	2 797,2
1979	2 796,7	2 796,7	2 797,6	2 796,8	2 797,0	2 797,0	2 797,4	2 796,9	2 796,3	2 795,2	2 795,1	2 795,0
1980	2 794,8	2 798,9	2 797,9	2 800,2	2 799,9	2 799,4	2 799,8	2 799,5	2 798,5	2 798,5	2 798,0	2 797,1

Fuente: Propia

4.4. Resultados Obtenidos al realizar la Proyección.

Al tener como último dato de medición del volumen y nivel de espejo de agua de la Laguna Aricota en el mes de Abril del 2011, la simulación comienza a partir de esa fecha.

Como se mencionó anteriormente, al existir pocos datos con los cuales se realizó el entrenamiento, es una fuerte razón por la cual la proyección se realiza sólo por 10 años.

Teniendo como resultados las Tablas 22 y 23:

Tabla 22. Resultados: Proyección – Volumen (MMC)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2011				163,61	165,78	160,96	172,23	170,33	168,21	165,17	158,87	155,07
2012	153,86	163,16	163,62	165,65	162,47	158,56	154,81	158,35	157,68	152,15	150,51	145,75
2013	142,92	153,30	165,34	162,02	163,04	163,84	159,31	158,45	155,22	152,96	152,20	148,04
2014	143,98	145,62	152,70	153,53	147,43	147,06	146,66	144,77	139,90	137,30	134,53	130,11
2015	137,30	140,49	144,44	139,12	141,72	139,64	140,11	138,51	132,94	129,85	126,35	124,24
2016	136,08	133,80	141,22	137,69	135,31	136,12	130,52	127,57	126,97	122,28	119,22	119,42
2017	127,44	135,94	131,31	131,33	129,44	129,80	124,95	124,98	122,91	119,37	115,34	110,92
2018	116,63	140,15	142,83	142,00	136,99	134,99	130,60	127,74	121,93	122,35	117,73	112,58
2019	111,90	107,88	122,93	121,32	120,85	116,22	110,27	108,90	106,27	106,94	106,35	100,55
2020	98,96	100,01	97,24	92,55	91,42	86,23	82,80	78,68	76,53	74,12	69,49	63,80

Fuente: Propia

Tabla 23. Resultados: Proyección – Nivel de Espejo (msnm)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2011				2 768,8	2 769,1	2 768,3	2 769,4	2 769,1	2 768,8	2 768,3	2 766,9	2 766,3
2012	2 766,0	2 765,4	2 765,2	2 765,3	2 764,9	2 764,3	2 763,7	2 763,9	2 763,8	2 762,7	2 762,5	2 761,6
2013	2 761,0	2 762,0	2 763,2	2 762,7	2 762,8	2 762,8	2 762,1	2 761,9	2 761,4	2 761,0	2 760,9	2 760,3
2014	2 759,7	2 759,9	2 760,9	2 761,2	2 759,8	2 759,7	2 759,7	2 759,5	2 758,6	2 758,2	2 757,8	2 757,0
2015	2 757,5	2 757,3	2 757,3	2 756,3	2 756,6	2 756,3	2 756,3	2 756,1	2 755,0	2 754,5	2 754,0	2 753,7
2016	2 754,8	2 754,4	2 752,6	2 752,2	2 751,8	2 751,9	2 750,8	2 750,3	2 750,2	2 749,4	2 748,8	2 748,8
2017	2 749,3	2 750,1	2 749,1	2 749,1	2 748,8	2 748,8	2 747,9	2 747,8	2 747,5	2 746,9	2 746,3	2 745,6
2018	2 746,2	2 750,9	2 750,0	2 749,8	2 748,9	2 748,5	2 747,8	2 747,4	2 746,2	2 746,2	2 745,4	2 744,5
2019	2 744,4	2 741,8	2 743,3	2 743,1	2 743,1	2 742,3	2 741,0	2 740,8	2 740,4	2 740,5	2 740,4	2 739,2
2020	2 737,7	2 737,1	2 736,5	2 735,7	2 735,6	2 734,7	2 734,1	2 733,4	2 733,2	2 732,8	2 732,0	2 730,9

Fuente: Propia

Según los resultados obtenidos, se observa que el volumen de agua de la Laguna Aricota, disminuye de una manera alarmante, con un valor de volumen de agua esperado de 63,68 MMC. Viendo conveniente realizar constantemente proyectos de afianzamiento hídrico, para que el Ciclo Hidrológico de la Laguna Aricota, no se vea afectada de una manera negativa, por la acción del hombre.

V. DISCUSIONES

5.1. Validación de Objetivos e Hipótesis.

Al validar una o más subhipótesis se dirá que los objetivos correspondientes fueron alcanzados.

5.1.1. Sub-hipótesis y Objetivos Específicos.

1. Sub-Hipótesis 01 y Objetivo Específico 01.

O₁: Analizar el comportamiento y el tipo de Distribución que sigue cada variable hidrológica, que interviene en el Ciclo Hidrológico de la Laguna Aricota, y así obtener el tipo de distribución a la cual se ajusta cada una de ellas.

H₁: Las Variables Hidrológicas generadas aleatoriamente, influyen favorablemente a la simulación, ya que poseen un algo grado de confiabilidad.

Para validar la hipótesis anterior, se utilizó la Prueba de T-Student, para cada una de las variables hidrológicas. Se tiene como ejemplo la aplicación de la Prueba de T-Student con la Precipitación Mensual. Se utilizó como base el **Tabla 24**, para cada una de los resultados obtenidos de la generación de variables hidrológicas aleatorias.

Tabla 24. Validación de Sub-Hipótesis 01: Precipitación

Mes	MEDICIONES			SIMULADOS			T	G.L.	Ttabla
	n_1	\bar{x}_1	S_1^2	n_2	\bar{x}_2	S_2^2			
Ene	17	26,40	785,61	11	30,30	715,37	-0,37	26	2,78
Feb	17	42,20	1253,73	11	42,35	768,17	-0,01	26	2,78
Mar	16	21,85	634,41	11	26,74	237,65	-0,57	25	2,79
Abr	16	1,15	17,07	11	1,16	0,95	-0,01	25	2,79
May	16	0,00	0,00	11	0,00	0,00	--	--	--
Jun	17	0,00	0,00	11	0,00	0,00	--	--	--
Jul	14	0,00	0,00	11	0,00	0,00	--	--	--
Ago	13	0,00	0,00	11	0,00	0,00	--	--	--
Sep	12	0,00	0,00	11	0,00	0,00	--	--	--
Oct	16	0,00	0,00	11	0,00	0,00	--	--	--
Nov	15	0,29	0,31	11	0,42	0,48	-0,53	24	2,80
Dic	15	7,13	54,18	11	8,65	49,12	-0,53	24	2,80

Fuente: Propia

“Al aplicar la Prueba de T-Student con cada una de las variables hidrológicas generadas aleatoriamente, se observa que los valores de los T calculados son menores que los de la tabla de T-Student, por lo que se afirma que los valores simulados son Homogéneos en la media, con un Nivel de Significancia de 0.01. Asimismo las Variables Hidrológicas generadas aleatoriamente, influyen favorablemente a la simulación, ya que se observa que posee un alto grado de confiabilidad.”

2. Sub-Hipótesis 02 y Objetivo Especifico 02.

O₂: Generar las variables hidrológicas aleatorias, según el tipo de distribución a la que pertenecen; y así obtener el Volumen y Nivel de Espejo simulados, utilizando Redes Neuronales Artificiales.

H₂: La aplicación de Redes Neuronales Artificiales, favorece el proceso de la Simulación, permitiendo calcular tanto el Volumen como el Nivel de Espejo de Agua de la Laguna Aricota, con un alto grado de confiabilidad.

Aplicando de igual manera la prueba de T-Student, tenemos:

“Según los resultados, el entrenamiento de la RNA permite calcular el incremento del volumen y del nivel de espejo, cuyos valores tienen un Error mínimo al ser comparados con los valores reales, por lo que se afirma que el uso de un RNA favorece el proceso de simulación obteniendo así un Volumen y un Nivel de Espejo de Agua de la Laguna que son confiables, con un nivel de significancia de 0.01.”

3. Sub-Hipótesis 03 y Objetivo Especifico 03.

O₃: Obtener una proyección, mediante el uso del software de simulación, el valor del Volumen y del Nivel de Espejo de agua de la Laguna Aricota para el año 2020.

H₃: Los valores del volumen y nivel de espejo de agua, obtenidos mediante la simulación al 2020, demuestran que se necesitaran realizar proyectos para un afianzamiento hídrico de la Laguna Aricota.

“Al encontrarse en constate uso para la Generación Eléctrica, por medio de las Hidroeléctricas Aricota I y Aricota II, que generan energía eléctrica a Tacna, Moquegua e Ilo, y también al ser utilizada por usuarios de aguas abajo. Y al obtener los resultados obtenidos de la proyección para el año 2020, obteniendo un valor de volumen de 63,88 MMC, se ve la necesidad de seguir realizando proyectos para el afianzamiento hídrico de la Laguna Aricota, puesto que la Laguna Aricota es vital para el ecosistema de la Cuenca Locumba.”

5.1.2. Hipótesis Global y Objetivo General.

O: Simular el Ciclo Hidrológico de la Laguna Aricota, utilizando Redes Neuronales Artificiales.

H₁: Los resultados obtenidos, por medio del software desarrollado para la simulación del ciclo hidrológico de la laguna Aricota, poseen un alto grado de confiabilidad.

“Al validar las sub-hipótesis, de que las variables hidrológicas generadas de forma aleatoria y que el uso de una RNA favorece al proceso de simulación, así como favorece al momento de realizar los cálculos empíricos, y así obtener valores confiables con respecto al incremento del volumen y del nivel de espejo, se puede afirmar que los resultados obtenidos por medio del software desarrollado para la simulación del ciclo hidrológico de la laguna Aricota, poseen un alto grado de confiabilidad, ya que en cada paso se obtienen valores confiables, que hacen que el todo (el software de simulación) obtenga resultados confiables.”

VI. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Existe la necesidad de realizar estudios y proyectos de afianzamiento del recurso hídrico de la laguna Aricota, ya que como se observa en los resultados obtenidos para el volumen, este disminuye constantemente en la temporada en la que no hay precipitaciones, el cual también es afectado directamente por el bombeo que realizan las hidroeléctricas, lo que genera que el volumen de agua de la laguna Aricota disminuya más de lo que aumenta.
- Se observa que el comportamiento de cada variable hidrológica es distinto en cada mes, es la razón por la que se realizó un análisis mensual de cada una de ellas, para así obtener que tipo de distribución sigue, y cuales son los parámetros que se requiere para la generación aleatoria de cada variable hidrológica.
- Al existir pocas mediciones, en lo que se refiere a las Filtraciones, no se puede realizar un análisis mensual, ya que el número de datos utilizados para el análisis sería demasiado bajo (llegando en algunos casos a sólo dos mediciones por mes), por lo que el grado de confiabilidad sería también demasiado bajo.

- El uso de una RNA mejora y facilita rotundamente la simulación del ciclo hidrológico de la laguna Aricota, ya que al haber obtenido los valores aleatorios de las nuevas variables hidrológicas, estas mismas son usadas para alimentar la RNA y así mejorar el entrenamiento de la misma.

6.2. Recomendaciones.

- Si se quiere conseguir una simulación con mayor grado de confiabilidad que la que actualmente posee, es necesario seguir realizando las mediciones de las distintas variables hidrológicas que afectan el ciclo hidrológico de la Laguna Aricota, y seguir entrenando a la R.N.A para que los valores resultantes posean el menor margen de Error posible.
- Al contar con mayor información, tanto el análisis del comportamiento de las variables hidrológicas, como el entrenamiento de la RNA, alcanzarán un mayor grado de confiabilidad.
- Obteniendo los resultados de la proyección para el año 2020, es necesario seguir realizando proyectos de afianzamiento hídrico de la laguna Aricota, además de las demás vertientes de la Región Tacna, ya que como se conoce Tacna es una zona desértica y no sólo la Cuenca Locumba sino también todo Tacna sufra fuertes pérdidas de este recurso que es el agua en un futuro.

VII. REFERENCIA BIBLIOGRÁFICA

Aracil, J. (s.f.). *Dinámica de Sistemas* (4ta ed.). Isdefe. 10 - 12.

Asteasuain, F., Contreras, B., Estévez, E., & Fillotrani, P. (2003). *Programación Orientada a Aspectos: Metodología y Evaluación*. Obtenido de <http://www.lafhis.dc.uba.ar/~ferto/docs/cacic2003.pdf>

Ávila Baray, H. (2006). *Introducción a la Metodología de la Investigación*. Obtenido de <http://www.eumed.net/libros/2006c/203/>

Azarang, M., & García Dunna, E. (1998). *Simulación y Análisis de Modelos Estocásticos* (1era ed.). McGraw Hill Interamericana S.A. 1, 3, 11 - 12.

Bertona, L. (2005). *Entrenamiento de Redes Neuronales basado en Algoritmos Evolutivos*. Tesis de Grado en Ingeniería Informática, Universidad de Buenos Aires. 5 - 7, 13 - 23.

Boza Condorena, E. (2005). *Efecto de la topología de redes neuronales de Backpropagation en la optimización de procesos químicos Vía modelos matemáticos neuronales empíricos*. Obtenido de http://www.ciiq.org/varios/peru_2005/Trabajos/VI/6.01.pdf

Canavos, G. (1998). *Probabilidad y Estadística: Aplicaciones y Métodos* (1era ed.). McGraw Hill Interamericana S.A. 130 - 174.

Cantú, M. (2005). *La Biblia de Delphi 7* (1era ed.). Anaya Multimedia. 170 - 217, 403 - 454.

Casas Castillo, C., & Alarcón Jordán, M. (1999). *Meteorología y Clima* (1era ed.). Centre de Publicacions del Campus Nord. 63 - 83, 117 - 152.

Day, R. (1998). *How to Write & Publish a Scientific Paper* (5ta ed.). The Oryx Press.

Dieterich, H. (2001). *Nueva guía para la investigación científica* (1era ed.). Offset Libra, S.A.

Domínguez Valente, C., López Octaviano, M., Martínez Martínez, G., Reyes Martínez, B., & Vázquez Hernández, M. (2008). *Simulación Digital. Tecnológico de Estudios Superiores de Jocotitlan*. Obtenido de <http://www.monografias.com/trabajos-pdf/simulacion-digital/simulacion-digital.pdf>

Filali Bouami, M. (2005). *Desarrollo y Optimización de Nuevos Modelos de Redes Neuronales basadas en Funciones de Base Radial*. Tesis Doctoral, Universidad de Granada. 36 - 54, 57 - 101, 122 - 134.

Gallego Fernández, I., & Medina Llinás, M. (1993). *Algorítmica y programación para Ingenieros* (1era ed.). Servei de Publicacions de la UPC y CPDA. 165 - 209.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (1991). *Metodología de la Investigación* (1era ed.). McGraw Hill Interamericana S.A.

Herrera, R., Caldera, R., & Martínez, M. (1999). *Análisis y Diseño de Sistemas con el Lenguaje de Modelaje Unificado (UML)*. Obtenido de <http://www.freelibros.com/2011/04/disen-y-analisis-de-sistemas-con-uml-randall-agenor-herrera-briones.htm>

Izaurieta, F., & Savedra, C. (s.f.). *Redes Neuronales Artificiales*. Obtenido de <http://encontrarpdf.net/preview/aHR0cDovL3d3dy51dGEuY2wvY2hhcmxhcY92b2x1bWVuMTYvSW5kaWNIL0NoLWNzYWF2ZWRYYS5wZGY=>

Letelier, P., & Penadés, C. (s.f.). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Obtenido de <http://www.willydev.net/descargas/masyxp.pdf>

Martínez, P., Gallardo, C., & Sánchez, J. (2010). *Estructuras de datos. Teoría*. Servicio de Publicaciones EUI.

Mijares, A. (1989). *Fundamentos de Hidrología de Superficie* (1era ed.). LIMUSA S.A. 13, 17 - 67.

Montaño Moreno, J. (s.f.). *Redes Neuronales Artificiales aplicadas al Análisis de Datos*. Tesis Doctoral, Universidad de Les Illes Balears. 40 - 61.

- Morales González, I. (2010). *Comparación teórico práctica entre modelos Estadísticos y el Perceptrón Multicapa*. Título de Estadístico, Pontificia Universidad Católica de Valparaíso. 22.
- Moreno, A., Armengol, E., Béjar, J., Belanche, L., Cortés, U., & Gavalda, G. (1994). *Aprendizaje Automático* (1era ed.). Servei de Publicacions de la UPC y CPDA. 227 - 256.
- Nanía, L. (2003). *La Cuenca y Los Procesos Hidrológicos*. Obtenido de <http://es.scribd.com/doc/49439706/Lacuencaylos-2003>
- Ras Sabidó, A. (1993). *Estadística Aplicada per a enginyeria* (1era ed.). Servei de Publicacions de la UPC y CPDA. 25 - 51.
- Rodríguez Ávila, E. (s.f.). *El Correcto y Completo Desarrollo de un Algoritmo*. Obtenido de <http://homepage.mac.com/eravila/algoritmos.pdf>
- Russell, S., & Norvig, P. (2004). *Inteligencia Artificial: Un Enfoque Moderno* (2da ed.). Pearson Prentice Hall. 739 - 896.
- Sampieri, R. H. (1991). *Metodología de la Investigación*. México: McGraw - Hill Interamericana de México, S.A. de C.V.
- Tejerina, M. (2005). *Programación Orientada a Objetos*. Obtenido de <http://www.rinconcitodelphi.com/articulos/POO/POO.pdf>

Vega Valle, J. (2004). *Diseño e implementación de una herramienta para la enseñanza y aprendizaje de la teoría de colas*. Proyecto de fin de Carrera de Ingeniería Informática. 32 - 40.

Ven Te, C. (1994). *Hidrología Aplicada* (1era ed.). McGraw Hill Interamericana S.A. 2, 8, 9, 65, 73.

Vera Hernández, L. (2002). *Análisis del Aforo de la Estación Hidrométrica Obrajillo: Periodo 2000-2001*. Título Profesional de Ingeniero Mecánico de Fluidos, Universidad Nacional Mayor de San Marcos. 8 - 12.

Villanueva Espinoza, M. (2007). *Las Redes Neuronales Artificiales y su Importancia como Herramienta en la Toma de Decisiones*. Título Profesional de Ingeniero en Sistemas, Universidad Nacional Mayor de San Marcos. 19 - 43.

ANEXOS

ANEXO 1. INSTRUMENTOS

Anexo 1.1. Medición de los Indicadores.

1. Confiabilidad.

Para calcular la confiabilidad, tanto de la Red Neuronal Artificial como de la simulación del Ciclo hidrológico, se vio necesario aplicar la Prueba de t-Student.

1.1. Prueba de t-Student.

La prueba t-Student fue desarrollada en 1899 por el químico inglés William Sealey Gosset (1876-1937), mientras trabajaba en técnicas de control de calidad para las destilerías Guinness en Dublín. Debido a que en la destilería, su puesto de trabajo no era inicialmente de estadístico y su dedicación debía estar exclusivamente encaminada a mejorar los costes de producción, publicó sus hallazgos anónimamente firmando sus artículos con el nombre de "Student".

La prueba t-Student se utiliza para contrastar hipótesis sobre medias en poblaciones con distribución normal. También proporciona resultados

aproximados para los contrastes de medias en muestras suficientemente grandes cuando estas poblaciones no se distribuyen normalmente.

Cálculo de la prueba t-Student para la diferencia de medias suponiendo igualdad de varianzas

Para llevar a cabo el contraste:

$$H_0: \mu_1 - \mu_2 = 0 \dots\dots\dots (01)$$

$$H_1: \mu_1 - \mu_2 \neq 0 \dots\dots\dots (02)$$

Suponiendo igualdad de varianzas poblacionales, se construye el estadístico de contraste experimental t, dado por:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{EE(\bar{x}_1 - \bar{x}_2)} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{S^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}} \dots\dots\dots (03)$$

Que bajo la hipótesis nula sigue una distribución t-Student con grados de libertad: $gl = (n_1 + n_2 - 2)$

2. Porcentaje de Error de RNA.

Para calcular el % de Error de la RNA entrenada, *ver el punto 2.2.6.4.*

El cálculo del porcentaje de Error en la RNA, se encuentra implementada en el componente de la RNA.

ANEXO 2. MEDICIONES METEOROLÓGICAS

Anexo 2.1. Medición de la Precipitación Total Mensual.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1993												8,5
1994	10,5	90,1	0,0	16,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	6,8
1995	27,2	0,0	77,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,1	12,6
1996	0,0	1,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	26,6
1997	31,6	77,9	64,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,4
1998	10,3	27,8	1,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,8	2,1
1999	2,4	64,5	56,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,8
2000	91,8	58,8	29,8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	4,0
2001	27,9	129,7	42,4	1,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	
2002	13,4	37,5	21,8	0,2	0,0	0,0	0,0	0,0	0,0			3,4
2003	14,8	2,2	20,9	0,0	0,0	0,0			0,0	0,0	0,0	
2004	19,7	52,1	5,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2005	21,3	25,7			0,0	0,0	0,0	0,0	0,0	0,0	0,0	19,4
2006	33,4	52,8	12,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0		1,7
2007	48,3	19,1	1,8	0,0	0,0	0,0	0,0	0,0		0,0	0,8	6,4
2008	92,5	9,7	6,5	0,0	0,0	0,0	0,0	0,0		0,0	0,0	3,2
2009	1,7	51,1	9,1	0,0	0,0	0,0		0,0		0,0	0,7	
2010	2,0	17,0	0,7	0,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	8,1

Fuente: Proyecto Especial Tacna

Anexo 2.2. Medición de la Evaporación Total Mensual.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1993												90,2
1994	66,9	92,0	166,0	145,0	140,0	144,0	147,0	183,0	205,0	236,0	229,0	187,0
1995	178,0	184,0	127,0	160,0	150,0	139,0	155,0	181,0	201,0	212,0	222,0	215,0
1996	142,0	133,0	156,0	158,6	172,0	153,0	162,0	183,0	216,0	236,0	217,0	204,0
1997	83,0	42,0	148,0	159,0	133,0	134,0	156,1	180,0	192,0	214,0	187,0	196,0
1998	126,0	112,6	152,0	162,1	165,9	167,3	154,0	163,5	142,6	152,4	126,7	190,3
1999	135,9	82,8	156,0	151,9	116,2	142,0	152,8	171,6	190,1	208,5	191,3	182,2
2000	141,2	95,2	146,3	150,1	141,7	138,5	150,1	152,7	183,5	212,5	222,4	180,9
2001	98,1	76,6	103,3	129,9	140,1	106,0	145,6	155,6	198,4	243,8	224,8	208,9
2002	189,4	112,5	129,9	191,3	157,3	130,0	139,5	155,4	178,5	190,9	161,2	193,2
2003	167,5	136,3	129,2	155,3	137,7	127,4	146,0	168,4	198,7	224,4	216,8	178,4
2004	132,9	146,4	151,9	161,0	142,2	119,9	126,2	158,8	200,9	223,1	212,5	221,2
2005	161,5	113,2	156,8	150,2	149,2	123,4	139,2	157,6	183,3	217,5	206,2	177,4
2006	129,7	162,5	126,7	181,3	148,6	130,5	126,0	158,9	176,5	184,0	199,1	204,9
2009	194,9	118,4	171,0	171,9	165,6	139,8	144,0	150,7	203,4	224,0	207,3	225,8
2010	196,8	163,5	187,1	174,2	126,0	156,5	174,6	181,6	178,7	184,7	181,4	173,3

Fuente: Proyecto Especial Tacna

Anexo 2.3. Medición del Caudal Bombeado.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1967		0,214	0,007	0,295	0,600	0,207	0,276	0,059	0,072	0,043	0,213	0,280
1968	0,105	0,029	0,022	0,011		0,004	0,002		0,194	0,243	0,359	0,366
1969	0,099				0,002		0,016	0,017	0,175	0,002		
1970	0,088	0,246	0,226	0,162	0,249	0,273	0,130	0,015	0,311	0,295	0,206	0,346
1971	0,478	0,799	0,740	0,719	0,628	0,588	0,588	0,543	0,729	0,798	0,715	0,841
1972	0,609	0,934	0,565	0,331	0,285	0,131	0,160	0,189	0,298	0,202	0,291	0,357
1973	0,224	0,533	0,222	0,119	0,226	0,081	0,059	0,050	0,056	0,093	0,154	0,171
1974	0,334	0,158	0,148	0,182	0,133	0,013	0,117	0,577	0,651	0,263	0,482	0,161
1975	1,471	1,551	1,319	1,208	0,551	0,020	0,053	0,728		0,559	0,991	0,944
1976	0,964	1,060	0,959	0,944	0,767	0,864	1,020	1,855	2,338	3,438	3,195	2,045
1977	2,512	2,457	0,830		0,723	2,550	2,487	2,646	2,604	2,440	2,674	2,609
1978	2,209	1,442	1,656	1,053	1,297	1,376	1,414	1,179	1,473	1,843	1,857	1,594
1979	1,876	1,935	1,525	2,132	1,882	2,577	2,739	1,969	3,929	2,757	2,492	3,076
1980	2,007	1,990	1,911	1,792	1,945	1,771	1,196	2,179	2,866	2,785	2,332	1,143
1981	1,254	1,345	1,579	2,158	2,207	2,704	2,778	2,272	1,494	3,036	2,812	3,282
1982	2,531	3,592	2,869	3,172	3,857	3,837	3,826	2,179	2,283	2,812	2,324	2,543
1983	1,620	2,377	2,073	2,371	1,968	1,453	1,611	1,558	1,572	1,024	1,449	2,259
1984	3,114	3,136	2,983	3,098	3,333	3,018	2,853	2,917	3,080	2,159	2,457	3,122
1985	3,145	3,141	2,932	2,713	2,577	3,240	2,654	2,604	2,925	2,416	3,463	2,443
1986	2,257	2,587	2,564	2,576	2,335	3,340	1,879	2,167	2,382	1,769	3,175	2,100
1987	2,220		2,739	2,245	3,238	2,250	1,819	2,326	3,002	2,968	2,909	2,623
1988	2,852	2,906	2,572	3,090	2,697	2,386	2,349	2,933	2,475	2,164		2,062
1989	2,034	1,623	1,871	2,537	3,043	1,567	1,523	1,667	1,557	1,704	1,502	1,820
1990	1,663	1,456	2,002									
1991	1,430	1,350	0,710	1,350	1,280	1,420	1,420	1,380	1,450	1,380	1,400	1,504
1992	1,787	1,693	1,607	1,706	1,670	1,723	1,658	1,648	1,640	1,564	1,596	1,922
1993	1,554	1,408	1,350	1,498	1,588	1,657	1,619	1,632	1,550	1,420	1,228	1,638
1994	1,807	1,355	1,494	1,569	1,632	1,827	1,286	1,226	1,271	1,288	1,634	1,909
1995	1,918	2,251	1,953	1,561	1,558	1,416	1,428	1,431	1,563	1,647	1,407	1,517
1996	1,448	1,559	1,707	1,603	1,121	1,195	1,075	1,188	1,385	1,121	1,197	1,226
1997	1,154	1,149	1,152	1,203	1,173	1,136	1,177	0,971	1,217	1,145	1,038	1,287
1998	1,199	1,069	1,106	1,141	1,258	1,644	1,601	1,501	1,285	1,530	1,340	1,329
1999	1,156	1,372	1,825	1,460	1,410	1,425	1,506	1,568	1,516	1,512	1,479	1,440
2000	1,611	2,467	2,665	2,438	1,800	1,944	1,288	1,321	1,350	1,250	1,501	1,999
2001	1,691	1,683	1,759	1,955	1,494	1,240	1,244	1,256	1,215	1,450	1,355	1,297

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2002	1,339	1,209	1,230	1,125	1,043	1,072	1,179	1,293	1,557	1,761	1,688	1,698
2003	1,331	1,299	1,348	1,361	1,323	1,389	1,455	1,462	1,433	1,458	1,635	2,014
2004	0,867	1,180	1,160	1,170	1,171	1,170	1,484	1,477	1,492	1,458	1,483	1,483
2005	1,483	1,492	1,489	1,493	1,494	1,484	1,483	1,481	1,497	1,495	1,496	1,481
2006	1,476	0,291	0,233	0,555	0,771	1,031	1,198	1,076	1,071	1,054	1,042	1,013
2007	0,853	1,056	1,334	1,438	1,421	1,348	1,281	1,423	1,425	1,470	1,440	1,430
2008	1,480	1,480	1,500	1,500	1,490	1,480	1,470	1,490	1,500	1,440	1,450	1,450
2009	1,470	1,480	1,320	1,380	1,440	1,490	1,490	1,380	1,450	1,340	1,420	1,400
2010	1,380	1,420	1,450	1,440	1,430	1,440	1,400	1,390	1,410	1,390	1,380	1,350

Fuente: Proyecto Especial Tacna

Anexo 2.4. Medición de las Descargas Medias Mensuales del Río Callazas.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1963								0,704	0,749	0,688	0,657	0,768
1964	0,795	1,061	1,199	0,852	0,676	0,709	0,733	0,665	0,589	0,578	0,536	0,543
1965	0,532	0,869	0,573	0,536	0,586	0,615	0,629	0,546	0,545	0,502	0,507	0,490
1966	0,472	0,494	0,482	0,475	0,483	0,497	0,498	0,516	0,436	0,428	0,411	0,402
1967	0,549	1,131	2,439	0,756	0,666	0,604	0,581	0,591	0,580	0,538	0,496	0,470
1968	0,965	0,871			0,660	0,654	0,558	0,625	0,428	0,400	0,493	0,405
1969	0,469	1,462	1,179	0,462	0,437	0,422	0,446	0,505	0,359	0,339	0,367	0,381
1970	2,050	1,162	1,173	0,418	0,431	0,455	0,445	0,463	0,445	0,421	0,410	0,465
1971	0,954	3,440	1,991	0,563	0,577	0,572	0,573	0,564	0,564	0,371	0,395	
1972					0,838	0,663	0,664	0,564	0,569	0,554	0,499	0,996
1973	2,679	12,260	11,056	2,796	0,796	0,524	0,625		0,545	0,546	0,422	0,445
1991	3,942	1,084	2,685	0,387	0,358	0,341	0,334	0,392	0,284	0,180	0,164	0,138
1992	0,187	0,166	0,177	0,338	0,376	0,391	0,471	0,488	0,413	0,413	0,419	1,208
1993	2,109	0,931	1,207	0,695	0,443	0,436	0,432	0,551	0,494	0,482	0,466	0,485
1994	1,383	5,957	0,709	0,652	0,593	0,553	0,541	0,563	0,516	0,529	0,538	0,557
1995	0,860	0,587	1,288	0,640	0,620	0,645	0,658	0,856	0,632	0,614	0,556	0,542
1996	0,673	1,051	0,904	0,579	0,560	0,543	0,590	0,676	0,534	0,439	0,421	0,440
1997	0,878	3,703	2,682	0,832	0,645	0,640	0,604	0,631	0,546	0,421	0,407	0,459
1998	1,365	0,984	0,577	0,604	0,619	0,573	0,565	0,556	0,489	0,438	0,437	0,495
1999	0,702	7,913	5,625	1,597	0,853	0,725	0,576	0,587	0,512	0,484	0,450	0,438

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2000	1,381	4,184	3,176	0,761	0,645	0,710	0,709	0,702	0,582	0,554	0,461	0,548
2001	0,979	5,153	5,480	1,143	0,912	1,043	0,922	0,875	0,763	0,813	0,699	0,816
2002	0,905	3,035	5,147	1,965	1,426	1,432	2,138	1,736	0,726	0,683	0,636	0,641
2003	0,750	0,749	1,008	0,915	0,805	0,806	0,754	0,872	0,801	0,654	0,609	0,609
2004	0,745	1,599	0,523	0,710	0,850	0,816	0,891	0,877	0,556	0,636	0,577	0,596
2006	1,087	3,677	3,808	1,462	0,825	0,785	0,923	0,898	0,815	0,803	0,872	0,744
2007	2,689	2,505	4,214	1,233	0,822	0,796	0,698	0,639	0,620	0,736	0,732	0,619
2008	2,050	1,250	1,950	0,840	1,050	0,900	0,970	0,780	0,910	0,660	0,550	0,720
2009	1,099	3,116	2,921	0,962	0,834	0,810	0,719	0,760	0,705	0,612	0,574	0,673
2010	0,923	1,780	1,050	0,745	0,727	0,728	0,711	0,655	0,690	0,662	0,689	0,732

Fuente: Proyecto Especial Tacna

Anexo 2.5. Medición de las Descargas Medias Mensuales del Río Salado.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1963										0,494	0,491	0,613
1964	0,780	1,078	0,587	0,570	0,574	0,476	0,480	0,572	0,619	0,440	0,477	0,611
1965	0,468	1,190	0,482	0,469	0,459	0,567	0,455	0,401	0,567	0,536	0,566	0,503
1966	0,439	0,460	0,480	0,441	0,459	0,441	0,416	0,413	0,402	0,396	0,400	0,419
1967	0,343	1,370	2,031	0,735	0,474	0,467	0,447	0,395	0,404	0,372	0,353	0,368
1968	1,541	1,362				0,453	0,485	0,382	0,340	0,346	0,350	0,366
1969	0,559	1,560	1,470	0,408	0,343	0,360	0,342	0,348	0,367	0,347	0,306	0,295
1970	1,491	0,808	1,988	0,498	0,471	0,419	0,391	0,391	0,410	0,392	0,410	0,426
1971	0,779	1,857	1,195	0,435	0,438	0,471	0,471	0,460	0,428	0,405	0,355	0,347
1972	7,376	3,898	2,582	0,704	0,524	0,518	0,525	0,530		0,393	0,450	0,837
1973	3,428	8,307	4,291	0,797	0,563	0,588	0,565		0,469	0,562	0,287	
1991	2,658	1,206	1,673	0,320	0,220	0,283	0,193	0,243	0,201	0,171	0,207	0,134
1992	0,211	0,168	0,116	0,200	0,246	0,368	0,555	0,430	0,439	0,458	0,569	1,520
1993	2,500	1,258	0,670	0,825	0,882	0,918	0,922	1,008	0,983	0,908	0,882	0,909
1994	2,782	5,045	1,972	1,428	1,259	1,292	1,023	0,913	0,925	0,658	0,579	0,712
1995	1,335	1,036	2,582	0,737	0,620	0,738	0,726	0,644	0,638	0,379	0,281	0,449
1996	1,211	2,305	1,668	0,719	0,534	0,519	0,530	0,604	0,576	0,439	0,455	0,468
1997	1,431	5,472	3,387	0,980	1,064	0,920	0,716	0,653	0,809	0,699	0,512	0,578
1998	1,609	1,168	0,505	0,562	0,606	0,728	0,556	0,515	0,464	0,422	0,492	0,531
1999	0,329	3,854	5,192	0,672	1,246	0,931	0,571	0,697	0,537	0,525	0,560	0,568

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2000	1,284	3,177	2,506	1,129	0,950	0,873	0,747	0,767	0,604	0,474	0,409	0,256
2001	1,238	12,499	11,579	1,451	1,380	1,236	1,446	1,350	0,809	0,819	0,746	0,882
2002	1,013	2,253	4,013	2,116	1,673	1,187	1,523	1,226	0,763	0,767	0,598	0,605
2003	0,593	0,603	1,707	0,526	0,717	0,636	0,680	0,496	0,483	0,419	0,383	0,359
2004	1,043	2,798	1,420	0,673	0,468	0,631	0,749	0,902	0,852	0,459	0,481	0,377
2006	1,479	6,380	6,184	4,590	1,558	1,237	0,855	0,869	0,701	0,618	0,562	0,537
2007	2,376	1,708	2,334	0,681	0,704	0,717	0,652	0,643	0,602	0,563	0,681	0,679
2008	1,920	1,770	2,290	1,330	0,880	0,910	0,890	0,830	0,550	0,750	0,570	1,120
2009	0,487	1,584	1,145	1,444	1,409	1,000	0,984	0,723	0,709	0,714	0,663	0,548
2010	1,474	1,542	0,652	0,770	0,653	0,655	0,628	0,741	0,647	0,650	0,612	0,711

Fuente: Proyecto Especial Tacna

Anexo 2.6. Medición de las Descargas Mensuales de los Ríos Callazas y Salado.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1963	1,600	2,530	1,080	1,250	1,250	1,240	1,100	1,170	1,370	1,180	1,150	1,380
1964	1,580	2,140	1,780	1,420	1,250	1,190	1,210	1,200	1,210	1,020	1,020	1,140
1965	1,000	2,060	1,050	1,000	1,050	1,190	1,090	1,000	1,120	1,040	1,080	0,990
1966	0,870	0,950	0,960	0,920	0,940	0,940	0,920	0,960	0,840	0,830	0,810	0,820
1967	0,790	2,500	4,460	1,500	1,140	1,070	1,030	0,920	0,980	0,910	0,860	0,840
1968	2,510	2,230	9,520	1,780	1,090	1,010	1,010	0,960	0,770	0,750	0,840	0,780
1969	1,030	3,020	2,650	0,870	0,780	0,780	0,810	0,980	0,730	0,690	0,680	0,680
1970	3,550	1,970	3,760	0,910	0,900	0,880	0,860	0,900	0,860	0,810	0,830	0,950
1971	1,740	5,300	3,190	1,000	1,020	1,040	0,940	0,920	0,990	0,780	0,760	0,750
1972	5,180	7,060	8,770	4,690	1,360	1,180	1,180	1,090	1,030	0,940	0,950	1,840
1973	4,710	9,430	8,440	5,460	1,760	0,970	1,100	1,160	1,040	1,000	0,940	2,560
1974	6,050	9,080	6,880	3,520	1,210	0,730	0,950	1,110	1,070	1,180	1,030	0,940
1975	4,500	9,080	8,460	6,380	1,710	0,820	0,860	1,630	1,400	1,160	0,840	2,910
1976	5,280	7,330	5,560	3,570	1,390	1,200	1,140	1,410	0,830	1,800	1,290	1,220
1977	3,600	9,870	6,910	4,950	1,030	1,030	1,240	1,570	1,070	1,070	0,890	2,230
1978	2,850	2,810	2,270	1,810	1,490	1,500	1,080	1,181	1,380	1,156	1,370	1,470
1979	1,550	2,390	1,550	1,620	1,300	1,140	1,220	1,460	1,420	1,140	0,980	2,430
1980	1,320	1,300	1,660	1,270	1,310	1,230	0,710	0,730	0,930	1,340	1,460	1,010
1981	2,010	6,200	3,180	1,700	1,245	0,930	1,140	0,820	0,810	0,980	0,840	1,150

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1982	1,670	1,850	1,180	1,390	1,222	1,020	1,100	1,020	0,960	0,920	0,910	1,070
1983	1,270	1,640	1,260	0,858	1,214	1,040	0,744	0,935	0,816	0,980	0,970	1,320
1984	1,301	5,528	4,107	1,494	0,837	0,845	0,978	1,207	1,088	1,077	1,185	1,192
1985	2,140	8,645	5,091	1,678	1,075	0,990	0,960	1,040	1,000	1,128	0,915	1,281
1986	6,513	7,370	5,160	2,798	1,281	1,120	1,041	1,461	1,124	0,872	0,887	1,490
1987	8,400	5,672	3,270	2,190	1,055	1,030	1,040	1,090	1,116	0,902	0,841	1,300
1988	2,600	2,153	1,510	1,340	1,154	1,190	1,030	1,130	0,930	0,969	0,932	1,500
1989	2,530	3,620	3,840	2,240	1,189	1,200	1,180	1,160	1,190	1,000	0,985	1,210
1990	1,960	2,220	1,280	1,230	1,202	1,080	1,090	1,010	0,910	1,020	0,960	1,380
1991	2,200	2,730	2,090	1,630	1,206	1,150	1,130	1,060	0,970	0,980	1,090	1,130
1992	1,450	1,890	1,650	1,370	1,200	1,100	1,100	1,280	1,184	1,362	1,280	2,319
1993	4,607	2,134	2,549	1,522	1,708	1,813	1,726	1,842	1,619	1,664	1,472	1,511
1994	4,165	10,997	2,639	1,869	1,656	1,709	1,626	1,536	1,552	1,195	1,118	1,289
1995	2,188	1,622	3,902	1,377	1,252	1,395	1,395	1,524	1,315	1,058	0,951	1,102
1996	1,920	3,410	2,622	1,328	1,214	1,185	1,257	1,352	1,153	0,924	0,907	0,951
1997	2,339	9,201	6,084	1,833	1,734	1,577	1,366	1,345	1,437	1,215	1,012	1,131
1998	3,112	2,242	1,160	1,236	1,305	1,511	1,344	1,246	1,123	1,015	1,013	1,117
1999	1,051	11,796	10,760	2,298	2,120	1,675	1,168	1,305	1,070	1,032	1,030	1,022
2000	2,672	7,382	5,707	1,915	1,621	1,608	1,481	1,530	1,234	1,060	0,921	0,893
2001	2,238	17,677	17,095	2,633	2,324	2,349	2,486	2,415	1,728	1,667	1,497	1,738
2002	2,003	5,364	9,280	4,512	3,249	2,869	4,131	3,562	1,580	2,010	1,609	1,571
2003	1,703	1,632	2,938	1,691	1,755	1,693	1,614	1,588	1,445	1,203	1,083	1,053
2004	1,948	4,385	2,140	1,536	1,483	1,617	1,832	1,966	1,591	1,280	1,224	1,123
2006	2,716	10,227	10,172	6,172	2,543	2,191	1,967	1,947	1,676	1,531	1,584	1,411
2007	5,275	4,433	6,737	2,104	1,706	1,682	1,525	1,464	1,405	1,469	1,563	1,440
2008	4,070	3,090	4,320	2,240	2,120	1,995	2,040	1,760	1,580	1,540	1,230	1,950
2009	1,696	4,850	4,216	2,526	2,353	1,894	1,782	1,568	1,493	1,411	1,317	1,335
2010	2,708	4,042	1,882	1,635	1,570	1,814	1,779	1,826	1,687	1,632	1,511	1,624

Fuente: Proyecto Especial Tacna

Anexo 2.7. Medición de las Filtraciones en Confluencia Esquina - Puquina.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1992	0,300											
1999								0,439				
2001								1,269	1,145			1,080
2002					1,268	1,240	1,219	1,313		1,221	1,136	
2003	0,962		1,108	1,030		0,933	0,906	0,942	0,810		0,795	
2004			0,638	0,611	0,655	0,719	0,641	0,705	0,678	0,664	0,798	0,752
2005			0,572			0,570	0,645	0,575				
2006		0,653				0,935						
2007			0,820		0,852					0,841		
2008		0,676	0,645	0,631	0,688		0,723		0,670		0,647	
2009						0,640						0,620
2010		0,625			0,544	0,640	0,631		0,582			
2011		0,533	0,527			0,761						

Fuente: Proyecto Especial Tacna

Anexo 2.8. Medición de las Filtraciones en Confluencia con la CC.HH.

Aricota N° 2.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1992	0,300											
1999								0,300				
2001								0,300	0,300			0,316
2002					0,408	0,400	0,365	0,360		0,442	0,412	
2003	0,410		0,400	0,400		0,345	0,290	0,290	0,285		0,280	
2004			0,250	0,250	0,250	0,250	0,250	0,250	0,250	0,250	0,250	0,250
2005			0,250			0,250	0,250	0,250				
2006		0,250				0,250						
2007			0,190		0,190					0,190		
2008		0,190	0,190	0,190	0,190		0,190		0,190		0,190	
2009						0,190						0,190
2010		0,190			0,190	0,190	0,150		0,150			
2011		0,150	0,150			0,150						

Fuente: Proyecto Especial Tacna

Anexo 2.9. Medición del Volumen Útil de Agua.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1967							708,38	706,51	704,54	702,15	699,35	706,61
1968	698,42	699,97	710,99	710,26	718,51	708,82	705,37	703,60	701,42	698,94	696,25	693,05
1969	691,15	693,77	694,28	692,01	690,06	688,31	687,07	685,63	683,79	681,74	679,89	678,05
1970	680,20	681,53	682,86	680,61	678,26	675,80	674,68	674,07	671,22	668,47	666,23	663,70
1971	663,09	668,47	668,98	665,52	662,18	659,35	655,51	653,29	650,28	645,46	642,06	638,96
1972	651,18	659,35	680,10	690,06	687,28	685,84	683,89	681,84	679,69	677,44	674,58	672,23
1973	676,52	700,49	728,48	731,86	738,52	724,59	721,86	719,24	716,21	713,39	709,94	706,30
1974	718,40	745,09	761,75	759,39	755,01	750,84	747,22	742,97	737,99	733,44	729,01	725,43
1975	723,02	739,26	769,60	767,66	762,93	759,72	756,72	751,27	747,97	743,08	737,14	734,29
1976	744,46	758,54	769,92	766,26	761,11	756,08	749,88	742,01	733,44	720,81	708,90	701,32
1977	694,90	704,02	731,12	731,75	727,54	717,57	710,26	701,01	692,01	683,27	674,48	664,92
1978	666,95	669,18	663,40	659,65	655,11	651,08	647,76	656,22	651,78	646,06	640,86	636,27
1979	663,60	628,71	629,90	624,85	605,10	599,74	594,40	590,53	585,02	579,74	575,14	570,26
1980	565,69	560,86	558,11	553,11	548,31	543,44	540,07	535,69	529,94	523,93	518,68	514,28
1981	510,53	521,16	523,56	519,88	509,80	502,61	495,64	489,24	485,03	476,99	469,28	461,17
1982	457,14	451,54	445,37	438,63	431,58	425,09	419,14	413,89	410,27	403,06	397,48	391,92
1983	387,54	381,96	375,75	370,22	363,92	359,82	355,73	352,30	348,65	345,24	341,37	330,56
1984	329,16	334,23	336,11	330,72	322,80	315,95	309,53	303,30	296,59	292,02	287,26	280,96
1985	273,32	286,14	290,90	286,96	275,01	273,61	267,78	269,35	256,24	251,24	243,17	238,88
1986	246,27	256,17	263,22	261,27	257,10	250,25	246,56	242,74	237,06	233,29	226,09	222,45
1987	241,13	244,15	238,74	233,99	226,44	220,87	217,52	213,92	207,91	200,89	194,19	188,35
1988	184,57	180,81	176,04	170,16	164,72	160,19	155,39	149,52	144,49	140,72	137,10	133,19
1989	128,84	140,84	137,88	139,51	132,66	130,03	127,78	124,59	121,94	119,31	116,52	112,70
1990	109,31	106,51	102,94	99,84	91,42	88,23	85,33	82,45	80,39	77,70	74,97	75,45
1991	81,36	83,26	89,99	88,01	86,15	84,24	82,18	79,96	77,75	75,61	73,11	70,36
1992	63,52	60,85	58,87	55,72	52,64	49,94	43,44	40,90	42,28	39,87	38,35	33,40
1993	41,17	42,50	44,96	44,27	43,61	43,13	42,52	42,18	41,26	40,26	39,02	38,20
1994	39,53	62,58	62,55	61,54	59,90	58,92	58,53	58,20	57,06	54,97	52,08	50,03
1995	48,86	45,57	46,78	44,28	41,88	40,05	38,37	36,93	34,53	31,53	29,21	26,24
1996	25,12	26,21	25,20	31,34	30,22	28,97	27,98	27,02	26,09	24,75	22,97	21,12
1997	23,13	41,72	53,61	51,03	53,28	52,69	52,07	51,43	51,03	49,42	47,37	45,02
1998	50,80	54,82	53,10	51,08	49,07	47,24	45,63	44,40	42,82	41,37	39,73	38,10
1999	36,79	61,67	89,46	94,76	95,38	94,71	93,25	91,29	88,96	86,74	84,04	81,57
2000	87,74	105,49	120,74	121,15	119,44	117,62	116,27	114,81	112,83	110,39	107,68	105,21
2001	110,68	153,84	199,64	206,37	207,11	206,50	206,50	205,96	204,21	201,12	197,84	194,84

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
2002	191,39	202,05	224,28	232,49	233,40	232,21	233,82	233,54	229,07	224,07	219,32	214,18
2003	210,36	207,38	206,70	203,67	200,18	197,10	194,04	190,86	187,24	183,24	179,07	174,22
2004	174,47	184,09	182,39	180,69	178,81	176,86	174,99	172,86	170,23	166,40	162,52	158,29
2005	158,80	168,05	167,86	165,57	162,46	159,61	157,35	154,90	153,84	151,66	147,94	144,30
2006	148,25	169,91	194,77	204,88	205,06	204,21	201,72	200,31	198,10	195,17	192,32	189,28
2007	191,99	193,18	201,92	199,51	195,57	192,32	189,41	186,71	183,50	179,78	176,28	172,99
2008	184,15	183,11	182,19	179,78	177,32	175,12	172,80	170,29	167,67	164,61	161,19	157,98
2009	155,47	160,69	172,22	171,19	169,33	167,29	165,50	163,60	161,38	158,54	155,59	152,59
2010	149,98	149,42	147,20	144,36	141,84	139,03	136,66	134,48	132,07	129,36	126,38	123,40
2011	125,01	137,56	153,03	153,97								

Fuente: Proyecto Especial Tacna

Anexo 2.10. Medición del Nivel del Espejo de Agua.

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1967							2 833,9	2 833,7	2 833,6	2 833,3	2 833,1	2 833,8
1968	2 833,0	2 833,1	2 834,2	2 834,1	2 834,9	2 833,8	2 833,6	2 833,5	2 833,3	2 833,0	2 832,8	2 832,4
1969	2 832,3	2 832,5	2 834,6	2 832,3	2 832,2	2 832,0	2 831,9	2 831,7	2 831,5	2 831,3	2 831,2	2 831,0
1970	2 831,2	2 831,3	2 831,5	2 831,2	2 831,0	2 830,8	2 830,7	2 830,6	2 830,3	2 830,0	2 829,8	2 829,6
1971	2 929,5	2 830,0	2 830,1	2 829,8	2 829,4	2 829,1	2 828,8	2 828,5	2 828,2	2 827,8	2 827,4	2 827,1
1972	2 828,3	2 829,1	2 831,2	2 832,2	2 831,9	2 831,7	2 831,6	2 831,4	2 831,1	2 830,9	2 830,6	2 830,4
1973	2 830,8	2 833,2	2 835,8	2 836,2	2 836,8	2 835,5	2 835,2	2 835,0	2 834,7	2 834,4	2 834,1	2 833,7
1974	2 834,9	2 837,4	2 839,0	2 838,8	2 838,3	2 838,0	2 837,6	2 837,2	2 836,7	2 836,3	2 835,9	2 835,6
1975	2 835,3	2 836,9	2 839,7	2 839,5	2 839,1	2 838,8	2 838,5	2 838,0	2 837,7	2 837,2	2 836,7	2 836,4
1976	2 837,4	2 838,7	2 839,7	2 839,4	2 838,9	2 838,4	2 837,9	2 837,1	2 836,3	2 835,1	2 834,0	2 833,2
1977	2 832,6	2 833,5	2 836,1	2 836,2	2 835,8	2 834,8	2 834,1	2 833,2	2 832,3	2 831,5	2 830,6	2 829,7
1978	2 829,9	2 830,1	2 829,5	2 829,2	2 828,7	2 828,3	2 828,0	2 928,8	2 828,4	2 827,8	2 827,3	2 826,8
1979	2 829,6	2 876,1	2 826,2	2 825,7	2 823,7	2 823,1	2 822,6	2 822,2	2 821,6	2 821,1	2 820,6	2 820,1
1980	2 819,6	2 819,1	2 818,8	2 818,3	2 817,8	2 817,2	2 816,9	2 816,4	2 815,8	2 815,1	2 814,6	2 814,1
1981	2 813,4	2 814,8	2 815,1	2 814,7	2 813,6	2 812,8	2 812,0	2 811,3	2 810,9	2 810,0	2 809,1	2 808,2
1982	2 807,7	2 807,1	2 806,4	2 805,6	2 804,8	2 804,0	2 803,3	2 802,7	2 802,2	2 801,4	2 800,7	2 800,0
1983	2 799,5	2 798,8	2 798,1	2 797,4	2 796,6	2 796,1	2 795,6	2 795,2	2 794,7	2 794,3	2 793,8	2 792,4
1984	2 792,2	2 792,9	2 793,1	2 792,4	2 791,4	2 790,5	2 789,7	2 788,9	2 788,0	2 787,4	2 786,7	2 785,9
1985	2 784,8	2 786,6	2 787,2	2 786,7	2 785,1	2 784,9	2 784,1	2 783,3	2 782,5	2 781,8	2 780,6	2 780,0
1986	2 781,1	2 782,5	2 783,4	2 783,2	2 782,6	2 781,6	2 781,1	2 780,6	2 779,8	2 779,2	2 778,2	2 777,6

(continúa)

Año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
1987	2 780,3	2 780,8	2 780,0	2 779,3	2 778,2	2 777,4	2 776,9	2 776,4	2 775,5	2 774,5	2 773,4	2 772,6
1988	2 772,0	2 771,4	2 770,7	2 769,7	2 768,9	2 768,2	2 767,4	2 766,4	2 765,6	2 765,0	2 764,4	2 763,7
1989	2 763,0	2 765,0	2 764,5	2 764,8	2 763,7	2 763,2	2 762,8	2 762,3	2 761,8	2 761,4	2 760,9	2 760,3
1990	2 759,7	2 759,2	2 758,5	2 758,0	2 756,5	2 755,9	2 755,4	2 754,8	2 754,5	2 754,0	2 753,5	2 753,5
1991	2 754,6	2 755,0	2 756,2	2 755,9	2 755,5	2 755,2	2 754,8	2 754,4	2 754,0	2 753,6	2 753,1	2 752,6
1992	2 752,0	2 751,5	2 751,2	2 750,5	2 749,9	2 749,4	2 748,1	2 747,6	2 747,1	2 746,6	2 746,3	2 746,3
1993	2 748,4	2 748,9	2 749,7	2 749,5	2 749,3	2 749,1	2 748,9	2 748,8	2 748,5	2 748,2	2 747,8	2 747,5
1994	2 747,9	2 742,9	2 752,9	2 752,7	2 752,4	2 752,2	2 752,2	2 752,1	2 751,9	2 751,5	2 751,0	2 750,6
1995	2 750,4	2 749,9	2 750,0	2 749,6	2 749,2	2 748,2	2 748,6	2 748,4	2 748,0	2 747,5	2 746,9	2 746,4
1996	2 746,2	2 746,4	2 746,2	2 745,9	2 745,6	2 745,4	2 745,2	2 745,0	2 744,8	2 744,5	2 744,1	2 743,7
1997	2 744,1	2 748,0	2 750,3	2 750,4	2 750,2	2 750,1	2 750,0	2 749,9	2 749,8	2 749,5	2 749,1	2 748,6
1998	2 749,8	2 750,5	2 750,2	2 749,8	2 749,4	2 749,1	2 748,7	2 748,5	2 748,2	2 747,9	2 747,6	2 747,2
1999	2 747,0	2 751,9	2 757,0	2 758,0	2 758,1	2 758,0	2 757,7	2 757,3	2 756,9	2 756,5	2 756,0	2 755,6
2000	2 756,7	2 759,9	2 762,5	2 762,5	2 762,3	2 761,9	2 761,7	2 761,5	2 761,1	2 760,7	2 760,2	2 759,7
2001	2 760,8	2 767,9	2 775,0	2 776,0	2 776,1	2 776,0	2 776,0	2 775,9	2 775,7	2 775,2	2 774,7	2 774,3
2002	2 773,8	2 775,4	2 778,6	2 779,8	2 779,9	2 779,8	2 780,0	2 780,0	2 779,3	2 778,6	2 777,9	2 777,2
2003	2 776,6	2 776,2	2 776,1	2 775,6	2 775,1	2 774,6	2 774,2	2 773,7	2 773,1	2 772,5	2 771,9	2 771,1
2004	2 771,2	2 772,7	2 772,4	2 772,1	2 771,8	2 771,5	2 771,3	2 770,9	2 770,5	2 769,9	2 769,3	2 768,6
2005	2 768,7	2 770,2	2 770,1	2 769,8	2 769,3	2 768,8	2 768,5	2 768,1	2 767,9	2 767,6	2 767,0	2 766,4
2006	2 767,0	2 770,5	2 774,3	2 775,8	2 775,8	2 775,7	2 775,3	2 775,1	2 774,8	2 774,3	2 773,9	2 773,4
2007	2 773,9	2 774,0	2 775,3	2 775,0	2 774,4	2 773,9	2 773,5	2 773,1	2 772,6	2 772,0	2 771,5	2 770,9
2008	2 772,7	2 772,5	2 772,4	2 772,0	2 771,6	2 771,3	2 770,9	2 770,5	2 770,1	2 769,6	2 769,1	2 768,6
2009	2 768,2	2 769,0	2 770,8	2 770,7	2 770,4	2 770,1	2 769,8	2 769,5	2 769,1	2 768,7	2 768,2	2 767,7
2010	2 767,3	2 767,2	2 766,9	2 766,4	2 766,0	2 765,5	2 765,1	2 764,8	2 764,4	2 763,9	2 763,4	2 762,9
2011	2 763,2	2 765,3	2 767,8	2 767,9								

Fuente: Proyecto Especial Tacna

ANEXO 3. CÓDIGO FUENTE

Anexo 3.1. Funciones Generales.

- Conjunto de funciones que se utilizan en varios módulos del software desarrollado. Funciones como: Función Chi-Cuadrado, Métodos de Ordenación (Burbuja), Métodos de Búsqueda.

```
Unit UnitFunciones;

interface

uses
  Math, UnitClass_Distribucion;

type
  TArrayInteger= array of integer;
  TArrayDouble= array of double;
  TArrayString= array of string;
  TArrayBoolean= array of boolean;
  tMatrizInteger= array of TArrayInteger;
  tMatrizDouble= array of TArrayDouble;
  tTipo_Dist=(tNone, tNormal, tExponencial, tLog_Normal, tGamma, tUniforme);

function Tabla_Chi2(GI: integer; Alfa: real): real;
function Burbuja(Numeros: TArrayInteger): TArrayInteger;
function Busqueda(Numeros: TArrayInteger; Numero: integer; var Posicion: integer): boolean;

implementation

function Tabla_Chi2(GI: integer; Alfa: real): real;
var
  Chi: tChi_square_distribution;
  q: extended;
begin
  Chi:=tChi_square_distribution.create(GI);
  q:=1-Alfa;
  result:=Chi.quantile(q);
  Chi.free;
end;

function Burbuja(Numeros: TArrayInteger): TArrayInteger;
var
  i, j, temp: integer;
begin
  for i:=High(Numeros)-1 downto Low(Numeros) do
  begin
    for j:=Low(Numeros) to High(Numeros)-1 do
    begin
      if Numeros[j]>Numeros[j+1] then
      begin
        temp:=Numeros[j];
        Numeros[j]:=Numeros[j+1];
        Numeros[j+1]:=temp;
      end;
    end;
  end;
end;
```

```

        end;
    end;
end;
result:=Numeros;
end;

function Busqueda(Numeros: TArrayInteger; Numero: integer; var Posicion: integer): boolean;
var
    i: integer;
    Encontro: boolean;
begin
    Posicion:=0;
    Encontro:=false;
    for i:=Low(Numeros) to High(Numeros) do
    begin
        if Numero=Numeros[i] then
        begin
            Posicion:=i;
            Encontro:=True;
        end
        end;
    Result:=Encontro;
end;
end.

```

Anexo 3.2. Clases para el Análisis de las Variables Hidrológicas.

- Clase donde contiene los datos de las mediciones, y también donde se realizan los cálculos de: promedio, varianza, desviación estándar, Máximo Valor, Mínimo Valor, Rango, etc.

```

unit UnitClass_Medicion;

interface

uses
    Classes, Math, UnitFunciones;

type
    tCalculoValores=class
    private
        FValor: TArrayDouble;
        FCantidad: integer;
        FTotal: double;
        FPromedio: double;
        FVarianza: double;
        FDesv_Std: double;
        FMaximo: double;
        FMinimo: double;
        FRango: double;
        FNum_Intervalos: integer;
        FIncremento: double;
        FIntervalos: TArrayDouble;
    public
        constructor Create;

```

```

procedure Insertar(Valor: double);
procedure Calcular_Datos;
procedure Obtener_Distribucion;
published
property Valor: TArrayDouble read FValor write FValor;
property Cantidad: integer read FCantidad write FCantidad;
property Total: double read FTtotal write FTtotal;
property Promedio: double read FPromedio write FPromedio;
property Varianza: double read FVarianza write FVarianza;
property Desv_Std: double read FDesv_Std write FDesv_Std;
property Maximo: double read FMaximo write FMaximo;
property Minimo: double read FMinimo write FMinimo;
property Rango: double read FRango write FRango;
property Num_Intervalos: integer read FNum_Intervalos write FNum_Intervalos;
property Incremento: double read FIncremento write FIncremento;
property Intervalos: TArrayDouble read FIntervalos write FIntervalos;
end;

type
tArrayValorMes=array of tCalculoValores;

type
tMedicion=class
private
FValor_Mes: TArrayValorMes;
FTipo_Medicion: string;
public
constructor Create(Medicion: string);
published
property Tipo_Medicion: string read FTipo_Medicion write FTipo_Medicion;
property Valor_Mes: TArrayValorMes read FValor_Mes write FValor_Mes;
end;

implementation

constructor tCalculoValores.Create;
begin
FCantidad:=0;
SetLength(FValor, 0);

FTtotal:=0;
FPromedio:=0;
FVarianza:=0;
FDesv_Std:=0;
FMaximo:=0;
FMinimo:=0;
FRango:=0;
FNum_Intervalos:=0;
FIncremento:=0;
SetLength(FIntervalos, 0);
end;

procedure tCalculoValores.Insertar(Valor: Double);
begin
Inc(FCantidad);
SetLength(FValor, FCantidad);
FValor[FCantidad-1]:=Valor;
if FCantidad>1 then
Calcula_Datos;
end;

procedure tCalculoValores.Calcular_Datos;
var

```

```

temp: double;
i: integer;
begin
  if FCantidad>1 then
    begin
      FTtotal:=FloatCut(Sum(FValor),3);
      FPromedio:=FloatCut(Mean(FValor),3);
      FVarianza:=FloatCut(Variance(FValor),3);
      FDesv_Std:=FloatCut(StdDev(FValor),3);
      FMaximo:=FloatCut(Maximo_Valor(FValor),3);
      FMinimo:=FloatCut(Minimo_Valor(FValor),3);
      FRango:=FloatCut(FMaximo-FMinimo,3);

      if FMaximo>0 then
        begin
          temp:=1+3.32*Log10(FCantidad);
          FNum_Intervalos:=Trunc(temp);
          if FNum_Intervalos+0.2<temp then
            FNum_Intervalos:=FNum_Intervalos+1;
          end
        end
        else
          FNum_Intervalos:=1;

          FIncremento:=FloatCut(FRango/FNum_Intervalos,3);
        end
      else
        begin
          FTtotal:=FValor[0];
          FPromedio:=FValor[0];
          FVarianza:=0;
          FDesv_Std:=0;
          FMinimo:=FValor[0];
          FMaximo:=FValor[0];
          FRango:=0;
          FNum_Intervalos:=0;
          FIncremento:=0;
        end;
      end;

      procedure tCalculoValores.Obtener_Distribucion;
      begin

      end;

      constructor tMedicion.Create(Medicion: string);
      var
        i: integer;
      begin
        SetLength(FValor_Mes, 13);
        FTipo_Medicion:=Medicion;

        for i:= 0 to 12 do
          FValor_Mes[i]:=tCalculoValores.Create;
        end;
      end.

```

- Clase donde se realizan las Pruebas de Bondad de Ajuste, de cada Variable Hidrológica, obteniendo así el tipo de distribución a la que pertenecen.

```

unit UnitClass_Bondad;

interface

uses
  Classes, Math, UnitClass_Distribucion, UnitFunciones, UnitClass_Medicion;

type
  tPrueba_Ajuste=class
  private
    FPromedio: double;
    FVarianza: double;
    FMaximo: double;
    FMinimo: double;
    FCantidad: integer;
    FIntervalos: TArrayDouble;
    FNum_Intervalos: integer;
    FFrecuencia_Observada: TArrayInteger;
    FFrecuencia_Esperada: TArrayDouble;
    FProbabilidad_Acumulada: TArrayDouble;
    FProbabilidad_Esperada: TArrayDouble;
    FSemi_Chi: TArrayDouble;
    FChi_Calculado: double;
    FChi_Tabla: double;
    FEstado_Prueba: boolean;
  public
    constructor Create(Intervalos: TArrayDouble; Frecuencia_Observada: TArrayInteger; Valores:
tCalculoValores);
    procedure Intervalo_Unico;
    procedure Calcular_PF_Esperadas;
    procedure Calcular_Chi;
    procedure Realiza_Prueba(Alfa: double; Grados_Libertad: integer);
  published
    property Frecuencia_Esperada: TArrayDouble read FFrecuencia_Esperada write FFrecuencia_Esperada;
    property Probabilidad_Acumulada: TArrayDouble read FProbabilidad_Acumulada write
FProbabilidad_Acumulada;
    property Probabilidad_Esperada: TArrayDouble read FProbabilidad_Esperada write
FProbabilidad_Esperada;
    property Semi_Chi: TArrayDouble read FSemi_Chi write FSemi_Chi;
    property Chi_Calculado: double read FChi_Calculado write FChi_Calculado;
    property Chi_Tabla: double read FChi_Tabla write FChi_Tabla;
    property Estado_Prueba: boolean read FEstado_Prueba write FEstado_Prueba;
  end;

type
  tAjuste_Normal=class(tPrueba_Ajuste)
  private
    FN: tNormal_distribution;
  public
    procedure Calcular_PEA;
  end;

type
  tAjuste_Exponencial=class(tPrueba_Ajuste)

```

```

private
    FE: tExponential_distribution;
public
    procedure Calcular_PEA;
end;

type
tAjuste_LogNormal=class(tPrueba_Ajuste)
private
    FLN: tLognormal_distribution;
public
    procedure Calcular_PEA;
end;

type
tAjuste_Gamma=class(tPrueba_Ajuste)
private
    FG: tGamma_distribution;
public
    procedure Calcular_PEA;
end;

type
tAjuste_Uniforme=class(tPrueba_Ajuste)
private
    FU: tUniform_distribution;
public
    procedure Calcular_PEA;
end;

type
tBondad_Ajuste=class
private
    FValores: tCalculoValores;
    FIntervalos: TArrayDouble;
    FFrecuencia_Observada: TArrayInteger;
    FDist_Normal: TAjuste_Normal;
    FDist_Exponencial: TAjuste_Exponencial;
    FDist_Log_Normal: TAjuste_LogNormal;
    FDist_Gamma: TAjuste_Gamma;
    FDist_Uniforme: TAjuste_Uniforme;
    FTipo_Distribucion: TTipo_Dist;
    FValue_Distribucion: double;
    procedure Calcular_Intervalos;
    procedure Calcular_Frecuencias_Observadas;
    procedure Escoger_Dist_Optima;
public
    constructor Create(Valores: tCalculoValores);
    procedure Ejecutar_Prueba(Alfa: double);
published
    property Frec_Observada: TArrayInteger read FFrecuencia_Observada;
    property Intervalos: TArrayDouble read FIntervalos write FIntervalos;
    property Dist_Normal: TAjuste_Normal read FDist_Normal write FDist_Normal;
    property Dist_Exponencial: TAjuste_Exponencial read FDist_Exponencial write FDist_Exponencial;
    property Dist_Log_Normal: TAjuste_LogNormal read FDist_Log_Normal write FDist_Log_Normal;
    property Dist_Gamma: TAjuste_Gamma read FDist_Gamma write FDist_Gamma;
    property Dist_Uniforme: TAjuste_Uniforme read FDist_Uniforme write FDist_Uniforme;
    property Tipo_Distribucion: TTipo_Dist read FTipo_Distribucion write FTipo_Distribucion;
    property Value_Distribucion: double read FValue_Distribucion write FValue_Distribucion;
end;

implementation

```

```

constructor tPrueba_Ajuste.Create(Intervalos: TArrayDouble; Frecuencia_Observada: TArrayInteger;
Valores: tCalculoValores);
begin
    FIntervalos:=Intervalos;
    FFrecuencia_Observada:=Frecuencia_Observada;
    Valores.Calcular_Datos;
    SetLength(FFrecuencia_Esperada, Valores.Num_Intervalos);
    SetLength(FProbabilidad_Esperada, Valores.Num_Intervalos);
    SetLength(FProbabilidad_Acumulada, Valores.Num_Intervalos);
    SetLength(FSemi_Chi, Valores.Num_Intervalos);
    FPromedio:=Valores.Promedio;
    FVarianza:=Valores.Varianza;
    FMaximo:=Valores.Maximo;
    FMinimo:=Valores.Minimo;
    FNum_Intervalos:=Valores.Num_Intervalos;
    FCantidad:=Valores.Cantidad;
end;

procedure tPrueba_Ajuste.Calcular_PF_Esperadas;
var
    i: integer;
begin
    FProbabilidad_Esperada[0]:=FProbabilidad_Acumulada[0];
    FFrecuencia_Esperada[0]:=FProbabilidad_Esperada[0]*FCantidad;

    for i:=1 to FNum_Intervalos-1 do
        begin
            FProbabilidad_Esperada[i]:=Abs(FProbabilidad_Acumulada[i]-FProbabilidad_Acumulada[i-1]);
            FFrecuencia_Esperada[i]:=FProbabilidad_Esperada[i]*FCantidad;
        end;
    end;

procedure tPrueba_Ajuste.Calcular_Chi;
var
    i: integer;
    Total: double;
begin
    Total:=0;
    for i:=0 to FNum_Intervalos-1 do
        begin
            if FFrecuencia_Esperada[i]>0 then
                FSemi_Chi[i]:=(sqr(FFrecuencia_Observada[i]-FFrecuencia_Esperada[i])/FFrecuencia_Esperada[i])
            else
                FSemi_Chi[i]:=0;
                Total:=Total+FSemi_Chi[i];
            end;
        end;
    Chi_Calculado:=Total;
end;

procedure tPrueba_Ajuste.Realiza_Prueba(Alfa: double; Grados_Libertad: integer);
begin
    FChi_tabla:=Tabla_Chi2(Grados_Libertad, Alfa);
    if Chi_Calculado>0 then
        begin
            if FChi_Calculado<FChi_Tabla then
                FEstado_Prueba:=true
            else
                FEstado_Prueba:=false;
            end
        else
            FEstado_Prueba:=false;
        end;
end;

```

```

procedure tPrueba_Ajuste.Intervalo_Unico;
begin
  FProbabilidad_Acumulada[0]:=0;
  FFrecuencia_Esperada[0]:=0;
  FProbabilidad_Esperada[0]:=0;
  FChi_Calculado:=0;
end;

procedure tAjuste_Normal.Calcular_PEA;
var
  i: integer;
begin
  FN:=tNormal_distribution.create(FPromedio, FVarianza);
  for i:=Low(FIntervalos) to High(FIntervalos)-1 do
    FProbabilidad_Acumulada[i]:=FloatCut(FN.cdf(FIntervalos[i+1]), 3);
  FN.free;
end;

procedure tAjuste_Exponencial.Calcular_PEA;
var
  i: integer;
begin
  FE:=tExponential_distribution.create(1/FPromedio);
  for i:=Low(FIntervalos) to High(FIntervalos)-1 do
    FProbabilidad_Acumulada[i]:=FloatCut(FE.cdf(FIntervalos[i+1]), 3);
  FE.free;
end;

procedure tAjuste_LogNormal.Calcular_PEA;
var
  i: integer;
begin
  FLN:=tLognormal_distribution.create(FPromedio, FVarianza);
  for i:=Low(FIntervalos) to High(FIntervalos)-1 do
    FProbabilidad_Acumulada[i]:=FloatCut(FLN.cdf(FIntervalos[i+1]), 3);
  FLN.free;
end;

procedure tAjuste_Gamma.Calcular_PEA;
var
  i: integer;
begin
  FG:=tGamma_distribution.create(FPromedio, sqrt(FPromedio/FVarianza));
  for i:=Low(FIntervalos) to High(FIntervalos)-1 do
    FProbabilidad_Acumulada[i]:=FloatCut(FG.cdf(FIntervalos[i+1]), 3);
  FG.free;
end;

procedure tAjuste_Uniforme.Calcular_PEA;
var
  i: integer;
begin
  FU:=tUniform_distribution.create(FMinimo, FMaximo);
  for i:=Low(FIntervalos) to High(FIntervalos)-1 do
    FProbabilidad_Acumulada[i]:=FloatCut(FU.cdf(FIntervalos[i+1]), 3);
  FU.free;
end;

procedure tBondad_Ajuste.Escoger_Dist_Optima;
begin
  if FDist_Normal.Estado_Prueba then
    begin
      if FDist_Normal.Chi_Calculado<FValue_Distribucion then

```

```

begin
    FValue_Distribucion:=FDist_Normal.Chi_Calculado;
    FTipo_Distribucion:=tNormal;
end;
end;
if FDist_Exponencial.Estado_Prueba then
begin
    if FDist_Exponencial.Chi_Calculado<FValue_Distribucion then
    begin
        FValue_Distribucion:=FDist_Exponencial.Chi_Calculado;
        FTipo_Distribucion:=tExponencial;
    end;
end;
if FDist_Log_Normal.Estado_Prueba then
begin
    if FDist_Log_Normal.Chi_Calculado<FValue_Distribucion then
    begin
        FValue_Distribucion:=FDist_Log_Normal.Chi_Calculado;
        FTipo_Distribucion:=tLog_Normal;
    end;
end;
if FDist_Gamma.Estado_Prueba then
begin
    if FDist_Gamma.Chi_Calculado<FValue_Distribucion then
    begin
        FValue_Distribucion:=FDist_Gamma.Chi_Calculado;
        FTipo_Distribucion:=tGamma;
    end;
end;
if FDist_Uniforme.Estado_Prueba then
begin
    if FDist_Uniforme.Chi_Calculado<FValue_Distribucion then
    begin
        FValue_Distribucion:=FDist_Uniforme.Chi_Calculado;
        FTipo_Distribucion:=tUniforme;
    end;
end;
end;
end;

constructor tBondad_Ajuste.Create(Valores: tCalculoValores);
begin
    FValores:=Valores;
    FValores.Calcular_Datos;
    SetLength(FIntervalos, FValores.Num_Intervalos+1);
    SetLength(FFrecuencia_Observada, FValores.Num_Intervalos);
    FTipo_Distribucion:=tNone;
    FValue_Distribucion:=9999;
end;

procedure tBondad_Ajuste.Calcular_Intervalos;
var
    i: integer;
begin
    FIntervalos[0]:=FValores.Minimo;
    for i:=1 to FValores.Num_Intervalos do
        FIntervalos[i]:=FIntervalos[i-1]+FValores.Incremento;
    FIntervalos[FValores.Num_Intervalos]:=FValores.Maximo;
end;

procedure tBondad_Ajuste.Calcular_Frecuencias_Observadas;
var
    i, j: integer;
begin

```

```

for i:=Low(FFrecuencia_Observada) to High(FFrecuencia_Observada) do
  FFrecuencia_Observada[i]:=0;

for j:=Low(FFrecuencia_Observada) to High(FFrecuencia_Observada) do
begin
  if j=High(FFrecuencia_Observada) then
  begin
    for i:=Low(FValores.Valor) to High(FValores.Valor) do
    begin
      if ((FValores.Valor[i])>=FIntervalos[j]) and
        ((FValores.Valor[i])<=FIntervalos[j+1]) then
        FFrecuencia_Observada[j]:=FFrecuencia_Observada[j]+1;
      end;
    end
  else
  begin
    for i:=Low(FValores.Valor) to High(FValores.Valor) do
    begin
      if ((FValores.Valor[i])>=FIntervalos[j]) and
        ((FValores.Valor[i])<FIntervalos[j+1]) then
        FFrecuencia_Observada[j]:=FFrecuencia_Observada[j]+1;
      end;
    end;
  end;
end;

procedure tBondad_Ajuste.Ejecutar_Prueba(Alfa: double);
begin
  Calcular_Intervalos;
  FDist_Normal:=TAjuste_Normal.Create(FIntervalos, FFrecuencia_Observada, FValores);
  FDist_Exponencial:=TAjuste_Exponencial.Create(FIntervalos, FFrecuencia_Observada, FValores);
  FDist_Log_Normal:=TAjuste_LogNormal.Create(FIntervalos, FFrecuencia_Observada, FValores);
  FDist_Gamma:=TAjuste_Gamma.Create(FIntervalos, FFrecuencia_Observada, FValores);
  FDist_Uniforme:=TAjuste_Uniforme.Create(FIntervalos, FFrecuencia_Observada, FValores);
  Calcular_Frecuencias_Observadas;
  if FValores.Maximo>0 then
  begin
    FDist_Normal.Calcular_PEA;
    FDist_Exponencial.Calcular_PEA;
    FDist_Log_Normal.Calcular_PEA;
    FDist_Gamma.Calcular_PEA;
    FDist_Uniforme.Calcular_PEA;
    FDist_Normal.Calcular_PF_Esperadas;
    FDist_Exponencial.Calcular_PF_Esperadas;
    FDist_Log_Normal.Calcular_PF_Esperadas;
    FDist_Gamma.Calcular_PF_Esperadas;
    FDist_Uniforme.Calcular_PF_Esperadas;
    FDist_Normal.Calcular_Chi;
    FDist_Exponencial.Calcular_Chi;
    FDist_Log_Normal.Calcular_Chi;
    FDist_Gamma.Calcular_Chi;
    FDist_Uniforme.Calcular_Chi;
    FDist_Normal.Realiza_Prueba(Alfa, (High(FIntervalos)-Low(FIntervalos))-1);
    FDist_Exponencial.Realiza_Prueba(Alfa, (High(FIntervalos)-Low(FIntervalos))-1);
    FDist_Log_Normal.Realiza_Prueba(Alfa, (High(FIntervalos)-Low(FIntervalos))-1);
    FDist_Gamma.Realiza_Prueba(Alfa, (High(FIntervalos)-Low(FIntervalos))-1);
    FDist_Uniforme.Realiza_Prueba(Alfa, (High(FIntervalos)-Low(FIntervalos))-1);
  end
  else
  begin
    FDist_Normal.Intervalo_Unico;
    FDist_Exponencial.Intervalo_Unico;
    FDist_Log_Normal.Intervalo_Unico;
  end;
end;

```

```

    FDist_Gamma.Intervalo_Unico;
    FDist_Uniforme.Intervalo_Unico;
end;

Escoger_Dist_Optima;

if FTipo_Distribucion=tNone then
begin
    if (FDist_Normal.Chi_Calculado<FValue_Distribucion) and
        (FDist_Normal.Chi_Calculado>0) then
    begin
        FValue_Distribucion:=FDist_Normal.Chi_Calculado;
        FTipo_Distribucion:=tNormal;
    end;
    if (FDist_Exponencial.Chi_Calculado<FValue_Distribucion) and
        (FDist_Exponencial.Chi_Calculado>0) then
    begin
        FValue_Distribucion:=FDist_Exponencial.Chi_Calculado;
        FTipo_Distribucion:=tExponencial;
    end;
    if (FDist_Log_Normal.Chi_Calculado<FValue_Distribucion) and
        (FDist_Log_Normal.Chi_Calculado>0) then
    begin
        FValue_Distribucion:=FDist_Log_Normal.Chi_Calculado;
        FTipo_Distribucion:=tLog_Normal;
    end;
    if (FDist_Gamma.Chi_Calculado<FValue_Distribucion) and
        (FDist_Gamma.Chi_Calculado>0) then
    begin
        FValue_Distribucion:=FDist_Gamma.Chi_Calculado;
        FTipo_Distribucion:=tGamma;
    end;
    if (FDist_Uniforme.Chi_Calculado<FValue_Distribucion) and
        (FDist_Uniforme.Chi_Calculado>0) then
    begin
        FValue_Distribucion:=FDist_Uniforme.Chi_Calculado;
        FTipo_Distribucion:=tUniforme;
    end;
end;
end;
end.

```

- Para la Implementación de la siguiente clase se utilizó el libro de (Canavos, 1998), dicha clase permite calcular la Distribución de Probabilidad de la Normal, Exponencial, Logaritmo Normal, Gamma y Uniforme, y las respectivas probabilidades inversas; permitiendo así realizar las pruebas de bondad de ajuste, y la generación de las variables hidrológicas aleatorias para las simulaciones.

```

Unit UnitClass_Distribucion;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  math;

type

  EArgumentError = class(Exception);

  range = record
    lower, upper:extended;
  end;

  parameter = record
    value: extended;
    integertype: boolean;
    bound: range;
  end;

  parameterarray = array[1..maxparametercount] of parameter;

  tDistribution = class
  private
    Fparametercount: 0..maxparametercount;
    Fparameters: parameterarray;
    support: range;
    function parameter_ok(i:integer):boolean;
  public
    constructor Create;
    destructor free;
    function Setparameters(par1,par2,par3:extended):boolean; virtual;
    function in_support(x:extended):boolean; virtual;
    function density(x:extended):extended; virtual;
    function probability(x:extended): extended; virtual;
    function cdf(x:extended):extended; virtual;
    function prob_between(a,b:extended):extended; virtual;
    function quantile(var q:extended):extended; virtual; abstract;
    function expectation_exists:boolean;virtual;
    function variance_exists:boolean;virtual;
    function Mean:extended; virtual; abstract;
    function Variance:extended; virtual; abstract;

```

```

function SD:extended; virtual;
function random_value:extended; virtual;
end;

tContinuous_distribution = class(tDistribution)
public
function quantile(var q:extended):extended; override;
end;

tNormal_distribution = class(tContinuous_distribution)
private
mu, sigma2, sigma, const_part, random_const: extended;
public
function Setparameters(par1,par2,par3:extended):boolean; override;
constructor create(newmu, newsigma2:extended);
function density(x:extended):extended; override;
function cdf(x:extended):extended; override;
function random_value:extended; override;
function Mean:extended; override;
function Variance:extended; override;
end;

tGamma_distribution = class(tContinuous_distribution)
private
alpha, lambda, const_part, random_const1, random_const2: extended;
public
function Setparameters(par1,par2,par3:extended):boolean; override;
constructor create(newalpha, newlambda:extended);
function in_support(x:extended):boolean; override;
function density(x:extended):extended; override;
function cdf(x:extended):extended; override;
function random_value:extended; override;
function Mean:extended; override;
function Variance:extended; override;
end;

tChi_square_distribution = class(tGamma_distribution)
private
df: extended;
public
function Setparameters(par1,par2,par3:extended):boolean; override;
constructor create(newdf:extended);
end;

tExponential_distribution = class(tGamma_distribution)
private
theta: extended;
public
function Setparameters(par1,par2,par3:extended):boolean; override;
constructor create(newtheta:extended);
function quantile(var q:extended):extended; override;
end;

tLognormal_distribution = class(tContinuous_distribution)
private
mu, sigma2, sigma, const_part: extended;
norm: tNormal_distribution;
public
function Setparameters(par1,par2,par3:extended):boolean; override;
constructor create(newmu, newsigma2:extended);
destructor free;
function in_support(x:extended):boolean; override;
function density(x:extended):extended; override;

```

```

function cdf(x:extended):extended; override;
function random_value:extended; override;
function Mean:extended; override;
function Variance:extended; override;
end;

tUniform_distribution = class(tContinuous_distribution)
private
  alpha, beta:extended;
public
  function Setparameters(par1,par2,par3:extended):boolean; override;
  constructor create(newalpha, newbeta:extended);
  function density(x:extended):extended; override;
  function cdf(x:extended):extended; override;
  function random_value:extended; override;
  function Mean:extended; override;
  function Variance:extended; override;
end;

implementation

function approx(x, y:extended):Boolean;
Begin
  result:= abs(x-y) < eps;
End;

Function nearint(x: extended): Boolean;
Begin
  result:= (Abs(x) < maxlongint) and approx(x,round(x));
End;

function ExtToStr(x:extended):string;
begin
  result:=FormatFloat('0.#####',x);
end;

function Check(condition:boolean; err_nr:integer):boolean;
begin
  if condition then
    raise EArgumentError.Create(LoadStr(err_nr));
  result:=not condition;
end;

function Checkval(val1:extended; err_nr:integer; val2:extended):boolean;
begin
  result:=false;
  case err_nr of
    err_val_l,err_val2_l,err_val3_l: result:=val1<val2;
    err_val_le,err_val2_le,err_val3_le: result:=val1<=val2;
    err_val_g: result:=val1>val2;
    err_val_ge,err_val2_ge,err_val3_ge: result:=val1>=val2;
  end;
  if result then
    raise EArgumentError.Create(LoadStr(err_nr)+ExtToStr(val2));
  result:=not result;
end;

Function factorial(n: extended): extended;
Var
  i:integer;
Begin
  result:= 1;
  for i:=2 to round(n) do

```

```

    result:=result/i;
    if result*maxlongint>1 then
        result:=round(1/result)
    else
        result:=1/result;
    End;

Function log_choose(n, m: extended): extended;
Var
    i: integer;
Begin
    result:=0;
    if Checkval(n,err_val_le,0) and Check(not nearint(n),err_val_int)
        and Check(n>maxlongint,err_overflow) and Checkval(m,err_val_l,0)
        and Check(not nearint(m),err_val_int) and Check(m>n,err_val1_l_val2)
    then
        begin
            If n - m < m Then
                m := n - m;
            For i := 1 To Round(m) Do
                result := result + Ln((n - i + 1) / i);
            end;
        End;

function Floor(x: Extended): Integer;
begin
    Result := Trunc(x);
    if Frac(x) < 0 then
        Dec(Result);
end;

Function log_gamma(x:extended):extended; {Numerical Recipes in Pascal}
var
    y,tmp,ser: extended;
    j: integer;
    cof: array[1..6] of extended;
begin
    cof[1]:=76.18009172947146; cof[2]:=-86.50532032941677;
    cof[3]:=24.01409824083091; cof[4]:=-1.231739572450155;
    cof[5]:=0.1208650973866179E-2; cof[6]:=-0.5395239384953E-5;
    y:=x-1; tmp:=y+5.5; tmp:=(y+0.5)*ln(tmp)-tmp; ser:=1;
    for j:=1 to 6 do
        begin
            y:=y+1; ser:=ser+cof[j]/y;
        end;
    result:=tmp+ln(2.5066282746310005*ser);
end;

Function incgamma(x, a: extended):extended; {Numerical Recipes In Pascal,p.181}
function gser(x, a:extended):extended; { Numerical Recipes In Pascal, p.181}
var n:integer; sum,del,ap:extended;
begin
    if approx(0,x) then result:=0
    else
        begin
            ap:=a; sum:=1/a; del:=sum; n:=1;
            repeat
                ap:=ap+1;
                del:=del*x/ap;
                sum:=sum+del;
                inc(n);
            until (abs(del) < abs(sum)*eps) or (n>100);
            result:=sum*exp(-x+a*ln(x)-log_gamma(a));
        end;
end;

```

```

end;
end;

function gcf(x, a:extended):extended; { Numerical Recipes In Pascal, p.181}
var n:integer; gold,g,fac,b1,b0,anf,ana,a1,a0:extended;
begin
g:=0; gold:=1; a0:=1; a1:=x; b0:=0; b1:=1; fac:=1; n:=1;
repeat
ana:=n-a;
a0:=(a1+a0*ana)*fac; b0:=(b1+b0*ana)*fac; anf:=n*fac;
a1:=x*a0+anf*a1; b1:=x*b0+anf*b1;
if a1<>0 then
begin
gold:=g;
fac:=1/a1;
g:=b1*fac;
end;
inc(n);
until abs((g-gold)/g)<eps;
result:=exp(-x+a*ln(x)-log_gamma(a))*g;
end;
begin {incgamma}
result:=1;
if Checkval(x,err_val_l,0) and Check(x>maxvalue,err_overflow)
and Checkval(a,err_val_le,0) and Check(a>maxvalue,err_overflow)
then
if x < a+1 then
result:= gser(x,a)
else
result:= 1 - gcf(x,a);
end; {incgamma}

Function incbeta(x, a, b: extended): extended;
Var
bt:extended;

function betacf(x,a,b:extended):extended;
var tem, qap, qam, qab,d, bz, bpp,bp,bm,az,app, am,aold,ap:extended;
m:integer;
begin
am:=1; bm:=1; az:=1;
qab:=a+b; qap:=a+1; {these q's will be used in factors which occur in the coeffs}
qam:=a-1; bz:=1-qab*x/qap; m:=1;
repeat
tem:=2*m;
d:=m*(b-m)*x/((qam+tem)*(a+tem));
ap:=az+d*am; bp:=bz+d*bm;
d:=-(a+m)*(qab+m)*x/((a+tem)*(qap+tem));
app:=ap+d*az; bpp:=bp+d*bz;
aold:=az;
am:=ap/bpp; bm:=bp/bpp; az:=app/bpp;
bz:=1; inc(m);
until abs(az-aold) < eps*abs(az);
result:=az;
end;
Begin {incbeta}
try
if x<=0 then
result:=0
else
if x>=1 then
result:=1
else

```

```

begin
  bt:=exp(a*ln(x)+b*ln(1-x)+log_gamma(a+b)-log_gamma(a)-log_gamma(b));
  if x < (a+1)/(a+b+2) then
    result:= bt*betacf(x,a,b)/a
  else
    result:=1.0 - bt*betacf(1-x,b,a)/b;
  end;
except
  result:=1;
end;
end; {incbeta}

Function randomdraw:extended;
const
  m1=2147483563; a1=40014; m2=2147483399;
  a2=40692; q1=53668; q2=52774;
var
  Z,k:integer;
begin
  k:=seed1 div q1;
  seed1:=a1*(seed1-k*q1) - k*12211;
  if seed1<0 then
    seed1:=seed1+m1;
  k:=seed2 div q2;
  seed2:=a2*(seed2-k*q2) - k*3791;
  if seed2<0 then
    seed2:=seed2+m2;
  Z:=seed1-seed2;
  if Z<1 then
    Z:=Z+m1-1;
  result:=Z*4.656613E-10;
end;

constructor tDistribution.create;
begin
  Fparametercount:=2;
  support.lower := -maxvalue;
  support.upper := maxvalue;
end;

destructor tDistribution.free;
begin
  inherited destroy;
end;

function tDistribution.Setparameters(par1,par2,par3:extended):boolean;
var
  i:1..maxparametercount+1;
begin
  Fparameters[1].value:=par1;
  Fparameters[2].value:=par2;
  Fparameters[3].value:=par3;
  i:=1;
  while (i<=Fparametercount) and parameter_ok(i) do
    inc(i);
  result:=i>Fparametercount;
  support.lower:=-maxvalue; support.upper:=maxvalue;
end;

function tDistribution.parameter_ok(i:integer):boolean;
begin
  with Fparameters[i] do
    begin

```

```

if integertype then
  result:= Checkval(value,{err_val_l}id_error,bound.lower)
  and Checkval(value,{err_val_g}id_error+4*i-1,bound.upper)
  and Check(not nearint(value),{err_val_int}id_error+4*i)
else
  result:= Checkval(value,{err_val_le}id_error+4*i-3,bound.lower)
  and Checkval(value,err_val_ge{id_fout+4*i-1},bound.upper);
end;
end;

function tDistribution.SD:extended;
begin
  Check(not variance_exists,err_variance);
  result:=sqrt(Variance);
end;

function tDistribution.in_support(x:extended):boolean;
begin
  result:=(x>=support.lower) and (x<=support.upper);
end;

function tDistribution.density(x:extended):extended;
begin
  result:=0;
end;

function tDistribution.probability(x:extended): extended;
begin
  result:=0;
end;

function tDistribution.cdf(x:extended):extended;
begin
  if x<support.lower then
    result:=0
  else
    if x>=support.upper then
      result:=1
    else
      result:=prob_between(support.lower,x);
    end;
  end;
end;

function tDistribution.prob_between(a,b:extended):extended;
var
  prob:extended;
begin
  if a<support.lower then
    a:=support.lower;
  if b>support.upper then
    b:=support.upper;
  if b >= maxvalue then
    prob := 1 - cdf(a) + probability(a)
  else
    if a <= -maxvalue then
      prob := cdf(b)
    else
      prob := cdf(b) - cdf(a) + probability(a);
  if prob < 0 then
    result:= 0
  else
    if prob > 1 then
      result:= 1
    else

```

```

    result:= prob;
end;

function tDistribution.random_value:extended;
var
  x:extended;
begin
  x:=randomdraw; result:=quantile(x);
end;

function tDistribution.expectation_exists:boolean;
begin
  result:=true;
end;

function tDistribution.variance_exists:boolean;
begin
  result:=true;
end;

function tContinuous_Distribution.quantile(var q:extended):extended;
var
  pdf,dx,G,xhi,xlo,x,xold:extended;
begin
  if (not Checkval(q,err_val_l,0) or approx(0,q)) then
    quantile:=support.lower
  else
    if (not Checkval(q,err_val_g,1) or approx(1,q)) then
      quantile:=support.upper
    else
      begin
        xlo:=support.lower; xhi:=support.upper;
        if expectation_exists then
          x:=Mean
        else
          x:=(xlo+xhi)/2;

        if variance_exists then
          dx:=2*SD
        else
          dx:=8;

        if x+dx<xhi then
          xhi:=x+dx;
        if x-dx>xlo then
          xlo:=x-dx;

        while cdf(xhi)<q do
          begin
            xlo:=xhi; xhi:=x+8*(xhi-x);
            if xhi>support.upper then
              xhi:=support.upper;
          end;
          { the `while' body was not executed }
          if approx(xhi-xlo,2*dx) then
            while cdf(xlo)>q do
              begin
                xhi:=xlo; xlo:=x+8*(xhi-x);
                if xlo<support.lower then
                  xlo:=support.lower;
              end;
            x:=(xlo+xhi)/2;          { xlo<=x<=xhi and cdf(xlo)<q<cdf(xhi) }

```

```

repeat
  G:=cdf(x)-q;
  pdf:=density(x);
  xold:=x;
  if G<=0 then
    xlo:=x;
  if G>=0 then
    xhi:=x;
  { if Newton-Raphson would go outside [xlo,xhi] }
  if (xhi-xlo)*pdf<=abs(2*G) then
    begin
      x:=(xlo+xhi)/2;      { bisection }
    end
  else
    begin
      x:=x-G/pdf;        { Newton-Raphson }
    end;
  until abs(x-xold) < eps;
  result:=x;
end;
end;

constructor tNormal_distribution.Create(newmu, newsigma2:extended);
begin
  Fparametercount:=2;
  with Fparameters[1] do
    begin
      bound.lower:=-maxvalue; bound.upper:=maxvalue;
      integertype:=false; value:=newmu;
    end;
  with Fparameters[2] do
    begin
      bound.lower:=0; bound.upper:=maxvalue;
      integertype:=false; value:=newsigma2;
    end;
  Setparameters(newmu,newsigma2,0);
end;

function tNormal_distribution.Setparameters(par1,par2,par3:extended):boolean;
begin
  result:=inherited Setparameters(par1,par2,par3);
  mu:=par1;
  sigma2:=par2;
  sigma:=sqrt(sigma2);
  const_part:= -(ln(2*pi)+ln(par2))/2;
  random_const:=sqrt(2/exp(1));
end;

function tNormal_distribution.density(x:extended):extended;
begin
  if in_support(x) then
    result:= exp(const_part-sqr(x-mu)/(2*sigma2))
  else
    result:=0;
end;

function tNormal_distribution.cdf(x:extended):extended;
var
  z:extended;
begin
  z:=(x-mu)/sigma;
  if in_support(x) then
    if z>=0 then

```

```

    result:=0.5+0.5*incgamma(0.5*sqr(z),0.5)
  else
    result:=0.5-0.5*incgamma(0.5*sqr(z),0.5)
  else
    result:=inherited cdf(x);
end;

function tNormal_distribution.random_value:extended;
var
  x,u,v,x2:extended;
begin {Devroye: Non-Uniform Random Variate Generation, p.197,199}
  repeat
    u:=randomdraw;
    v:=random_const*(2*randomdraw-1);
    x:=v/u; x2:=sqr(x);
    until (x2<=6-8*u+2*sqr(u)) or ((x2<2/u-2*u) and (x2<=-4*ln(u)));
    result:=mu+sigma*x;
  end;

function tNormal_distribution.Mean:extended;
begin
  result:=mu;
end;

function tNormal_distribution.Variance:extended;
begin
  result:=sigma2;
end;

constructor tGamma_distribution.create(newalpha, newlambda:extended);
begin
  Fparametercount:=2;
  with Fparameters[1] do
    begin
      bound.lower:=0; bound.upper:=maxvalue;
      integertype:=false;
    end;
  with Fparameters[2] do
    begin
      bound.lower:=0; bound.upper:=maxvalue;
      integertype:=false;
    end;
  Setparameters(newalpha,newlambda,0);
end;

function tGamma_distribution.Setparameters(par1,par2,par3:extended):boolean;
var z:extended;
begin
  result:=inherited Setparameters(par1,par2,par3);
  alpha:=par1; lambda:=par2;
  const_part := alpha*ln(lambda)-log_gamma(alpha);
  support.lower := 0; support.upper:=maxvalue;
  if alpha>1 then
    begin {needed for simulation: random_const1=a-; random_const2:=a+}
      z:=sqr(2*alpha-1);
      random_const1:=(1-z)*sqrt(exp((alpha-1)*(ln(alpha-z)-ln(alpha-1))+z-1));
      random_const2:=(1+z)*sqrt(exp((alpha-1)*(ln(alpha+z)-ln(alpha-1))-z-1));
    end
  else if alpha<1 then
    begin
      random_const1:=(exp(1)+alpha)/exp(1); random_const2:=1/alpha;
    end;
end;

```

```

function tGamma_distribution.in_support(x:extended):boolean;
begin
  if alpha<1
  then result:=inherited in_support(x) and (x>support.lower)
  else result:=inherited in_support(x);
end;

function tGamma_distribution.density(x:extended):extended;
begin
  if in_support(x) and (x>0)
  then result:=exp(const_part+(alpha-1)*ln(x)-lambda*x)
  else
  if approx(1,alpha) and (x=0)
  then result:=lambda
  else result:=0;
end;

function tGamma_distribution.cdf(x:extended):extended;
begin
  if in_support(x)
  then
  if alpha<=500
  then result:=incgamma(x*lambda, alpha)
  else
  result:=(1+incgamma(sqrt(power(lambda*x/alpha,1/3)-1+1/(9*alpha))*4.5*alpha,0.5))/2
  else result:=inherited cdf(x);
end;

function tGamma_distribution.random_value:extended;
var u,v,w,x,x1,x2:extended; accept,qac,qrc:boolean;
begin
  if alpha=1 {exponential distribution}
  then x:=-ln(randomdraw)
  else
  if alpha>1 {unimodal}
  then {Devroye: Non-Uniform Random Variate Generation, p.203}
  begin
    repeat
      u:=randomdraw; v:=random_const1+(random_const2-random_const1)*randomdraw;
      x:=v/u; x1:=x+alpha-1; x2:=sqrt(x);
      if x>=0 then
      begin
        qac:=sqrt(x1)*(-2*sqrt(u)+8*u-6)<=-x2*(x+x1);
        qrc:=x1*(2*sqrt(u)-2)>=-u*x2;
      end
    else
      begin
        qac:=x1*(-2*sqrt(u)+8*u-6)<=-x2;
        qrc:=(alpha-1)*(2*sqrt(u)-2)>=-u*x2;
      end;
    until qac or (not qrc and (x+alpha-1>0) and (2*ln(u)+x<=(alpha-1)*ln(1+x/(alpha-1))));
    x:=x+alpha-1;
  end
  else
  begin {Devroye: Non-Uniform Random Variate Generation, p.425}
    repeat
      u:=randomdraw; v:=random_const1*u; w:=randomdraw;
      if v<=1
      then
      begin
        x:=power(v,random_const2);
        accept:=ln(w)<=-x;
      end
    until accept;
  end
end;

```

```

        end
    else
    begin
        x:=-ln(random_const2*(random_const1-v));
        accept:=w<=power(x,alpha-1);
    end;
    until accept;
    end;
    result:=x/lambda;
end;

function tGamma_distribution.Mean:extended;
begin
    result:=alpha/lambda;
end;

function tGamma_distribution.Variance:extended;
begin
    result:=alpha/sqr(lambda);
end;

constructor tChi_square_distribution.create(newdf:extended);
begin
    Inherited create(newdf/2,1/2);
end;

function tChi_square_distribution.Setparameters(par1,par2,par3:extended):boolean;
begin
    result:=inherited Setparameters(par1,par2,par3);
    df:=2*par1;
end;

constructor tExponential_distribution.create(newtheta:extended);
begin
    Inherited create(1,newtheta);
end;

function tExponential_distribution.Setparameters(par1,par2,par3:extended):boolean;
begin
    result:=inherited Setparameters(par1,par2,par3);
    Fparametercount:=1; Fparameters[1].value:=par2;
    theta:=par2;
end;

function tExponential_distribution.quantile(var q:extended):extended;
begin
    if q<=0 then result:=support.lower
    else if q>=1 then result:=support.upper
    else result:=-ln(1-q)/theta;
end;

constructor tLognormal_distribution.create(newmu, newsigma2:extended);
begin
    Fparametercount:=2;
    With Fparameters[1] do
    begin
        bound.lower:=-maxvalue; bound.upper:=maxvalue;
        integertype:=false;
    end;
    With Fparameters[2] do
    begin
        bound.lower:=0; bound.upper:=maxvalue;
        integertype:=false;
    end;
end;

```

```

    end;
    norm:=tNormal_distribution.create(newmu,newsigma2);
    Setparameters(newmu,newsigma2,0);
end;

destructor tLognormal_distribution.free;
begin
    norm.free;
    inherited free;
end;

function tLognormal_distribution.Setparameters(par1,par2,par3:extended):boolean;
begin
    result:=inherited Setparameters(par1,par2,par3);
    mu:=par1; sigma2:=par2; sigma:=sqrt(sigma2);
    support.lower:=0;
    const_part:=-ln(2*pi)+ln(sigma2)/2;
    norm.Setparameters(mu,sigma2,0);
end;

function tLognormal_distribution.in_support(x:extended):boolean;
begin
    result:=inherited in_support(x) and (x>0);
end;

function tLognormal_distribution.density(x:extended):extended;
begin
    if in_support(x)
    then result:=exp(const_part-ln(x)-sqr(ln(x)-mu)/sigma2/2)
    else result:=0;
end;

function tLognormal_distribution.cdf(x:extended):extended;
begin
    if in_support(x)
    then result:=norm.cdf(ln(x))
    else result:=inherited cdf(x);
end;

function tLognormal_distribution.random_value:extended;
begin
    result:=exp(norm.random_value);
end;

function tLognormal_distribution.mean:extended;
begin
    result:=exp(mu+sigma2/2);
end;

function tLognormal_distribution.variance:extended;
begin
    result:=exp(2*mu+2*sigma2)-exp(2*mu+sigma2);
end;

constructor tUniform_distribution.create(newalpha, newbeta:extended);
begin
    Fparametercount:=2;
    With Fparameters[1] do
        begin
            bound.lower:=-maxvalue; bound.upper:=newbeta;
            integertype:=false;
        end;
    With Fparameters[2] do

```

```

begin
  bound.lower:=newalpha; bound.upper:=maxvalue;
  integertype:=false;
end;
Setparameters(newalpha,newbeta,0)
end;

function tUniform_distribution.Setparameters(par1,par2,par3:extended):boolean;
begin
  Fparameters[1].bound.upper:=par2;
  Fparameters[2].bound.lower:=par1;
  result:=inherited Setparameters(par1,par2,par3);
  alpha:=par1; beta:=par2;
  support.lower:=alpha; support.upper:=beta;
end;

function tUniform_distribution.density(x:extended):extended;
begin
  if in_support(x) then
    result:=1/(beta-alpha)
  else
    result:=0;
end;

function tUniform_distribution.cdf(x:extended):extended;
begin
  if in_support(x) then
    result:=(x-alpha)/(beta-alpha)
  else
    result:=inherited cdf(x);
end;

function tUniform_distribution.random_value:extended;
begin
  result:=alpha+(beta-alpha)*randomdraw;
end;

function tUniform_distribution.mean:extended;
begin
  result:=(alpha+beta)/2;
end;

function tUniform_distribution.variance:extended;
begin
  result:=sqr(beta-alpha)/12;
end;

end.

```

Anexo 3.3. Componente RNA

- Componente que se encarga de realizar una RNA y así poder evitar realizar cálculos que empíricos, que dificultarían la simulación del Ciclo Hidrológico de la Laguna Aricota.

```
unit Red_Neuronal;

interface

uses
  Windows, Messages, SysUtils, Classes;

type
  TVector_Neuronas = array of Real;
  TRed_Neuronal = class;

  TNCapa=class(TCollectionItem)
  private
    function GetNumero_Neuronas: Integer;
    procedure SetNumero_Neuronas(const Value: Integer);
  protected
    v: TVector_Neuronas;
    InCentro, InRadio, Minima_Salida, Diametro: Real;
  public
    Pesos, dp: array of array of Real;
    i: ^TVector_Neuronas;
    o: TVector_Neuronas;
    constructor Create(Collection: TCollection); override;
    procedure Calcular_Salida;
    procedure BackProp(var d: TVector_Neuronas; Razon_Aprendizaje, Momento: Real);
    function Sigmoid(x: Real): Real;
    function Deriv(x: Real): Real;
    procedure Inicializar_Pesos;
    procedure SetEntrada_Min_Max(x, y: Real);
    procedure SetSalida_Min_Max(x, y: Real);
    function GetSalida_Min: Real;
    function GetSalida_Max: Real;
  published
    property Numero_Neuronas: Integer read GetNumero_Neuronas write SetNumero_Neuronas;
  end;

  TNCapaClass = class of TNCapa;

  TNCapas=class(TCollection)
  private
    FRed_Neuronal: TRed_Neuronal;
    function GetItem(Index: Integer): TNCapa;
    procedure SetItem(Index: Integer; Value: TNCapa);
  protected
    function GetOwner: TPersistent; override;
    procedure Update(Item: TCollectionItem); override;
  public
    constructor Create(Red_Neuronal: TRed_Neuronal);
    function Add: TNCapa;
    function AddItem(Item: TNCapa; Index: Integer): TNCapa;
    function Insert(Index: Integer): TNCapa;
```

```

    property Items[Index: Integer]: TNCapa read GetItem write SetItem; default;
end;

TRed_Neuronal = class(TComponent)
private
    function GetNumero_Entradas: Integer;
    procedure SetNumero_Entradas(const Value: Integer);
    function GetNumero_Salidas: Integer;
    procedure SetNumero_Salidas(const Value: Integer);
    procedure SetNCapas(const Value: TNCapas);
    function GetSalida_Max: Real;
    function GetSalida_Min: Real;
    procedure SetEntrada_Max(const Value: Real);
    procedure SetEntrada_Min(const Value: Real);
    procedure SetSalida_Max(const Value: Real);
    procedure SetSalida_Min(const Value: Real);
protected
    FEntrada_Min, FEntrada_Max: Real;
    FSalida_Min, FSalida_Max: Real;
    FNCapas: TNCapas;//Capas Ocultas
    FCapa_Sale: TNCapa;//Capa Salida
    FRazon_Aprendizaje, FMomento: Real;//Const's en el Algoritmo BackProp
    function GetNCapaClass: TNCapaClass; virtual;
    function CreateNCapa: TNCapa; virtual;
    procedure Conectar_Capas;
    procedure Inicializar_Pesos;
    function GetCapa_Entrada: TNCapa;
    function GetCapa_Salida: TNCapa;
public
    Entrada, Salida: TVector_Neuronas;//Use this to access In and Out. Don't use [0]
    Salida_Deseada: TVector_Neuronas;//Set this prior to calling BackProp
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    procedure Borrar_Capas;
    procedure Cargar_Capas(Neuronas_Capa: array of integer; Neuronas_Salida: integer);
    function Agregar_Capa: TNCapa; overload;
    procedure Calcular_Salida;
    function BackProp: Real; overload;
    function GetAvgError: Real;
    function GetMaxError: Real;
published
    property Numero_Entradas: Integer read GetNumero_Entradas write SetNumero_Entradas;
    property Numero_Salidas: Integer read GetNumero_Salidas write SetNumero_Salidas;
    property Entrada_Min: Real read GetEntrada_Min write SetEntrada_Min;
    property Entrada_Max: Real read GetEntrada_Max write SetEntrada_Max;
    property Salida_Min: Real read GetSalida_Min write SetSalida_Min;
    property Salida_Max: Real read GetSalida_Max write SetSalida_Max;
    property NCapas: TNCapas read FNCapas write SetNCapas;
    property Razon_Aprendizaje: Real read FRazon_Aprendizaje write FRazon_Aprendizaje;
    property Momento: Real read FMomento write FMomento;
end;

procedure Register;

implementation

{$R *.dcr}

procedure Register;
begin
    RegisterComponents('IA', [TRed_Neuronal]);
end;

```

```

constructor TNCapa.Create(Collection: TCollection);
begin
    Numero_Neuronas:=3;
    InCentro:=0;
    InRadio:=1;
    Minima_Salida:=-1;
    Diametro:=2;
    inherited;
end;

procedure TNCapa.SetEntrada_Min_Max(x, y: Real);
begin
    InCentro:=(x+y)/2;
    InRadio:=(y-x)/2;
end;

procedure TNCapa.SetSalida_Min_Max(x, y: Real);
begin
    Minima_Salida:=x;
    Diametro:=y-x;
end;

function TNCapa.GetSalida_Min: Real;
begin
    Result:=Minima_Salida;
end;

function TNCapa.GetSalida_Max: Real;
begin
    Result:=Minima_Salida+Diametro;
end;

procedure TNCapa.Calcular_Salida;
var
    j,k: Integer;
begin
    for j:=1 to Length(o)-1 do
        begin
            v[j]:=0;
            for k:=0 to Length(i^)-1 do
                v[j]:=v[j]+Pesos[j,k]*i^[k];
            o[j]:=Sigmoid(v[j]);
        end;
    end;
end;

procedure TNCapa.BackProp (var d: TVector_Neuronas; Razon_Aprendizaje, Momento: Real);
var
    j,k: Integer;
begin
    for j:=1 to Length(o)-1 do
        d[j]:=d[j]*Deriv(v[j]);
    for j:=1 to Length(o)-1 do
        for k:=0 to Length(i^)-1 do
            begin
                dp[j,k]:=dp[j,k]*Momento + Razon_Aprendizaje*d[j]*i^[k];
                Pesos[j,k]:=Pesos[j,k]+dp[j,k];
            end;
        for j:=2 to Length(o)-1 do
            d[1]:=d[1]+d[j];
            SetLength(d, Length(i^));
            for j:=2 to Length(d)-1 do
                d[j]:=d[1];
            end;
        end;
    end;
end;

```

```

procedure TNCapa.Inicializar_Pesos;
var
  j,k: Integer;
begin
  SetLength(Pesos, Length(o), Length(i^));
  SetLength(dp, Length(o), Length(i^));
  for j:=1 to Length(o)-1 do
    for k:=0 to Length(i^)-1 do
      begin
        Pesos[j,k]:=lnRadio*(2*(2.4/(Length(i^)-1))*random(1000)/1000-1);
        dp[j,k]:=0;
      end;
    end;
end;

function TNCapa.Deriv(x: Real): Real;
begin
  result:=(Diametro/lnRadio)*(exp(-(x-lnCentro)/lnRadio))/(sqr(1+exp(-(x-lnCentro)/lnRadio)));
end;

function TNCapa.Sigmoid(x: Real): Real;
begin
  result:=Diametro/(1+exp(-(x-lnCentro)/lnRadio))+Minima_Salida
end;

function TNCapas.Add: TNCapa;
begin
  Result := TNCapa(inherited Add);
end;

function TNCapas.AddItem(Item: TNCapa;
  Index: Integer): TNCapa;
begin
  if Item = nil then
    Result := FRed_Neuronal.CreateNCapa
  else
    begin
      Result := Item;
      if Assigned(Item) then
        begin
          Result.Collection := Self;
          if Index < 0 then
            Index := Count - 1;
          Result.Index := Index;
        end;
      end;
    end;
end;

function TNCapas.Insert(Index: Integer): TNCapa;
begin
  Result := AddItem(nil, Index);
end;

function TRed_Neuronal.GetNCapaClass: TNCapaClass;
begin
  Result:=TNCapa;
end;

constructor TNCapas.Create(Red_Neuronal: TRed_Neuronal);
begin
  if Red_Neuronal <> nil then
    inherited Create(Red_Neuronal.GetNCapaClass)
  else

```

```

    inherited Create(TNCapa);
    FRed_Neuronal := Red_Neuronal;
end;

procedure TNCapas.Update(Item: TCollectionItem);
begin
    inherited;
    FRed_Neuronal.Conectar_Capas;
    FRed_Neuronal.Inicializar_Pesos;
end;

procedure TRed_Neuronal.SetNCapas(const Value: TNCapas);
begin
    FNCapas.Assign( Value );
end;

function TNCapas.GetItem(Index: Integer): TNCapa;
begin
    Result := TNCapa(inherited GetItem(Index));
end;

function TNCapas.GetOwner: TPersistent;
begin
    Result := FRed_Neuronal;
end;

procedure TNCapas.SetItem(Index: Integer; Value: TNCapa);
begin
    inherited SetItem(Index, Value);
end;

function TRed_Neuronal.GetNumero_Entradas: Integer;
begin
    Result:=Length(Entrada)-1;
end;

procedure TRed_Neuronal.SetNumero_Entradas(const Value: Integer);
begin
    SetLength(Entrada, Value+1);
end;

constructor TRed_Neuronal.Create(AOwner: TComponent);
begin
    Inherited;
    Numero_Entradas:=2;
    Entrada[0]:=-1;
    Numero_Salidas:=1;
    Salida[0]:=-1;
    FRazon_Aprendizaje:=0.1;
    FMomento:=0.1;
    FNCapas:=TNCapas.Create( Self );
    FCapa_Sale:=TNCapa.Create(nil);
    Conectar_Capas;
    randomize;
end;

destructor TRed_Neuronal.Destroy;
begin
    Borrar_Capas;
    FNCapas.Free;
    FCapa_Sale.Free;
    Inherited Destroy;
end;

```

```

end;

procedure TRed_Neuronal.Borrar_Capas;
begin
    NCapas.Clear;
end;

procedure TRed_Neuronal.Cargar_Capas(Neuronas_Capa: array of integer;
    Neuronas_Salida: integer);
var
    i: integer;
begin
    Borrar_Capas;
    for i:=Low(Neuronas_Capa) to High(Neuronas_Capa) do
        begin
            NCapas.Add;
            NCapas[i].Numero_Neuronas:=Neuronas_Capa[i];
        end;
    FCapa_Sale.Numero_Neuronas:=Neuronas_Salida;
    Conectar_Capas;
    Inicializar_Pesos;
end;

procedure TRed_Neuronal.Calcular_Salida;
var
    i, j: Integer;
begin
    for j:=0 to FNCapas.Count-1 do
        FNCapas.Items[j].Calcular_Salida;
        FCapa_Sale.Calcular_Salida;
        for i:=1 to Length(Salida)-1 do
            Salida:=FCapa_Sale.o;
        end;
    end;

procedure TRed_Neuronal.Conectar_Capas;
var
    i: Integer;
begin
    GetCapa_Entrada.SetEntrada_Min_Max(FEntrada_Min, FEntrada_Max);
    GetCapa_Salida.SetSalida_Min_Max(FSalida_Min, FSalida_Max);

    GetCapa_Entrada.i:=@Entrada;
    if NCapas.Count>0 then
        begin
            for i:=0 to NCapas.Count-2 do
                NCapas.Items[i+1].i:=@(NCapas.Items[i].o);
                FCapa_Sale.i:=@(NCapas.Items[NCapas.Count-1].o);
            end;
        end;
end;

function TRed_Neuronal.GetCapa_Entrada: TNCapa;
begin
    if NCapas.Count>0 then
        Result:=NCapas.Items[0]
    else
        Result:=FCapa_Sale;
    end;
end;

function TRed_Neuronal.GetCapa_Salida: TNCapa;
begin
    Result:=FCapa_Sale;
end;

```

```

function TRed_Neuronal.GetSalida_Max: Real;
begin
    Result:=FSalida_Max;
end;

function TRed_Neuronal.GetSalida_Min: Real;
begin
    Result:=FSalida_Min;
end;

procedure TRed_Neuronal.SetEntrada_Max(const Value: Real);
begin
    FEntrada_Max:=Value;
    GetCapa_Entrada.SetEntrada_Min_Max(GetCapa_Entrada.GetEntrada_Min, Value);
end;

procedure TRed_Neuronal.SetEntrada_Min(const Value: Real);
begin
    FEntrada_Min:=Value;
    GetCapa_Entrada.SetEntrada_Min_Max(Value, GetCapa_Entrada.GetEntrada_Max);
end;

procedure TRed_Neuronal.SetSalida_Max(const Value: Real);
begin
    FSalida_Max:=Value;
    GetCapa_Salida.SetSalida_Min_Max(GetCapa_Salida.GetSalida_Min, Value);
end;

procedure TRed_Neuronal.SetSalida_Min(const Value: Real);
begin
    FSalida_Min:=Value;
    GetCapa_Salida.SetSalida_Min_Max(Value, GetCapa_Salida.GetSalida_Max);
end;

function TRed_Neuronal.CreateNCapa: TNCapa;
var
    LClass: TNCapaClass;
begin
    LClass := GetNCapaClass;
    Result := LClass.Create(NCapas);
end;

procedure TRed_Neuronal.Inicializar_Pesos;
var
    i: Integer;
begin
    for i:=0 to NCapas.Count-1 do
        NCapas.Items[i].Inicializar_Pesos;
    FCapa_Sale.Inicializar_Pesos;
end;

function TRed_Neuronal.GetNumero_Salidas: Integer;
begin
    Result:=Length(Salida)-1;
end;

procedure TRed_Neuronal.SetNumero_Salidas(const Value: Integer);
begin
    SetLength(Salida, Value+1);
    SetLength(Salida_Deseada, Value+1);
    if FCapa_Sale<> nil then
        FCapa_Sale.Numero_Neuronas:=Value;
    if Length(Salida)>0 then

```

```

    Salida[0]:=-1;
end;

function TNCapa.GetNumero_Neuronas: Integer;
begin
    Result:=Length(o)-1;
end;

procedure TNCapa.SetNumero_Neuronas(const Value: Integer);
begin
    SetLength(o, Value+1);
    o[0]:=-1;
    SetLength(v, Value+1);
end;

function TRed_Neuronal.Agregar_Capa: TNCapa;
var
    NCapa: TNCapa;
begin
    NCapa:=NCapas.Add;
    Conectar_Capas;
    NCapa.Inicializar_Pesos;
    Result:=NCapa;
end;

function TRed_Neuronal.BackProp: Real;
var
    j: Integer;
    d: TVector_Neuronas;
begin
    Calcular_Salida;
    SetLength(d, Length(Salida));
    for j:=1 to Length(Salida)-1 do
        d[j]:=Salida_Deseada[j]-Salida[j];
        FCapa_Sale.BackProp(d, FRazon_Aprendizaje, FMomento);
    for j:=FNCapas.Count-1 downto 0 do
        FNCapas.Items[j].BackProp(d, FRazon_Aprendizaje, FMomento);
    Result:=GetMaxError;
end;

function TRed_Neuronal.GetAvgError: Real;
var
    i: Integer;
begin
    Result:=0;
    for i:=1 to High(Salida_Deseada) do
        Result:=Result+sqr(Salida_Deseada[i]-Salida[i]);
    Result:=sqrt(Result);
end;

function TRed_Neuronal.GetMaxError: Real;
var
    i: Integer;
begin
    Result:=0;
    for i:=1 to High(Salida_Deseada) do
        if Abs(Salida_Deseada[i]-Salida[i]) > Result then
            Result:=Abs(Salida_Deseada[i]-Salida[i]);
    end;
end.

```

Anexo 3.4. Clases para el Entrenamiento de la RNA

- Clase con funciones necesarias para realizar un correcto entrenamiento de una RNA utilizando el componente “TRed_Neuronal”.

```
unit UnitClass_Entrenamiento;

interface

uses
  UnitFunciones;

type
  tAnio_Contador=class
  private
    FValor: TArrayInteger;
    FTotal: TArrayInteger;
    FCantidad: integer;
    FMinimo: integer;
    FMaximo: integer;
  public
    constructor Create;
    procedure Insertar(valor: integer);
  published
    property Valor: TArrayInteger read FValor write FValor;
    property Total: TArrayInteger read FTotal write FTotal;
    property Minimo: integer read FMinimo write FMinimo;
    property Maximo: integer read FMaximo write FMaximo;
  end;

  type
  tAnio_Mes_Contador=class
  private
    FAnio: integer;
    FMeses: TArrayInteger;
  public
    constructor Create;
  published
    property Anio: integer read FAnio write FAnio;
    property Meses: TArrayInteger read FMeses write FMeses;
  end;

  type
  TArrayAnio_Mes=array of tAnio_Mes_Contador;

  type
  tAnioMes=class
  private
    FAnio: integer;
    FMes: string;
    FId_Mes: integer;
  published
    property Anio: integer read FAnio write FAnio;
    property Mes: string read FMes write FMes;
    property Id_Mes: integer read FId_Mes write FId_Mes;
  end;

implementation
```

```

{tAnio_Contador}
constructor tAnio_Contador.Create;
begin
    FCantidad:=0;
end;

procedure tAnio_Contador.Insertar(valor: Integer);
var
    temp: integer;
procedure Insertar;
begin
    Inc(FCantidad);
    SetLength(FValor, FCantidad);
    SetLength(FTotal, FCantidad);
    FValor[FCantidad-1]:=valor;
    FTotal[FCantidad-1]:=0;

    if FCantidad>1 then
    begin
        FValor:=Burbuja(FValor);
        FMinimo:=FValor[Low(FValor)];
        FMaximo:=FValor[High(FValor)];
    end;
end;
begin
    if FCantidad>1 then
    begin
        if not(Busqueda(FValor, valor, temp)) then
            Insertar
        end
    else
        Insertar;
    end;
end;
{Fin de tAnio_Contador}

{tAnio_Mes_Contador}
constructor tAnio_Mes_Contador.Create;
begin
    SetLength(FMeses, 12);
end;
{Fin de tAnio_Mes_Contador}

end.

```

Anexo 3.5. Código Fuente: del Entrenamiento de RNA

- Función con la cual se realiza el entrenamiento de la RNA

```
procedure TFrmEntrenar_RNA.btnEntrenar;
var
  i, j, k: Integer;
begin
  ProgressBar.Max:=Total_Datos;
  ProgressBar.Position:=0;
  for k:=1 to 10000 do
  begin
    for i:=0 to Total_Datos-1 do
    begin
      for j:=Low(FVariables_Entrada) to High(FVariables_Entrada) do
        Red_Neuronal.Entrada[j+1]:=FVariables_Entrada[j,i];

      for j:=Low(FVariables_Salida) to High(FVariables_Salida) do
        Red_Neuronal.Salida_Deseada[j+1]:=FVariables_Salida[j,i];

      Red_Neuronal.BackProp;
      ProgressBar.Position:=ProgressBar.Position+1;
    end;
  end;
  Mostrar_Resultados;
end;
```

Anexo 3.6. Código Fuente: de la Simulación.

- Función con la cual se realiza simulación del ciclo hidrológico de la laguna Aricota, generando las variables hidrológicas aleatorias, si como el uso de la RNA entrenada previamente, para así obtener el volumen y nivel de espejo de agua estimados.

```
procedure TFrmSimulacion.Simulacion;
label fin_for, fin_for1, fin_for2;
var
  i, j, k: integer;
  Nombre_Mes: string;
  Tamano_Barra: integer;
  Mes_S_Inicial, Mes_S_Final: integer;
  FFecha_Inicial: tAnioMes;
  FFecha_Final: tAnioMes;
  VA_Generada: double;
  FN: tNormal_distribution;
  FE: tExponential_distribution;
  FLN: tLognormal_distribution;
  FG: tGamma_distribution;
  FU: tUniform_distribution;
begin
  FFecha_Inicial:=tAnioMes.Create;
```

```

FFecha_Inicial.Mes:=CboMesI.Text;
FFecha_Inicial.Anio:=StrToInt(txtAnioI.text);
FFecha_Inicial.Id_Mes:=CboMesI.ItemIndex;
FFecha_Final:=tAnioMes.Create;
FFecha_Final.Mes:=CboMesF.Text;
FFecha_Final.Anio:=StrToInt(txtAnioF.text);
FFecha_Final.Id_Mes:=CboMesF.ItemIndex;
Tamano_Barra:=0;
for i:=FFecha_Inicial.Anio to FFecha_Final.Anio do
begin
  {Si es el Año Inicial}
  if i=FFecha_Inicial.Anio then
  begin
    {Si el Mes Inicial es Diciembre}
    if FFecha_Inicial.Id_Mes=11 then
      goto fin_for1 {Saltar al Siguiente Año y no Simular}
    else
      Mes_S_Inicial:=FFecha_Inicial.Id_Mes+1; {Pasar al siguiente Mes}
    end
  else
    Mes_S_Inicial:=0;
  if i=FFecha_Final.Anio then
    Mes_S_Final:=FFecha_Final.Id_Mes
  else
    Mes_S_Final:=11;
  {Recorrido por Meses}
  for j:=Mes_S_Inicial to Mes_S_Final do
    Inc(Tamano_Barra);
  fin_for1:
end;

ProgressBar.Max:=ADOQVariablesH.RecordCount*Tamano_Barra;
{Recorrido de la Simulación}
ADOQVariablesH.First;
repeat
  {Recorrido por Años}
  for i:=FFecha_Inicial.Anio to FFecha_Final.Anio do
  begin
    {Si es el Año Inicial}
    if i=FFecha_Inicial.Anio then
    begin
      {Si el Mes Inicial es Diciembre}
      if FFecha_Inicial.Id_Mes=11 then
        goto fin_for {Saltar al Siguiente Año y no Simular}
      else
        Mes_S_Inicial:=FFecha_Inicial.Id_Mes+1; {Pasar al siguiente Mes}
      end
    else
      Mes_S_Inicial:=0;

    if i=FFecha_Final.Anio then
      Mes_S_Final:=FFecha_Final.Id_Mes
    else
      Mes_S_Final:=11;

    {Recorrido por Meses}
    for j:=Mes_S_Inicial to Mes_S_Final do
    begin
      ProgressBar.Position:=ProgressBar.Position+1;

      if ADOTMedicion_Laguna.FieldByName(' analisis_total').AsBoolean then
        Nombre_Mes:=DMLaguna.Meses[12]
      else

```

```

Nombre_Mes:=DMLaguna.Meses[j];

if ADODSSimular.Locate('titulo;mes', VarArrayOf([ADOQVariablesH.FieldValues['titulo'],
Nombre_Mes]), []) then
begin
if ADODSSimular.FieldValues['codigo_distribucion']='tNone' then
begin
VA_Generada:=0;
end
else if ADODSSimular.FieldValues['codigo_distribucion']='tNormal' then
begin
FN:=tNormal_distribution.create(ADODSSimular.FieldValues['promedio'],
ADODSSimular.FieldValues['varianza']);
VA_Generada:=abs(FloatCut(FN.random_value,3));
FN.Destroy;
end
else if ADODSSimular.FieldValues['codigo_distribucion']='tExponencial' then
begin
FE:=tExponential_distribution.create(1/ADODSSimular.FieldValues['promedio']);
VA_Generada:=FloatCut(FE.random_value,3);
FE.Destroy;
end
else if ADODSSimular.FieldValues['codigo_distribucion']='tLog_Normal' then
begin
FLN:=tLognormal_distribution.create(ADODSSimular.FieldValues['promedio'],
ADODSSimular.FieldValues['varianza']);
VA_Generada:=FloatCut(FLN.random_value,3);
FLN.Destroy;
end
else if ADODSSimular.FieldValues['codigo_distribucion']='tGamma' then
begin
FG:=tGamma_distribution.create(ADODSSimular.FieldValues['promedio'],
sqr(ADODSSimular.FieldValues['promedio']/ADODSSimular.FieldValues['varianza']));
VA_Generada:=FloatCut(FG.random_value,3);
FG.Destroy;
end
else if ADODSSimular.FieldValues['codigo_distribucion']='tUniforme' then
begin
FU:=tUniform_distribution.create(ADODSSimular.FieldValues['minimo'],
ADODSSimular.FieldValues['maximo']);
VA_Generada:=FloatCut(FU.random_value,3);
FU.Destroy;
end;
end;

if j=Mes_S_Inicial then
begin
ADOTSimulacion.Insert;

ADOTSimulacion.FieldValues['id_laguna_medicion']:=ADODSSimular.FieldValues['id_laguna_medicion'];
ADOTSimulacion.FieldValues['anio']:=i;
ADOTSimulacion.FieldValues[DMLaguna.Meses[j]]:=VA_Generada;
ADOTSimulacion.Post;
end
else
begin
ADOTSimulacion.Edit;
ADOTSimulacion.FieldValues[DMLaguna.Meses[j]]:=VA_Generada;
ADOTSimulacion.Post;
end;
end;

fin_for:

```

```

end;

{Recorrido siguiente}
ADOQVariablesH.Next;
until ADOQVariablesH.Eof;

ADODSDatos_Entrada.Close;
ADODSDatos_Entrada.Open;
{Recorrido por Años}
for i:=FFecha_Inicial.Anio to FFecha_Final.Anio do
begin
  {Si es el Año Inicial}
  if i=FFecha_Inicial.Anio then
  begin
    {Si el Mes Inicial es Diciembre}
    if FFecha_Inicial.Id_Mes=11 then
      goto fin_for2 {Saltar al Siguiente Año y no Simular}
    else
      Mes_S_Inicial:=FFecha_Inicial.Id_Mes+1; {Pasar al siguiente Mes}
    end
  else
    Mes_S_Inicial:=0;

    if i=FFecha_Final.Anio then
      Mes_S_Final:=FFecha_Final.Id_Mes
    else
      Mes_S_Final:=11;

    {Recorrido por Meses}
    for j:=Mes_S_Inicial to Mes_S_Final do
    begin
      ADOQVariablesH.First;
      k:=1;
      repeat
        if ADODSDatos_Entrada.Locate('id_laguna_medicion;anio',
          VarArrayOf([ADOQVariablesH.FieldValues['id_laguna_medicion'], i]), []) then
          begin
            Red_Neuronal.Entrada[k]:=ADODSDatos_Entrada.FieldValues[DMLaguna.Meses[j]]/ADODSDatos_Entrada.Fie
            ldValues[entre];
            end;
            inc(k);
            ADOQVariablesH.Next;
          until ADOQVariablesH.Eof;
          ADOQVariablesH.First;

          Red_Neuronal.Calcular_Salida;

          k:=1;
          ADOQRNA.First;
          repeat
            Valores_Iniciales[k-1]:=floatcut(Valores_Iniciales[k-1]+Red_Neuronal.Salida[k],3);
            if j=Mes_S_Inicial then
              begin
                ADOTSimulacion.Insert;

                ADOTSimulacion.FieldValues['id_laguna_medicion']:=ADOQRNA.FieldValues['id_laguna_medicion'];
                ADOTSimulacion.FieldValues['anio']:=i;
                ADOTSimulacion.FieldValues[DMLaguna.Meses[j]]:=Valores_Iniciales[k-1];
                ADOTSimulacion.Post;
              end
            else
              begin

```

```

if ADOTSimulacion.Locate('id_laguna_medicion;anio',
  VarArrayOf([ADOQRNA.FieldValues['id_laguna_medicion'], i]), []) then
begin
  ADOTSimulacion.Edit;
  ADOTSimulacion.FieldValues[DMLaguna.Meses[j]]:=Valores_Iniciales[k-1];
  ADOTSimulacion.Post;
end;
end;

  Inc(k);
  ADOQRNA.Next;
until ADOQRNA.Eof;
end;

  fin_for2:
end;
Actualizar;
end;

```