

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
Facultad 5, Laboratorio de Investigaciones en Gestión de Proyectos  
Facultad 1, Departamento Técnicas de Programación



## **SISTEMA BASADO EN TÉCNICAS DE SOFT COMPUTING PARA LA EVALUACIÓN DE LA EJECUCIÓN DE PROYECTOS**

Trabajo final presentado en opción al título de  
Máster en Gestión de Proyectos Informáticos

**Autora: Ing. Anié Bermudez Peña**

**Tutor: Dr.C Pedro Yobanis Piñero Pérez**

La Habana, Junio de 2015

## DEDICATORIA

*A mis padres Ania Peña Leyva y Edi Bermudez González por su cariño eterno y apoyo incondicional. Por estar siempre a mi lado en todo momento, especialmente en los difíciles, y por sustentar todas las decisiones que he tomado. Los quiero mucho.*

*A mis abuelos.*

## AGRADECIMIENTOS

*Este trabajo no hubiera sido posible sin la ayuda de mi tutor José Alejandro Lugo García, quien además de guía constante e incansable, me brindó su apoyo, tiempo, experiencia y amistad a lo largo del desarrollo de esta investigación. A él va dedicado mi más profundo agradecimiento.*

*Muchas gracias a mi tutor Pedro Yobanis Piñero Pérez por sus valiosos aportes y por ser el líder de esta Maestría de Excelencia.*

*Muchísimas gracias a mi familia por su confianza, paciencia, cariño y sobre todo por su apoyo incondicional.*

*Muchas gracias al profesor Gil Cruz Lemus por su apoyo en los momentos que más lo necesité.*

*Gracias a los profesores de la maestría, particularmente a Surayne.*

*A mis amigos, los de ahora y los de siempre, gracias por estar pendiente de mis avances, por su cariño y ánimo, en especial a Amaris, Luis, Yusmary, Dunia, Dianly, Daimerys, Raudel...*

*Gracias a mi colectivo de asignatura Inteligencia Artificial.*

*Finalmente, gracias a todos aquellos que de una forma u otra me han acompañado y apoyado durante esta investigación.*

## **DECLARACIÓN DE AUTORÍA**

Declaro por este medio que yo Anié Bermudez Peña, con carné de identidad 84072304195, soy la autora principal del trabajo final de maestría Sistema basado en técnicas de soft computing para la evaluación de la ejecución de proyectos, desarrollado como parte de la Maestría en Gestión de Proyectos Informáticos y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como sus derechos patrimoniales con carácter exclusivo.

Para que así conste, firmo la presente declaración jurada de autoría en La Habana a los \_\_\_\_ días del mes de junio del año 2015.

---

Nombre y Firma

## RESUMEN

Durante el control de la ejecución a sus proyectos, las organizaciones emplean disímiles herramientas para asistir a los especialistas en la toma de decisiones. Sin embargo, aún son insuficientes en ambientes con incertidumbre en la información y condiciones cambiantes en los estilos de gestión. Las deficiencias en la adaptación y el aprendizaje automático durante el control de la ejecución de proyectos, afectan la calidad de la clasificación de los mismos como apoyo a la toma de decisiones. Una alternativa de solución es la introducción de técnicas de *soft computing*, las cuales aportan robustez, eficiencia y adaptabilidad a las herramientas. La presente investigación propone un sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos, que contribuya a mejorar la calidad de la clasificación de los proyectos. La propuesta de solución realiza el aprendizaje automático y ajuste de sistemas de inferencia borrosos para la evaluación de proyectos. Los resultados se logran a partir de la ejecución de siete algoritmos: WM, ANFIS, HYFIS, FIR.DM, FS.HGD, GFS.THRIFT y GFS.LT.RS; los cuales son basados en partición espacial, redes neuronales, gradiente descendente y algoritmos genéticos. La validación del sistema propuesto, integrado con la Suite de Gestión de Proyectos Xedro-GESPRO, evidencia una mejora en la calidad de la evaluación de los proyectos.

**Palabras clave:** aprendizaje automático, evaluación de la ejecución de proyectos, sistema de inferencia borroso, *soft computing*.

## ABSTRACT

*During their projects execution control, organizations employ dissimilar tools to support managers in decision-making. However, are still insufficient in environments with uncertain information and changing conditions in management styles. The deficiencies in adaptation and machine learning for controlling the projects execution, affects the quality of their classification in aiding decision making. An alternative solution is the introduction of soft computing techniques, which provide robustness, efficiency and adaptability at tools. This research proposes a soft computing techniques-based system for project performance evaluation, which contributes to improve the quality of projects classification. The proposed system allows the machine learning and adjusting of fuzzy inference systems to the projects evaluation. The results are obtained from the execution of seven algorithms: WM, ANFIS, HYFIS, FIR.DM, FS.HGD, GFS.THRIFT and GFS.LT.RS; which are based on space partitioning, neural networks, gradient descent and genetic algorithms. The validation of the proposed system, integrated to the Xedro-GESPRO projects management tool, ratifies an improvement in the quality of the projects evaluation.*

**Keywords:** *fuzzy inference system, machine learning, project performance evaluation, soft computing.*

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1. TENDENCIAS EN EL CONTROL DE LA EJECUCIÓN DE PROYECTOS .....	9
1.1    Análisis bibliográfico.....	9
1.2    Escuelas de gestión de proyectos .....	10
1.2.1    Guía de los Fundamentos para la Dirección de Proyectos (PMBOK) .....	10
1.2.2    Modelo de Madurez y Capacidad Integrado (CMMI) .....	10
1.2.3    Proyectos en Entornos Controlados (PRINCE2) .....	11
1.2.4    Organización Internacional de Normalización (ISO) .....	11
1.2.5    Normas Cubanas para la gestión de proyectos (NC) .....	13
1.2.6    Programa de Mejora de la UCI.....	13
1.3    Herramientas informáticas para la gestión de proyectos .....	14
1.3.1    Comparación de herramientas a nivel mundial.....	14
1.3.2    Evaluación de proyectos con Xedro-GESPRO .....	15
1.4    Tratamiento de la incertidumbre durante el control de proyectos.....	17
1.4.1    Sistemas de inferencia borrosos .....	17
1.4.2    Estrategias para la adaptación y el aprendizaje automático de reglas.....	19
1.4.3    Técnicas para la generación inicial de reglas borrosas y su optimización .....	20
1.4.4    Herramientas informáticas para la aplicación de <i>soft computing</i> .....	24
1.4.5    Técnicas de <i>soft computing</i> aplicadas en la gestión de proyectos.....	26
1.5    Conclusiones del capítulo .....	27
CAPÍTULO 2. SISTEMA BASADO EN TÉCNICAS DE <i>SOFT COMPUTING</i> .....	29
2.1    Principios de funcionamiento del sistema.....	29
2.2    Componentes del sistema .....	29
2.3    Técnicas de adaptación y aprendizaje de reglas borrosas .....	31
2.3.1    Sistema borroso basado en el método de Wang y Mendel (WM) .....	33
2.3.2    Sistema borroso basado en redes adaptativas (ANFIS) .....	34
2.3.3    Sistema neuro-borroso híbrido (HYFIS) .....	36
2.3.4    Sistema borroso basado en método de descenso (FIR.DM) .....	38
2.3.5    Sistema borroso basado en heurística y gradiente descendente (FS.HGD) .....	40
2.3.6    Sistema borroso genético basado en la técnica de Thrift (GFS.THRIFT) .....	41
2.3.7    Sistema borroso genético basado en ajuste lateral (GFS.LT.RS).....	42
2.4    Conclusiones del capítulo .....	44
CAPÍTULO 3. APLICACIÓN DEL SISTEMA Y ANÁLISIS DE RESULTADOS .....	45
3.1    Síntesis de aplicación.....	45
3.1.1    Caracterización del entorno.....	45
3.1.2    Base de casos para el aprendizaje.....	45
3.1.3    Parámetros de los experimentos .....	47
3.2    Resultados experimentales .....	47
3.2.1    Resultados de la ejecución de cada técnica .....	47
3.2.2    Resultados de las pruebas estadísticas .....	50
3.2.3    Sistema de inferencia borroso obtenido .....	59
3.3    Valoración social y económica .....	60
3.4    Conclusiones del capítulo .....	62
CONCLUSIONES.....	64
RECOMENDACIONES .....	65
REFERENCIAS BIBLIOGRÁFICAS .....	66
ANEXOS .....	71

## **INTRODUCCIÓN**

### **Antecedentes y situación problemática**

La revolución tecnológica implantada a fines del siglo XX, que integró la informática y las telecomunicaciones, dio lugar a múltiples aplicaciones científicas y comerciales. Este desarrollo propició el acercamiento entre grandes mercados, exigiendo una alta preparación profesional y el desarrollo de nuevas estrategias empresariales a escala mundial (Castro Díaz-Balart, 2004).

Las aplicaciones informáticas actuales han cambiado radicalmente los objetivos, medios, técnicas, herramientas y formas de solucionar los problemas de gestión; hoy resuelven problemas que hace tan solo veinte años no se soñaban resolver. Es de suponer que, en los próximos veinte años, esa tendencia continúe, manteniéndose permanente "tan solo el cambio". Existe un grupo de sistemas que se han clasificado en el rubro de sistemas de apoyo a la toma de decisiones, hechos para respaldar el trabajo individual y las decisiones en grupo (Blanco, 2011).

La toma de decisiones es una actividad que está presente en prácticamente todas las áreas de la gestión; se define como la selección de una alternativa entre un conjunto de ellas, a partir de la evaluación de indicadores ponderados aplicando un sistema de conocimientos. La misma se trata como el proceso de evaluación que maneja múltiples alternativas, con un resultado esperado en forma de acciones futuras (PMI, 2013). La toma de decisiones es parte de un proceso en el que la decisión es sólo un paso y su calidad depende del nivel de información que brinde el sistema y la experiencia de la persona que la ejecute.

Para ordenar sus trabajos, las organizaciones orientan los objetivos de producción hacia la gestión por proyectos. Un proyecto es un conjunto de procesos constituidos por actividades cotejadas, con fechas de inicio y fin, definidas para conseguir un objetivo, que puede ser la obtención de un producto o servicio determinado. La aplicación apropiada de conocimientos, procesos, técnicas, habilidades y herramientas tiene un impacto significativo en el éxito de los proyectos.

La competitividad en el área de la gestión de proyectos se intensifica a nivel mundial, es por ello que las organizaciones requieren ser cada vez más eficientes en la planificación de sus recursos y actividades, así como en la ejecución y control de los mismos, con el fin de lograr con calidad los objetivos trazados. Durante la Dirección Integrada de Proyectos (DIP), el control de ejecución por cortes representa el subproceso más importante por el nivel de información que genera y por su incidencia en la toma de decisiones (Delgado, 2011).

Un factor fundamental para lograr el éxito de los proyectos es la correcta aplicación e integración de los procesos de dirección: inicio, planificación, ejecución, seguimiento y control, y cierre. Especialmente el proceso de seguimiento y control se encarga de medir y supervisar regularmente el avance, a fin de identificar las variaciones respecto al plan de gestión del proyecto, de tal forma que se tomen medidas correctivas cuando sea necesario para cumplir con los objetivos del mismo. La integración, en el contexto de la dirección de proyectos, consiste en tomar decisiones sobre dónde concentrar recursos y esfuerzos cada día, anticipando las posibles polémicas de modo que puedan ser tratadas antes de que se conviertan en críticas. El esfuerzo de integración también implica hacer concesiones entre objetivos y alternativas en competencia.

En el desarrollo de un proyecto, la ejecución constituye una de las etapas más importantes, por su complejidad en la toma de decisiones para lograr los objetivos del mismo, en el menor plazo de tiempo posible, en el marco del presupuesto y con la calidad requerida por el cliente y las partes interesadas. Para lograr el éxito del proyecto es necesario desarrollar un trabajo de dirección en equipo donde el papel del director como líder es decisivo, evaluando por cortes un conjunto de indicadores. Dichos indicadores deben cubrir las áreas de conocimiento claves de la gestión de proyectos: costo, tiempo, calidad, logística y recursos humanos (ISO, 2003).

Entre las habilidades que debe tener una persona para dirigir proyectos de desarrollo está sin lugar a dudas la combinación de definición de problemas, identificación de alternativas y toma de decisiones; todo con la meta de respaldar los objetivos estratégicos de la organización. Una de las opciones para apoyar a las personas durante los procesos de ejecución y control de un proyecto es la utilización de herramientas que permitan dilucidar la situación actual y proponer determinadas alternativas para darle solución al problema.

La DIP, apoyada por los sistemas informáticos profesionales, permite llevar a cabo un cronograma de ejecución y sobre el mismo mostrar el avance del proyecto a través de los indicadores de gestión, destacando los problemas y sus causas para sobre ellas realizar la toma de decisiones a partir de una estrategia de prioridades previamente establecidas (Delgado, 2011). Al realizar una caracterización de la DIP se identifican los siguientes aspectos:

- Presencia de datos numéricos y lingüísticos.
- Vaguedad en los conceptos para la toma de decisiones.
- Incertidumbre e imprecisión en los datos y apreciación de las personas.
- Condiciones cambiantes en el entorno de ejecución de los proyectos.
- Evaluación de los proyectos de forma estática y basada en el conocimiento de expertos.

La mala gestión de proyectos y las insuficiencias en el manejo de los datos en la DIP causan innumerables pérdidas económicas con elevado impacto social. Entre las principales causas de fracaso en esta área se pueden citar:

- Falta de conocimiento de buenas prácticas para la gestión de proyectos.
- Poca experiencia en el control y seguimiento de proyectos.
- Dificultades en el tratamiento de la ambigüedad e incertidumbre de los datos.
- Cambios en los miembros del equipo de proyecto lo que provoca que se no se mantenga el conocimiento adquirido por las personas.
- Debilidades de las herramientas para la evaluación automática o semiautomática de proyectos.
- Inadaptabilidad de los mecanismos de evaluación y control en organizaciones con cambiantes estilos de gestión provocados por la madurez alcanzada durante su mejora continua.

Una alternativa de solución a los problemas antes mencionados, es la aplicación de aprendizaje automático. Este abarca los mecanismos por los cuales el conocimiento es adquirido a través de la experiencia y se muestra como un campo interdisciplinario donde intervienen: la estadística, la lógica, las matemáticas, las estructuras neuronales, la información teórica, la inteligencia artificial y el *soft computing* (computación blanda) (Mitchell, 1997, Moreno et al., 1994).

En (Zadeh, 1994) se expone la definición de *soft computing* como un sistema que aprovecha la tolerancia ante la imprecisión, la incertidumbre y la veracidad parcial de los hechos con vista a lograr un manejo adecuado, robustez, solución de bajo coste y mejor afinidad con la realidad. En (Bonissone et al., 1999) se actualiza la definición de *soft computing* como el conjunto de metodologías computacionales que tiene entre sus principales miembros la lógica borrosa, la teoría de las redes neuronales, el razonamiento probabilístico y la computación evolutiva.

Las técnicas de *soft computing* aportan a las herramientas informáticas una aproximación al razonamiento humano, a través de la utilización del conocimiento y la experiencia acumulada (Bello and Verdegay, 2010). La hibridación de varias técnicas de *soft computing* permite ganar en eficiencia, adaptabilidad y equilibrio adecuado entre poder de predicción e interpretación. Este tipo de técnicas recurren a la experiencia acopiada en las organizaciones, estándares definidos, resultados obtenidos, entre otros tipos de conocimiento, los cuales son integrados y empleados para dar apoyo a la toma de decisiones. Dichas técnicas son robustas ante entornos con entradas ruidosas y tienen una alta tolerancia a la imprecisión de los datos con los que operan; permiten desarrollar soluciones de bajo costo y mayor capacidad de modelación (Piñero, 2005).

Enmarcado en nuestro país, la industria cubana experimenta dificultades en el desarrollo exitoso de sus proyectos (Castro Ruz, 2011). Las actuales políticas socioeconómicas hacen un llamado a perfeccionar los trabajos de planificación, erradicando la espontaneidad, la improvisación, la superficialidad y el incumplimiento de dichos planes. De igual manera, demandan el fortalecimiento de los procesos de control sistemáticos de proyectos y el incremento de la soberanía tecnológica. En los Lineamientos de la Política Económica y Social definidos por el Partido Comunista de Cuba se reclama la urgencia de contar con mecanismos eficientes de control que permitan conocer el estado real de los proyectos en las empresas y contribuir a la toma de decisiones (PCC, 2011).

Atemperado a las condiciones de la actualización del modelo económico cubano, el Decreto No. 327 "Reglamento del proceso inversionista" tiene por objetivo regular los elementos esenciales del proceso inversionista (Consejo de Ministros, 2015). Dicho decreto demanda que todos los sujetos que intervienen en el proceso inversionista posean preparación en las técnicas de evaluación, selección y gestión de la tecnología que garantice el enfoque integral de la inversión y la organización eficiente del proceso inversionista. Este decreto hace un llamado a optimizar la eficiencia económico-social del proceso de inversión.

Por otra parte, las estrategias del Ministerio de la Informática y las Comunicaciones incluyen el fomento de oportunidades de negocios, promocionar y crear las condiciones de penetración de nuestros productos de *software* y servicios informáticos en otros países (MIC, 2011). Para contribuir al logro de estos objetivos, se necesita provocar un cambio de pensamiento y reconocer aún más el valor agregado que puede aportar la informática durante los procesos de dirección.

La presente investigación se enmarca en la Universidad de las Ciencias Informáticas (UCI). En el año 2009 se comienza a desarrollar un paquete de herramientas para la gestión de proyectos (GESPRO) basado en *Redmine* y *software* libres propios de la institución, teniendo en cuenta aspectos como la soberanía tecnológica, la seguridad de los datos, así como su extensibilidad y capacidad para brindar ayuda en la toma de decisiones (Piñero, 2013). Desarrollada por el Laboratorio de Investigaciones en Gestión de Proyectos de la UCI, la actual Suite de Gestión de Proyectos Xedro-GESPRO 14.05 se ha implantado en la red de centros productivos de la UCI como solución integral para la gestión de los proyectos que en la misma se desarrollan.

Xedro-GESPRO se encuentra alineado con los estándares propuestos por el *Project Management Institute* (PMI, 2013) y el *Software Engineering Institute* (SEI, 2010), brindando las interfaces adecuadas para que los usuarios introduzcan al sistema las medidas sugeridas por estos modelos y según las necesidades específicas de información en cada organización. Sin embargo, al igual que sucede con otras herramientas de gestión de proyectos, muchos datos que se manejan con Xedro-GESPRO son suministrados por seres humanos, lo cual implica que exista imprecisión, vaguedad e incertidumbre en la información contenida.

En (Lugo, 2012) se formaliza un modelo para el control de la ejecución de proyectos basado en indicadores y lógica borrosa, logrando su integración con Xedro-GESPRO. Sin embargo, la base de reglas de inferencia borrosas del modelo que evalúa el estado de los proyectos es creada a partir de la visión de varios expertos humanos, donde se relacionan diferentes parámetros de forma estática. Con la implementación del criterio de los expertos en gestión de proyectos para la construcción de las reglas se introducen elementos de ruido en el método de evaluación, debido a su fundamento subjetivo y dependiente de la experiencia.

A partir de dicho sistema estático de evaluación de proyectos, se está en presencia de un problema donde existe vaguedad en la definición de los conjuntos borrosos, debido a que los expertos pueden equivocarse acerca de la localización de determinados puntos característicos de las funciones de pertenencia. Además, las reglas que miden los indicadores, al ser estáticas, no permiten la adaptación del sistema de evaluación según el rigor con que se controlan los proyectos, dado por el nivel de madurez alcanzado por la organización. Esto implica que los datos que brinda el sistema

sobre el estado de los proyectos no sean suficientes como apoyo a la toma de decisiones para controlar correctamente su ejecución y que se vea afectada la calidad de la clasificación de los proyectos en *Bien*, *Regular* y *Mal*.

Partiendo de una recomendación de (Lugo, 2012) se construye una base de casos de proyectos terminados con el propósito de aplicar técnicas de *soft computing* para el aprendizaje y mejora del sistema de evaluación de proyectos. Desde diciembre de 2012 hasta la actualidad se han recopilado en una base de casos los indicadores y la evaluación de proyectos (Escobar and et.al., 2013), no obstante aún se carece de herramientas que empleen este conocimiento y lo integren en sistemas de apoyo a la toma de decisiones para la gestión de proyectos.

### **Problema**

Teniendo en cuenta lo enunciado anteriormente se plantea el siguiente problema científico: las deficiencias en la adaptación y el aprendizaje automático de los sistemas para la evaluación de la ejecución de proyectos afectan la calidad de la clasificación de los mismos como apoyo a la toma de decisiones.

### **Objeto de investigación**

La investigación se enmarca en la siguiente área de conocimiento: evaluación de la ejecución de proyectos.

### **Objetivo general**

Para resolver la problemática se plantea como objetivo general de la investigación: desarrollar un sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos, que contribuya a mejorar la calidad de la clasificación de los mismos como apoyo a la toma de decisiones.

### **Objetivos específicos**

- Elaborar un marco teórico en torno a las tendencias en el control de la ejecución de proyectos y la aplicación de técnicas de *soft computing*; analizando escuelas, experiencias y herramientas útiles desarrolladas en los últimos años.
- Aplicar técnicas de *soft computing* en la construcción y aprendizaje de sistemas de inferencia borrosos para la evaluación de la ejecución de proyectos.
- Validar el sistema propuesto a partir de su integración con la Suite de Gestión de Proyectos Xedro-GESPRO.

### **Campo de acción**

La investigación se centra en: el empleo de técnicas de *soft computing* para la evaluación de la ejecución de proyectos.

### **Tipo de investigación**

Se aplica el tipo de investigación Descriptiva, ya que existe suficiente conocimiento precedente acerca del problema para plantear una hipótesis a nivel descriptivo. La representación del problema es clara en lo referente a la caracterización del fenómeno en sus aspectos externos. Se tienen conocimientos previos en cuanto a mecanismos para medir los indicadores aplicados en el control de la ejecución de proyectos. Se persigue especificar las propiedades más importantes en cuanto a la adaptación y el aprendizaje automático de sistemas de inferencia borrosos para realizar la evaluación de la ejecución de proyectos.

### **Hipótesis**

Si se desarrolla un sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos, entonces se mejorará la calidad de la clasificación de los mismos como apoyo a la toma de decisiones.

### **Análisis de las variables independiente y dependiente**

A continuación se muestran las variables con su correspondiente análisis operacional (Tabla 1).

Tabla 1. Análisis de variables.

<b>Variable independiente</b>	<b>Dimensión</b>	<b>Indicador</b>	<b>Índice</b>
Sistema basado en técnicas de <i>soft computing</i> para la evaluación de la ejecución de proyectos	Funcionamiento del sistema	Reglas borrosas generadas	Cantidad de reglas
		Integración del sistema con herramienta de gestión de proyectos	Sí No
<b>Variable dependiente</b>	<b>Dimensión</b>	<b>Indicador</b>	<b>Índice</b>
Calidad de la clasificación de proyectos como apoyo a la toma de decisiones	Calidad de la evaluación de proyectos	Clasificaciones correctas	Porcentaje
		Falsos positivos	Porcentaje
		Falsos negativos	Porcentaje
		Error cuadrático medio	MSE
		Raíz del error cuadrático medio	RMSE
		Error porcentual de la media absoluta simétrica	SMAPE

### **Métodos de investigación**

Para el desarrollo de la investigación se aplican los siguientes métodos científicos de búsqueda y procesamiento de la información:

- Método hipotético-deductivo: utilizado para la elaboración de la hipótesis de la investigación.
- Método analítico-sintético: aplicado en el análisis de las teorías y los trabajos publicados sobre adaptación y aprendizaje automático de reglas de inferencia borrosas; así como para conocer los antecedentes y las tendencias actuales referidas a la aplicación de *soft computing* en el área de la gestión de proyectos.
- Método teórico de modelación: aplicado en el diseño del esquema de funcionamiento del sistema mediante componentes, en el boceto de las topologías de las redes neuronales y en las funciones de pertenencia utilizadas en los conjuntos borrosos.
- Métodos empíricos de medición y experimental: utilizados en la ejecución de los algoritmos con diferentes parámetros para comparar sus resultados en cuanto a la calidad de la evaluación de proyectos y para validar el sistema propuesto.

### **Población y muestra**

A continuación se especifica el universo y el proceso de selección de la muestra.

Los escenarios para la toma de decisiones equivalen a una **población** compuesta por los 20 centros productivos donde está implantada la Suite de Gestión de Proyectos Xedro-GESPRO. La **muestra** se corresponde con el 70% de los escenarios descritos, lo cual equivale a los 14 centros productivos de la UCI (método no probabilístico intencional). Las evidencias utilizadas para el aprendizaje del sistema cubren un total de 204 proyectos pertenecientes a los 14 centros productivos de la UCI.

### **Diseño de experimentos**

El principal tipo de experimento a realizar es pre-experimento ya que es propio de la investigación descriptiva y se caracteriza por conclusiones extraídas de la variación de la variable dependiente en relación con su historia anterior. Se realiza el Experimento 1 del tipo cuasi experimento post prueba con grupo de control para comparar la cantidad de reglas generadas por cada técnica de *soft computing*. Se realizan 12 experimentos con pre y post prueba para analizar la calidad de la evaluación de proyectos (en el conjunto de entrenamiento y el de prueba) en cuanto a los seis indicadores: clasificaciones correctas, falsos positivos, falsos negativos, error cuadrático medio, raíz del error cuadrático medio y error porcentual de la media absoluta simétrica. Además, se realiza un experimento con pre y post prueba para analizar la calidad de la evaluación de proyectos del sistema seleccionado respecto al sistema estático anterior.

**Experimento 1:** Análisis de resultados obtenidos al aplicar el sistema propuesto en cuanto a la cantidad de reglas generadas por cada técnica de *soft computing*. El diseño del experimento queda de la siguiente forma:

G X O

Simbología utilizada:

- G: grupo de experimentación compuesto por una base de casos con 204 proyectos de la UCI que utilizan Xedro-GESPRO.
- X: aplicación de la propuesta al grupo de experimentación.
- O: observación de la cantidad de reglas generadas por cada técnica de *soft computing* después de aplicar la propuesta.

**Experimentos 2, 4, 6, 8, 10 y 12:** Análisis de resultados obtenidos al aplicar el sistema propuesto al conjunto de proyectos de entrenamiento, respecto a los indicadores: clasificaciones correctas, falsos positivos, falsos negativos, error cuadrático medio, raíz del error cuadrático medio y error porcentual de la media absoluta simétrica, respectivamente. El diseño de los experimentos queda de la siguiente forma:

G O1 X O2

Simbología utilizada:

- G: grupo de experimentación compuesto por el conjunto de datos de entrenamiento correspondiente al 75% de los 204 proyectos de la UCI que utilizan Xedro-GESPRO.
- O1: observación de la calidad de la evaluación de proyectos, en un punto de referencia inicial, según el indicador contemplado en el experimento.
- X: aplicación de la propuesta al grupo de experimentación.
- O2: observación de la calidad de la evaluación de proyectos, después de aplicar la propuesta, según el indicador contemplado en el experimento.

**Experimentos 3, 5, 7, 9, 11, 13:** Análisis de resultados obtenidos al aplicar el sistema propuesto al conjunto de proyectos de prueba, respecto a los indicadores: clasificaciones correctas, falsos positivos, falsos negativos, error cuadrático medio, raíz del error cuadrático medio y error porcentual de la media absoluta simétrica, respectivamente. El diseño de los experimentos queda de la siguiente forma:

G O1 X O2

Simbología utilizada:

- G: grupo de experimentación compuesto por el conjunto de datos de prueba correspondiente al 25% de los 204 proyectos de la UCI que utilizan Xedro-GESPRO.
- O1: observación de la calidad de la evaluación de proyectos, en un punto de referencia inicial, según el indicador contemplado en el experimento.
- X: aplicación de la propuesta al grupo de experimentación.

- O2: observación de la calidad de la evaluación de proyectos, después de aplicar la propuesta, según el indicador contemplado en el experimento.

**Experimento 14:** Análisis de resultados obtenidos al aplicar el sistema propuesto con respecto al sistema estático anterior, teniendo en cuenta la calidad de la evaluación de proyectos. El diseño del experimento queda de la siguiente forma:

G O1 X O2

Simbología utilizada:

- G: grupo de experimentación compuesto por 204 proyectos de la UCI que utilizan Xedro-GESPRO.
- O1: observación de la calidad de la evaluación de proyectos dada por el sistema estático.
- X: aplicación de la propuesta al grupo de experimentación.
- O2: observación de la calidad de la evaluación de proyectos dada por el sistema propuesto.

### **Análisis estadístico a realizar**

El método estadístico a utilizar en la investigación se basa en la comparación de muestras relacionadas, para ello se aplica el Test de Friedman (N muestras) y el Test de Wilcoxon (2 muestras) ya que se dispone de suficientes conjuntos de datos. Estas técnicas permiten identificar las diferencias significativas y comparar varias muestras relacionadas durante la validación del sistema propuesto.

### **Instrumentos**

El instrumento a utilizar para medir las variables operacionales es la herramienta pgAdmin III. Se valida a través de experimentos, comparando la calidad de la evaluación de proyectos entre la base de conocimiento brindada por los expertos, el sistema estático anterior y los sistemas basados en técnicas de *soft computing* obtenidos con cada algoritmo de aprendizaje empleado. Para ello se requiere contar con una base de casos de proyectos que contenga la evaluación de los expertos y la del sistema estático.

### **Novedad y aporte práctico de la investigación**

Los beneficios potenciales de la propuesta se relacionan con la mejora de la calidad de la clasificación de proyectos como apoyo a la toma de decisiones en organizaciones con cambiantes estilos de gestión provocados por la madurez alcanzada durante su mejora continua.

- Aplicación de técnicas de *soft computing* en la evaluación de la ejecución de proyectos, elevando la robustez, adaptabilidad y equilibrio entre poder de predicción e interpretación de las herramientas de gestión de proyectos.
- Desarrollo de la biblioteca AnalysisPro.SC.PMC basada en *software* libre con la implementación de técnicas de *soft computing* para el aprendizaje y la experimentación de sistemas de inferencia borrosos para la evaluación de la ejecución de proyectos.
- Integración del sistema basado en técnicas de *soft computing* con la Suite de Gestión de Proyectos Xedro-GESPRO.

### **Listado de publicaciones y participación en eventos que avalan la investigación**

1. **Bermudez, A.** *Propuesta de estrategia para la gestión del conocimiento en un departamento docente.* Duodécima Semana Tecnológica de FORDES, Taller Buenas Prácticas en Formación y Desarrollo, La Habana, 2012.
2. **Bermudez, A.** *Gestión del conocimiento en un colectivo docente.* Tercer Encuentro de Mujeres Creadoras, La Habana, 2012.
3. **Bermudez, A.** *Optimización de reglas de inferencia borrosas para la evaluación de proyectos de desarrollo de software.* VIII Peña Tecnológica de Jóvenes Profesionales, La Habana, 2013.

4. **Bermudez, A.**; Lugo, J.A.; Piñero, P.Y. *Optimización de reglas borrosas para el apoyo a la toma de decisiones en la Dirección Integrada de Proyectos*. Congreso Internacional COMPUMAT 2013, Taller de Inteligencia Artificial, ISBN: 978-959-286-022-3, La Habana, 2013.
5. **Bermudez, A.**; Lugo, J.A.; Piñero, P.Y. *Aplicación de ANFIS para la mejora del control de proyectos en Xedro-GESPRO*. I Conferencia Científica Internacional de la Universidad de las Ciencias Informáticas (UCIENCIA 2014), I Taller Internacional de Gestión de Proyectos, ISBN: 978-959-286-026-1, La Habana, 2014.
6. **Bermudez, A.**; Lugo, J.A.; Pérez, I.; Piñero, P.Y.; Cruz, G. *Sistema neuro-borroso de apoyo al control de la ejecución de proyectos*. Revista Cubana de Ingeniería, vol. 5, no. 2, pp. 41-51, ISSN: 2223-1781, La Habana, 2014.
7. **Bermudez, A.**; Lugo, J.A.; Cruz, G.; Piñero, P.Y. *Sistema de inferencia borroso basado en redes neuronales para la evaluación de proyectos*. 17 Convención Científica de Ingeniería y Arquitectura 2014, X Seminario Euro Latinoamericano de Sistemas de Ingeniería, ISBN: 978-959-261-467-3, La Habana, 2014.
8. **Bermudez, A.**; Lugo, J.A. *Sistema de optimización de reglas borrosas para el control de la ejecución de proyectos*. XIX Escuela Latino Americana de Verano en Investigación Operativa (ELAVIO 2015), Escuela Politécnica Nacional en Quito, Ecuador, 2015.
9. **Bermudez, A.**; Lugo, J.A.; Piñero, P.Y. *Sistema de inferencia borroso basado en redes adaptativas para la evaluación de proyectos*. Revista Ingeniería y Universidad, vol. 19, no. 2, ISSN: 2011-2769, Colombia, 2015.

### **Estructura de la tesis**

La investigación se divide en las siguientes partes:

En el **Capítulo 1** se elabora un marco teórico en torno a las tendencias en el control de la ejecución de proyectos, analizando escuelas, experiencias y herramientas útiles desarrolladas en los últimos años. Se expone la fundamentación teórica y de actualidad sobre la aplicación de técnicas de *soft computing* en la gestión de proyectos.

En el **Capítulo 2** se presentan los componentes del sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos. Se abordan los principios necesarios para el funcionamiento del sistema y se presentan los parámetros a utilizar en la aplicación de las técnicas. Se describen las técnicas aplicadas: WM, ANFIS, HYFIS, FIR.DM, FS.HGD, GFS.THRIFT y GFS.LT.RS; las cuales son basadas en partición espacial, redes neuronales, gradiente descendente y algoritmos genéticos.

En el **Capítulo 3** se muestra una síntesis de la aplicación del sistema propuesto integrado a la herramienta Xedro-GESPRO. Se describe la base de casos de proyectos preclasificados por expertos como recurso para el aprendizaje y la validación. Se exponen los experimentos para validar el comportamiento de la variable dependiente: calidad de la clasificación de proyectos. Se realiza una valoración del impacto económico y social de la propuesta.

Finalmente se abordan las conclusiones generales, las recomendaciones y los anexos de la investigación.

## CAPÍTULO 1. TENDENCIAS EN EL CONTROL DE LA EJECUCIÓN DE PROYECTOS

En el ámbito del desarrollo de proyectos se vienen aplicando disímiles maneras de asistir a los especialistas en la toma de decisiones. A escala mundial, el enfoque competitivo en la gestión de proyectos obliga a extremar los rasgos de identidad y las ventajas de los productos y servicios de brindan las organizaciones. Esto conlleva a la necesidad de diseñar herramientas que se adapten de forma continua a las condiciones cambiantes de los negocios dentro de la organización.

El presente capítulo tiene por objetivo elaborar un marco teórico en torno a las tendencias en el control de la ejecución de proyectos; analizando escuelas, experiencias y herramientas útiles desarrolladas en los últimos años. Se examinan las escuelas de gestión de proyectos, con especial interés en el tratamiento de la incertidumbre, la mejora continua, la optimización, la adaptación y el aprendizaje que proponen para la evaluación durante el control de la ejecución de proyectos. Se presentan los elementos fundamentales de los sistemas de inferencia borrosos como mecanismo para el tratamiento de la incertidumbre en la información. Se realiza un análisis comparativo de las principales tendencias en la generación y adaptación de reglas borrosas, con especial interés en las técnicas para modificar las funciones de pertenencia de los conjuntos borrosos, enunciando de cada una sus ventajas y limitaciones. Se analiza la actualidad a nivel mundial sobre la aplicación de técnicas de *soft computing* en el área de la gestión de proyectos y la forma de evaluar los mismos mediante la herramienta Xedro-GESPRO. Finalmente se presentan las conclusiones del capítulo.

### 1.1 Análisis bibliográfico

A continuación se relaciona en forma de resumen la principal bibliografía consultada (Tabla 2), la cual es objeto de análisis y referencia en la presente investigación. Para elaborar el marco teórico se utilizaron herramientas bibliográficas, dígame: motores de búsqueda, directorios de materias y bases de datos bibliográficas, como: IEEE, Elsevier, DOAJ, Scielo, EBSCO y Google Académico.

Tabla 2. Resumen cuantitativo de la bibliografía consultada.

Tipo de publicación	Últimos 5 años		Años anteriores	
Libros y monografías	7	28%	18	72%
Tesis de doctorado	1	25%	3	75%
Tesis de maestría	2	100%	-	-
Artículos de revistas en SCOPUS y <i>Web of Science</i>	9	27%	24	73%
Artículos de otras revistas	5	83%	1	17%
Memorias de eventos	6	55%	5	45%
Páginas oficiales y artículos publicados en la web	9	100%	-	-
Reportes técnicos y conferencias	7	41%	10	59%
Entrevistas personales	2	100%	-	-

La aplicación de técnicas para la adaptación y el aprendizaje automático de reglas de inferencia borrosas es un tema ampliamente abordado en diversas formas de literatura; de ahí que la mayoría de los libros, tesis de doctorado y artículos de revistas de primer nivel consultados sobre el tema se reportan en años anteriores al 2010. Sin embargo, según la bibliografía consultada, la aplicación de técnicas de *soft computing* en la gestión de proyectos es un tema poco abordado; se encuentran algunos artículos de primer nivel en el período de los últimos 5 años.

La bibliografía consultada más relevante se reporta en los epígrafes 1.4.3 y 1.4.5; donde se abordan las técnicas para la generación y optimización de reglas borrosas, así como las corrientes actuales sobre la aplicación de técnicas de *soft computing* en el área de la gestión de proyectos.

## **1.2 Escuelas de gestión de proyectos**

Las escuelas de gestión de proyectos están constituidas por las normas, buenas prácticas, metodologías y estándares más representativos tanto en el plano internacional, nacional, como en la propia institución donde se desarrolla la presente investigación. En esta sección se expone un estudio comparativo de las escuelas referente a si proponen o no mecanismos para realizar la mejora continua en la organización, con especial interés en el tratamiento de la incertidumbre, la optimización, la adaptación y el aprendizaje de los sistemas que promueven para el control de la ejecución de proyectos, así como las herramientas informáticas que suscitan. En las siguientes subsecciones se abordan algunas descripciones y análisis de dichas escuelas.

### **1.2.1 Guía de los Fundamentos para la Dirección de Proyectos (PMBOK)**

PMBOK (*Project Management Body of Knowledge*), desarrollado por *Project Management Institute* (PMI, 2013), proporciona pautas para el ciclo de vida del proyecto y los procesos relacionados. Constituye uno de los estándares de gestión de proyectos más reconocidos a nivel internacional.

PMBOK propone controlar la ejecución del proyecto durante todo su ciclo de vida para proporcionar una retroalimentación constructiva, descubrir situaciones desconocidas o no resueltas y favorecer la mejora continua. Para esto, sugiere el uso de indicadores relacionados con las áreas de conocimiento de Alcance, Costo, Tiempo y Calidad. Aborda el tratamiento de la incertidumbre en los riesgos del proyecto, así como en el análisis de reserva para la estimación de los costos y en la gestión del cronograma del proyecto mediante el método de la cadena crítica.

El enfoque de gestión de la calidad que presenta el PMBOK, reconoce la importancia de la mejora continua en la calidad de la dirección del proyecto, así como la del producto y del proceso. Aborda los ciclos de vidas adaptativos en el desarrollo del proyecto basados en métodos iterativos e incrementales que respondan a niveles altos de cambio y a la participación continua de los interesados. Propone técnicas de optimización de recursos, costos, productos y procesos del ciclo de vida del proyecto. Recomienda reunir las lecciones aprendidas y archivar la información del proyecto para su uso futuro por parte de la organización e implementar las actividades aprobadas de mejora de procesos.

PMBOK refleja la importancia de los sistemas informáticos para mejorar o restringir las opciones de la dirección de proyectos, destacando su influencia de manera positiva o negativa sobre los resultados del proyecto. Aborda las herramientas de *software* para definir cronogramas, gestionar las configuraciones, recopilar y distribuir información, administrar costos y recursos, calcular indicadores de desempeño y gestionar registros de proyectos que se utilizan a lo largo del proceso de monitoreo y control del proyecto.

Se puede concluir que PMBOK registra de manera general la evaluación y optimización del desempeño del proyecto, el tratamiento de la incertidumbre, la mejora continua y el trabajo con las lecciones aprendidas. Sin embargo, no propone mecanismos concretos ni herramientas que aseguren la adaptación de la evaluación de proyectos dado el nivel de madurez alcanzado por la organización.

### **1.2.2 Modelo de Madurez y Capacidad Integrado (CMMI)**

CMMI (*Capability Maturity Model Integration*), desarrollado por *Software Engineering Institute* (SEI, 2010), es un modelo para la mejora y la evaluación de procesos de desarrollo de *software*. Propone cinco niveles de madurez: 1-Inicial, 2-Administrado, 3-Definido, 4-Administrado cuantitativamente y 5-Optimizado. En el nivel 5 del modelo se aborda la optimización de los procesos dentro de la organización, enfocada hacia la mejora continua en el desempeño de los procesos a través de reformas innovadoras e incrementales. Plantea que los proyectos deben actualizar periódicamente la definición de sus procesos para incorporar los últimos cambios realizados por la organización a los procesos estándares cuando estos se beneficien. Sin embargo, la mejora continua debería estar

presente en todo momento, no solamente cuando la organización alcanza su máximo nivel de madurez según CMMI.

El modelo reconoce que las evaluaciones de proceso y producto deben obtenerse a partir del análisis e interpretación de las métricas definidas en la organización, de las cuales son derivables indicadores. Provee mecanismos para evaluar la efectividad del curso del entrenamiento con respecto a los objetivos de rendimiento establecidos por la organización, el proyecto o el individuo. Sin embargo, el modelo se enfoca en la descripción verbal de los procesos y no define métodos concretos para el aprendizaje y la adaptación durante el control de la ejecución de proyectos según el nivel de madurez alcanzado por la organización.

Expone la importancia de las herramientas informáticas como apoyo a la toma de decisiones del equipo de desarrollo del proyecto para administrar el mismo, diseñar, gestionar requisitos, evaluar y probar. Refleja la importancia de la mejora tecnológica para dar soporte a la optimización de los procesos como una gestión sistemática de la organización. Aborda los sistemas de información como herramienta para acelerar y compartir el aprendizaje, sobre todo en procesos ágiles e innovadores donde se depende de la participación de empleados que estén adaptados al negocio y a los objetivos de la organización. Sin embargo, deja a las organizaciones la responsabilidad de implementarlas, no haciendo alusión al tratamiento de la imprecisión, vaguedad e incertidumbre contenida en la información que se gestiona.

### **1.2.3 Proyectos en Entornos Controlados (PRINCE2)**

La metodología PRINCE2 (*Projects in Controlled Environments*), establece un conjunto de buenas prácticas en torno a la gestión de proyectos, que cubre el control, la administración y la organización de proyectos (PRINCE2, 2009). Propone utilizar indicadores como Retorno de la Inversión del proyecto, Costos y Riesgos Mayores asociados al proyecto, para los que señala como buena práctica la acción de recopilarlos y almacenarlos para futuras consultas en los procesos de comienzo e inicio de proyecto.

Para favorecer la mejora continua, PRINCE2 propone llevar un registro de lecciones aprendidas que contenga una colección de información respecto a las buenas y malas experiencias de los directores y especialistas de procesos y productos durante el desarrollo de los proyectos. Al concluir cada proyecto, dicho registro se formaliza en reportes que describen las lecciones aprendidas y las estadísticas del control de la calidad de los productos desarrollados. Estos reportes son almacenados centralmente para el beneficio de proyectos futuros.

Aunque PRINCE2 carece de un nivel de optimización para evaluar el nivel de madurez de la gestión de sus proyectos, al adoptar esta metodología en la organización se puede combinar con P3M3. Este último es el Modelo de Madurez de Gestión de Portafolios, Programas y Proyectos (*Portfolio, Programme and Project Management Maturity Model*). P3M3 cuenta con un nivel 5 para la optimización, el cual establece pautas para la mejora continua en los procesos de la organización con gestión proactiva de los problemas y apoyo en herramientas informáticas como apoyo a la toma de decisiones.

Sin embargo, PRINCE2 no propone métodos concretos para el aprendizaje y la adaptación de la evaluación de la ejecución de proyectos dado el nivel de madurez alcanzado por la organización durante su mejora continua. Tampoco hace alusión al tratamiento de la imprecisión, vaguedad e incertidumbre contenida en la información que se gestiona en la organización.

### **1.2.4 Organización Internacional de Normalización (ISO)**

La norma **ISO 10006:2003** desarrollada por *International Standards Organization* (Organización Internacional de Normalización) proporciona orientación para gestionar la calidad durante el desarrollo de los proyectos (ISO, 2003). En su capítulo 8 aborda los procesos relacionados con la medición y el análisis, así como la mejora continua del rendimiento del proyecto. Plantea que la

dirección es responsable de la búsqueda de mejoras de la calidad de sus procesos mediante el aprendizaje basado en la experiencia y la gestión del conocimiento.

Menciona la necesidad de definir indicadores de rendimiento, cómo medirlos y adoptar disposiciones para su evaluación periódica con el fin de monitorear el progreso. Señala que se deben facilitar las acciones preventivas, correctivas y confirmar que los objetivos del proyecto permanecen válidos dentro del cambiante entorno que le rodea. Expone que la organización debe garantizar que el sistema de gestión de información sea diseñado para identificar y coleccionar la información relevante de sus proyectos, en orden de mejorar los procesos de gestión. Plantea que las evaluaciones de rendimiento y progreso deben ser llevadas a cabo para evaluar el estado del proyecto y deben cubrir todos los procesos y proveer una oportunidad para evaluar el éxito de los objetivos del proyecto.

Aborda el manejo de la incertidumbre en relación a los costos y al control del cronograma del proyecto. Para ello propone que cuando la estimación de la duración o de los costos envuelve una incertidumbre significativa, esta debe ser identificada, evaluada, documentada y mitigada. El progreso del proyecto debe ser analizado en orden de identificar tendencias y posibles incertidumbres en el resto del trabajo del proyecto. Reconoce la importancia de los sistemas informáticos como apoyo a la toma de decisiones. En orden de mejorar el rendimiento de la organización del proyecto, plantea se debe proveer al personal de las herramientas, técnicas y métodos apropiados para facilitar el monitoreo y control de los procesos.

La norma **ISO 21500:2013** establece una guía para la dirección y gestión de proyectos; describe conceptos y procesos que forman parte de las buenas prácticas (Zandhuis and Stellingwerf, 2013). Identifica los procesos de dirección y gestión (Inicio, Planificación, Implementación, Control, Cierre) agrupados por grupos de materia (Integración, Parte interesada, Alcance, Recurso, Tiempo, Costo, Riesgo, Calidad, Adquisiciones, Comunicación) que permiten distribuir y gestionar la información relevante del proyecto estableciendo entradas y salidas. Puede ser usada como acoplamiento entre diferentes procesos de negocio y gestión de proyectos. Complementa la norma ISO 10006 para la gestión de la calidad, especialmente en el área de la mejora continua: realizando los procesos de perfeccionamiento necesarios en las operaciones con un mínimo de perturbación durante la producción y los servicios.

Plantea que las prácticas y métodos de gestión de proyectos desarrollados por la organización requieren ser mejorados en dependencia de la experiencia alcanzada, sin que implique realizar cambios drásticos. Enfatiza en el rol que juegan las condiciones del entorno durante la ejecución de proyectos en orden de maximizar el valor agregado de sus entregables. En los métodos y prácticas de gestión propone ajustar sólo lo necesario para el proyecto y entorno específico, documentando los cambios o adiciones transparentemente. Promueve que durante la gestión de proyectos se realice una mejorada toma de decisiones en todos los niveles de la organización a través de una comunicación efectiva. Para ello se requieren de herramientas que apoyen la toma de decisiones y permitan evaluar los proyectos a diferentes niveles.

El proceso de recolección de lecciones aprendidas se centra en la gestión del conocimiento. Para ello incluye un acápite en el proceso de Cierre (sub-grupo Integración) sobre el acopio de la experiencia alcanzada en el proyecto orientada hacia la toma de medidas preventivas en la organización. Esto permite al equipo de proyecto y a las partes interesadas, afrontar situaciones desafiantes de manera proactiva para lograr un mejor rendimiento. La gestión de lecciones aprendidas enriquece la comunicación, creando un entendimiento común del proyecto y una visión compartida para lograr un trabajo en equipo exitoso.

Sin embargo, al igual que sucede con las escuelas analizadas hasta este momento, no propone mecanismos concretos que aseguren la adaptación en el control de la ejecución de proyectos dado

el nivel de madurez alcanzado por la organización durante su mejora continua, así como tampoco el tratamiento a la incertidumbre durante la evaluación a los proyectos.

### **1.2.5 Normas Cubanas para la gestión de proyectos (NC)**

Las normas cubanas para la gestión de proyectos están determinadas por un conjunto de resoluciones y decretos aprobados por ministerios. La especificación de dichas normas se orienta al tipo de proyecto que desarrolla el ministerio en cuestión. En el caso de los proyectos de desarrollo de *software* le compete al Ministerio de la Informática y las Comunicaciones (MIC) la determinación de las normas a seguir.

Para dar a conocer la actualización de las normas cubanas se hace público el boletín NCOonline (Normas Cubanas Online) en (CubaIndustria, 2014). Hasta enero de 2015 existen registradas 4527 normas vigentes en Cuba, de las cuales 310 se encuentran relacionadas con la gestión de proyectos de diversas ramas de la industria, siendo NC ISO 10006:2007 la norma más actual. Se destaca el reconocimiento de las normas ISO como la opción más viable a seguir para organizar los procesos de la producción. Esto se debe fundamentalmente a que ISO ofrece guías y permite obtener certificaciones y evaluaciones que ayudan a mejorar la imagen de la empresa ante el mercado nacional e internacional.

El Decreto No. 327 (Consejo de Ministros, 2015) tiene como objeto contribuir a la eficiencia, racionalidad e integralidad del proceso inversionista en Cuba. Establece realizar la evaluación técnico-económica final para medir si los supuestos de diseño de la inversión fueron alcanzados al fin del período de puesta en explotación. Para ello, propone comparar los indicadores técnicos, económicos y energéticos alcanzados con los previstos en el estudio de factibilidad, lo cual posibilita la adopción de medidas de corrección inmediatas.

Sin embargo, al igual que sucede con la mayoría de los estándares internacionales, las normas cubanas relacionadas con la gestión de proyectos representan guías abstractas que permiten ordenar y perfeccionar el trabajo, pero no ofrecen soluciones concretas. Esto hace que surjan brechas entre la práctica social y el marco normativo existente. Por otra parte, dichas normas no proponen mecanismos ni herramientas específicas que aseguren la adaptación en el control de la ejecución de proyectos dado el nivel de madurez alcanzado por la organización durante su mejora continua.

### **1.2.6 Programa de Mejora de la UCI**

El Programa de Mejora en la UCI comienza en el 2008 con el objetivo de alcanzar la certificación internacional del nivel 2 de CMMI. Esta meta se logra en el 2011 y convierte a la UCI en la primera empresa cubana certificada con este modelo y una de las pocas en el área del Caribe. Para ello se necesitaron definir procedimientos que permitieran dar cumplimiento a los requisitos que establece esta área de proceso (Ramos, 2011).

El desarrollo de *software* dentro de la actividad productiva de la UCI se caracteriza por el uso de diferentes metodologías de desarrollo, entre robustas y ágiles. A pesar de la variedad de metodologías usadas, se ha comprobado que muy pocos proyectos la aplican en su totalidad. Es por ello que se decide definir una metodología de desarrollo única para la actividad productiva de la UCI, que responde a una variación del Proceso Unificado Ágil (*Agile Unified Process*, AUP) (Edeki, 2013) y apoyado en el modelo CMMI v1.3 (SEI, 2010).

AUP describe la forma de desarrollar aplicaciones de *software* de negocio usando conceptos válidos en el Proceso Unificado de Rational (*Rational Unified Process*, RUP) (Jacobson et al., 2000). AUP aplica técnicas ágiles incluyendo desarrollo dirigido por pruebas, modelado ágil, gestión de cambios ágil y refactorización de base de datos para mejorar la productividad. Dada la situación actual de la institución y la adopción de una metodología de desarrollo ágil, es necesario destacar la importancia de utilizar herramientas de apoyo a la toma de decisiones que permitan adaptar la evaluación de

los proyectos. Dichas herramientas deben facilitar la identificación de problemas y sus causas, así como proponer determinadas alternativas para darle solución.

El grupo de Normalización y Métricas de CALISOFT (Centro de Calidad para Soluciones Tecnológicas), en coordinación con el Laboratorio de Investigaciones en Gestión de Proyectos de la UCI, establecieron un plan para la automatización de la recolección de medidas y obtención de indicadores mediante reportes a través de la Suite de Gestión de Proyectos Xedro-GESPRO (descrita en la sección 1.3.2).

Del análisis de las principales escuelas de gestión de proyectos se puede concluir que en su mayoría, de una forma u otra, hacen alusión al establecimiento y monitoreo de métricas para el control, seguimiento y evaluación de proyectos. Todas registran de manera general el tratamiento de la incertidumbre en cierta información, la importancia de la mejora continua, el trabajo con lecciones aprendidas y la influencia de las herramientas informáticas para el éxito de los proyectos como apoyo a la toma de decisiones. Sin embargo, se enfocan en la descripción verbal de los procesos y no definen métodos concretos para el aprendizaje y la adaptación de la evaluación de la ejecución de proyectos según la mejora continua dada por el nivel de madurez alcanzado por la organización.

### **1.3 Herramientas informáticas para la gestión de proyectos**

Los avances tecnológicos en la esfera de la gestión de proyectos han posibilitado conocer en un instante la acción que se debe tomar para solucionar determinado problema. Esto se debe gracias a las herramientas informáticas de apoyo a la toma de decisiones, que están fundamentadas en indicadores claros y medibles.

#### **1.3.1 Comparación de herramientas a nivel mundial**

En las últimas décadas han sido desarrolladas numerosas herramientas para la gestión de proyectos; en la Tabla 3 se muestra un extracto del estudio comparativo realizado sobre las herramientas más representativas según (Stang, 2013). El análisis se centra en la capacidad de las herramientas para gestionar las áreas de conocimiento fundamentales de la gestión de proyectos. Además se tiene en cuenta el tipo de licencia (propietaria o código abierto) y si incluían funcionalidades para manejar la incertidumbre de la información.

Tabla 3. Estudio comparativo de herramientas de gestión de proyectos.

Herramientas Informáticas	Áreas de conocimiento abordada							Software de código abierto	Tratamiento de incertidumbre	Aprendizaje automático
	Alcance	Tiempo	Costo	Calidad	Recursos Humanos	Integración	Adquisición			
Collabtive	Si	Si	No	No	Si	No	No	Si	No	No
dotProject	Si	Si	No	Si	Si	No	No	Si	No	No
LibrePlan	Si	Si	Si	Si	Si	Si	No	Si	No	No
OnepointProject	Si	Si	Si	No	Si	No	No	Si	No	No
Plandora	Si	Si	Si	No	Si	No	No	Si	No	No
Project.net	Si	Si	Si	Si	Si	No	Si	Si	No	No
Project-Open	Si	Si	Si	Si	Si	No	No	Si	No	No
OpenERP	Si	Si	Si	Si	Si	Si	Si	Si	No	No
TeamLab	Si	Si	Si	No	Si	No	No	Si	No	No
Redmine	Si	Si	Si	Si	Si	Si	No	Si	No	No
web2project	Si	Si	Si	No	Si	No	No	Si	No	No
Achievo	Si	Si	Si	Si	No	Si	No	Si	No	No

La sociedad cubana se encuentra inmersa en un proceso de migración a *software* libre de código abierto que permita alcanzar la soberanía tecnológica y la seguridad en la información. Sólo algunas de las herramientas de gestión de proyectos son basadas en *software* libre y brindan funcionalidades para el análisis y la generación de reportes. Además, algunos de los indicadores propuestos por las escuelas antes mencionadas son implementados para su cálculo automático.

La facilidad de aprendizaje automático constituye otra característica importante a considerar, ya que de esta manera se pueden adaptar mejor a las particularidades de cada entorno. Sin embargo, no se reportan evidencias en la generalidad de las herramientas analizadas que demuestren la implementación de funcionalidades relacionadas con el aprendizaje automático y el tratamiento de la incertidumbre en la información.

A partir del estudio de herramientas se puede concluir que aunque algunas satisfacen las necesidades de sus clientes, no siempre cubren todas las expectativas por temas tan específicos como precio, licencia, soporte o insuficiencias en la captura de ciertos datos relacionados con los procesos de la gestión de proyectos. La gran mayoría de estas herramientas no contemplan el tratamiento de la incertidumbre, ni el aprendizaje automático para la evaluación de proyectos.

### **1.3.2 Evaluación de proyectos con Xedro-GESPRO**

Como parte de la decisión de estandarizar los procesos de gestión de proyectos en la red de centros de desarrollo de *software* de la UCI, el Laboratorio de Investigaciones en Gestión de Proyectos implementa la Suite Xedro-GESPRO. Esta es una herramienta de *software* libre, genérico y adaptable, capaz de asistir a los usuarios en la gestión de los proyectos de cualquier tipo de organización. Se basa en tecnología Web y permite tratar con un fondo de recursos compartidos, administrado en función de las prioridades de los proyectos (Piñero, 2013).

Xedro-GESPRO se ha implantado en más de 20 centros como solución integral para la gestión de proyectos. Permite realizar la gestión mediante el control por cortes de varios indicadores correspondientes a: costo, tiempo, calidad, logística y desempeño de las personas que ejecutan las tareas. La evaluación de dichos indicadores puede ser cualitativa (bien, regular o mal) y cuantitativa, aplicando escalas a través de la lógica borrosa que permite ponderar los indicadores, con el objetivo de lograr una valoración integral en una fecha corte y determinar si se ha cumplido con lo planificado. Las evaluaciones de los proyectos en los cortes se recogen en un tablero de control (Abelardo et al., 2014) y se reflejan en la base de datos de proyectos en ejecución.

El sistema además provee reportes que contribuyen en gran medida al seguimiento y control del proyecto (Santiesteban et al., 2014). Muestra las tareas del cronograma de ejecución que presentan dificultades, identificando las causas con sus indicadores. Estos últimos son aplicables al proyecto, a una tarea, tarea resumen o sub-proyecto y representan una forma efectiva de medir el avance del proyecto permitiendo detectar los problemas para la toma de decisiones.

Xedro-GESPRO permite la gestión de portafolios de proyectos con sus respectivos cronogramas de ejecución y controlar el avance de los mismos. Lo anterior se logra mediante el cálculo automático de un conjunto de indicadores claves de rendimiento que reflejan el comportamiento de los proyectos, destacando los problemas y las causas que los originan, para sobre ellas, tomar decisiones a partir de una estrategia de prioridades previamente establecida.

La presente investigación parte del modelo implementado por (Lugo, 2012), el cual es automatizado e integrado con Xedro-GESPRO, desde el análisis de los datos primarios, hasta la caracterización y evaluación de la ejecución de los proyectos a partir de sus indicadores. Estos últimos reflejan el rendimiento de los proyectos que se gestionan con ayuda de Xedro-GESPRO y abarcan las principales áreas de conocimiento de la gestión de proyectos (Tabla 4).

Tabla 4. Indicadores de gestión de proyectos calculados por Xedro-GESPRO (Lugo, 2012).

Indicadores	Áreas de conocimiento de la gestión de proyectos
Índice de rendimiento de la ejecución (IRE)	Gestión del alcance y los compromisos
Índice de rendimiento de la planificación (IRP)	Gestión del tiempo
Índice de rendimiento de los costos (IRC)	Gestión de costos
Índice de rendimiento de la eficacia (IREF)	Gestión de alcance y calidad
Índice de rendimiento de los recursos humanos (IRRH)	Gestión de recursos humanos
Índice de rendimiento de la logística (IRL)	Gestión de logística
Índice de calidad del dato (ICD)	Consistencia de la información

A continuación se expone una síntesis de cada indicador (Lugo, 2012):

- IRE: relación entre la sumatoria del impacto de los hitos de ejecución cerrados y la sumatoria del impacto de los hitos de ejecución del proyecto planificados hasta la fecha de corte.
- IRP: muestra el estado de avance del proyecto; es la relación entre la sumatoria de los porcentos de ejecución real y la sumatoria de los porcentos de ejecución planificada de cada una de sus tareas según la fecha de corte.
- IRC: muestra el estado del presupuesto del proyecto. Relación entre el costo planificado del proyecto y el costo real según la fecha de corte.
- IREF: muestra el estado de la eficacia del proyecto, dada por la relación entre la realización de sus requisitos, la evaluación de sus tareas teniendo en cuenta la prioridad y sus no conformidades.
- IRRH: muestra el estado del desempeño del recurso humano en torno al desarrollo, impacto y prioridad de las tareas asignadas.
- IRL: propone el análisis en torno al movimiento de recursos materiales asociados al proyecto, desde los proveedores hasta el usuario final.
- ICD: propone el análisis de la completitud y correcta entrada de los datos a la herramienta de gestión de proyectos.

El cálculo automático de indicadores para la gestión de proyectos en las organizaciones, avala la calidad en la etapa de selección y con ello la coherencia de los datos necesarios para el inicio de los procesos de aprendizaje automático. Sin embargo, en el modelo de Lugo, la declaración de las reglas borrosas es estática y basada en el conocimiento de un conjunto de expertos. El Laboratorio de Investigaciones en Gestión de Proyectos continúa estudiando variantes y perfeccionando Xedro-GESPRO para lograr un acercamiento a las pautas del PMBOK y CMMI de forma que se presente una propuesta concreta que pueda ser aplicada en las organizaciones con mejora continua, presencia de incertidumbre en los datos primarios y cambiantes estilos de gestión (Lugo et al., 2014). Ejemplo de estas investigaciones son las siguientes:

- *Extensiones para el control de proyectos basadas en el análisis de la dimensión geográfica* (Jiménez, 2013). Desarrolla extensiones para el control de la ejecución de proyectos basadas en el análisis de la dimensión geográfica. Contribuye a mejorar la capacidad de ayuda a la toma de decisiones con Xedro-GESPRO al trabajar con ocho indicadores relacionados con las áreas claves de la gestión de proyectos para la clasificación de zonas geográficas.
- *Experiencias en la integración de sistemas de gestión de proyectos a nivel de datos con PostgreSQL* (Aldana and et.al., 2013). Existen varios enfoques para la integración de sistemas informáticos de una organización, como la integración a nivel de ficheros, punto a punto, a nivel de base de datos, entre otros. Se presenta una propuesta de integración de Xedro-GESPRO con otros sistemas a nivel de base de datos con PostgreSQL.

- *Aprendizaje de reglas borrosas para la toma de decisiones en la gestión de proyectos* (Pérez and et.al., 2013). Presenta un modelo para aprender y generalizar en forma de reglas borrosas el conocimiento implícito en bases de datos de proyectos terminados, integrado con Xedro-GESPRO. Se basa en una estrategia de construcción de reglas híbrida, combinando la teoría de conjuntos aproximados y el algoritmo de estimación de distribuciones UMDA.
- *Limpieza de datos: aplicación al sistema de gestión de proyectos GESPRO* (Torres et al., 2013). Define un procedimiento para la limpieza de datos que son recolectados con el fin de utilizarlos en el descubrimiento de conocimiento. Se utiliza Xedro-GESPRO como herramienta de base para la gestión de los proyectos productivos.
- *Sistema de información basado en reportes* (Santiesteban et al., 2014). Propone un sistema de información basado en reportes integrado a Xedro-GESPRO que cubre las áreas de conocimiento fundamentales de la gestión de proyectos: costo, tiempo, calidad, recursos humanos, comunicaciones e integración. Tiene en cuenta los niveles en la toma de decisiones existentes en la organización, como: persona, proyecto y centro.
- *Estrategia de formación integrada en Gestión de Proyectos Informáticos* (Torres et al., 2014). Propone una estrategia de formación integrada en gestión de proyectos informáticos, mediante cinco componentes que responden a la formación, investigación, colaboración y desarrollo de *software*. Es posible de extender hacia otros escenarios de formación en gestión de proyectos.

Al igual que sucede con las restantes herramientas informáticas para la gestión de proyectos, Xedro-GESPRO presenta insuficiencias en la adaptación y el aprendizaje automático, lo cual afecta la calidad de la clasificación de proyectos y la toma de decisiones. Aún no se resuelven de forma práctica los problemas relacionados con la evaluación de proyectos en organizaciones con mejora continua en los estilos de gestión y presencia de incertidumbre en los datos primarios.

#### **1.4 Tratamiento de la incertidumbre durante el control de proyectos**

En esta sección se abordan los sistemas de inferencia borrosos como mecanismo para tratar la incertidumbre en la información durante el control de la ejecución de proyectos. Se exponen las principales estrategias para el aprendizaje y adaptación de reglas borrosas. Se realiza un análisis comparativo de técnicas para la generación y optimización de reglas, enunciando de cada una sus ventajas y limitaciones. Finalmente se muestra un resumen comparativo de las herramientas de minería de datos más utilizadas en el último año, útiles para la aplicación de *soft computing*.

##### **1.4.1 Sistemas de inferencia borrosos**

Los Sistemas de Inferencia Borrosos (SIB) imitan la forma de razonamiento humano, permitiendo manejar eficazmente la incertidumbre de la información y conceptos vagos e imprecisos como los empleados en la vida cotidiana. Se caracterizan por ser sistemas robustos y tolerantes a imprecisiones en los datos de entrada. El centro de los SIB es la base de reglas; cuando esta es construida solamente a partir del conocimiento adquirido por un experto en gestión de proyectos o grupo de ellos, podría no funcionar correctamente al ser aplicada. Las principales causas son debido a que los expertos pueden equivocarse respecto al número de reglas, en relación a los antecedentes y consecuentes que las constituyen o en cuanto a la localización de los parámetros correspondientes a las funciones de pertenencia de los conjuntos borrosos.

La presente investigación trabaja con los SIB de múltiples entradas y una sola salida, conocidos como MISO (*Multiple Input Single Output*) (Martín, 2001).

La base del funcionamiento de los SIB son los conjuntos de la Lógica Borrosa (Zadeh, 1976, Zadeh, 1978). Un conjunto borroso  $A$  en un universo de discurso  $U$  está caracterizado por una función de pertenencia  $\mu_x$ , la cual a cada elemento del dominio le asigna un grado de pertenencia al conjunto en el intervalo  $[0,1]$  y se representa de la forma  $\mu_x:U \rightarrow [0, 1]$  (Cox et al., 1998, Zadeh, 1976). De

esta manera un mismo elemento puede pertenecer a varios conjuntos simultáneamente con ciertos grados de pertenencia. Cada conjunto borroso tiene asociado además un término lingüístico de forma tal que la función de pertenencia de un conjunto está ligada a una palabra, como por ejemplo: *Bajo*, *Medio*, *Alto* (Figura 1). El universo de discurso de la variable lingüística *Indicador* representa el rango de valores que puede tomar dicha variable y aporta los límites mínimo y máximo correspondientes a los términos lingüísticos de los extremos.

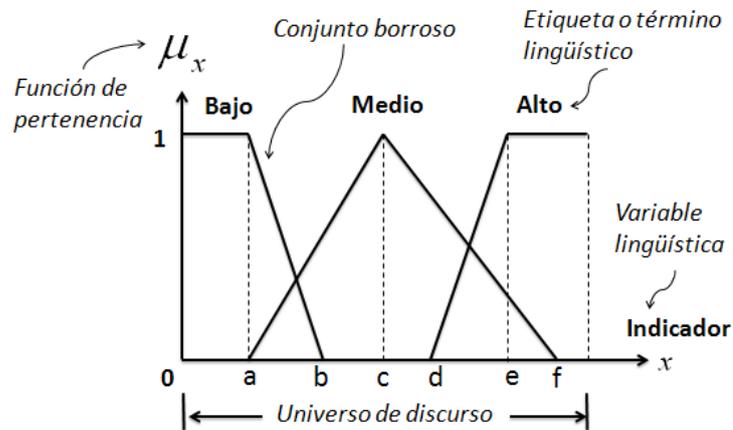


Figura 1. Términos relativos a los conjuntos borrosos.

Donde  $a$  y  $b$  son los parámetros de la función de pertenencia trapezoidal (específicamente del tipo: escalón decreciente) correspondiente al conjunto borroso *Bajo*;  $a$ ,  $c$  y  $f$  son los parámetros de la función de pertenencia triangular correspondiente al conjunto borroso *Medio*;  $d$  y  $e$  son los parámetros de la función de pertenencia trapezoidal (específicamente del tipo: escalón creciente) correspondiente al conjunto borroso *Alto*.

En el ámbito de investigación de los SIB, existen dos modelos fundamentales: Mamdani (Mamdani, 1977) y Sugeno (Sugeno, 1977). La principal diferencia de estos modelos reside en la forma de cálculo del consecuente de las reglas borrosas. El modelo de Mamdani tiene como base de funcionamiento la aplicación de un mecanismo fusificador, seguido de la inferencia de reglas borrosas y al final un mecanismo defusificador. En esta última fase existen distintas formas para aproximar el resultado lingüístico final a un valor numérico de mayor precisión; las principales variantes son el método del centroide o centro de gravedad, el máximo y el promedio de los máximos. El modelo de Sugeno (Figura 2) se caracterizan por tener entradas lingüísticas y procesar (agregar) los datos mediante un clasificador (reglas borrosas) devolviendo como salida los grados de pertenencia a los términos lingüísticos utilizados para expresar un resultado final. A diferencia del modelo anterior, este carece de defusificador, siendo la función de salida una combinación polinomial del resultado de agregar las variables de entrada.

Donde  $(x, y)$  son las entradas del sistema;  $A_i$  y  $B_i$  son las etiquetas lingüísticas de los conjuntos borrosos antecedentes;  $z_i$  son los consecuentes de las reglas;  $n$  corresponde a la cantidad de reglas;  $\{p_i, q_i, r_i\}$  es el conjunto de parámetros consecuentes o coeficientes en las ecuaciones de salida de las reglas y  $w_i$  son los pesos de las reglas borrosas.

Para resolver problemas de clasificación, con frecuencia se utilizan los SIB Sugeno Grado Cero por la estructura de las reglas que este tipo de sistema representa (Botía-Blaya, 2003). Este modelo también conocido como TSK (por sus autores Takagi, Sugeno y Kang) utiliza reglas de la forma: Si "x es A" y "y es B" entonces  $z=f(x, y)$  donde se aplican diferentes T-Normas y S-Normas (Takagi and Sugeno, 1985). Donde A y B son conjuntos borrosos en el antecedente y z usualmente  $f(x, y)$  es un polinomio cuyo grado determina el grado del modelo borroso. Cuando  $f$  es una constante, al sistema borroso correspondiente se le llama Sugeno Grado Cero.

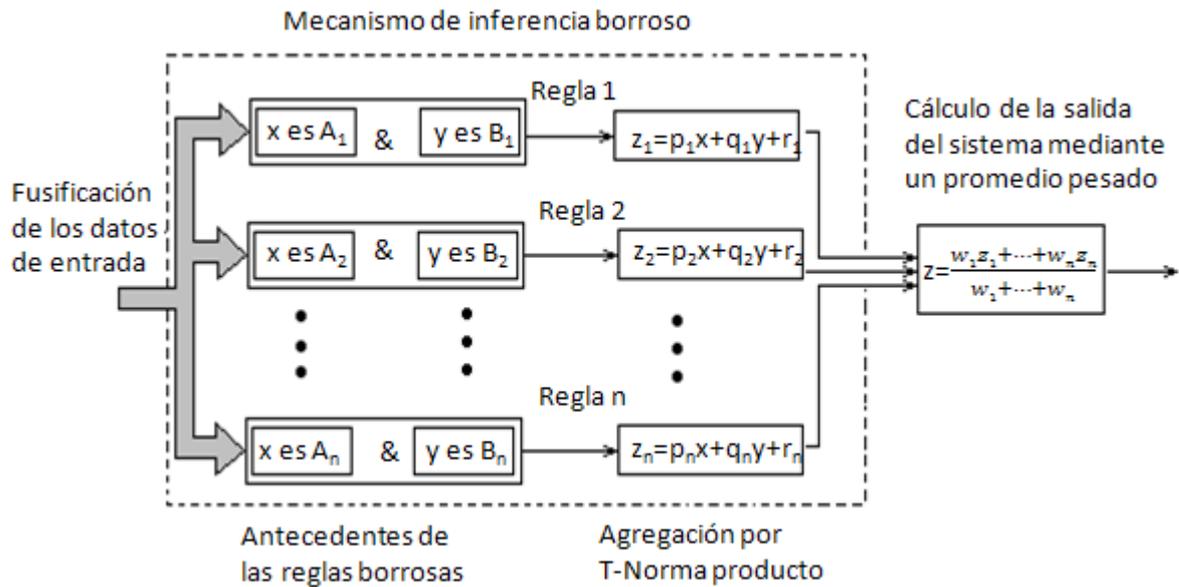


Figura 2. Representación de bloques que ilustra el funcionamiento de un SIB de tipo Sugeno.

Muchos SIB utilizan una base de reglas estáticas, donde estas son introducidas al sistema de forma manual; además, los conjuntos borrosos de las variables de entrada y salida se basan en la experiencia y el conocimiento de expertos humanos. En (Lugo, 2012) se propone un modelo para el control de la ejecución de proyectos utilizando indicadores y lógica borrosa mediante un SIB tipo Sugeno Grado Cero con 27 reglas estáticas, definidas según el juicio de expertos, al igual que las funciones de pertenencia que utiliza de tipo trapezoidal y triangular.

Con la implementación del criterio de los expertos para la construcción de las reglas y sus conjuntos borrosos se introducen elementos de ruido en el método de evaluación, debido a su fundamento subjetivo y dependiente de la experiencia. Además las reglas que miden los indicadores, al ser estáticas, no permiten la adaptación del sistema de evaluación según el rigor con que se controlan los proyectos, dado por el nivel de madurez alcanzado por la organización. Esto implica que los datos que brinda el sistema sobre el estado de los proyectos no sean suficientes como apoyo a la toma de decisiones para controlar correctamente su ejecución. Para adaptar las reglas borrosas es conveniente emplear técnicas de optimización que permitan ajustarlas de forma automática según el desarrollo del entorno de la organización.

#### 1.4.2 Estrategias para la adaptación y el aprendizaje automático de reglas

Planteados los elementos fundamentales de los SIB se prosigue con el análisis de los diferentes mecanismos empleados en la construcción y explotación de los mismos. Para ello se pueden aplicar varios enfoques en el momento de realizar el aprendizaje y la adaptación de las reglas de inferencia borrosas (Piñero, 2005):

- (a) generar reglas duras y transformarlas en borrosas,
- (b) generar la base de reglas borrosas candidatas y refinarlas, o
- (c) generar la base de reglas borrosas inicial y optimizarlas.

Uno de los algoritmos clásicos en la generación de la base de reglas borrosas candidatas y refinamiento lo constituye el propuesto por (Wang and Mendel, 1992) basado en partición de espacio. El método WM (*Wang & Mendel*) permite generar reglas borrosas a partir de datos numéricos y es capaz de aproximar funciones continuas. En una primera fase este algoritmo genera todas las posibles reglas que se obtienen de las combinaciones de los conjuntos borrosos, estos últimos predefinidos de antemano. En un segundo estado refina la base de reglas inicial seleccionando solamente aquellas que sean representativas del conjunto de entrenamiento. Como desventaja se le señala que necesita la definición previa de los conjuntos borrosos lo cual obliga a

la intervención de expertos. Otra desventaja consiste en que genera gran número de reglas como parte de la base inicial de reglas candidatas; algunas muy específicas y otras incluso sin sentido.

Sin dejar de lado el resto de los enfoques, la presente investigación se centra en el enfoque (c) ya que pretende generar y optimizar una base inicial de reglas borrosas para realizar la evaluación de proyectos según varios indicadores de control de ejecución. En este enfoque tiene una probada eficiencia el aprendizaje basado en la aplicación de técnicas de redes neuronales y algoritmos genéticos (Herrera, 2004).

Se parte de la necesidad de generar y ajustar las reglas borrosas a los casos de proyectos terminados que arriban a conclusiones con la implementación del sistema de indicadores y la evaluación de expertos de la organización.

Para optimizar la base de reglas borrosas inicial se pueden aplicar varias alternativas:

- (a) asignarle pesos a las reglas borrosas y ajustarlos,
- (b) modificar los parámetros de las funciones de pertenencia,
- (c) transformar la estructura de la base de reglas, o
- (d) alguna combinación de las anteriores.

Para optimizar la base de reglas borrosas se adopta como estrategia fundamental la (b); esta selección se debe a que básicamente se requiere mantener la estructura e importancia de las reglas, siendo necesario ajustar los parámetros de las funciones de pertenencia de los conjuntos borrosos.

### **1.4.3 Técnicas para la generación inicial de reglas borrosas y su optimización**

En esta sección se realiza un análisis de las tendencias en cuanto a la generación de la base inicial de reglas y el ajuste de los parámetros de las funciones de pertenencia de los conjuntos borrosos. Una comparación minuciosa entre estas estrategias se presenta en (Nauck et al., 1999, Piñero, 2005). En este enfoque sobresalen las siguientes tres variantes:

- (a) métodos derivados de Redes Neuronales Artificiales (RNA) basados en el gradiente,
- (b) métodos heurísticos que no usan información de gradiente,
- (c) algoritmos evolutivos, principalmente Algoritmos Genéticos (AG).

La estrategia (a) tiene su base en la probada efectividad de los mecanismos de aprendizaje de las RNA, son muy utilizados algoritmos como el *backpropagation* (propagación hacia atrás del error) (Freeman and Skapura, 1991). Su principal limitación está dada por la restricción que imponen a las funciones de pertenencia de los conjuntos borrosos de que sean derivables ya que basan su funcionamiento en el cálculo del gradiente (Grauel and Ludwig, 1999). Entre los algoritmos que usan esta estrategia para optimizar las reglas borrosas se encuentran:

- En (Nomura et al., 1992) se propone un algoritmo de aprendizaje de reglas de inferencia borrosas mediante un método de descenso (FIR.DM: *Fuzzy Inference Rules by Descent Method*). A partir de los datos de entrada-salida se obtienen de forma automática las reglas de inferencia que expresan la relación de los datos. Las funciones de pertenencia en la parte antecedente y el número real en la parte consecuente de las reglas de inferencia son ajustadas por medio del método de descenso. Permite que los conjuntos borrosos de cada regla sean modificados de forma independiente al resto de las reglas. La velocidad de aprendizaje y la capacidad de generalización de este método son superiores a las de una red neuronal de retropropagación tipo convencional (*backpropagation*). Su desventaja radica en que admite múltiples funciones de pertenencia para un mismo término lingüístico, efecto que atenta contra la interpretación del modelo borroso y de sus reglas.
- En (Jang, 1993) se introduce uno de los primeros modelos neuro-borrosos de tipo híbrido llamado Sistema de Inferencia Borroso basado en Redes Adaptativas (ANFIS: *Adaptive-Network-based Fuzzy Inference System*). Representa un SIB de tipo Sugeno por medio de una RNA multicapa con funciones de pertenencia tipo campana. La optimización se realiza al ajustar los parámetros

de las funciones de pertenencia de los antecedentes y de los consecuentes de las reglas. El aprendizaje se divide en dos etapas: modificar los consecuentes siguiendo la estrategia de los mínimos cuadrados y después modificar los parámetros de los antecedentes por medio del gradiente descendente. Estudios comparativos de diferentes estrategias de aprendizaje automático aplicadas en la toma de decisiones, muestran buenos resultados con los ANFIS (Kelemen et al., 2002). Las capacidades adaptativas de las redes ANFIS las hacen aplicables a una gran cantidad de problemas, donde se conozca la clasificación de los datos. Tiene como limitaciones que las funciones de pertenencia en los antecedentes requieren ser derivables y el costo computacional es alto, sobre todo si es grande la cantidad de conjuntos borrosos.

- En (Nauck et al., 1999) se presenta un sistema borroso conducido por redes neuronales (NNFR: *Neural Network Driven Fuzzy* –) que implementa otro algoritmo de optimización con fundamento neuro-borroso. Este es basado en una red neuronal perceptrón multicapas (MLP: *Multi Layer Perceptron*) con  $n$  unidades de entrada y  $k$  unidades de reglas en la capa oculta. Tiene como limitaciones que requiere conocer la base inicial de reglas y las funciones de pertenencia en los antecedentes han de ser derivables.
- La combinación de gradiente descendente con otros métodos, es una estrategia ampliamente aplicada (Pal et al., 2001), algunos ejemplos de algoritmos se abordan en (Jang et al., 1997, Castellano et al., 2005). El sistema de control inteligente basado en razonamiento aproximado generalizado (GARIC: *Generalized Approximate Reasoning-based Intelligent Control*) utiliza un modelo de red neuronal multicapa combinado con el aprendizaje por refuerzo y mezcla las estrategias (a) y (b) (Berenji, 2012, Kruse et al., 2013).
- En (Ishibuchi et al., 1994) se propone un método para la generación de reglas borrosas mediante heurística y gradiente descendente (FS.HGD: *Fuzzy System by Heuristics and Gradient Descent*). Utiliza reglas borrosas de la forma IF-THEN con números reales en los consecuentes. La técnica de partición de espacio se implementa para generar la parte antecedente, mientras que la parte consecuente inicial de cada regla está determinada por el valor de la media ponderada de los datos de entrenamiento dado. Mediante el método del gradiente descendente se actualiza el valor de la parte consecuente. El valor heurístico dado por el usuario afecta el valor del peso de cada dato.
- En (Kim and Kasabov, 1999) se propone un sistema de inferencia neuro-borroso híbrido (HYFIS: *Hybrid neural Fuzzy Inference System*) para construir y optimizar sistemas borrosos. El modelo propuesto integra el poder de aprendizaje de las redes neuronales con los sistemas de lógica borrosa y provee significados lingüísticos a las arquitecturas conexionistas. Las reglas borrosas, así como las funciones de pertenencia de entradas y salidas son optimizadas según los casos de entrenamiento mediante un esquema de aprendizaje híbrido compuesto por dos fases: la generación de reglas desde los datos y el ajuste de las reglas mediante el *backpropagation*. El método propuesto puede ser aplicado en aprendizaje adaptativo incremental para predecir y controlar sistemas dinámicos no lineales. Aunque el enfoque neuro-borroso proporciona un buen rendimiento para los sistemas dinámicos no lineales, no siempre puede satisfacer todos los requisitos de las aplicaciones del mundo real. En tales casos, la estrategia de modelado elegido debe ser integrada con otros enfoques para que los diseñadores de un sistema inteligente híbrido puedan elegir entre varios paradigmas, incluyendo la computación evolutiva.
- En (Wanga et al., 2003) se presenta un sistema neuro-borroso que combina el método del gradiente descendente con los mínimos cuadrados para optimizar una base de reglas de tipo Sugeno donde los consecuentes son una combinación lineal de los antecedentes. En este caso se optimizan los antecedentes utilizando el gradiente descendente y los consecuentes por medio de los mínimos cuadrados.

- En (Pedrycza and Reformata, 2005) se propone un método que en una primera fase construye una base de reglas borrosas potenciando las propiedades como optimizadores globales de los AG. En una segunda fase aplica un gradiente descendente para refinar las reglas. Este método sólo permite el trabajo con datos numéricos y con funciones de pertenencia campanas que se obtienen de aplicar el AG.

Ante las restricciones de los métodos de aprendizaje basados en el gradiente algunos autores apuestan por el empleo de heurísticas y siguen la estrategia (b). Modelos y sistemas que derivan su aprendizaje de las RNA y que constituyen ejemplos del enfoque alejado del gradiente son:

- Nauck presenta en su tesis doctoral (Nauck, 2000) algoritmos de aprendizaje basados en heurísticas que no imponen restricciones de derivación a las funciones del modelo. Estos algoritmos son NEFCLASS (*Neuro-Fuzzy data analysis a Classification System* – sistema de clasificación por análisis de datos neuro-borroso) y NEFPROX (*Function Approximation* – aproximación de funciones). Su principal deficiencia radica en que modifica los parámetros de las funciones de pertenencia teniendo en cuenta cada uno de los objetos del conjunto de entrenamiento cada vez. Esto provoca que patrones aprendidos se pierdan durante la optimización y hace dependiente el proceso de aprendizaje del orden en que aparezcan los objetos en el conjunto de entrenamiento. Además durante el aprendizaje y la modificación de los parámetros no se tiene en cuenta las relaciones entre diferentes conjuntos borrosos, se obvia el concepto de variable lingüística, por lo que propone un paso final de corrección de los conjuntos borrosos en vista a lograr configuraciones consistentes.
- Otros métodos heurísticos no basados en el gradiente se trabajan en (Pedrycza and Reformata, 2005, Sánchez and Otero, 2004, Jamei et al., 2004, Alcalá and Herrera, 2004, Spiegela and Sudkampb, 2003). Estos se basan en aplicar desplazamientos en los parámetros de los conjuntos borrosos en correspondencia con el error que produce cada regla durante la clasificación. La cantidad de reglas es un parámetro prefijado. Durante el aprendizaje se trata de forma aislada cada regla, no teniendo en cuenta la cooperación entre ellas, así como tampoco el concepto de base de reglas que es la solución final del problema. Para garantizar un correcto aprendizaje requiere que el SIB final use como T-Norma el producto y que establezca un proceso para defusificar antes del proceso de agregación. Además obliga a que la agregación esté basada en el operador suma.
- Piñero presenta en su tesis doctoral (Piñero, 2005) un algoritmo de aprendizaje basado en redes neuronales para ajustar los parámetros de las funciones de pertenencia. Estos parámetros se dividen en posición y amplitud, permitiendo determinar la localización y el solapamiento de los conjuntos borrosos. El algoritmo determina la responsabilidad en la clasificación de cada conjunto borroso y de acuerdo a ello modifica sus parámetros mediante el cálculo de un término delta que se suma o resta al parámetro en cuestión. La modificación de los parámetros de los conjuntos borrosos se realiza en correspondencia con el error cometido durante la clasificación del conjunto de entrenamiento y no de forma secuencial a medida que se procesan los casos. Este algoritmo tiene como principales ventajas que no requiere que las funciones de pertenencia sean derivables y tiene en cuenta la interrelación entre los conjuntos borrosos de una misma variable lingüística.

La estrategia (c) optimiza las reglas utilizando algoritmos evolutivos, especialmente AG. Este enfoque tiene como base la probada eficiencia de los algoritmos evolutivos en la optimización y las pocas restricciones que imponen a los modelos. Sin embargo los AG clásicos presentan dificultades en la optimización de funciones con interacciones no lineales entre las variables debido a que por su naturaleza están preparados para el trabajo con variables independientes (Muhlenbein et al., 1999, Ochoa et al., 2000, Soto, 2003, Cotta and Moscato, 2004, Pelikan et al., 2002, Fernández et al., 2015). Entre los parámetros de los conjuntos borrosos hay un alto grado de dependencia y la

combinación correcta de estos parámetros garantiza la capacidad de la variable lingüística de representar con eficiencia la naturaleza y la realidad de los datos. La aplicación de los operadores tradicionales de cruzamiento y mutación no tiene en cuenta la interrelación entre las variables. En los SIB juega un importante papel el concepto de variable lingüística y los conjuntos borrosos que la conforman, existiendo una inexorable interrelación entre ellos. Entre los algoritmos que usan esta estrategia para optimizar las reglas borrosas se encuentran:

- En (Thrift, 1991) se propone un algoritmo que implementa la técnica de Thrift basada en un algoritmo genético (GFS.THRIFT: *Genetic Fuzzy Systems based on Thrift's technique*) para generar y optimizar sistemas de inferencia borrosos. El método de Thrift se basa en el modelo de Mamdani (Mamdani, 1977) y considera una tabla como un cromosoma con indicadores de conjuntos borrosos sobre el dominio de salida. Un cromosoma se forma a partir de la tabla de decisiones y la producción de una cadena de números del conjunto de códigos. Utiliza operadores de cruce y mutación estándares para modificar dicha cadena de números. Sin embargo, este modelo funciona para estructuras de reglas simples y con pocas variables de entrada.
- El modelo de Pittsburg se basa en un AG clásico, donde cada cromosoma es una base de reglas y la función objetivo representa la calidad de las respuestas (Gómez-Skarmeta and Jiménez, 1999). Un cromosoma contiene todos los conjuntos borrosos de la base de reglas. Su ventaja radica en la fácil aplicación y la posibilidad de considerar el concepto de variable lingüística durante la optimización. Presenta la desventaja de tener un alto costo computacional debido a la necesaria evaluación de todas las bases de reglas en cada generación.
- El modelo de Michigan se basa en un AG donde cada cromosoma es una regla y cada población que se genere forma una base de reglas. El algoritmo fomenta la competencia entre las reglas de la misma población y entre generaciones donde los individuos de una misma generación cooperan para lograr una solución final. Ejemplos de SIB que aplican este algoritmo se abordan en (Ishibuchi et al., 2001, Jamei et al., 2004, Makrehchi et al., 2003). Su principal limitación radica en que obvia el concepto de variable lingüística. Una vez iniciada la optimización, los conjuntos borrosos se circunscriben a la regla donde se encuentran; se pierde la interrelación entre los conjuntos borrosos de una misma variable lingüística. Al finalizar la optimización pueden existir varios conjuntos borrosos correspondientes al mismo término, con diferentes parámetros. Esta situación influye negativamente en el poder de interpretación de los conjuntos borrosos y sus respectivos términos lingüísticos.
- El modelo de aprendizaje iterativo reduce el espacio de búsqueda concentrando la evolución en reglas aisladas en lugar de en la base de reglas. Existe competencia sólo entre las reglas de la misma población, se selecciona la mejor de acuerdo al criterio de calidad y se inserta en la base de reglas final. Presenta una cooperación entre las nuevas y las viejas reglas seleccionadas. Ejemplos de SIB que aplican este algoritmo se abordan en (González and Pérez, 1999, Córdon et al., 1999). La principal desventaja radica en que el aprendizaje se concentra en las reglas y se pierde el sentido de la optimización de los conjuntos borrosos aspecto que atenta contra el poder de interpretación del SIB.
- Algunas alternativas para solucionar las deficiencias de los AG se identifican en (Piñero, 2005), con variantes de algoritmos evolutivos que desde su concepción brindan solución al problema de interrelación entre las variables. Propone modificaciones para los Algoritmos de Estimación de Distribuciones (EDA: *Estimation of Distribution Algorithms*), siendo una herramienta natural para la construcción de variables lingüísticas y la determinación de los parámetros de sus respectivos conjuntos borrosos (Larrañaga et al., 2003). En los EDA, a diferencia de los algoritmos evolutivos clásicos, la población de soluciones candidatas se recrea en cada

generación según la distribución de probabilidad obtenida a partir de los mejores individuos de la generación anterior (Larrañaga et al., 2003). Dado que la población no se regenera a partir de individuos, sino desde las distribuciones de probabilidad obtenidas, no existen operadores de cruzamiento ni de mutación. El uso de los EDA en la optimización y generación de bases de reglas, reportan mejores resultados que los AG (Mahing and Muhlenbein, 2001, Larrañaga and Lozano, 2001, Larrañaga et al., 2003, Santana et al., 2001, Alba et al., 2004). Sin embargo por su alta complejidad computacional deben implementarse usando variantes distribuidas.

- En (Alcalá et al., 2007) se propone un algoritmo genético distribuido (GFS.LT.RS: *Genetic Fuzzy Systems by Lateral Tuning and Rule Selection*) para la adaptación de un sistema basado en reglas borrosas. Muestra un buen desempeño del enfoque propuesto en problemas de alta dimensión y su capacidad de cooperar con los métodos para eliminar las reglas innecesarias. Su enfoque de post-procesado realiza un ajuste lateral evolutivo de las funciones de pertenencia, combinado con estrategias de selección de reglas que mejoran la precisión del sistema, manteniendo un nivel de interpretación aceptable. Sin embargo, en algunos casos, el espacio de búsqueda considerado al combinar ambas técnicas es demasiado grande, lo cual puede derivar en un modelo no óptimo.

Cada una de estas estrategias tiene potencialidades y limitantes en su aplicación a diferentes situaciones. En la presente investigación se persigue aplicar varias técnicas de *soft computing* basadas en RNA, partición espacial, gradiente descendente y AG, con el objetivo de lograr un equilibrio entre las potencialidades de cada técnica para seleccionar finalmente el sistema basado en técnicas de *soft computing* más adecuado para el entorno de aplicación. El campo del *soft computing* ha evolucionado de un modo vertiginoso, constantemente se proponen nuevas técnicas y aplicaciones. La revisión de todas las propuestas que aparecen en la literatura en este campo está fuera del ámbito de este trabajo.

#### **1.4.4 Herramientas informáticas para la aplicación de *soft computing***

Al igual que para el caso de la gestión de proyectos, en las últimas décadas se han desarrollado numerosas herramientas informáticas útiles para la aplicación de técnicas de *soft computing*. A menudo a estas herramientas se les conoce como de Minería de Datos (MD). Dentro del procedimiento de extracción de conocimiento de bases de datos (KDD: *Knowledge Discovery in Databases*), la MD es la etapa que realiza el descubrimiento de conocimiento mediante la aplicación de procedimientos automáticos o semiautomáticos. Para ello se utilizan métodos de inteligencia artificial, aprendizaje automático, estadística y *soft computing* (Mathur and Nand, 2014).

En la Tabla 5 se muestra un resumen de las 15 herramientas de MD más utilizadas a nivel mundial en el último año, según una encuesta realizada por (KDnuggets, 2014). De cada herramienta se especifica en la primera columna la cantidad de votos obtenidos y el porcentaje de votos que recibió como herramienta de uso exclusivo. En la segunda columna se muestra el porcentaje de votos respecto a un total de 3285 recibidos en el 2014 y el porcentaje obtenido en el 2013. Según la encuesta, el 71% de los usuarios utilizan herramientas propietarias, mientras que el 78% herramientas libres.

En los dos primeros puestos se posicionan herramientas basadas en *software* libre: Rapidminer (Rapidminer Inc., 2015) y R (R Core Team, 2015). Rapidminer es una herramienta muy ágil que permite diseñar un proceso de minería de datos de manera visual e intuitiva, con la facilidad de extensión a partir de operadores que implementan técnicas de *soft computing*.

Otras herramientas como Weka (Waikato, 2015) y MATLAB (Mathworks, 2014) poseen iguales facilidades, a través de la incorporación de *plugins* y *toolboxes* respectivamente. Sin embargo, las conectividades ad-hoc a los orígenes de datos pueden ocasionar ineficiencias en el sistema de

información a emplear dado que aparecen variables a tener en cuenta como la conectividad, velocidad de la red, disponibilidad del servicio y arquitectura modular.

Por otra parte, R (R-Foundation, 2015) es un lenguaje de programación para análisis estadístico y gráfico basado en *software* libre con una alta capacidad para la integración directa a nivel de funciones con bases de datos, a través de lenguajes procedurales como PL/R (Conway, 2015), con facilidades de instalación tanto en entornos privativos como de código abierto. R cuenta con una amplia comunidad de desarrolladores, investigadores y usuarios, bajo la Licencia Pública General (GPL: *General Public Licence*) (Free Software Foundation, 1991). Se realizan actualizaciones sistemáticas mediante el R-Cran Packages (R-CRAN, 2015), un repositorio para la creación de paquetes o bibliotecas con algoritmos implementados listos para su aplicación en diferentes dominios de información genéricos o específicos.

Tabla 5. Herramientas de minería de datos más utilizadas (KDnuggets, 2014).

	Herramienta	Votos obtenidos	Porcentaje de votos obtenidos como herramienta de uso exclusivo	Software libre	Porcentaje de uso en 2014 en 2013
1.	RapidMiner	1453	35.1%	Si	44.2% 39.2%
2.	R	1264	2.1%	Si	38.5% 37.4%
3.	Excel	847	0.1%	No	25.8% 28.0%
4.	SQL	832	0.1%	No	25.3% na
5.	Python	639	0.9%	Si	19.5% 13.3%
6.	Weka	558	0.4%	Si	17.0% 14.3%
7.	KNIME	492	10.6%	Si	15.0% 5.9%
8.	Hadoop	416	0%	No	12.7% 9.3%
9.	SAS base	357	0%	No	10.9% 10.7%
10.	Microsoft SQL Server	344	0%	No	10.5% 7.0%
11.	Revolution Analytics R	300	13.3%	No	9.1% 4.5%
12.	Tableau	298	1.3%	No	9.1% 6.3%
13.	MATLAB	277	0%	No	8.4% 9.9%
14.	IBM SPSS Statistics	253	0.4%	No	7.7% 8.7%
15.	SAS Enterprise Miner	235	1.3%	No	7.2% 5.9%

El lenguaje R tiene entornos de desarrollo integrados a su disposición como R-Studio (RStudio, 2015), que facilitan la incorporación de paquetes del R-Cran para su explotación y prueba en proyectos experimentales. Entre estos paquetes se encuentra FRBS: *Fuzzy Rule-based Systems for Classification and Regression Tasks* (sistemas basados en reglas borrosas para la clasificación y regresión de tareas) (Septem, 2014, Riza et al., 2015), desarrollado por el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada. FRBS se basa en el concepto de lógica difusa propuesto por (Zadeh, 1994) y representa los sistemas borrosos para manejar diversos problemas mediante la implementación de técnicas de *soft computing*.

A pesar de la existencia de las herramientas analizadas, aún no se reporta en la literatura el desarrollo de soluciones que permitan una fácil integración a nivel de bases de datos entre estas herramientas y las plataformas informáticas de gestión de proyectos. Esto permitiría dotar a los sistemas de gestión de proyectos de mecanismos robustos para contribuir al tratamiento de la incertidumbre en la información, así como mejorar su adaptabilidad ante los cambiantes estilos de gestión provocados por la mejora continua en las organizaciones.

#### **1.4.5 Técnicas de *soft computing* aplicadas en la gestión de proyectos**

En esta sección se presenta la actualidad sobre la aplicación de técnicas de *soft computing* en el área de la gestión de proyectos con la intención de proporcionar una síntesis del marco teórico y una breve comparación de las investigaciones examinadas de los últimos años. Se orienta la atención hacia la adaptación y el aprendizaje automático de las herramientas para la evaluación de la ejecución de proyectos.

Varias investigaciones recientes han sugerido soluciones teóricas basadas en minería de datos, inteligencia artificial y *soft computing*, útiles para la gestión de proyectos. Aplicaciones específicas de estas técnicas en la gestión de proyectos son relativamente pocas en comparación con otras áreas de aplicación. A continuación se citan varios trabajos relacionados:

- En (HERACLES, 2004) se conjuga la utilización de técnicas de aprendizaje automático y modelado dinámico de proyectos *software* como una alternativa a la estimación y gestión de proyectos. Permite simular el comportamiento de un proyecto, obtener reglas de gestión basadas en la experiencia de la organización y generar reglas borrosas de forma automática. Sin embargo, una vez obtenidas las reglas de gestión, es el responsable del proyecto quien decide qué reglas son las más adecuadas; además las reglas borrosas sólo se pueden ajustar a nuevos entornos si se vuelven a generar, lo que implica ejecutar todo el proceso de generación nuevamente. Carece de mecanismos de optimización o ajuste de las reglas.
- En (Dweiri and Kablan, 2006) se combinan las medidas de costo, tiempo y calidad para obtener la eficiencia interna de desarrollo del proyecto. El sistema de toma de decisiones borroso que proponen para la evaluación de la eficiencia se implementa con MATLAB. Este es un modelo estático donde las reglas son introducidas al sistema de forma manual y los conjuntos borrosos de las variables de entrada y salida se basan en la experiencia y el conocimiento de expertos humanos. Su propuesta es ilustrada en un caso de estudio y carece de mecanismos de optimización o ajuste de las reglas.
- En (Bath, 2010) se propone un método basado en *soft computing* para clasificar proyectos en tres clases: simple, medio o complejo. El enfoque propuesto se divide en varias fases: pre-procesamiento de datos para eliminar la información irrelevante, fase de extracción de características (variables de entrada) y fase de análisis de relevancia para eliminar características irrelevantes. Sin embargo estos aportes teóricos y experimentales no dan solución al problema de investigación que compete a la evaluación de la ejecución de proyectos mediante indicadores de gestión.
- En (Gao, 2010) se construye un sistema de índices para evaluar el desempeño en la gestión de proyectos. Este es tratado como un problema de decisión multi-criterio que combina las teorías borrosas y los procesos analíticos en red (ANP: *Analytic Network Process*) para darle solución. Tiene en cuenta los índices de calidad, costo, tiempo y riesgos. En su investigación no incluyen técnicas para realizar aprendizaje y ajustar los parámetros de los conjuntos borrosos.
- En (Certa et al., 2010) se propone un sistema de inferencia borroso para el cálculo del indicador que evalúa el rendimiento del proyecto, conjugando dos entradas obtenidas según el análisis del valor ganado (rendimiento de los costos y rendimiento de la planificación). Tiene en cuenta ocho conjuntos borrosos para las variables lingüísticas de entrada y siete para la de salida, obteniendo

64 reglas borrosas que abarcan todas las combinaciones de las variables de entrada. Los conjuntos borrosos son representados con funciones triangulares y trapezoidales, tomando valores positivos y negativos; los parámetros de las funciones son definidos por expertos. Si bien es un interesante aporte, este es un modelo estático que no se ajusta de forma automática según el desarrollo de la organización.

- En (Lugo, 2012) se propone un modelo para el control de la ejecución de proyectos utilizando indicadores y lógica borrosa, solución aplicada sobre la herramienta Xedro-GESPRO. El modelo utiliza un SIB tipo Sugeno Grado Cero para la evaluación de proyectos mediante indicadores. Propone una base de 27 reglas definidas según el juicio de expertos. Utiliza funciones de pertenencia trapezoidales en los extremos y triangulares en el centro, cubriendo tres conjuntos borrosos (bajo, medio, alto) para cada indicador. El modelo de funcionamiento del SIB utiliza como T-Norma el Producto y como S-Norma el Máximo. Este es un modelo estático que no se ajusta según el desarrollo y madurez de la organización.
- En (Mewada et al., 2013) se expone un modelo de evaluación de esfuerzo de *software* aplicando ANFIS sobre datos con incertidumbre. Se muestra que esta técnica aporta muy buenos resultados en las predicciones y mejores que los obtenidos de aplicar otras técnicas como las redes neuronales. Brinda un breve panorama sobre la aplicación de técnicas tradicionales y de *soft computing* en la evaluación de esfuerzo de *software*. Para la implementación utiliza MATLAB sobre un total de 18 proyectos del conjunto de datos NASA. El modelo no es precisamente aplicado en la evaluación de la ejecución de proyectos.
- En (Liu et al., 2013) se presenta un sistema basado en RNA para la aplicación del modelo de madurez de gestión de proyectos organizacional. El sistema es entrenado y validado con MATLAB. Como resultado demuestran que se puede obtener una valoración precisa y objetiva. Sin embargo, estos resultados no son integrados con herramientas de gestión de proyectos, ni incluye técnicas para la generación de reglas borrosas con fines de evaluar la ejecución del proyecto.
- En (Govindarajan, 2014) se propone un sistema de inferencia borroso para evaluar la eficiencia en la gestión de proyectos de *software*. Para la implementación hacen uso del paquete Lógica Borrosa de MATLAB y *Simulink*. Utilizan cuatro funciones triangulares para los conjuntos borrosos de las dos variables de entrada (retraso del proyecto y cambios en el retraso del proyecto) y en la variable de salida (eficiencia del rendimiento del proyecto de *software*). Como trabajo futuro proponen utilizar ANFIS para ajustar el SIB creado, ya que este es estático.

De manera general las investigaciones analizadas no integran las siguientes características:

- Adaptación del sistema según la mejora continua en los estilos de gestión de la organización.
- Integración de la propuesta con herramientas de gestión de proyectos.

A pesar de las diversas aplicaciones de *soft computing* en sistemas de apoyo a la toma de decisiones en la gestión de proyectos, aún no se resuelven los problemas relacionados con la evaluación de proyectos en organizaciones con mejora continua en los estilos de gestión y presencia de incertidumbre en los datos.

### **1.5 Conclusiones del capítulo**

En el presente capítulo se examina el marco teórico en torno a las tendencias en el control de la ejecución de proyectos; se analizan escuelas, experiencias y herramientas utilizadas en los últimos años, dando cumplimiento al primer objetivo específico de la investigación. Se expone la fundamentación esencial, la adaptación y el aprendizaje automático de los sistemas de inferencia borrosos. Se analizan herramientas e investigaciones recientes sobre la aplicación de técnicas de

*soft computing* en el área de la gestión de proyectos. De los modelos abordados, de su resultado y discusión, se puede arribar a las siguientes conclusiones:

- 1) Las escuelas de gestión de proyectos analizadas abordan la mejora continua en la organización, el tratamiento de la incertidumbre, el trabajo con lecciones aprendidas y la influencia de las herramientas informáticas en el éxito de los proyectos. Sin embargo, se enfocan en la descripción verbal de los procesos y no definen métodos concretos para el aprendizaje y la adaptación de la evaluación de la ejecución de proyectos según la mejora continua dada por el nivel de madurez alcanzado por la organización.
- 2) A partir del estudio de herramientas de gestión de proyectos se concluye que la gran mayoría de ellas no contemplan funcionalidades relacionadas con la adaptación y el aprendizaje automático para la evaluación de proyectos y el tratamiento de la incertidumbre en la información, lo cual afecta la calidad de la clasificación de proyectos y por tanto la toma de decisiones. Se destaca además la importancia de utilizar *software* libre de código abierto.
- 3) Según la bibliografía consultada, la aplicación de técnicas de *soft computing* en el control de la ejecución de proyectos es un tema poco abordado. La mayoría de los modelos analizados no se integran con herramientas de gestión de proyectos, ni permiten su aprendizaje automático según la mejora continua en los estilos de gestión de la organización.
- 4) Sobre la base de las ideas anteriores, se concluye la necesidad de desarrollar un sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos utilizando tecnologías de *software* libre y que se integre con herramientas de gestión de proyectos. Con ello se persigue mejorar la calidad de la clasificación de los proyectos como apoyo a la toma de decisiones en organizaciones con mejora continua en los estilos de gestión y presencia de incertidumbre en los datos primarios.

## **CAPÍTULO 2. SISTEMA BASADO EN TÉCNICAS DE SOFT COMPUTING**

En el presente capítulo se define el sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos. Se abordan los principios necesarios para el funcionamiento del sistema y se presentan sus componentes. Se describen las técnicas de *soft computing* utilizadas para generar y ajustar los sistemas de inferencia borrosos. Por último, se presentan las conclusiones del capítulo.

### **2.1 Principios de funcionamiento del sistema**

En la investigación se aplican algoritmos basados en aprendizaje supervisado que permiten ajustar los parámetros graduables del sistema de inferencia borroso, de manera tal que la salida calculada por este, sea lo más parecida a la salida esperada. Para ello es necesario contar con datos de entrenamiento compuestos por patrones de entrada y salida esperada. El aprendizaje se realiza a partir de un conjunto de proyectos terminados de los cuales se conocen los valores numéricos de los indicadores claves de gestión y la evaluación dada por expertos de la organización. De dicha base de proyectos se extrae el conjunto de casos de entrenamiento y el conjunto de casos de validación.

Para la puesta en práctica del sistema se necesita conocer de antemano los indicadores de gestión que se van a utilizar en el aprendizaje, así como la cantidad de etiquetas o términos lingüísticos por cada indicador. La actual investigación trabaja en principio con los siete indicadores presentados en la Tabla 4: IRE, IRP, IRC, IREF, IRRH, IRL e ICD; los cuales abarcan las fundamentales áreas de conocimiento en la gestión de proyectos.

El cálculo automático de los indicadores y la evaluación de expertos, es un trabajo externo a la presente investigación (Lugo, 2012). Dichos datos deben someterse a un proceso de selección, limpieza y transformación, con el objetivo de obtener una base de casos limpia y representativa. El pre-procesamiento de datos es otro trabajo precedente a esta investigación y se encarga de prepararlos y persistirlos en una base de datos de proyectos terminados (Torres et al., 2013). Para ello, el científico de datos implementa las funciones necesarias de la limpieza de datos y selección de casos. En dicho proceso se eliminan los datos duplicados y los valores ausentes desechando los registros correspondientes.

### **2.2 Componentes del sistema**

En la presente investigación se desarrolla la biblioteca AnalysisPro.SC.PMC para el análisis de proyectos mediante técnicas de *soft computing*, aplicada en el control de la ejecución de proyectos. Dicha biblioteca permite la construcción, aprendizaje automático y selección de sistemas de inferencia borrosos para la evaluación de proyectos, integrada en la base de conocimiento de la organización. La biblioteca se basa en *software* libre y contiene funciones programadas en lenguaje PL/PGSQL y PL/R integrables en PostgreSQL. Como dependencias, utiliza el paquete FRBS (Riza et al., 2015) y funciones nativas de R (R Core Team, 2015).

La estrategia que se sigue para la adaptación y el aprendizaje automático del sistema consiste en generar la base inicial de reglas borrosas y optimizarlas realizando el ajuste de los parámetros de las funciones de pertenencia de los conjuntos borrosos. Con ello se persigue obtener una mejora continua en el control de la ejecución de los proyectos y elevar la calidad de las herramientas de gestión, como apoyo a la toma de decisiones.

El sistema propuesto dispone de tres componentes fundamentales (Figura 3): obtener de la base de datos de proyectos terminados los casos necesarios para el aprendizaje, aplicar diferentes técnicas de *soft computing* y emplear métricas estadísticas de rigor para seleccionar el sistema que mejor evalúe la ejecución de los proyectos en la organización. Este mecanismo se realiza cuando la alta gerencia determine la necesidad de actualizar dicho sistema de evaluación según los cambios producidos por la mejora continua en los estilos de gestión de la organización.

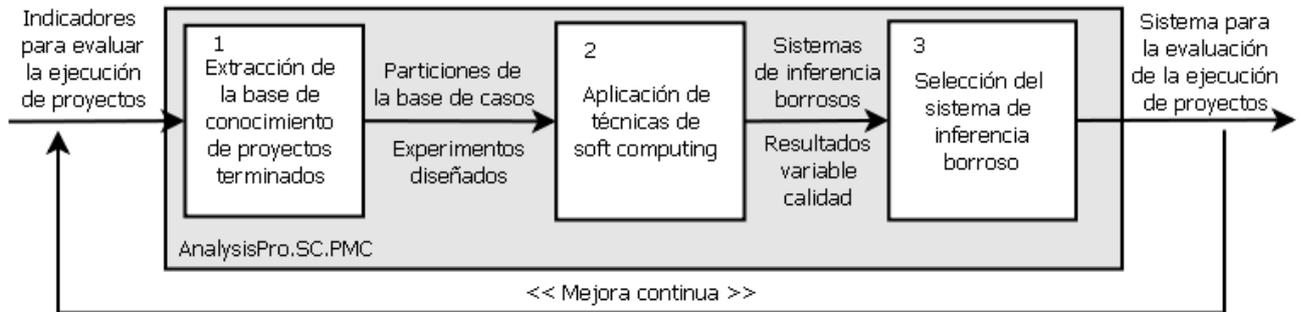


Figura 3. Sistema basado en técnicas de *soft computing* para la evaluación de proyectos.

### Descripción del sistema

**Componente 1:** Las entradas de este bloque se corresponden con los indicadores de gestión de proyectos calculados automáticamente y almacenados en la base de datos de la organización, según su nivel de madurez actual. Se propone por defecto la utilización de los indicadores abordados en la Tabla 4: IRE, IRP, IRC, IREF, IRRH, IRL e ICD. Se consulta la base de conocimiento de proyectos terminados, adquiriendo los indicadores pre-calculados y la evaluación de los expertos a los proyectos, que servirán de casos de entrenamiento. Dicha fuente de información sirve como base para la generación y optimización de las reglas borrosas mediante aprendizaje supervisado. La base de proyectos preclasificados por expertos se divide de forma aleatoria en diferentes particiones para obtener los conjuntos de entrenamiento y de validación. Estas particiones se utilizan posteriormente para realizar la validación cruzada entre los experimentos de cada técnica. Funciones de la biblioteca diseñan los experimentos correspondientes para cada técnica integrada con su respectiva partición de entrenamiento.

**Componente 2:** Se ejecutan los experimentos para las diferentes técnicas de *soft computing* (se explican en la sección 2.3). Se persisten en la base de datos los sistemas que son generados por las distintas técnicas, lo cual incluye la estructura de las reglas y los parámetros de sus conjuntos borrosos. También se almacena en dicha tabla de resultados: el nombre de la técnica aplicada en el experimento, los parámetros iniciales que se utilizaron para el aprendizaje, la cantidad de reglas generadas, así como los resultados de las variables para medir la calidad de la evaluación. Estas métricas son: MSE (*Mean Square Error* - Error cuadrático medio), RMSE (*Root Mean Square Error* - Raíz del error cuadrático medio) y SMAPE (*Symmetric Mean Absolute Percentage Error* - Error porcentual de la media absoluta simétrica), ver ecuaciones 1, 2 y 3 respectivamente. Además, al realizar la predicción se obtienen medidas de evaluación que se persisten en la base de datos; estas se corresponden con las clasificaciones correctas, los falsos positivos y los falsos negativos.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_{real} - Y_{pred})^2 \quad (1)$$

$$RMSE = \sqrt{MSE} \quad (2)$$

$$SMAPE = 100 * \frac{1}{n} \sum_{i=1}^n \frac{|Y_{real} - Y_{pred}|}{(Y_{real} + Y_{pred})/2} \quad (3)$$

Donde  $n$  es la cantidad de casos,  $Y_{real}$  es el valor de salida esperada y  $Y_{pred}$  es la salida calculada por el sistema.

**Componente 3:** Mediante funcionalidades implementadas en la biblioteca se ejecutan pruebas estadísticas de rigor y se selecciona el sistema de inferencia borroso mejor ajustado. Para ello se comparan los resultados de los algoritmos en cuanto a la calidad de las evaluaciones, haciendo uso de las métricas calculadas en el componente 2, las cuales permiten obtener la precisión de los pronósticos realizados. A estas medidas se les aplican los test de Friedman y Wilcoxon para detectar si existen diferencias significativas entre los resultados de los algoritmos. La salida final de

este componente consiste en el sistema de inferencia borroso que más se ajusta al entorno de la organización (mejor resultado de validación).

Posteriormente es posible pasar como entrada nuevos casos de entrenamiento, con el objetivo de lograr la mejora continua y adaptar el sistema de evaluación a los cambiantes entornos de gestión de proyectos. Entiéndase por entornos cambiantes cuando la organización modifica los tipos y cantidades de indicadores de gestión que utiliza. Para garantizar la mejora continua del sistema propuesto y que este incida positivamente en la adaptación de la organización a los nuevos estilos de gestión, se requiere que cada año se actualice la base de conocimiento de proyectos terminados. Dicha actualización se puede realizar cuando los directivos consideren necesario según la madurez actual de la organización.

Una vez presentados los componentes del sistema, se prosigue en la siguiente sección con los elementos fundamentales de cada técnica de *soft computing* empleada.

### **2.3 Técnicas de adaptación y aprendizaje de reglas borrosas**

Para la aplicación de las técnicas de *soft computing* se utiliza la funcionalidad *frbs.learn* del paquete FRBS. En esta sección se presentan los parámetros de las técnicas de *soft computing* aplicadas para generar y ajustar los sistemas de inferencia borrosos para la evaluación de la ejecución de proyectos. En las siguientes subsecciones se exponen los elementos de funcionamiento de las técnicas: WM, ANFIS, HYFIS, FIR.DM, FS.HGD, GFS.THRIFT y GFS.LT.RS; las cuales son basadas en partición espacial, redes neuronales, gradiente descendente y algoritmos genéticos.

Para aplicar cada técnica de *soft computing* se le asigna el correspondiente valor a los parámetros de control y se invoca de la siguiente forma la funcionalidad *frbs.learn*:

```
object <- frbs.learn(data.train, range.data, method.type, control.param)
```

En *object* se guarda el objeto que contiene el sistema de inferencia borroso, a partir del cual se realiza la predicción y se persisten los resultados obtenidos en la base de datos.

A continuación se describen los parámetros que se utilizan para la aplicación de las técnicas:

- Tipo de método (*method.type*): este parámetro determina el algoritmo de aprendizaje que se utilizará. Los siguientes métodos son ejecutados: ANFIS, HYFIS, WM, FIR.DM, FS.HGD, GFS.LT.RS y GFS.THRIFT. El método por defecto es WM.
- Datos de entrenamiento (*data.train*): matriz de  $m$  por  $n$  con los datos para el proceso de aprendizaje, donde  $m$  es el número de casos de entrenamiento y  $n$  es el número de variables ( $n-1$  variables de entrada o indicadores de gestión de proyectos y la última columna corresponde a la variable de salida). Estos datos son normalizados internamente a valores entre 0 y 1 por el algoritmo de aprendizaje en caso de que la técnica a aplicar lo requiera.
- Rango de los datos (*range.data*): matriz de 2 por  $n$  que contiene los posibles valores que pueden tomar los datos, donde  $n$  es el número de variables. La primera y segunda filas son el valor mínimo y máximo respectivamente. En caso de que el usuario no se le asigne valor a este parámetro, la funcionalidad *frbs.learn* internamente toma los valores mínimos y máximos de cada variable según los datos de entrenamiento.
- Cantidad de etiquetas lingüísticas (*num.labels*): matriz de 1 por  $n$ , cuyos elementos representan el número de etiquetas o términos lingüísticos para cada una de las  $n$  variables. Por defecto toma valores de 3 o 7 etiquetas por cada variable en dependencia de la técnica a aplicar.
- Máximo número de iteraciones (*max.iter*): este valor es utilizado como condición de parada. Por defecto toma el valor de 10.
- Tamaño del paso (*step.size*): se utiliza para realizar las modificaciones del gradiente descendente, toma valores entre 0 y 1. Por defecto toma el valor de 0,01.
- Tipo de T-Norma (*type.tnorm*): representa la forma en que se van a realizar las operaciones de conjunción y contribución sobre dos parámetros  $x_1$  y  $x_2$  (Baczynski et al., 2013). Por defecto es MIN, puede tomar los siguientes valores:

1. MIN: es el tipo por defecto, se calcula como:  $\min(x_1, x_2)$ .
  2. HAMACHER: significa el producto de Hamacher, se calcula como:  $(x_1 * x_2) / (x_1 + x_2 - x_1 * x_2)$ .
  3. YAGER: representa la clase de Yager, se calcula como:  $1 - \min(1, ((1-x_1) + (1-x_2)))$ .
  4. PRODUCT: representa el producto, se calcula como:  $(x_1 * x_2)$ .
  5. BOUNDED: representa el producto acotado, se calcula como:  $\max(0, x_1 + x_2 - 1)$ .
- Tipo de S-Norma (*type.snorm*): forma en que se van a realizar las operaciones de disyunción y global sobre dos parámetros  $x_1$  y  $x_2$  (Baczynski et al., 2013). Por defecto es MAX, puede tomar los siguientes valores:
    1. MAX: es el tipo por defecto, se calcula como:  $\max(x_1, x_2)$ .
    2. HAMACHER: significa la suma de Hamacher, se calcula como:  $(x_1 + x_2 - 2x_1 * x_2) / 1 - x_1 * x_2$ .
    3. YAGER: representa la clase de Yager, se calcula como:  $\min(1, (x_1 + x_2))$ .
    4. SUM: representa la suma, se calcula como:  $(x_1 + x_2 - x_1 * x_2)$ .
    5. BOUNDED: representa la suma acotada, se calcula como:  $\min(1, x_1 + x_2)$ .
  - Tipo de función de implicación (*type.implication.func*): considerando una regla de la forma  $a \rightarrow b$  (Baczynski et al., 2013), puede tomar los siguientes valores según la comparación entre  $a$  y  $b$  (por defecto es ZADEH):
    1. DIENES\_RESHER se calcula como:  $(b > 1 - a ? b : 1 - a)$ .
    2. LUKASIEWICZ se calcula como:  $(b < a ? 1 - a + b : 1)$ .
    3. ZADEH se calcula como:  $(a < 0.5 || 1 - a > b ? 1 - a : (a < b ? a : b))$ .
    4. GOGUEN se calcula como:  $(a < b ? 1 : b/a)$ .
    5. GODEL se calcula como:  $(a <= b ? 1 : b)$ .
    6. SHARP se calcula como:  $(a <= b ? 1 : 0)$ .
    7. MIZUMOTO se calcula como:  $(1 - a + a * b)$ .
    8. DUBOIS\_PRADE se calcula como:  $(b == 0 ? 1 - a : (a == 1 ? b : 1))$ .
    9. MIN se calcula como:  $(a < b ? a : b)$ .

Donde el signo ? corresponde al *entonces* y los dos puntos corresponden al *sino*, para darle el valor a la salida de la función de implicación en dependencia de la comparación que se realiza entre  $a$  y  $b$ .
  - Tipo de defusificación (*type.defuz*): se utiliza para convertir los grados de pertenencia de la variable consecuente a salida numérica; el método por defecto es WAM. Puede tomar los siguientes valores:
    1. WAM: método de la media ponderada.
    2. FIRST.MAX: significa la primera máxima.
    3. LAST.MAX: significa la última máxima.
    4. MEAN.MAX: significa la media máxima.
    5. COG: centro de gravedad modificado.
  - Valor heurístico (*alpha.heuristic*): representa el valor heurístico, es un número real positivo con valor por defecto igual a 1.
  - Tipo de función de pertenencia (*type.mf*): hay cinco formas de funciones de pertenencia implementadas. Están representadas por una matriz de dimensión 5 por  $n$ , donde  $n$  es una multiplicación del número de términos lingüísticos y el número de variables de entrada. Las filas de la matriz representan: la primera fila es el tipo de función de pertenencia, la segunda hasta la quinta fila indican los puntos de esquina para construir las funciones. El valor predeterminado es GAUSSIAN. Puede tomar los siguientes valores:
    1. TRIANGLE: se refiere a la forma triangular, tiene tres parámetros ( $a, b, c$ ), donde  $b$  es el punto de centro del triángulo,  $a$  y  $c$  son los puntos izquierdo y derecho, respectivamente.
    2. TRAPEZOID: se refiere a la forma trapezoidal, tiene cuatro parámetros ( $a, b, c, d$ ).
    3. GAUSSIAN: se refiere a la forma gaussiana, tiene dos parámetros (media y varianza).
    4. SIGMOID: se refiere a la forma sigmoide, tiene dos parámetros ( $\gamma$  y  $c$ ) para la representación de la pendiente de la función y la distancia desde el origen, respectivamente.

5. BELL: se refiere a la campana generalizada, tiene tres parámetros (a, b, c).

- Tamaño de la población (*popu.size*): cantidad de cromosomas que se toman inicialmente en cada generación, el valor predeterminado es 10.
- Cantidad máxima de generaciones (*max.gen*): un entero positivo para determinar el número máximo de generaciones del algoritmo genético, el valor predeterminado es 10.
- Porcentaje de mutación (*persen\_mutant*): representa la probabilidad de mutación, el valor predeterminado es 0.3.
- Porcentaje de cruzamiento (*persen\_cross*): representa la probabilidad de cruce, el valor predeterminado es 0.6.
- Modo de ajuste (*mode.tuning*): tipo de ajuste lateral que puede tomar los valores: "LOCAL" o "GLOBAL". El valor por defecto es "GLOBAL".
- Selección de reglas (*rule.selection*): un valor booleano que representa si realiza selección de reglas o no. Puede tomar los valores: "TRUE" o "FALSE", por defecto es "TRUE".

Una vez descritos los parámetros que se utilizan en las diferentes funcionalidades, se procede con la exposición de los elementos principales de las técnicas de *soft computing* aplicadas.

### **2.3.1 Sistema borroso basado en el método de Wang y Mendel (WM)**

En (Wang and Mendel, 1992) se propone un algoritmo para generar el conjunto de reglas borrosas candidatas y refinarlas, basado en la técnica de partición de espacio. Requiere de la existencia de un conjunto de datos de entrada-salida que reflejen el comportamiento del problema a resolver y de una definición de las particiones borrosas de entrada y salida. La estructura de regla que maneja es la habitual de los sistemas de inferencia borrosos de tipo Mamdani:

*Si  $X_1$  es  $A_1$  y...y  $X_n$  es  $A_n$  entonces  $Y$  es  $B$*

Donde  $X_i$ ,  $Y$  son variables lingüísticas de entrada y salida respectivamente;  $A_i$  y  $B$  son etiquetas lingüísticas asociadas a los conjuntos borrosos antecedentes y consecuente.

De este modo, la base de reglas generadas a partir del proceso de Wang y Mendel presenta un número de reglas acotado superiormente por el producto del número de términos lingüísticos asociados a las variables de entrada del sistema.

El proceso de aprendizaje consta de las siguientes cinco etapas:

Paso 1: dividir igualmente los espacios de entrada y salida de los datos numéricos dados en regiones difusas referentes a los intervalos de cada término lingüístico. Por ejemplo, el término lingüístico "alto" tiene la región difusa [1; 3] y función de pertenencia triangular con los puntos  $a = 1$ ,  $b = 2$  y  $c = 3$  donde  $b$  es el medio de triángulo y su grado de pertenencia es 1.

Paso 2: generar reglas borrosas según los casos de entrenamiento proporcionados. Se calculan los grados de pertenencia a los conjuntos borrosos para cada caso de entrenamiento y se determina el término lingüístico que tiene el grado máximo en cada variable. Se repite con cada caso de entrenamiento para construir reglas borrosas que cubran todos los casos.

Paso 3: asignar un grado de importancia a cada regla generada con el propósito de resolver los conflictos entre las reglas generadas. Los grados o pesos de las reglas se determinan mediante la suma del grado de pertenencia de los antecedentes y consecuente. En este caso se puede utilizar el operador de agregación (T-Norma) Producto.

Paso 4: crear una base de reglas combinada a partir de las reglas generadas y las normas lingüísticas proporcionadas por los expertos.

Paso 5: obtener una base de regla final después de eliminar las reglas redundantes con los grados más bajos. En el caso en que todas las reglas existentes en el conjunto preliminar que presenten la misma combinación de antecedentes tengan asociado el mismo consecuente, dicha regla es automáticamente insertada (una sola vez) en la base de reglas final. En cambio, en el caso en que

existan reglas conflictivas o redundantes, es decir, reglas con el mismo antecedente y distintos valores en el consecuente, la regla insertada será aquella que presente mayor grado de importancia. Para la aplicación de WM con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, tipo de función de pertenencia, T-Norma, S-Norma, tipo de defusificación y tipo de función de implicación. Los datos de entrenamiento son normalizados internamente a valores entre 0 y 1.

### 2.3.2 Sistema borroso basado en redes adaptativas (ANFIS)

ANFIS representa un sistema neuro-borroso híbrido de tipo Sugeno (Jang, 1993). Los sistemas neuro-borrosos (SNB) utilizan una combinación de los paradigmas de Lógica Borrosa y RNA. Por un lado, la lógica borrosa permite trabajar con información incierta y expresar el conocimiento de un experto humano mediante reglas *If-then* simples, descritas en lenguaje natural. Mientras que las RNA persiguen simular la capacidad de razonamiento humano a través de su estructura y organización; aprovechando su capacidad de aprendizaje y habilidad de generalización. La unión de ambas técnicas permite ganar en robustez, eficiencia, adaptabilidad y equilibrio entre poder de predicción e interpretación (Nauck, 2005) (Nauck and Kruse, 1998, Nauck, 2000, Zadeh, 1994). Los aspectos que caracterizan a las RNA y que permiten distinguir los diversos modelos son: la topología de la red, el modelo de las neuronas y el algoritmo de aprendizaje.

ANFIS utiliza una arquitectura de red multicapa orientada a nodos en la cual todos los parámetros de las funciones de pertenencia residen dentro de los nodos. La representación orientada a nodos de los sistemas borrosos facilita desde su implementación la interpretación del sistema al no tener en cuenta pesos en la conexiones, cuyo significado semántico con frecuencia es difícil de explicar. En la Figura 4 se muestra una síntesis de la topología de la red ANFIS ajustada para la evaluación de proyectos. Donde  $n$  es la cantidad de entradas a la red y se corresponde con los indicadores de gestión de proyectos ( $Ind_1 \dots Ind_n$ ). Las neuronas de la capa 1 representan los conjuntos borrosos, en este caso se adoptan tres para cada entrada (bien, regular y mal), pero pudiera ser otra cantidad con otras etiquetas lingüísticas.

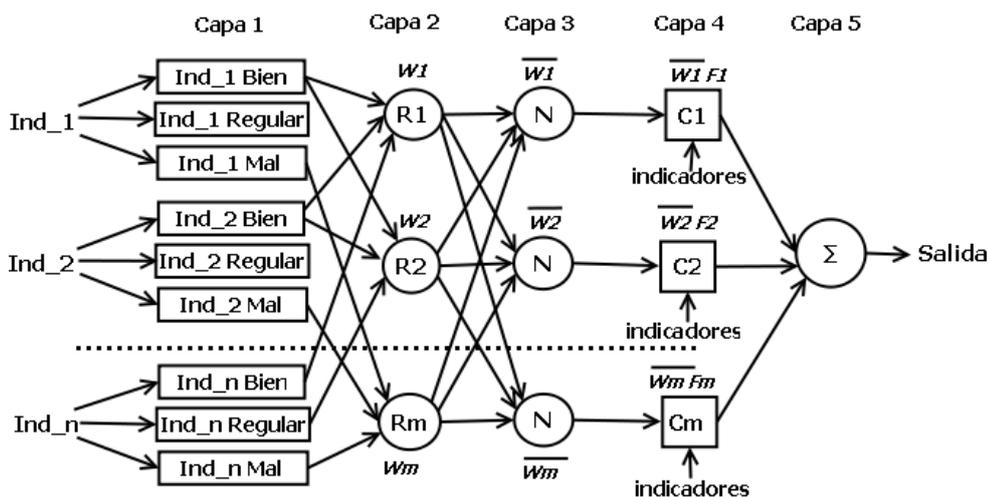


Figura 4. Arquitectura resumida del ANFIS diseñado para la evaluación de proyectos (adaptado de (Jang, 1993)).

Cada nodo de la capa 1 recibe los valores numéricos de cada indicador y calcula el grado de pertenencia del valor recibido al conjunto borroso que representa. Las neuronas de las capas 2, 3 y 4 corresponden a las  $m$  reglas borrosas ( $R1 \dots Rm$ ), ellas calculan la activación, la normalizan ( $N$ ) y computan la salida de cada regla ( $C1 \dots Cm$ ). El nodo de la capa 5 ( $\Sigma$ ) calcula la salida global del sistema para dar la evaluación del proyecto.

Los nodos en la primera y cuarta capa están dibujados como bloques cuadrados, lo cual significa que son modificables; es decir, que tienen parámetros que pueden variar. Los nodos en el resto de las capas están dibujados como bloques circulares, lo cual significa que son fijos y no dependen de ningún parámetro variable. Las capas de la red no están completamente conectadas sino que las conexiones tienen lugar a partir de la presencia de las reglas borrosas y las relaciones que estas representan dentro del sistema.

El modelo de las neuronas en ANFIS queda definido de manera diferente para cada capa de la red. A continuación se especifica su funcionamiento (entradas, salida y cálculo realizado):

Capa 1. (Fusificación): cada nodo se conecta a exactamente una unidad de entrada y calcula el grado de pertenencia del valor recibido al conjunto borroso correspondiente (Ecuación 4). Las entradas a esta capa son valores numéricos de las variables de entrada y las salidas son grados de pertenencia a los conjuntos borrosos. Las funciones de pertenencia asociadas a estos conjuntos borrosos deben ser continuas y derivables a tramos, para poder aplicarle el gradiente descendiente durante el algoritmo de aprendizaje.

$$O_i^1 = \mu_{A_i}(x) \quad (4)$$

Donde  $x$  es la entrada al nodo  $i$  y  $A_i$  es la etiqueta lingüística asociada con esta función de nodo.  $O_i^1$  es el grado de pertenencia ( $\mu$ ) a un conjunto borroso  $A$  ( $A = A_1, A_2$ ), el cual especifica el grado en el cual una entrada  $x$  dada satisface el cuantificador  $A_i$ .

Capa 2. (Conjunción): cada nodo en esta capa representa una regla, que se conecta a sus correspondientes antecedentes de la capa 1 y obtiene como entrada los grados de pertenencia. Se calcula el grado de activación de la regla asociada, aplicando un operador T-Norma para modelar la operación lógica "and" (Ecuación 5).

$$w_i = T(\mu_{A_i}(x), \mu_{B_i}(y)), i = 1, 2 \quad (5)$$

Donde  $w_i$  representa el grado de verosimilitud o la intensidad de disparo que tiene la regla.

Capa 3. (Normalización): cada nodo calcula el grado de activación normalizado de la regla  $i$ -ésima con respecto a la suma de las intensidades de disparo de todas las reglas (Ecuación 6).

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \quad (6)$$

Capa 4. (Defusificación): cada nodo calcula su salida según el producto entre el grado de activación normalizado por la salida individual de cada regla (Ecuación 7).

$$\bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (7)$$

Donde  $w_i$  normalizada es la salida de la capa 3 y  $\{p_i, q_i, r_i\}$  es el conjunto de parámetros consecuentes.

Capa 5. (Agregación): el único nodo de esta capa calcula la salida global del sistema como la suma pesada de todas las señales individuales ( $z_i$ ) que entran a él (Ecuación 8).

$$\text{Salida Global} = \frac{\sum_i w_i z_i}{\sum_i w_i} \quad (8)$$

El tipo de RNA que utiliza ANFIS es *feedforward* (encadenamiento hacia adelante). Las funciones de pertenencia de los antecedentes requieren ser derivables, notando muy buenos resultados con las de tipo campanas (Kelemen et al., 2002).

El algoritmo de aprendizaje de ANFIS consta de los siguientes pasos (Jang, 1993):

Paso 1: los parámetros de las funciones de pertenencia de los conjuntos borrosos son inicializados y es presentado al sistema un vector de entrada-salida. Se calculan las salidas para cada capa de la red propagándolas hacia adelante hasta los consecuentes, según el modelo de las neuronas anteriormente descrito.

Paso 2: se calcula el error como la diferencia entre la salida computada y la salida esperada. Una de las medidas más empleadas para calcular el error de entrenamiento es la suma del error cuadrado para cada consecuente.

Paso 3: ajustar los parámetros de las funciones de pertenencia en los consecuentes siguiendo la estrategia de los mínimos cuadrados. En el paso hacia atrás, se propagan las señales de error y el vector gradiente es acumulado para cada caso de entrenamiento. Los parámetros en la capa 1 son actualizados por el método descendente.

Paso 4: volver al paso 2 si el error es mayor que la tolerancia, si no finalizar.

Jang reporta que este enfoque híbrido basado en dos etapas mejora los resultados obtenidos y converge más rápidamente que la variante de aplicar solamente el gradiente descendente (método *backpropagation*) para optimizar todos los parámetros (Jang et al., 1997).

Para la aplicación de ANFIS con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, máximo número de iteraciones, tamaño del paso, T-Norma, S-Norma y tipo de función de implicación. Esta funcionalidad normaliza internamente los datos de entrenamiento a valores entre 0 y 1.

### 2.3.3 Sistema neuro-borroso híbrido (HYFIS)

HYFIS es un modelo neuro-borroso de tipo híbrido utilizado para construir y optimizar sistemas borrosos (Kim and Kasabov, 1999). Representa una red neuronal de cinco capas que es funcionalmente equivalente a un sistema de inferencia borroso con reglas de tipo Mamdani (Arango, 2012). Permite adaptar las funciones de pertenencia de los conjuntos borrosos y las reglas de acuerdo a los casos de entrenamiento. En la Figura 5 se muestra la topología de la red HYFIS ajustada para la evaluación de proyectos. Los nodos de las capas 1 y 5 representan los estados de entrada y salida respectivamente. Los nodos de las capas ocultas corresponden a las funciones de pertenencia y reglas borrosas.

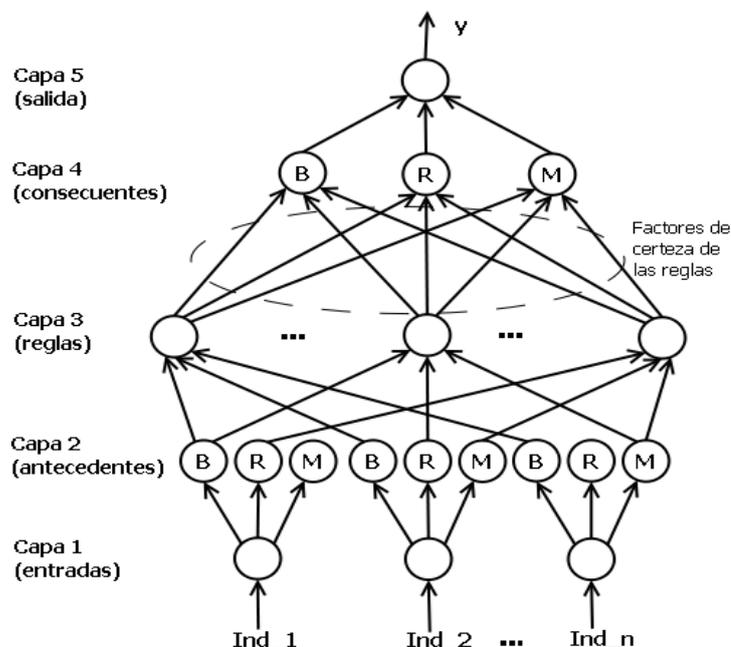


Figura 5. Arquitectura resumida del HYFIS diseñado para la evaluación de proyectos (adaptado de (Kim and Kasabov, 1999)).

Donde  $Ind_1, \dots, Ind_n$  son los indicadores de gestión de proyectos y entradas de la red;  $B$ ,  $R$  y  $M$  corresponden a los conjuntos borrosos, en este caso se adoptan tres para cada variable antecedente o consecuente (bien, regular y mal);  $y$  es la salida de la red.

El modelo de las neuronas de HYFIS se especifica a continuación:

Capa 1. (Entradas): cada nodo representa una variable lingüística de entrada correspondiente a un indicador de gestión de proyectos. Recibe el valor numérico del indicador asociado y lo transmite hacia la siguiente capa de la red neuronal.

Capa 2. (Fusificación): cada nodo se conecta a exactamente una unidad de entrada y calcula el grado de pertenencia ( $y_i$ ) del valor recibido ( $x_i$ ) al conjunto borroso que representa. Las funciones de pertenencia utilizadas son de tipo gaussiana y se modelan mediante una función exponencial, con los parámetros  $c$  y  $\sigma$ , que representan el centro y el ancho respectivamente (Ecuación 9). La salida de cada nodo de esta capa ( $y_j^{(2)}$ ) constituye el grado de pertenencia de  $x$  al conjunto borroso.

$$y_i^{(2)} = \exp\left[-\frac{(x_i - c)^2}{\sigma^2}\right] \quad (9)$$

Capa 3. (Conjunción): cada nodo representa el antecedente conjugado de una regla del sistema. Obtiene como entradas los grados de pertenencia de los antecedentes de la regla y su salida se calcula según la (Ecuación 10) aplicando el operador T-Norma MIN (podría tomar otro tipo).

$$y_r^{(3)} = \min_{j \in I_r} (y_j^{(2)}) \quad (10)$$

Donde  $y_r^{(3)}$  es la salida del nodo  $r$  en la capa 3;  $I_r$  representa los índices de los nodos de la capa 2 que conforman los antecedentes de la regla  $r$  y  $y_j^{(2)}$  es la salida del nodo  $j$  en la capa 2.

Capa 4. (Defusificación): representa los factores de certeza asociados a los consecuentes de las reglas del sistema neuro-borroso. Los nodos de las capas 3 y 4 están completamente conectados por conexiones cuyos pesos son adaptativos. Dichos pesos  $w_{rj}$ , entre los  $r$  nodos de la capa 4 y los  $j$  nodos de la capa 3, representan los factores de certeza de las reglas, los cuales infieren en los valores borrosos de salida. Cada nodo calcula su salida según la (Ecuación 11) aplicando el operador S-Norma Máximo (podría utilizarse otro tipo de S-Norma).

$$y_r^{(4)} = \max_{j \in I_r} (w_{rj} * y_j^{(3)}) \quad (11)$$

Donde  $I_r$  representa los índices de los nodos de la capa 3 que están unidos al nodo  $r$  de la capa 4;  $y_j^{(3)}$  es la salida del nodo  $j$  en la capa 3.

Capa 5. (Agregación): representa la variable de salida del sistema, cuyo dominio es cubierto por varios conjuntos borrosos. El único nodo de esta capa realiza un proceso de concertación de la salida al calcular según la Ecuación 12 el centro del área del conjunto borroso (podría aplicar otro tipo de operador para defusificar).

$$y_l^{(5)} = \frac{\sum_{k \in I_k} y_k^{(4)} \sigma_{lk} c_{lk}}{\sum_{k \in I_k} y_k^{(4)} \sigma_{lk}} \quad (12)$$

Donde  $I_k$  representa los índices de los nodos de la capa 4 que están unidos al nodo de la capa 5;  $\sigma_{lk}$  y  $c_{lk}$  son el centro y ancho respectivamente de la función de pertenencia de la variable lingüística de salida representada por  $l$  en la capa 4;  $y_k^{(4)}$  es la salida del nodo  $k$  en la capa 4.

Los pesos de los enlaces entre los nodos de las capas 4 y 5 tienen valor de 1. Los únicos pesos que tienen modificaciones durante el aprendizaje son los comprendidos entre las capas 3 y 4, donde residen los factores de certeza de las reglas borrosas.

El algoritmo de aprendizaje de HYFIS consta de dos fases, en la primera se estructura la base de reglas mediante el módulo de adquisición de conocimientos. En la segunda fase se ajustan los parámetros de las funciones de pertenencia para lograr un nivel de rendimiento adecuado. Una ventaja de este enfoque es la facilidad para modificar la base de reglas borrosas a medida que se disponga de nuevos datos. Cuando un nuevo caso de entrenamiento está disponible, se crea una regla para este y se añade a la base de reglas borrosas. El algoritmo de aprendizaje incremental que utiliza HYFIS consta de los siguientes pasos (Kim and Kasabov, 1999):

Paso 0: inicializar el esquema de aprendizaje compuesto por los datos de entrenamiento y la cantidad de etiquetas lingüísticas para cada variable de entrada y salida. Se seleccionan al azar entre [-1,+1] los pesos iniciales de las conexiones entre las capas 3 y 4.

Paso 1: extraer el conjunto de reglas borrosas a partir de los datos de entrada-salida. Se fija el número de funciones de pertenencia asociadas a cada variable según la cantidad de etiquetas lingüísticas. Los valores iniciales de los parámetros se establecen de tal manera que los centros de las funciones de pertenencia estén igualmente espaciados a lo largo del rango de cada variable y cubran todo el espacio. Generar una regla para cada caso de entrenamiento.

Paso 2: modificar las reglas y añadir nuevas si es necesario. Asignarle peso a cada regla y eliminar las redundantes (con antecedentes iguales) dejando la que tenga el peso mayor. La estrategia a seguir para calcular el peso de la regla es dada por la multiplicación de los factores de certeza de sus antecedentes. Para cada regla borrosa se comprueba que esté contenida en la estructura de la red, si es así, entonces se actualiza; si no, se añade a la red.

Paso 3: segunda fase de aprendizaje para ajustar los parámetros de las funciones de pertenencia. Este paso se realiza con el conjunto de datos completo en función de la frecuencia de la corriente de datos entrante. Se utiliza un algoritmo de aprendizaje de descenso de gradiente para minimizar la función de error. Se ajustan los pesos de las conexiones entre las capas 3 y 4, así como los parámetros de las funciones de pertenencia.

Paso 4: repetir pasos del 1 al 3.

Para la aplicación de HYFIS con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, máximo número de iteraciones, tamaño del paso, T-Norma, S-Norma, tipo de defusificación y tipo de función de implicación. Esta funcionalidad normaliza internamente los datos de entrenamiento a valores entre 0 y 1.

### **2.3.4 Sistema borroso basado en método de descenso (FIR.DM)**

FIR.DM se basa en un sistema de inferencia borroso tipo Sugeno Grado Cero que actualiza sus parámetros a partir de un entrenamiento supervisado basado en el método de gradiente descendente (Nomura et al., 1992). Utiliza como información inicial un vector de datos que consta de dos partes: vector de entrada y su respectiva salida, los cuales se pueden describir mediante reglas borrosas como la mostrada en la Ecuación 13.

$$\text{Si } x_i \text{ es } A_{11} \wedge \dots \wedge x_m \text{ es } A_{1m} \text{ entonces } y \text{ es } w_i \quad (13)$$

Donde  $x_1 \dots x_m$  son las variables de entrada al sistema con valores en el intervalo [-1,1];  $A_{11} \dots A_{1m}$  son los conjuntos borrosos de la parte antecedente;  $y$  es la salida del sistema y  $w_i$  representa el peso o aporte de la regla  $i$  a la solución del sistema.

Las funciones de pertenencia utilizadas tienen forma de triángulo isósceles (Figura 6) y se definen mediante dos parámetros:  $b_{ij}$  representa el ancho de la base y  $a_i$  el centro del triángulo.

La metodología de FIR.DM se divide en un módulo de reconocimiento y de entrenamiento, los cuales se describen a continuación (Nomura et al., 1992).

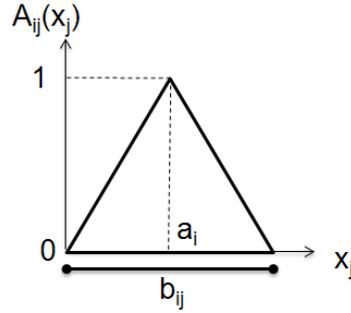


Figura 6. Función de pertenencia en forma de triángulo (Nomura et al., 1992).

El proceso de reconocimiento consta de los siguientes pasos:

Paso 1: Fusificar un valor  $x_j$  de entrada (Ecuación 14). Los grados de pertenencia  $A_{ij}(x_j)$  toman valores en el intervalo  $[-1, 1]$ .

$$A_{ij}(x_j) = 1 - \frac{2 * |x_j - a_{ij}|}{b_{ij}} \quad (14)$$

Donde  $b_{ij}$  representa el ancho de la base y  $a_i$  el centro del triángulo según el conjunto borroso  $A_{ij}$ .

Paso 2: Calcular la intensidad de disparo ( $\mu_i$ ) de cada regla mediante el producto de los grados de pertenencia ( $A_{ij}(x_j)$ ) de los antecedentes que componen cada una de las reglas (Ecuación 15).

$$\mu_i = A_{i1}(x_1) * A_{i2}(x_2) * \dots * A_{im}(x_m) \quad (15)$$

Paso 3: Defusificar la salida ( $y$ ) mediante el promedio de los pesos  $w_i$  de cada regla y su intensidad de disparo  $\mu_i$  (Ecuación 16).

$$y = \frac{\sum_{i=1}^n \mu_i w_i}{\sum_{i=1}^n \mu_i} \quad (16)$$

El proceso de entrenamiento consiste en la optimización de los parámetros del sistema borroso de manera iterativa a partir de los valores calculados por el sistema ( $y$ ) y los deseados ( $y_r$ ) como resultado de una entrada ( $u$ ) al sistema. Las condiciones iniciales que necesita esta metodología son: los conjuntos borrosos espaciados linealmente, las bases entre conjuntos adyacentes solapadas entre sí y los pesos iniciales de las reglas borrosas en 0.5 (López, 2008). El algoritmo necesita un vector de dimensión  $p$  definido como  $Z = (z_1, z_2, \dots, z_p)$  que contiene los parámetros ajustables. Dicho vector en el método de descenso minimiza la función objetivo de error  $E$  (Ecuación 17) hasta que esta sea menor que un valor de aceptación  $\varepsilon$ .

$$E = \frac{1}{2} (y - y_r)^2 \quad (17)$$

La optimización de los parámetros para la adaptación del sistema borroso de un estado actual  $t$  a un estado siguiente  $t+1$  es dada mediante las ecuaciones 18, 19 y 20. Estas son obtenidas despejando parcialmente la Ecuación 17 con respecto a  $a_{ij}$ ,  $b_{ij}$  y  $w_i$  para ajustar los centros, bases y pesos respectivamente según el error calculado.

$$a_{ij}(t+1) = a_{ij}(t) - K_a * \frac{\mu_i}{\sum_{i=1}^n \mu_i} (y - y^r) * (w_i(t) - y) * \text{sgn}(x_j - a_{ij}(t)) \frac{2}{b_{ij}(t) * A_{ij}} \quad (18)$$

$$b_{ij}(t+1) = b_{ij}(t) - K_b * \frac{\mu_i}{\sum_{i=1}^n \mu_i} (y - y^r) * (w_i(t) - y) * \frac{1 - A_{ij}(x_j)}{A_{ij}(x_j)} * \frac{1}{b_{ij}(t)} \quad (19)$$

$$w_i(t+1) = w_i(t) - K_w * \frac{\mu_i}{\sum_{i=1}^n \mu_i} (y - y^r) \quad (20)$$

Donde  $K_a$ ,  $K_b$  y  $K_w$  son constantes que representan el valor de aprendizaje o grado de modificación independiente para cada tipo de parámetro y  $\text{sgn}()$  es la función signo.

El algoritmo de aprendizaje consta de los siguientes pasos (Nomura et al., 1992):

1. Cargar la base de entrenamiento.
2. Calcular la aportación de cada caso de entrenamiento a los pesos de las reglas ( $w_i$ ).
3. Calcular para cada  $w_i$  el promedio de las aportaciones.

4. Aplicar el incremento promedio a cada centro, base y peso ( $a_{ij}$ ,  $b_{ij}$ ,  $w_i$ ).
5. Calcular el resultado de la función de error  $E$  utilizando los nuevos valores de los parámetros.
6. Si  $E > \epsilon$  regresar al paso 1, en caso contrario se llegó a la solución deseada.

Para la aplicación de FIR.DM con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, máximo número de iteraciones, tamaño del paso, T-Norma, S-Norma y tipo de función de implicación. Esta funcionalidad internamente utiliza funciones de pertenencia triángulo y normaliza los datos de entrenamiento a valores entre 0 y 1.

### 2.3.5 Sistema borroso basado en heurística y gradiente descendente (FS.HGD)

FS.HGD implementa un sistema de inferencia borroso que actualiza sus parámetros a partir de un entrenamiento supervisado basado en un método de gradiente descendente y heurística (Ishibuchi et al., 1994). Utiliza el error cuadrático medio. La metodología de FS.HGD se divide en un módulo de reconocimiento y otro de entrenamiento (Chang and Hsing, 1999), los cuales se describen a continuación. Considera reglas ( $R_i$ ) de la forma mostrada en (Ecuación 21).

$$R_i: \text{Si } x_1 \text{ es } A_{i1} \text{ y } x_2 \text{ es } A_{i1} \dots \text{ y } x_n \text{ es } A_{in} \text{ entonces } y \text{ es } b_i \quad (21)$$

Donde  $x_1 \dots x_n$  son las variables de entrada,  $n$  es el número de variables de entrada,  $A_{in}$  son los conjuntos borrosos,  $i=1 \dots N$  (cantidad de reglas),  $y$  es la variable de salida y  $b_i$  es un número real.

El proceso de reconocimiento se realiza para  $k$  iteraciones y consta de los siguientes pasos:

Paso 1: presentar un caso de entrenamiento ( $x_p, y_p$ ) al modelo. Determinar el valor heurístico de  $b_i$  según el promedio pesado de la salida esperada  $y_p$  (en caso de ser la primera iteración del algoritmo) o del error  $e_p$  con respecto al peso de la regla  $w_i(x_p)$  (Ecuación 22).

$$\begin{array}{ll} \text{para } k=1 & \text{para } k \geq 2 \\ b_i = \sum_{p=1}^m (w_i(x_p) * y_p) / \sum_{p=1}^m w_i(x_p) & b_i(k) = \sum_{p=1}^m (w_i(x_p) * e_p(k-1)) / \sum_{p=1}^m w_i(x_p) \end{array} \quad (22)$$

Donde  $i = 1 \dots N$  y el peso de las reglas ( $w_i$ ) se calcula según (Ecuación 23).

$$w_i = A_{i1}(x_1) * A_{i2}(x_2) * \dots * A_{in}(x_n) \quad (23)$$

Donde  $A_{in}(x_n)$  es el grado de pertenencia de la variable de entrada  $x_n$  al conjunto borroso  $A_{in}$ .

El error de la salida para cada caso de entrenamiento  $p$  se calcula según (Ecuación 24).

$$e_p(k-1) = y_p - y_p(k-1) \quad (24)$$

Paso 2: calcular  $y_p(k)$  según (Ecuación 25) con  $b_i = b_i(1) + b_i(2) + \dots + b_i(k)$ ; para todo caso de entrenamiento  $p$ .

$$y_p(k) = \sum_{i=1}^N w_i(x_p) * \sum_{l=1}^k b_i(l) / \sum_{i=1}^N w_i(x_p) \quad (25)$$

Paso 3: actualizar  $e_p(k) = y_p - y_p(k)$  para todo caso de entrenamiento  $p$ .

Paso 4: si el error cuadrático medio (MSE) es menor que la cota de precisión fijada inicialmente ( $\epsilon$ ), detener el aprendizaje; sino continuar en la siguiente iteración ( $k=k+1$ ).

El proceso de entrenamiento se encarga del aprendizaje mediante retro propagación del error y heurística; se realiza para  $k+1$  iteraciones y consta de los siguientes pasos:

Paso 1: presentar un caso de entrenamiento ( $x_p, e_p(k,j)$ ) al modelo; donde  $e_p$  corresponde al error del caso  $p$  para la entrada  $j$  en la iteración  $k$ .

Paso 2: para cada regla calcular el valor heurístico de  $b_{ij}$  (Ecuación 26).

$$b_{ij}(k+1) = \frac{\sum_{p=1}^m (w_i(x_p) * (1 - A_{ij}(x_{jp})) * e_p(k, j))}{\sum_{p=1}^m (w_i(x_p) * (1 - A_{ij}(x_{jp}))^2)} \quad (26)$$

Donde  $i = 1 \dots N$  y  $j = 0 \dots n$ .

Paso 3: calcular la salida estimada  $y_p(k+1)$  para todo caso de entrenamiento  $p$  (Ecuación 27).

$$y = \frac{\sum_{i=1}^N (w_i * y_i)}{\sum_{i=1}^N w_i} = \frac{\sum_{i=1}^N (w_i * (\sum_{j=0}^n (1 - A_{ij}(x_j)) * b_{ij}))}{\sum_{i=1}^N w_i} \quad (27)$$

Donde  $A_{i0}(x_0) = 0$ ;  $i = 1 \dots N$ .

Paso 4: actualizar error para cada caso de entrenamiento  $p$  (Ecuación 28).

$$e_p(k, j+1) = y_p - y_p(k+1) \quad (28)$$

Paso 5: Modificar la parte de los consecuentes  $b_i$  del modelo mediante el gradiente descendente (Ecuación 29).

$$b_i(t+1) = b_i(t) + \beta \left( -\frac{\partial E}{\partial b_1} \right) = b_i(t) - \beta_b * (y_c - y_d) * w_i / \sum_{i=1}^R w_i \quad (29)$$

Donde  $\beta_b$  es el factor de aprendizaje fijado inicialmente.

Paso 6: si el error cuadrático medio (MSE) es menor que la cota de precisión fijada inicialmente ( $\epsilon$ ), detener el aprendizaje; sino continuar en la siguiente iteración.

Para la aplicación de FS.HGD con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, máximo número de iteraciones, tamaño del paso, valor heurístico, T-Norma y tipo de función de implicación. Esta funcionalidad normaliza internamente los datos de entrenamiento a valores entre 0 y 1.

### 2.3.6 Sistema borroso genético basado en la técnica de Thrift (GFS.THRIFT)

En (Thrift, 1991) se propone una técnica para generar reglas y optimizar sistemas de inferencia borrosos, fundamentada en un algoritmo genético (AG) básico. Los AG son algoritmos de búsqueda y optimización de propósito general que utilizan principios inspirados por la genética natural para evolucionar las soluciones de problemas (Golberg, 1989). Permiten explotar la información acumulada sobre el espacio de búsqueda que manejan y redirigir la exploración hacia sub-espacios útiles. En la Figura 7 se muestra el mecanismo de funcionamiento de los AG de forma general. Este parte de una población inicial de individuos que codifican soluciones candidatas al problema (denominados cromosomas). Esta población evoluciona en el tiempo mediante un proceso de competición y variación controlada, aplicando operadores genéticos de selección, cruce y mutación.

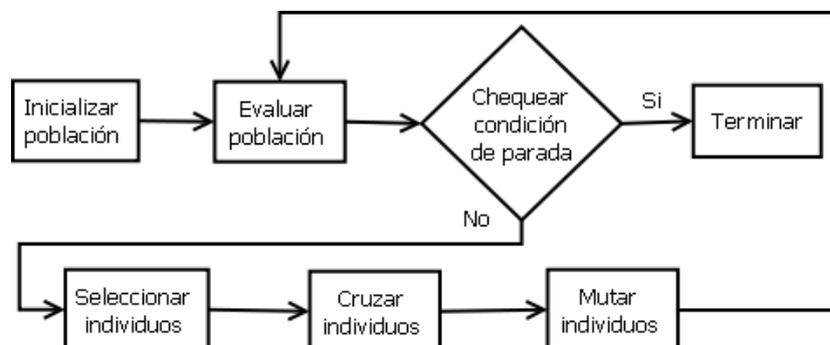


Figura 7. Procedimiento básico de un algoritmo genético.

GFS.THRIFT considera el sistema borroso tipo Mamdani. Representan los conjuntos borrosos en forma de tabla como genotipo con alelos. El fenotipo es producido por la fusificación, composición máxima y la defusificación. Los valores de los genes se representan como {Bien, Regular, Mal, \_};

donde el símbolo “\_” indica que no hay entrada en la posición donde este aparece. Un cromosoma (genotipo) es formado por una tabla de decisión dirigida por las filas que produce una cadena de números para los conjuntos codificados. Para un sistema de pocas dimensiones o variables de entrada, la tabla de decisión borrosa se puede especificar como la mostrada en la Figura 8a, para dos variables de entrada con 3 etiquetas cada una (B, R y M, correspondientes a bien, regular y mal). Las particiones borrosas se representan con funciones de pertenencia tipo triángulo en el centro y trapecio en los extremos como las mostradas en la Figura 8b.

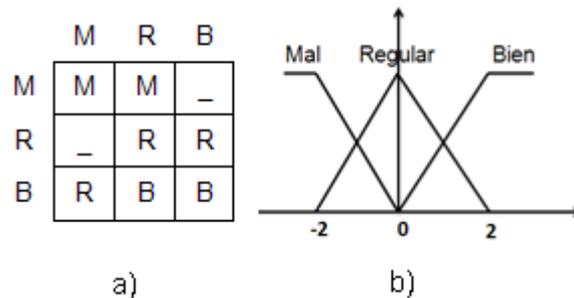


Figura 8. a) Tabla de decisión borrosa para dos variables de entrada b) Particiones borrosas.

Para la selección se aplica una estrategia elitista por lo cual la mejor solución de una generación dada es promovida directamente a la siguiente generación del AG. Un modelo con alto grado de elitismo consiste en utilizar una población intermedia con todos los padres ( $N$ ) y todos los descendientes, para seleccionar los  $N$  mejores.

El operador de cruzamiento que se utiliza es el estándar de un punto de cruce. El operador de mutación cambia un código borroso cualquiera a su nivel superior o inferior, o al código vacío “\_”; si su valor es “\_”, se cambia a un código no vacío seleccionado de forma aleatoria. La probabilidad de mutación es fijada en 0.01 y la de cruzamiento en 0.7.

Se realiza una cantidad de iteraciones definidas inicialmente para una población de soluciones generada mediante traslaciones a partir de la tabla de decisión. El operador de defusificación aplicado es una simplificación del centroide: calcula el promedio pesado de los puntos centrales en los conjuntos borrosos de la variable de salida.

Para la aplicación de GFS.THRIFT con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, tamaño de la población, cantidad máxima de generaciones, porcentaje de mutación, porcentaje de cruzamiento, T-Norma, tipo de defusificación y tipo de función de implicación. Esta funcionalidad normaliza internamente los datos de entrenamiento a valores entre 0 y 1. Utiliza como condición de parada una cantidad de generaciones máxima fijada por el usuario y una cota de error correspondiente al RMSE.

### 2.3.7 Sistema borroso genético basado en ajuste lateral (GFS.LT.RS)

En (Alcalá et al., 2007) se propone un mecanismo para realizar el ajuste lateral evolutivo de las funciones de pertenencia con el objetivo de obtener modelos lingüísticos con mayores niveles de precisión. Se presenta la cooperación del ajuste lateral junto con un mecanismo de selección de reglas borrosas, lo que reduce el espacio de búsqueda y facilita la legibilidad del sistema. Utiliza un esquema de representación de reglas basado en 2-tupla lingüística (Herrera and Martínez, 2000), que permite el desplazamiento lateral de las etiquetas (leves traslados a la izquierda/derecha de la función de pertenencia original). Las particiones borrosas se representan con funciones de pertenencia de tipo triángulo simétrico.

El desplazamiento lateral representado en una 2-tupla lingüística  $(s_i, a_i)$  se denomina traducción simbólica de una etiqueta; la cual toma valor en el intervalo  $[-0.5, 0.5)$ , expresando el dominio de una etiqueta cuando se mueve entre las dos etiquetas laterales adyacentes. Implica el

desplazamiento lateral de la función de pertenencia asociada. Se considera  $S$  como un conjunto de etiquetas que representan las particiones borrosas. Formalmente, para representar la traducción simbólica de una etiqueta  $S$  se tiene un 2-tupla como la mostrada en la Ecuación 30.

$$(s_i, a_i), s_i \in S, a_i \in [-0.5, 0.5] \quad (30)$$

La Figura 9 muestra la traducción simbólica de una etiqueta representada por el par  $(S_2, -0.3)$  junto con el desplazamiento lateral respecto a la posición inicial de su función de pertenencia.

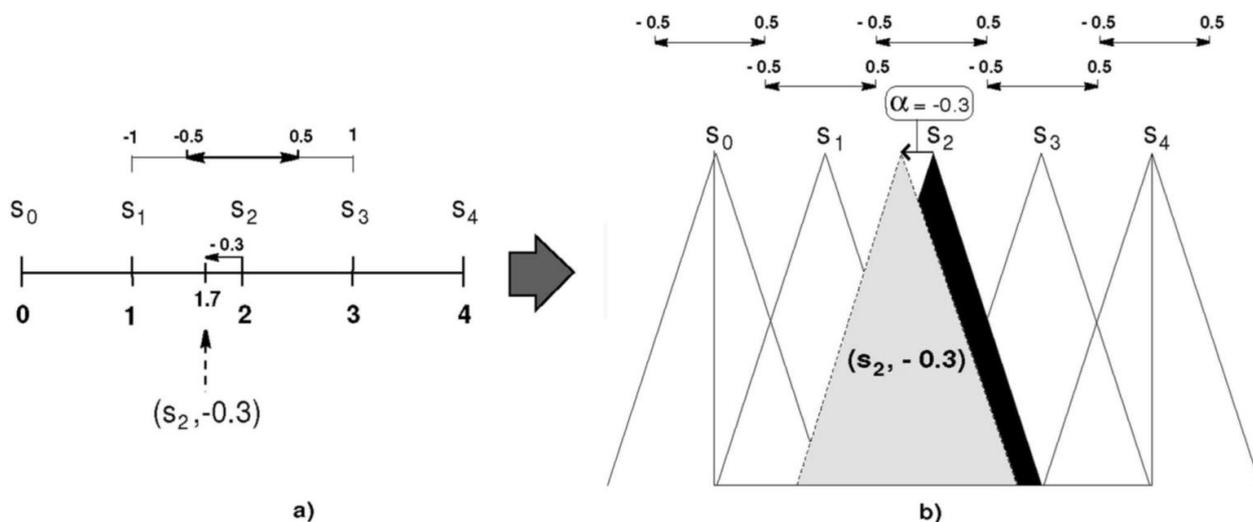


Figura 9. a) Traducción simbólica de una etiqueta lingüística b) desplazamiento lateral de la función de pertenencia (Alcalá et al., 2007).

La representación de reglas mediante las 2-tuplas lingüísticas se muestra en la Ecuación 31 y se interpreta como: "Si  $x_1$  es más que Regular y  $x_2$  es un poco menos que Bien entonces y es un poco menos que Bien".

$$R_1: \text{Si } x_1 \text{ es } (R, 0.3) \text{ y } x_2 \text{ es } (B, -0.2) \text{ entonces y es } (B, -0.1) \quad (31)$$

Este tipo de representación disminuye la complejidad durante el ajuste de las funciones de pertenencia, al considerar para cada etiqueta sólo un parámetro en lugar de los tres parámetros correspondiente a la forma de triángulo clásica.

El modo del ajuste lateral puede ser de dos formas: global o local, siendo el primero más interpretable y el segundo más exacto. El ajuste global es aplicado a nivel de partición lingüística, donde un par  $(x_i, \text{etiqueta})$  toma el mismo valor de ajuste en todas las reglas. Por ejemplo, " $x_i$  es (Bien, 0.3)" presenta el mismo valor para todas las reglas que consideran la pareja " $x_i$  es Bien". El ajuste local se realiza a nivel de regla, donde un par  $(x_i, \text{etiqueta})$  es ajustado de forma diferente en cada regla, basado en la medida de calidad asociada al método de ajuste (usualmente el error). Se consideran diferentes parámetros de desplazamiento para cada etiqueta, por ejemplo, para un par  $(x_i, \text{Bien})$  en una regla la variable  $x_i$  puede representarse con la 2-tupla  $(\text{Bien}, 0.3)$  (más que Bien) y en otra con  $(\text{Bien}, -0.2)$  (poco menos que Bien). En este caso se pierde el sentido de interpretación global pero aporta una mayor libertad para encontrar una exactitud mejor del sistema borroso y mayor exploración del espacio de búsqueda.

El esquema de codificación considera parámetros reales en los genes. En un cromosoma están contenidos todos los parámetros modificables y la estructura de la base de reglas. Para ambos modos de ajuste, la codificación de los genes es con números reales. Para obtener la población inicial se toma el primer individuo con todos sus parámetro de traslación en 0 y para el resto de los individuos se generan de forma aleatoria en el intervalo  $[-0.5, 0.5]$ .

La evaluación del cromosoma depende del tipo de sistema a modelar. Cuando se dispone de un conjunto de datos de entrenamiento de entrada-salida, se utiliza el error cuadrado medio (MSE). Cuando las reglas son creadas según el conocimiento de expertos humanos y/o se dispone de un

modelo de validación del sistema, cada cromosoma se evalúa según la medida de desempeño del sistema al realizar la simulación.

Este AG utiliza un enfoque de reinicio para escapar de los óptimos locales. En este caso, el mejor cromosoma se mantiene y los restantes se generan al azar dentro de los intervalos de variación correspondientes [0.5, 0.5). Este procedimiento de reinicio se realiza cuando se alcanza un valor umbral o todos los individuos que conviven en la población son muy similares.

El método de selección de reglas permite agregar múltiples reglas y/o seleccionar un subconjunto de reglas en orden de minimizar la extensión del conjunto de reglas. En la codificación de los cromosomas se incluye un gen para cada regla que determina si la misma se selecciona o no (alelos 1 y 0 respectivamente).

Para la aplicación de GFS.LT.RS con la biblioteca AnalysisPro.SC.PMC, mediante la funcionalidad *frbs.learn*, se requiere asignarle valor a los parámetros: datos de entrenamiento, cantidad de etiquetas lingüísticas, rango de los datos, tamaño de la población, cantidad máxima de generaciones, porcentaje de mutación, modo de ajuste, T-Norma, tipo de defusificación, tipo de función de implicación y selección de reglas. Esta funcionalidad normaliza internamente los datos de entrenamiento a valores entre 0 y el número de etiquetas menos 1. Esta variante de implementación de GFS.LT.RS realiza mutación y no cruzamiento. Aplica como condición de parada una cantidad de generaciones máxima fijada por el usuario y una cota de error.

## **2.4 Conclusiones del capítulo**

En el presente capítulo se aplican técnicas de *soft computing* en la construcción y aprendizaje de sistemas de inferencia borrosos para la evaluación de la ejecución de proyectos, dando cumplimiento al segundo objetivo específico de la investigación. Según el análisis de potencialidades y limitaciones de la propuesta, se llega a las siguientes conclusiones:

- 1) La propuesta de solución, para realizar el aprendizaje, requiere de una base de casos de proyectos terminados que contenga la evaluación de los expertos.
- 2) Se desarrolla la biblioteca AnalysisPro.SC.PMC basada en técnicas de *soft computing* para la construcción, aprendizaje automático y selección de sistemas de inferencia borrosos para la evaluación de la ejecución de proyectos. La misma se integra a la base de conocimiento de la organización.
- 3) El sistema propuesto es automatizado completamente, desde la consulta a la base de conocimiento de proyectos terminados, hasta la aplicación de pruebas estadísticas de rigor para seleccionar el sistema que mejor evalúe la ejecución de los proyectos.
- 4) A diferencia de las soluciones más cercanas consultadas en la literatura, el sistema propuesto efectúa el tratamiento de la incertidumbre en la información y la integración con herramientas informáticas para la gestión de proyectos, empleando tecnologías de *software* libre.
- 5) Con la implementación de la propuesta se pretende mejorar la calidad de la clasificación de los proyectos como apoyo a la toma de decisiones en organizaciones con mejora continua en los estilos de gestión.

## **CAPÍTULO 3. APLICACIÓN DEL SISTEMA Y ANÁLISIS DE RESULTADOS**

En el presente capítulo se analizan los resultados de aplicación del sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos. Se exponen las características del entorno de aplicación, de la base de casos utilizada en el aprendizaje y los parámetros de los algoritmos ejecutados. Se desarrollan en detalle los experimentos para validar el comportamiento de la variable dependiente: calidad de la clasificación de proyectos. Se presenta el sistema de inferencia borroso seleccionado. Se realiza una valoración del impacto económico y social de la propuesta. Finalmente se presentan las conclusiones del capítulo.

### **3.1 Síntesis de aplicación**

La construcción de sistemas de inferencia borrosos se destina a la actual versión 14.05 de la Suite de Gestión de Proyectos Xedro-GESPRO, la cual está siendo utilizada por la red de centros de desarrollo de la UCI y otras entidades nacionales para la gestión de sus proyectos. A continuación, se realiza una breve caracterización del entorno de gestión sobre el cual se ejecuta la propuesta de solución. Seguidamente se describe la base de casos utilizada para el aprendizaje y se presentan los valores de los parámetros de cada técnica aplicada.

#### **3.1.1 Caracterización del entorno**

La construcción de los sistemas de inferencia borrosos se realiza a partir de un conjunto de casos correspondiente a 204 proyectos, recuperados de una base de datos de proyectos terminados y en ejecución que se gestionan con la herramienta Xedro-GESPRO. Dichos casos son recopilados por el Laboratorio de Investigaciones en Gestión de Proyectos, en un período de fechas de corte entre marzo y abril de 2014, correspondiente a proyectos iniciados entre octubre de 2007 y abril de 2014. Los proyectos utilizados en el aprendizaje corresponden a los centros de desarrollo de *software* de la UCI, la cual se encuentra ubicada en el nivel 2 de CMMI. Debido al nivel de madurez actual de gestión de proyectos en dicha organización, sólo se tienen en cuenta los indicadores: IRE, IRP, ICD, IRRH e IREF. En la base de casos no está presente la información referente a los indicadores de logística y costo debido a que en la organización aún no se manejan estas áreas de conocimiento.

Los indicadores almacenados se calculan previamente mediante un mecanismo automatizado que se ejecuta semanalmente. Esto permite contar con una base de conocimiento confiable, actualizada y obtenida con un mínimo de esfuerzo y de errores en las operaciones de cálculo. Para garantizar el aprendizaje automático sobre los datos existentes en la base de conocimiento, partiendo de un enfoque supervisado, se evalúa el conjunto de proyectos por un grupo de expertos. Con el objetivo de contar con una base de conocimiento acorde al nivel de madurez de la gestión en la organización, se recomienda realizar la evaluación por los expertos con período semestral o anual (momento en que los directivos de la organización lo consideren necesario). Lo anterior implica, que la propuesta de solución debe ejecutarse con esa misma frecuencia para generar un sistema de inferencia borroso actualizado y con esto mejorar la calidad de la clasificación de proyectos durante la mejora continua de la organización.

#### **3.1.2 Base de casos para el aprendizaje**

Para el aprendizaje de los sistemas de inferencia borrosos se utiliza la base de casos de proyectos terminados y en ejecución extraída de la plataforma informática Xedro-GESPRO y procesada en el Laboratorio de Investigaciones en Gestión de Proyectos (Tabla 6). La base de casos contiene los valores de los indicadores en una fecha de corte (calculados de forma automática) y la evaluación de proyectos ofrecida por los expertos (con valores: 1 para *Mal*, 2 para *Regular* y 3 para *Bien*). Los indicadores que se analizan corresponden a las áreas de conocimiento claves de la gestión de proyectos: Integración, Tiempo, Calidad y Recursos Humanos.

Donde los indicadores de gestión, según el valor que toman los mismos, se interpretan de la siguiente forma en relación con la evaluación de la ejecución del proyecto:

- Índice de rendimiento de la ejecución: si toma valor  $< 1$  entonces la ejecución del proyecto es baja; si es  $= 0$  está atrasada; si es  $= 1$  es buena; si es  $> 1$  está adelantada.
- Índice de rendimiento de la producción: si es  $< 1$  entonces el proyecto está atrasado; si es  $= 0$  está atrasado por no haber comenzado; si es  $= 1$  está en tiempo; si es  $> 1$  está adelantado.
- Índice de rendimiento de la eficacia: es mejor al estar más cercano a 1.
- Índice de rendimiento de los recursos humanos: es mejor al estar más cercano a 100.
- Índice de calidad del dato: es mejor al estar más cercano a 1.

Tabla 6. Muestra de la base de casos utilizada en el aprendizaje.

Proyectos	IRE	IRP	IREF	IRRH	ICD	Evaluación
P1	0,45	0,58	0,3	50,45	0,96	2
P2	1,01	1,01	0,54	79	0,97	3
P3	0,47	0,46	0,2	10	0,95	1
...	...	...	...	...	...	...
P204	0,47	0,46	0,2	10	0,2	1

La base de casos utilizada presenta la siguiente distribución: 63 proyectos evaluados de Bien (31%), 92 de Regular (45%) y 49 de Mal (24%); no contiene valores nulos o fuera de rango. En la Tabla 7 se muestran las características del conjunto de datos utilizado en el aprendizaje, en cuanto a los valores mínimo, máximo, promedio y desviación estándar para cada variable. Las dos primeras filas de la tabla se corresponden con los rangos de los datos de entrenamiento (cota mínima y cota máxima). Además se muestran los valores mínimo, máximo, promedio y desviación estándar según la clasificación de los proyectos en Bien, Regular y Mal.

Tabla 7. Características de la base de casos utilizada en el aprendizaje.

Característica		IRE	IRP	IREF	IRRH	ICD	Evaluación
Cota mínima		0	0	0	0	0	1
Cota máxima		2	2	1	100	1	3
Valor mínimo		0	0	0	0	0	1
Valor máximo		1.07	1.072	1	82.3	0.805	3
Promedio		0.74	0.87	0.91	50.82	0.45	2.06
Desviación estándar		0.38	0.27	0.25	24.23	0.19	0.73
Bien	Valor mínimo	0	0.98	0.97	75.25	0.39	3
	Valor máximo	1.02	1.07	1	82.3	0.80	3
	Promedio	0.92	1.01	0.98	78.97	0.57	3
	Desviación estándar	0.24	0.01	0.01	1.02	0.08	0
Regular	Valor mínimo	0	0.58	0.96	31.6	0.2	2
	Valor máximo	1.07	1.04	1	51.77	0.76	2
	Promedio	0.73	0.94	0.99	50.62	0.50	2
	Desviación estándar	0.38	0.09	0.01	2.05	0.11	0
Mal	Valor mínimo	0	0	0	0	0	1
	Valor máximo	1.05	1	1	29	0.76	1
	Promedio	0.52	0.57	0.69	15	0.21	1
	Desviación estándar	0.41	0.41	0.44	11.13	0.21	0

La base de proyectos se divide de forma aleatoria en 20 particiones diferentes. Cada partición dispone del 75% de los 204 casos para entrenar y el 25% restante para validar. Dichas particiones de la base de casos se utilizan en el procedimiento de validación cruzada de los experimentos, realizando 20 ejecuciones de cada técnica de *soft computing*.

### 3.1.3 Parámetros de los experimentos

Para la aplicación de las técnicas de *soft computing*, mediante la funcionalidad *frbs.learn*, se le asignan los valores mostrados en la Tabla 8 a cada parámetro requerido por cada técnica.

Tabla 8. Parámetros de aprendizaje utilizados con cada técnica.

Técnica	Valor de cada parámetro
WM	num.labels=3; type.mf="TRAPEZOID"; type.tnorm="HAMACHER"; type.snorm="HAMACHER"; type.defuz="WAM"; type.implication.func = "MIN"
ANFIS	num.labels=3; max.iter=200; step.size=0.1; type.tnorm="YAGER"; type.snorm="YAGER"; type.implication.func="DUBOIS_PRADE"
HYFIS	num.labels=3; max.iter=200; step.size=0.1; type.tnorm="YAGER"; type.snorm="YAGER"; type.defuz="WAM"; type.implication.func="DUBOIS_PRADE"
FIR.DM	num.labels=3; max.iter=200; step.size=0.01; type.tnorm="YAGER"; type.snorm="YAGER"; type.implication.func = "ZADEH"
FS.HGD	num.labels=3; max.iter=200; step.size=0.1; alpha.heuristic=1; type.tnorm="YAGER"; type.implication.func = "DUBOIS_PRADE"
GFS.THRIFT	num.labels=3; popu.size=50; max.gen=100; persen_cross=0.8; persen_mutant=0.15; type.tnorm="YAGER"; type.defuz="WAM"; type.implication.func="DUBOIS_PRADE"
GFS.LT.RS	num.labels=3; popu.size=50; max.gen=100; persen_mutant=0.15; mode.tuning="LOCAL"; type.tnorm="YAGER"; type.defuz="WAM"; type.implication.func="DUBOIS_PRADE"; rule.selection=TRUE

Una vez mostrado un extracto de la base de casos y planteados los valores de los parámetros de cada algoritmo, se procede con los resultados de los experimentos ejecutados.

## 3.2 Resultados experimentales

En la presente sección se muestran los resultados de las métricas calculadas en cada una de las ejecuciones de las siete técnicas sobre las 20 particiones. Se aplican los test de Friedman y Wilcoxon con el fin de comparar el desempeño de cada algoritmo y se analizan los resultados obtenidos. Por último se muestra el sistema de inferencia borroso generado con la mejor variante.

### 3.2.1 Resultados de la ejecución de cada técnica

Los resultados obtenidos con la ejecución de cada técnica en las 20 particiones de datos (Anexo 1) permiten comparar el desempeño de los algoritmos, teniendo en cuenta las siguientes variables: cantidad de reglas borrosas generadas, porcentaje de clasificaciones correctas, cantidad de falsos positivos, cantidad de falsos negativos, MSE, RMSE y SMAPE.

En las Figuras 10-16 se muestran las predicciones de los casos de entrenamiento y los de validación para una de las 20 ejecuciones realizadas con cada algoritmo.

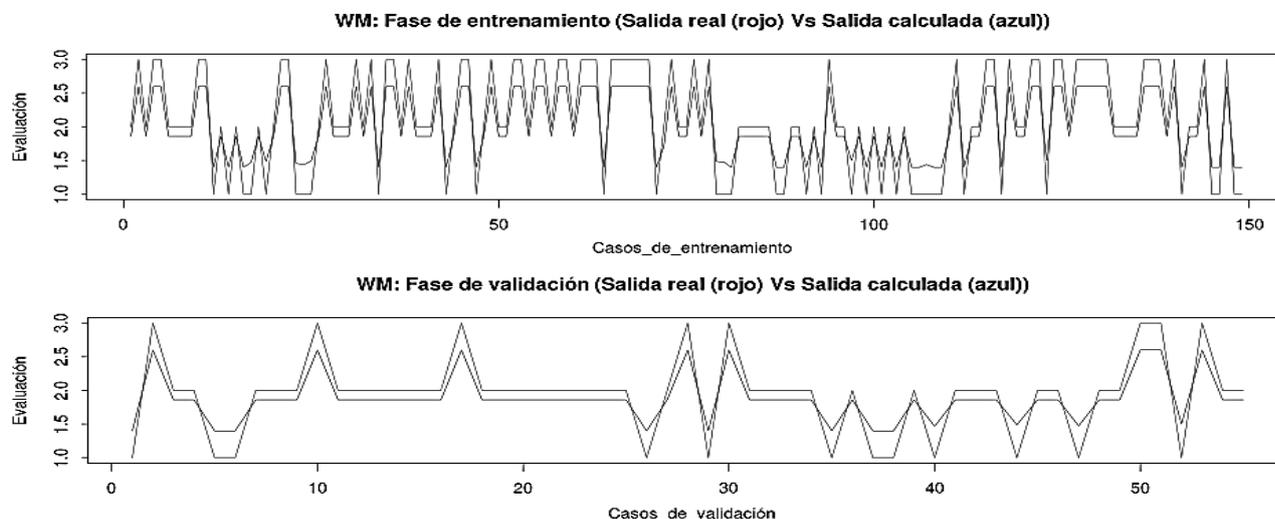


Figura 10. Predicciones de casos según WM.

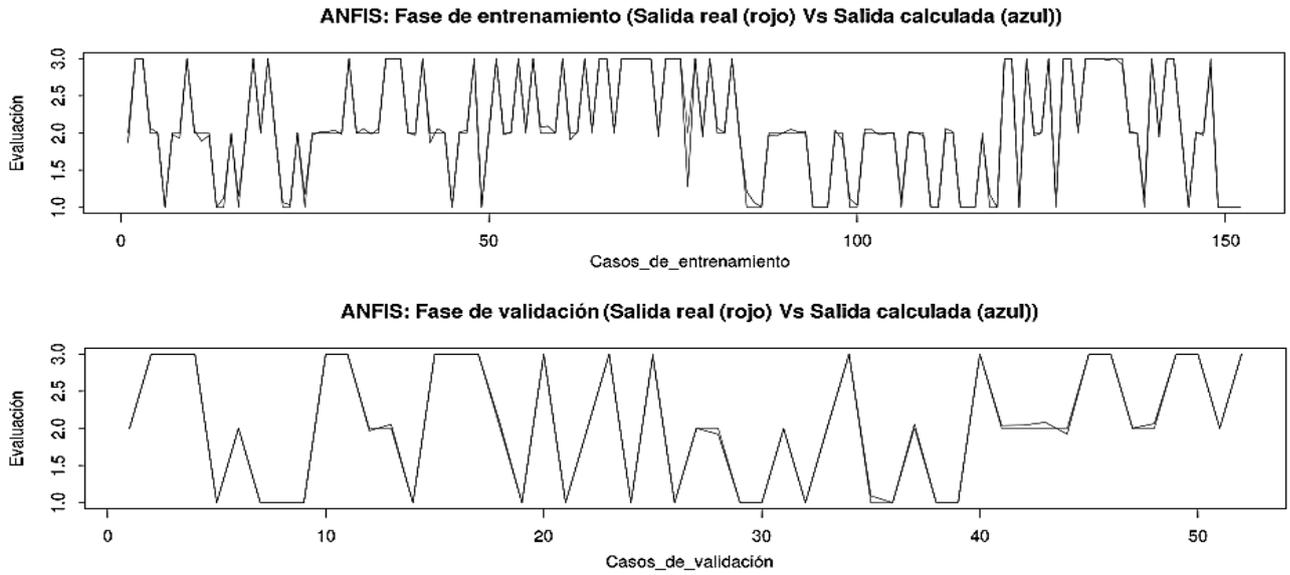


Figura 11. Predicciones de casos según ANFIS.

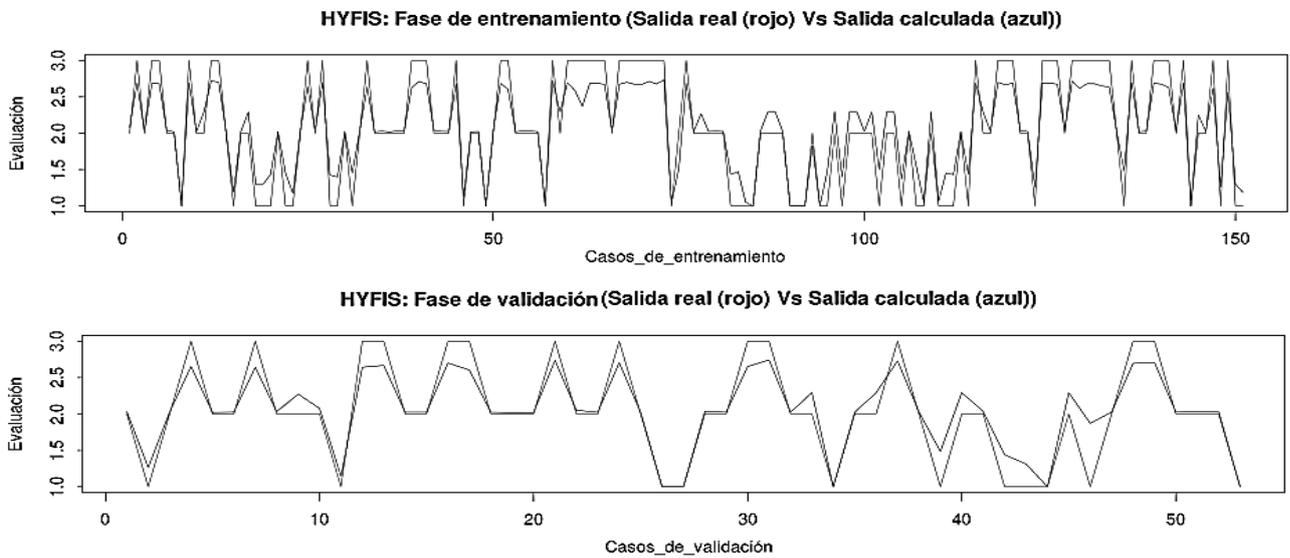


Figura 12. Predicciones de casos según HYFIS.

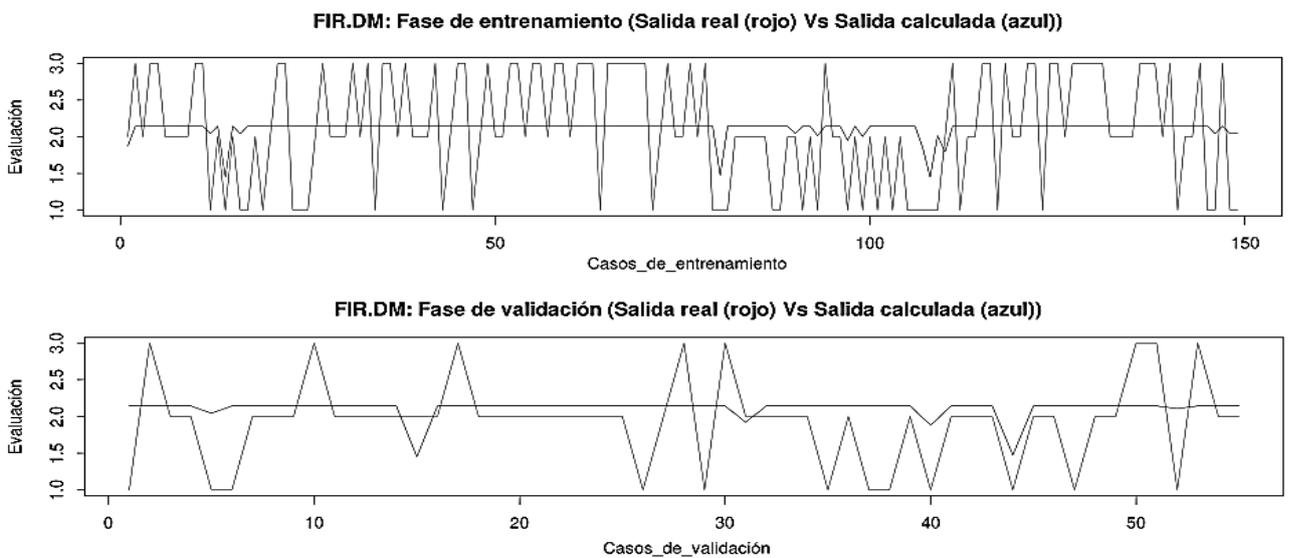


Figura 13. Predicciones de casos según FIR.DM.

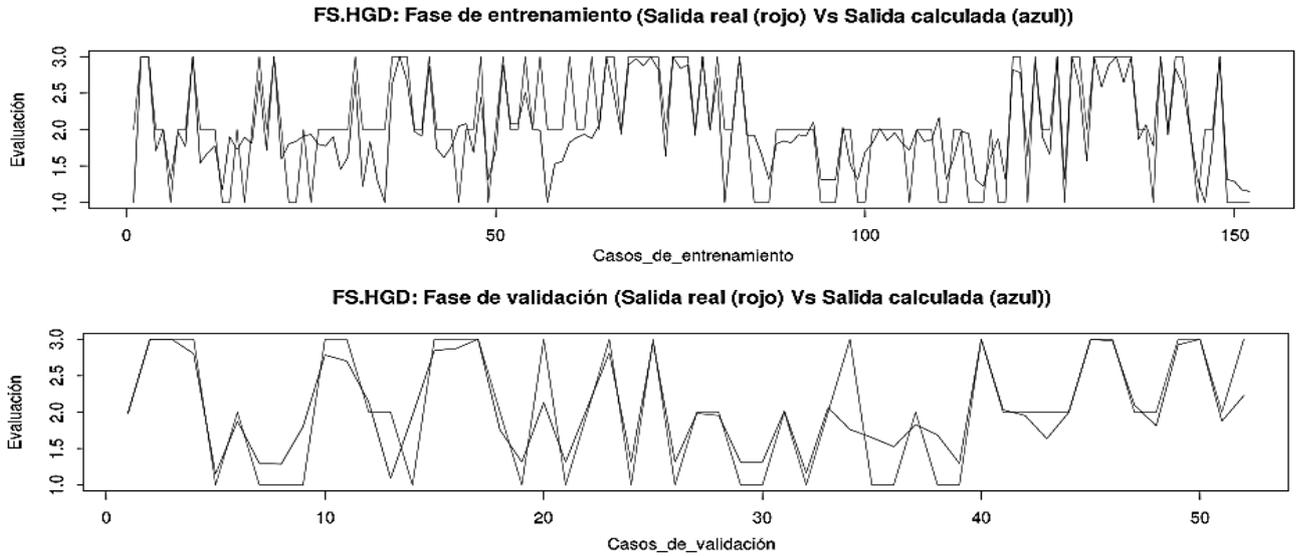


Figura 14. Predicciones de casos según FS.HGD.

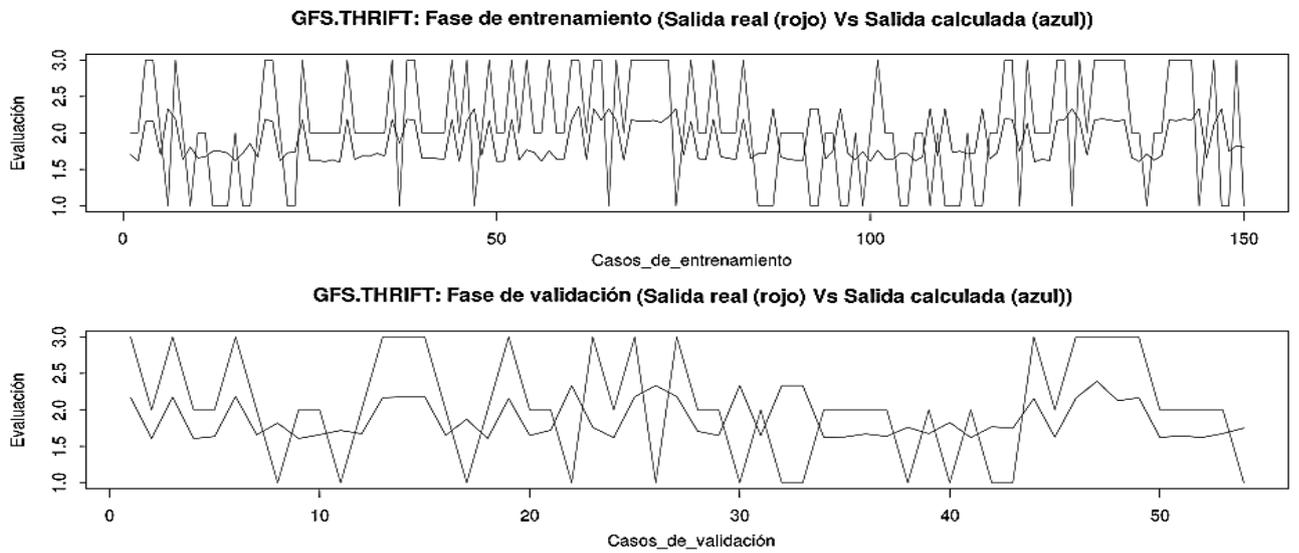


Figura 15. Predicciones de casos según GFS.THRIFT.

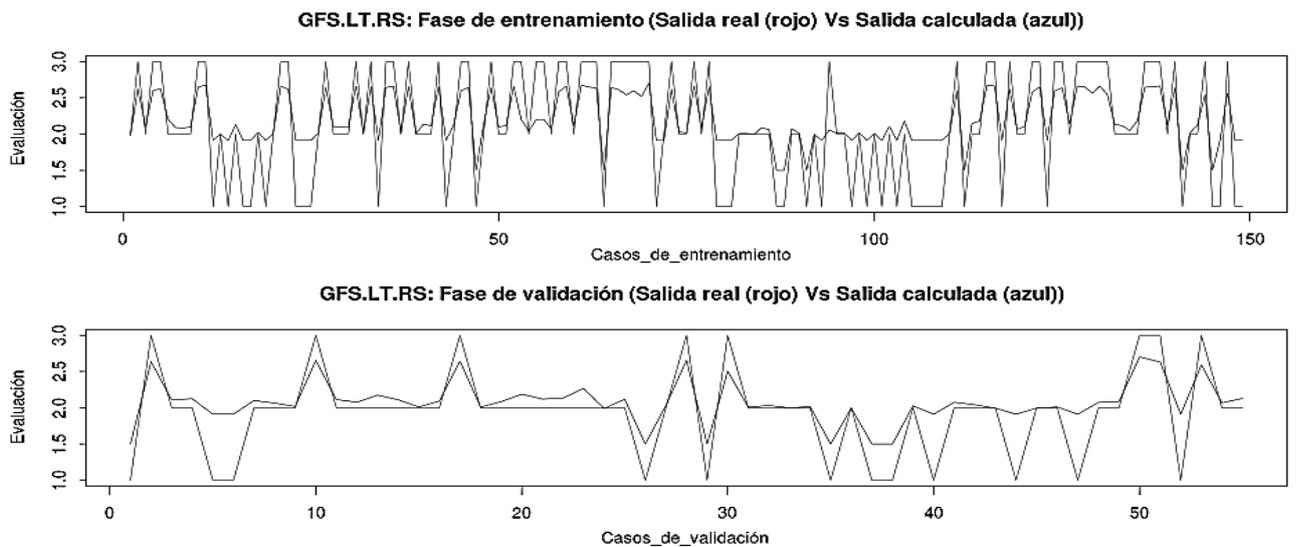


Figura 16. Predicciones de casos según GFS.LT.RS.

Se puede apreciar de forma gráfica que con la técnica ANFIS se obtienen buenos resultados, teniendo en cuenta que las evaluaciones de proyectos calculadas (azul) son bastante parecidas a

las evaluaciones esperadas (rojo). Sin embargo, con FIR.DM las evaluaciones de proyectos obtenidas son muy diferentes de las esperadas, tanto para los casos de entrenamiento como para los de validación. El resto de las técnicas se comportan de manera intermedia en comparación con las dos anteriormente analizadas.

### 3.2.2 Resultados de las pruebas estadísticas

Para la comparación de las siete técnicas se utilizan los test de Friedman y Wilcoxon según se especifica a continuación. En ambos casos se considera el umbral igual a 0.05 para determinar si existen diferencias significativas. Con el fin de validar los algoritmos ejecutados, se considera cada par de algoritmos como dos muestras relacionadas. Para determinar si existen diferencias significativas o no, se aplica el test no paramétrico de Wilcoxon para dos muestras relacionadas, con el método de Monte Carlo con un 99% de intervalo de confianza. Para ello se utiliza la herramienta PgAdmin III e implementan funciones en lenguaje PL/R.

Durante la aplicación de las pruebas, en los casos que el test de Friedman encuentra diferencias significativas entre los siete algoritmos, mediante el test de Wilcoxon se forman diferentes grupos de algoritmos. Los grupos cumplen con las siguientes propiedades: los algoritmos de un mismo grupo no tienen diferencias significativas entre ellos; los algoritmos en los grupos menores (Grupo1 < Grupo2 <...) son los que reportan mejores resultados mientras que los ubicados en los grupos más altos son los de peor resultado para la variable analizada. A continuación se relacionan los experimentos de validación realizados, indicando en cada caso los objetivos que persiguen y los resultados obtenidos.

**Experimento 1:** Comparar las siete técnicas respecto a la variable: cantidad de reglas borrosas generadas.

**Objetivo:** Determinar si existen diferencias significativas en la cantidad de reglas generadas por las siete técnicas. En caso de existir diferencias significativas determinar cuáles técnicas reportan la menor cantidad de reglas y cuáles la mayor.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre la base de datos de gestión de proyectos y respecto a la variable: cantidad de reglas borrosas generadas. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: GFS\_LT\_RS; Grupo2: FIR\_DM; Grupo3: FS\_HGD; Grupo4: ANFIS; Grupo5: WM; Grupo6: HYFIS; Grupo7: GFS\_THRIFT. Al estar ubicado en el grupo 1, se puede afirmar que el algoritmo GFS\_LT\_RS genera la menor cantidad de reglas borrosas, significativamente menor al resto de los algoritmos.

Como complemento del análisis, en la Figura 17 se muestra el promedio de la cantidad de reglas generadas por cada técnica en las 20 ejecuciones.

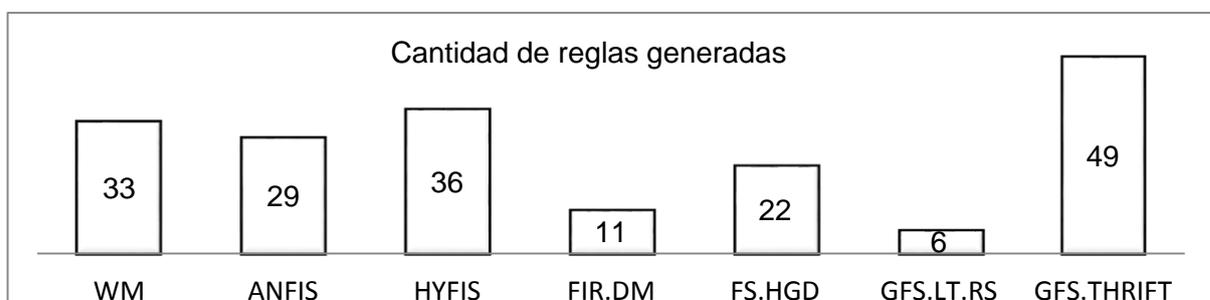


Figura 17. Cantidad promedio de reglas generadas por cada técnica.

Esta variable no influye directamente en la calidad de la clasificación de los proyectos, pero si en el funcionamiento del sistema en el momento de la evaluación desde el punto de vista del tiempo

utilizado para ejecutar el sistema de inferencia borroso. A mayor cantidad de reglas el sistema demora más tiempo en hacer la evaluación de nuevos proyectos.

**Experimento 2:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento en cuanto a la variable: porcentaje de clasificaciones correctas obtenidas durante el entrenamiento.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos mediante las siete técnicas, respecto al porcentaje de clasificaciones correctas durante el entrenamiento. En caso de existir diferencias significativas, determinar cuáles técnicas reportan los mejores resultados.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: porcentaje de clasificaciones correctas. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: HYFIS y GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: FIR\_DM y GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al porcentaje de clasificaciones correctas durante el entrenamiento.

Como complemento del análisis, en la Figura 18 se muestran de forma gráfica los porcentajes de clasificaciones correctas obtenidas con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS y WM obtienen los valores mayores.

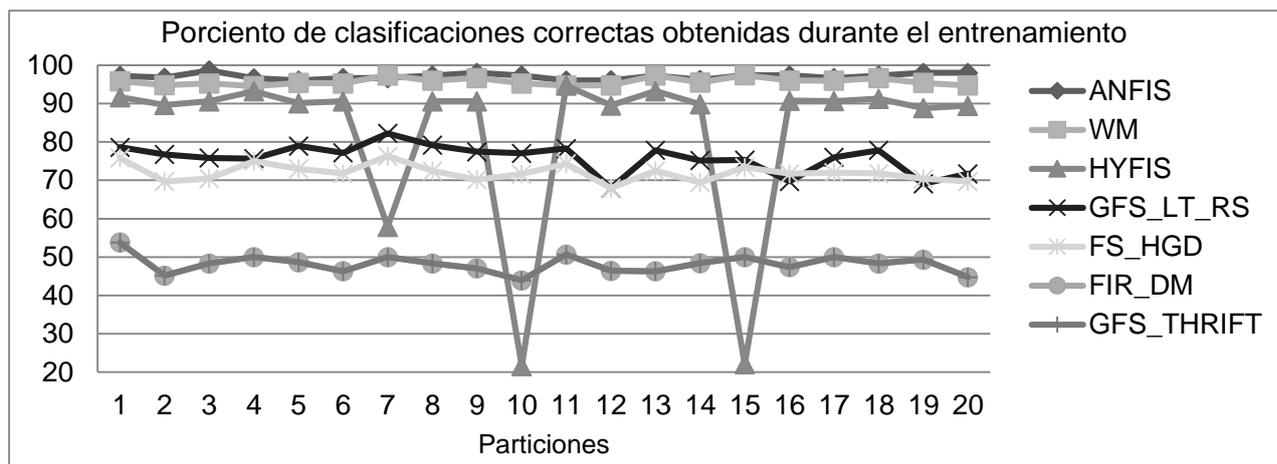


Figura 18. Porcentaje de clasificaciones correctas obtenidas con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

**Experimento 3:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba en cuanto a la variable: porcentaje de clasificaciones correctas durante las pruebas.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto al porcentaje de clasificaciones correctas durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: porcentaje de clasificaciones correctas. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: HYFIS; Grupo4: GFS\_LT\_RS; Grupo5: FS\_HGD; Grupo6: FIR\_DM y GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al porcentaje de clasificaciones correctas sobre el conjunto de casos de prueba.

Como complemento del análisis, en la Figura 19 se muestran de forma gráfica los porcentajes de clasificaciones correctas obtenidas con cada técnica durante las pruebas, en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores mayores.

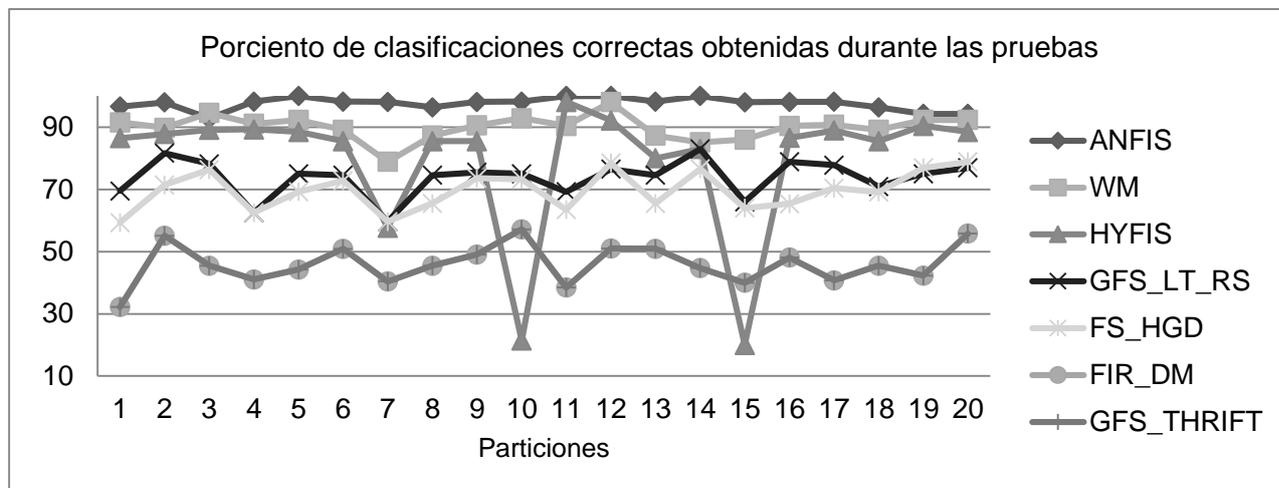


Figura 19. Porcentaje de clasificaciones correctas obtenidas con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

**Experimento 4:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento en cuanto a la variable: cantidad de falsos positivos.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la cantidad de falsos positivos durante el entrenamiento. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Como complemento del análisis, en la Figura 20 se muestra de forma gráfica: la cantidad de falsos positivos obtenidos con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores menores.

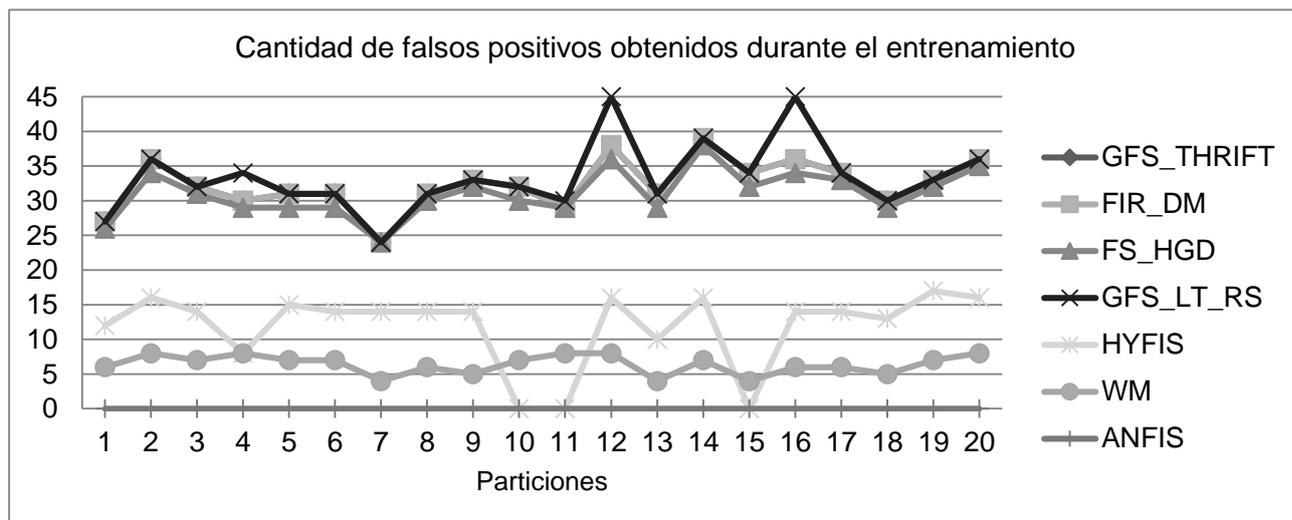


Figura 20. Cantidad de falsos positivos obtenidos con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: cantidad de falsos positivos. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se

aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: HYFIS; Grupo4: FS\_HGD; Grupo5: FIR\_DM, GFS\_THRIFT y GFS\_LT\_RS.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto a la cantidad de falsos positivos durante el entrenamiento.

**Experimento 5:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba en cuanto a la variable: cantidad de falsos positivos.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la cantidad de falsos positivos durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: cantidad de falsos positivos. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM y HYFIS; Grupo3: FS\_HGD; Grupo4: FIR\_DM, GFS\_THRIFT y GFS\_LT\_RS.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto a la cantidad de falsos positivos durante las pruebas.

Obtener falsos positivos (clasificación de *Bien* cuando el proyecto en realidad está *Mal*) en el área de la gestión de proyectos es perjudicial. Evaluar de *Bien* de un proyecto que en realidad no lo está puede causar perjuicios en la toma de decisiones y provocar planificaciones irreales.

Como complemento del análisis, en la Figura 21 se muestra de forma gráfica: la cantidad de falsos positivos obtenidos con cada técnica durante las pruebas en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores menores.

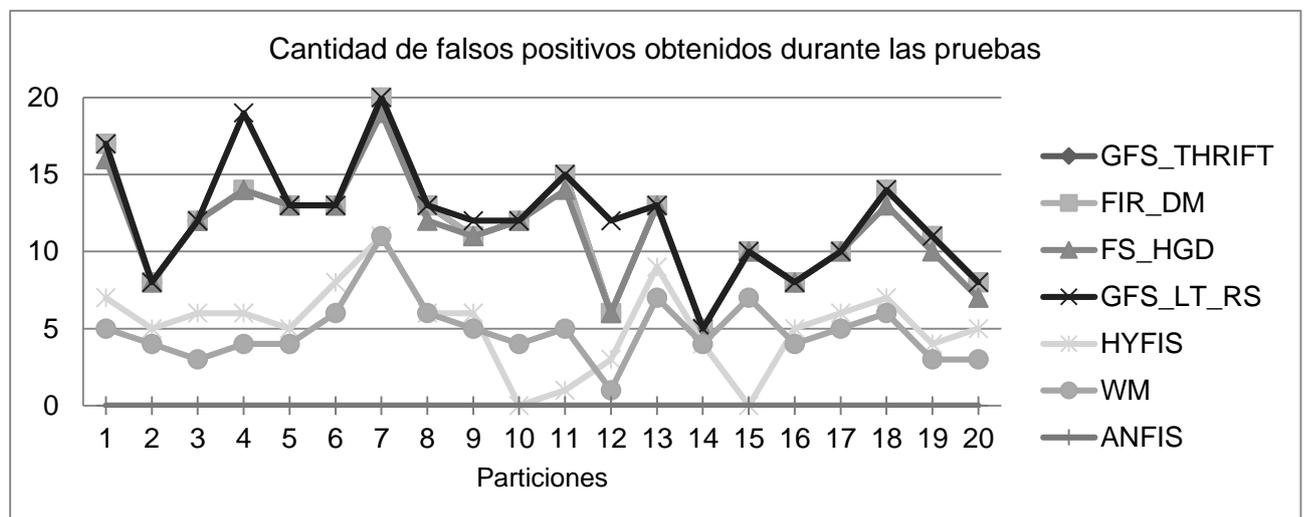


Figura 21. Cantidad de falsos positivos obtenidos con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

**Experimento 6:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento en cuanto a la variable: cantidad de falsos negativos.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la cantidad de falsos negativos durante el entrenamiento. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: cantidad de falsos negativos. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: WM; Grupo2: GFS\_LT\_RS; Grupo3: ANFIS; Grupo4: HYFIS y FS\_HGD; Grupo5: FIR\_DM y GFS\_THRIFT.

Se puede afirmar que el algoritmo WM obtiene resultados significativamente mejores en cuanto a la cantidad de falsos negativos durante el entrenamiento.

Como complemento del análisis, en la Figura 22 se muestra de forma gráfica: la cantidad de falsos negativos obtenidos con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que WM, GFS\_LT\_RS y ANFIS obtienen los valores menores.

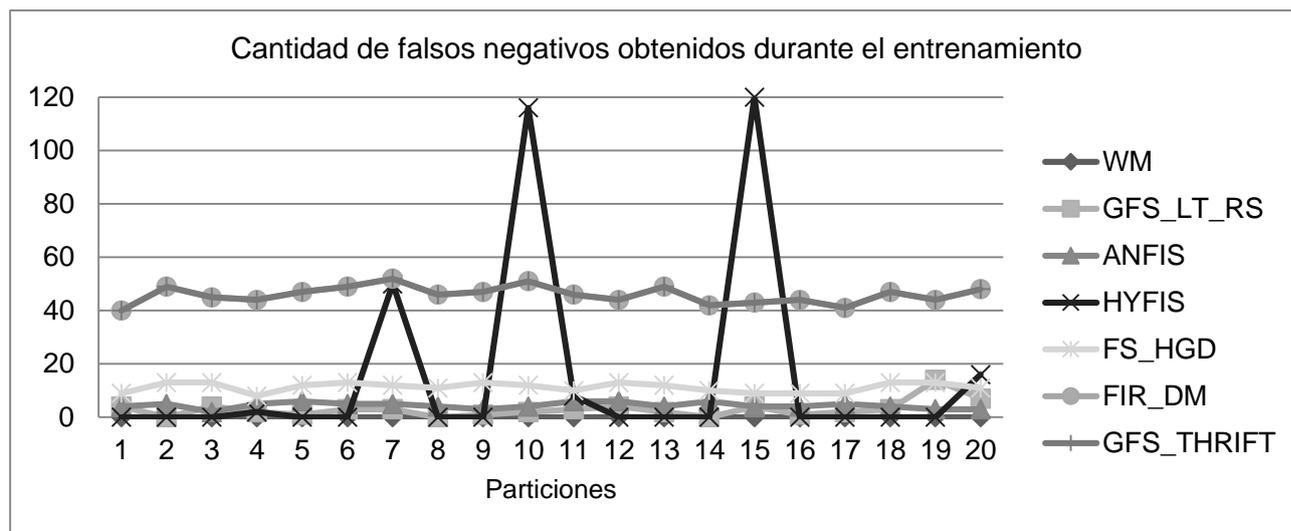


Figura 22. Cantidad de falsos negativos obtenidos con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

**Experimento 7:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba en cuanto a la variable: cantidad de falsos negativos.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la cantidad de falsos negativos durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: cantidad de falsos negativos. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: WM; Grupo2: ANFIS y GFS\_LT\_RS; Grupo3: HYFIS y FS\_HGD; Grupo4: FIR\_DM y GFS\_THRIFT.

Se puede afirmar que el algoritmo WM obtiene resultados significativamente mejores en cuanto a la cantidad de falsos negativos durante las pruebas. Obtener falsos negativos (clasificación de *Mal* cuando el proyecto en realidad está *Bien*) en el área de la gestión de proyectos no es tan perjudicial como los falsos positivos, pero evaluar de *Mal* un proyecto que en realidad no lo esté puede conllevar a tomar malas decisiones. ANFIS alcanza un valor promedio pequeño (2,79) en el porcentaje de falsos negativos, posicionándose en segundo grupo.

Como complemento del análisis, en la Figura 23 se muestra de forma gráfica: la cantidad de falsos negativos obtenidos con cada técnica durante las pruebas en las 20 ejecuciones realizadas. Se puede apreciar que WM, GFS\_LT\_RS y ANFIS obtienen los valores menores.

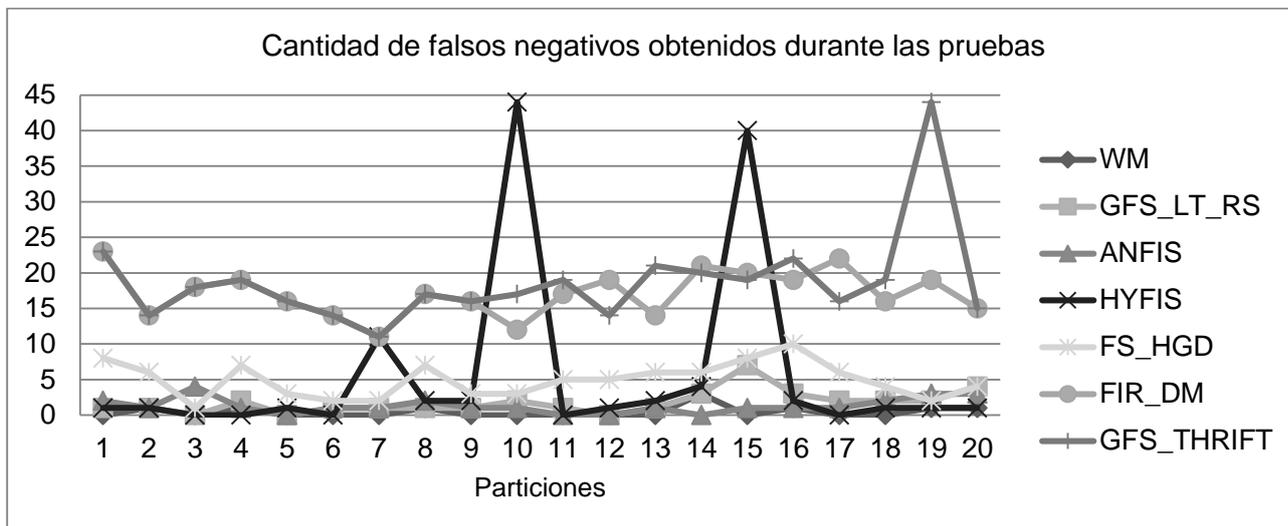


Figura 23. Cantidad de falsos negativos obtenidos con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

**Experimento 8:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento en cuanto a la variable: error cuadrático medio.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto al error cuadrático medio durante el entrenamiento. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: error cuadrático medio. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: HYFIS; Grupo6: FIR\_DM; Grupo7: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al error cuadrático medio durante el entrenamiento.

Como complemento del análisis, en la Figura 24 se muestra de forma gráfica: el error cuadrático medio obtenido con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que WM, GFS\_LT\_RS y ANFIS obtienen los valores menores.

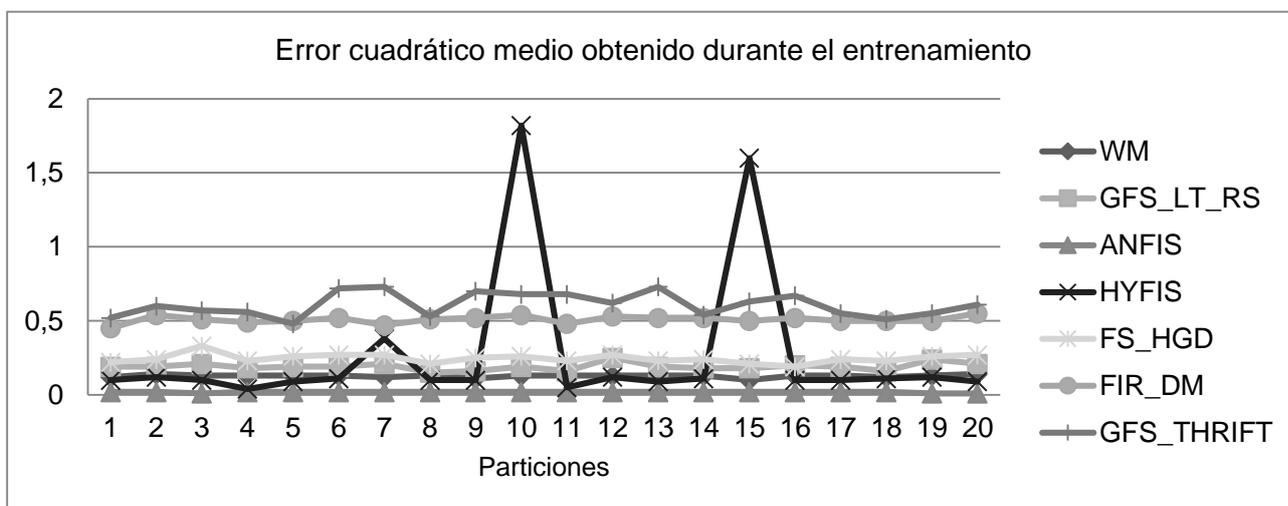


Figura 24. Error cuadrático medio obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

**Experimento 9:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba, en cuanto a la variable: error cuadrático medio.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto al error cuadrático medio durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: error cuadrático medio. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: HYFIS; Grupo6: FIR\_DM; Grupo7: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al error cuadrático medio durante las pruebas.

Como complemento del análisis, en la Figura 25 se muestra de forma gráfica: el error cuadrático medio obtenido con cada técnica durante las pruebas en las 20 ejecuciones realizadas. Se puede apreciar que WM, GFS\_LT\_RS y ANFIS obtienen los valores menores.

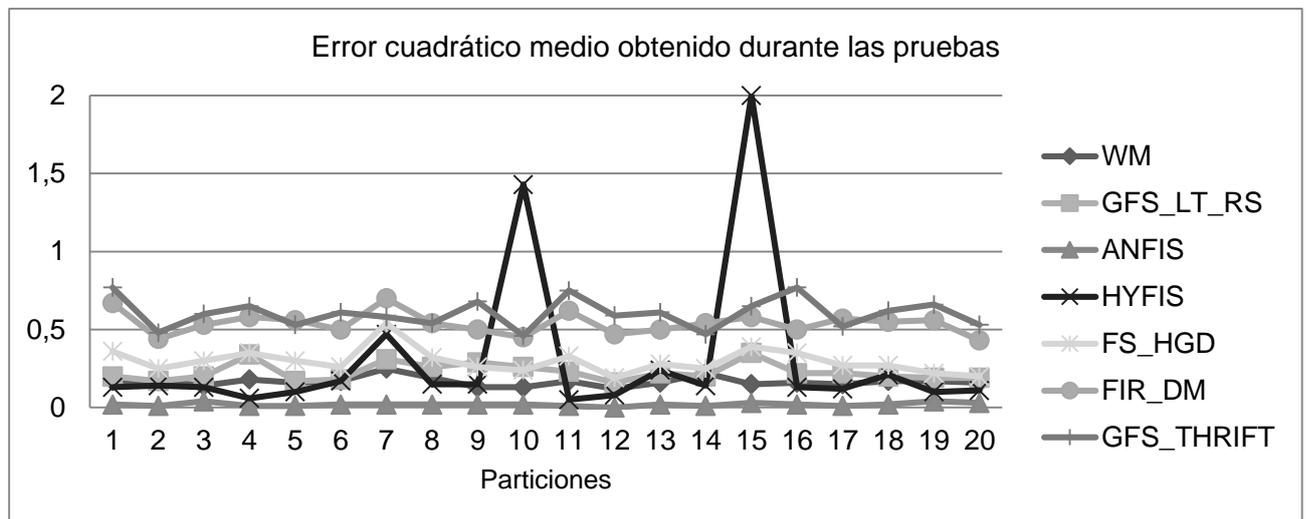


Figura 25. Error cuadrático medio obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

**Experimento 10:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento, en cuanto a la variable: raíz del error cuadrático medio.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la raíz del error cuadrático medio durante el entrenamiento. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: raíz del error cuadrático medio. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM y HYFIS; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: FIR\_DM; Grupo6: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto a la raíz del error cuadrático medio durante el entrenamiento.

Como complemento del análisis, en la Figura 26 se muestra de forma gráfica: la raíz del error cuadrático medio obtenido con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que WM y ANFIS obtienen los valores menores.

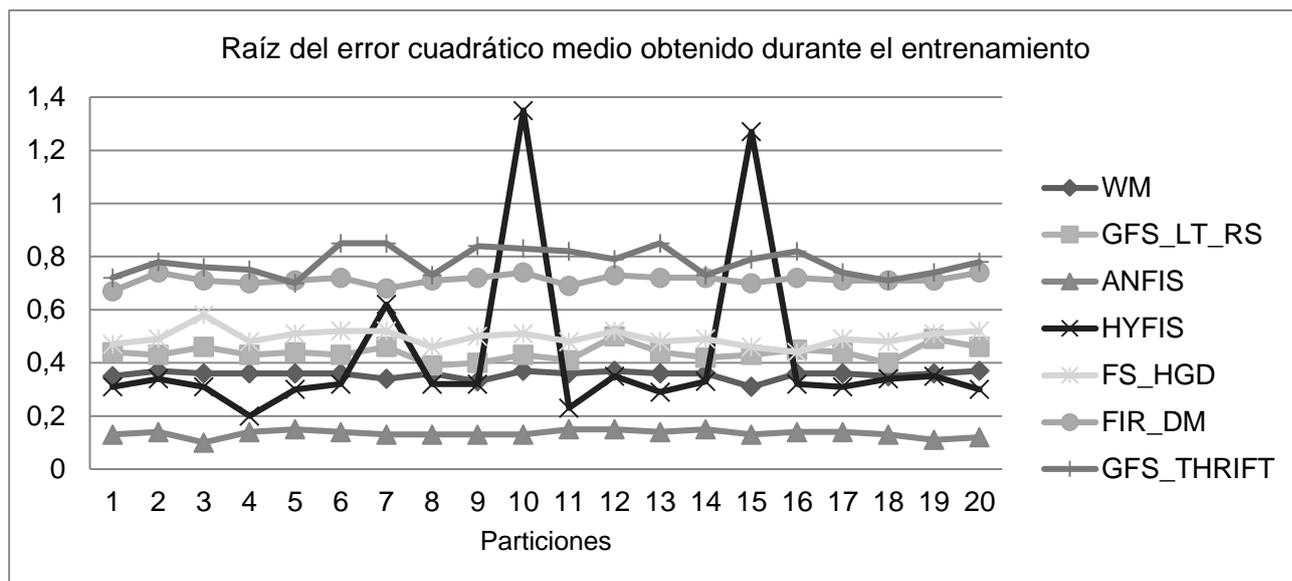


Figura 26. Raíz del error cuadrático medio obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

**Experimento 11:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba, en cuanto a la variable: raíz del error cuadrático medio.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto a la raíz del error cuadrático medio durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Como complemento del análisis, en la Figura 27 se muestra de forma gráfica: la raíz del error cuadrático medio obtenido con cada técnica durante las pruebas en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores menores.

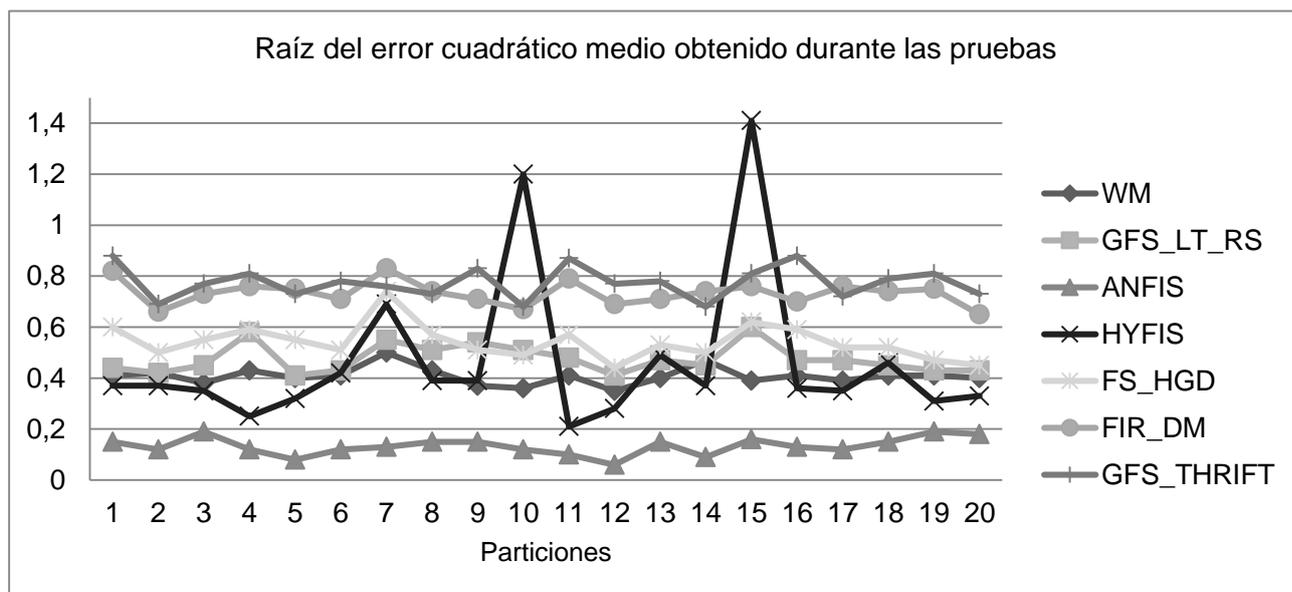


Figura 27. Raíz del error cuadrático medio obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: raíz del error cuadrático medio.

Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM y HYFIS; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: FIR\_DM; Grupo6: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto a la raíz del error cuadrático medio durante las pruebas.

**Experimento 12:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de entrenamiento en cuanto a la variable: error porcentual de la media absoluta simétrica.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto al error porcentual de la media absoluta simétrica durante el entrenamiento. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de entrenamiento y respecto a la variable: error porcentual de la media absoluta simétrica. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: HYFIS; Grupo6: FIR\_DM; Grupo7: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al error porcentual de la media absoluta simétrica durante el entrenamiento.

Como complemento del análisis, en la Figura 28 se muestra de forma gráfica: el error porcentual de la media absoluta simétrica obtenido con cada técnica durante el entrenamiento en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores menores.

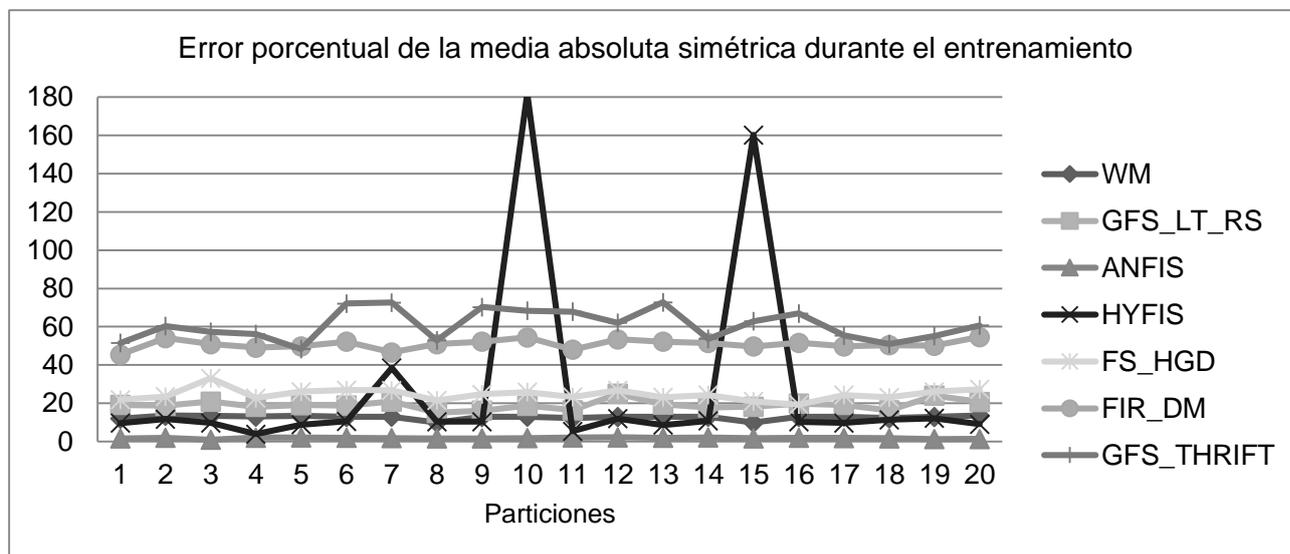


Figura 28. Error porcentual de la media absoluta simétrica obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de entrenamiento.

**Experimento 13:** Comparar las siete técnicas respecto a los resultados en la clasificación del conjunto de casos de prueba en cuanto a la variable: error porcentual de la media absoluta simétrica.

Objetivo: Determinar si existen diferencias significativas en los resultados de clasificar la base de datos aplicando las siete técnicas, respecto al error porcentual de la media absoluta simétrica durante las pruebas. En caso de existir diferencias significativas determinar cuáles técnicas reportan los mejores resultados y cuáles los peores.

Se aplica el Test de Friedman comparando las muestras obtenidas de aplicar los algoritmos del 1 al 7 sobre el conjunto de casos de prueba y respecto a la variable: error porcentual de la media absoluta simétrica. Como resultado se encuentran diferencias significativas respecto a los algoritmos y se aplica el Test de Wilcoxon convenientemente para comparar los algoritmos dos a dos, determinando finalmente los siguientes grupos: Grupo1: ANFIS; Grupo2: WM; Grupo3: GFS\_LT\_RS; Grupo4: FS\_HGD; Grupo5: HYFIS; Grupo6: FIR\_DM; Grupo7: GFS\_THRIFT.

Se puede afirmar que el algoritmo ANFIS obtiene resultados significativamente mejores en cuanto al error porcentual de la media absoluta simétrica durante las pruebas.

Como complemento del análisis, en la Figura 29 se muestra de forma gráfica: el error porcentual de la media absoluta simétrica obtenido con cada técnica durante las pruebas en las 20 ejecuciones realizadas. Se puede apreciar que ANFIS obtiene los valores menores.

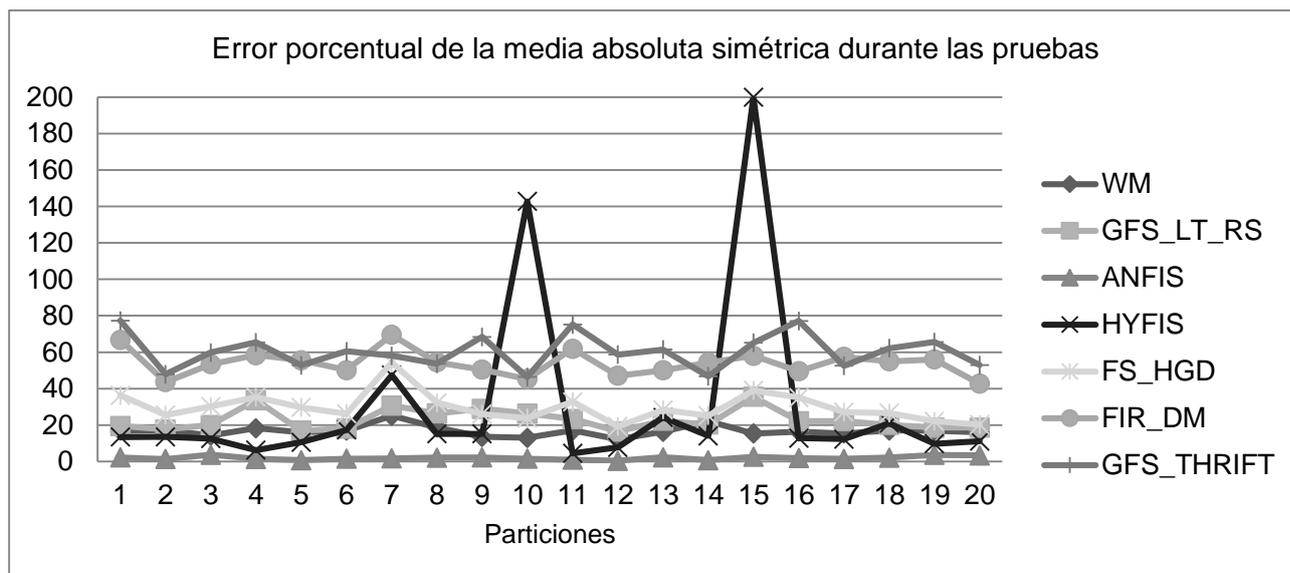


Figura 29. Error porcentual de la media absoluta simétrica obtenido con cada técnica durante las 20 ejecuciones sobre el conjunto de datos de prueba.

Finalmente se pudo constatar que la técnica ANFIS genera el mejor sistema para la evaluación de proyectos puesto que alcanza los mejores resultados respecto a la correlación de las variables: menores errores, mayor cantidad de clasificaciones correctas y menor cantidad de falsos positivos y negativos.

### 3.2.3 Sistema de inferencia borroso obtenido

Al aplicar las siete técnicas se obtuvo como mejor sistema de inferencia borroso el generado con ANFIS. A continuación se muestra la estructura del sistema de evaluación obtenido, para lo cual se especifican sus 28 reglas borrosas.

- R1: Si IRE es Medio y IRP es Medio y ICD es Alto y IRRH es Medio y IREF es Medio entonces EP es Medio
- R2: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Alto entonces EP es Alto
- R3: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Alto entonces EP es Medio
- R4: Si IRE es Bajo y IRP es Bajo y ICD es Bajo y IRRH es Bajo y IREF es Bajo entonces EP es Bajo
- R5: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Medio entonces EP es Medio
- R6: Si IRE es Bajo y IRP es Medio y ICD es Alto y IRRH es Medio y IREF es Bajo entonces EP es Medio
- R7: Si IRE es Medio y IRP es Medio y ICD es Alto y IRRH es Bajo y IREF es Bajo entonces EP es Bajo
- R8: Si IRE es Bajo y IRP es Alto y ICD es Alto y IRRH es Bajo y IREF es Bajo entonces EP es Bajo
- R9: Si IRE es Alto y IRP es Medio y ICD es Alto y IRRH es Medio y IREF es Medio entonces EP es Bajo
- R10: Si IRE es Medio y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Medio entonces EP es Bajo
- R11: Si IRE es Medio y IRP es Medio y ICD es Alto y IRRH es Bajo y IREF es Alto entonces EP es Bajo
- R12: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Bajo entonces EP es Alto
- R13: Si IRE es Medio y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Bajo entonces EP es Medio
- R14: Si IRE es Bajo y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Bajo entonces EP es Medio

- R15: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Bajo y IREF es Alto entonces EP es Bajo  
 R16: Si IRE es Bajo y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Bajo entonces EP es Alto  
 R17: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Bajo entonces EP es Medio  
 R18: Si IRE es Bajo y IRP es Bajo y ICD es Alto y IRRH es Bajo y IREF es Bajo entonces EP es Bajo  
 R19: Si IRE es Medio y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Alto entonces EP es Medio  
 R20: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Medio entonces EP es Alto  
 R21: Si IRE es Bajo y IRP es Bajo y ICD es Bajo y IRRH es Bajo y IREF es Alto entonces EP es Bajo  
 R22: Si IRE es Bajo y IRP es Alto y ICD es Alto y IRRH es Medio y IREF es Alto entonces EP es Medio  
 R23: Si IRE es Bajo y IRP es Bajo y ICD es Bajo y IRRH es Medio y IREF es Alto entonces EP es Bajo  
 R24: Si IRE es Bajo y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Alto entonces EP es Alto  
 R25: Si IRE es Alto y IRP es Alto y ICD es Alto y IRRH es Bajo y IREF es Bajo entonces EP es Bajo  
 R26: Si IRE es Bajo y IRP es Bajo y ICD es Bajo y IRRH es Bajo y IREF es Medio entonces EP es Bajo  
 R27: Si IRE es Medio y IRP es Medio y ICD es Alto y IRRH es Medio y IREF es Alto entonces EP es Medio  
 R28: Si IRE es Medio y IRP es Alto y ICD es Alto y IRRH es Alto y IREF es Medio entonces EP es Alto

Para comprobar el sistema propuesto para la evaluación de la ejecución de proyectos y el sistema anterior construido según el juicio de los expertos, se realiza un último experimento de validación.

**Experimento 14:** Comparar el sistema obtenido con ANFIS y el sistema estático anterior respecto a los resultados en la clasificación de proyectos, en cuanto a las variables: porcentaje de clasificaciones correctas (%CC), porcentaje de falsos positivos (%FP), porcentaje de falsos negativos (%FN), y errores (MSE, RMSE y SMAPE).

En la Tabla 9 se muestra la media de los valores obtenidos para las variables de los experimentos entre las muestras tomadas antes y después de aplicado el sistema.

Tabla 9. Diferencia entre los sistemas antes y después de aplicada la propuesta.

Sistema para la evaluación de proyectos	%CC	%FP	%FN	MSE	RMSE	SMAPE
Sistema estático (Antes)	47,49	17,64	32,84	0,65	0,80	65,35
Sistema propuesto según ANFIS (Después)	95,66	0	2,94	0,03	0,14	2,35

Al realizar un análisis de los resultados mostrados en la Tabla 9 se observa la diferencia en cada una de las variables con respecto a ambos sistemas borrosos. Con ello se evidencia una mejora del sistema propuesto con respecto al sistema anterior, el cual es estático y basado en el criterio de los expertos. Se demuestra que al aplicar la técnica ANFIS, existe un incremento en la cantidad de proyectos clasificados correctamente; elemento que sugiere una mejora en la calidad de las clasificaciones de los proyectos de la organización.

Finalmente se realiza una comparación de las herramientas de gestión de proyectos estudiadas en el capítulo 1 (epígrafe 1.3.1), incluyendo la herramienta Xedro-GESPRO 14.05 para la cual se le propone la inclusión del sistema desarrollado (ver Anexo 2). La tabla comparativa muestra como Xedro-GESPRO 14.05 abarca las diferentes áreas de conocimiento y además incluye el tratamiento de la incertidumbre y el aprendizaje automático, permitiendo elevar su robustez, competitividad y valor agregado.

Una vez analizados los resultados del sistema generado con la mejor variante, se procede con el análisis social y económico de la propuesta.

### 3.3 Valoración social y económica

Con la integración del sistema propuesto en la Suite de Gestión de Proyectos Xedro-GESPRO, se beneficiarán en el entorno de la UCI un total de 14 centros de desarrollo de tecnologías de la información, donde se gestionan aproximadamente 200 proyectos anualmente. Además, son potenciales favorecidas las empresas: XETID, COPEXTEL, CALISOFT y ALBET, las cuales utilizan la herramienta Xedro-GESPRO. En la Tabla 10 se muestra la distribución de proyectos por centros

de la UCI que utilizan esta herramienta, clasificados según el tipo de cliente (datos tomados en junio de 2014 por la Dirección General de Producción UCI).

Tabla 10. Proyectos potenciales beneficiarios con la aplicación de la propuesta en la UCI.

<b>Centro</b>	<b>Desarrollo Red UCI</b>	<b>Exportación</b>	<b>Nacional</b>	<b>Informatización UCI</b>	<b>Sub-proyecto</b>	<b>Total</b>
CDAE	3	0	4	1	6	<b>14</b>
CEDIN	2	3	0	0	2	<b>7</b>
CEGEL	0	0	6	0	16	<b>22</b>
CEIGE	1	0	8	2	11	<b>22</b>
CESIM	4	2	10	0	8	<b>24</b>
CESOL	7	0	1	2	0	<b>10</b>
CIDI	1	0	10	2	0	<b>13</b>
CISED	2	1	1	2	5	<b>11</b>
DATEC	2	3	10	0	30	<b>45</b>
FORTES	4	0	1	1	0	<b>6</b>
GEYSED	3	1	7	0	0	<b>11</b>
ISEC	3	1	6	1	0	<b>11</b>
TLM	1	1	4	1	0	<b>7</b>
VERTEX	4	3	4	0	1	<b>12</b>
<b>Total</b>	<b>37</b>	<b>15</b>	<b>72</b>	<b>12</b>	<b>79</b>	<b>215</b>

Los nombres de los centros mostrados en la Tabla 10 son:

- CDAE - Centro de Consultoría y Desarrollo de Arquitecturas Empresariales
- CEDIN - Centro de Informática Industrial
- CEGEL - Centro de Gobierno Electrónico
- CEIGE - Centro de Informatización de Entidades
- CESIM - Centro de Informática Médica
- CESOL - Centro Gestión de Información y Tecnologías Libres
- CIDI - Centro de Investigación y Desarrollo de Internet
- CISED - Centro de Identificación y Seguridad Digital
- DATEC - Centro Tecnologías de Gestión de Datos
- FORTES - Centro de Tecnologías para la Formación
- GEYSED - Centro de Geoinformática y Señales Digitales
- ISEC - Centro de Informatización de la Seguridad Ciudadana
- TLM - Centro de Telemática
- VERTEX - Entornos Interactivos 3D

La toma de decisiones con ayuda de Xedro-GESPRO 14.05 se realiza de forma íntegra, ágil y cómoda, elevando la calidad de vida de sus usuarios. Resultan favorecidos un promedio de 4000 usuarios de la herramienta, con diferentes niveles de especialización y roles. Estos usuarios son identificados como potenciales beneficiarios de la propuesta de solución.

El sistema permitirá incidir en la disminución de eventualidades negativas desde el punto de vista de las evaluaciones incorrectas e insatisfacciones en la clasificación de los proyectos. Esto se traduce en una disminución de tiempos empleados por los usuarios para tomar decisiones objetivas en los escenarios de aplicación de la propuesta. Además, permitirá ingresos monetarios a la economía nacional por concepto de ventas del módulo de toma de decisiones del sistema Xedro-GESPRO, donde se incluye la biblioteca AnalysisPro.SC.PMC.

Su importancia en el orden económico se encuentra reflejada en la implementación práctica de los Lineamientos de la Política Económica y Social del Estado (PCC, 2011) en el panorama cubano actual, como contribución a la gestión del conocimiento y al control de proyectos. Específicamente se refiere a los lineamientos 8, 12 y 20 del capítulo I Modelo de Gestión Económica. Estos se relacionan con la elevación de la eficiencia, eficacia y el control en el empleo del personal, los recursos materiales y financieros, el fortalecimiento del sistema de control interno, para lograr los resultados esperados en cuanto al cumplimiento del plan con eficiencia y orden.

Otra de las ventajas del sistema propuesto radica en el uso del *software* libre como requisito para alcanzar la soberanía tecnológica, lo cual ayuda a garantizar el desarrollo nacional de manera integral y sustentable. El entorno computacional y funcionalidades desarrolladas sobre tecnologías de *software* de código abierto promueven y reflejan estos preceptos. Esto implica las siguientes ventajas para la herramienta de gestión: dominio pleno de las funcionalidades, detección de errores y correcciones a tiempo, mejora continua sobre la base del desarrollo colaborativo.

El análisis económico se basa en el costo de desarrollo de la investigación e implementación del sistema, teniendo en cuenta solamente el salario de las personas implicadas (estuvieron inmersos en esta actividad tres profesores). En la Tabla 11 se resume el costo aproximado por etapas de la investigación; su monto total asciende a 8847,60 pesos cubanos (CUP).

Tabla 11. Costo asociado al desarrollo del sistema propuesto.

Etapas	Trabajadores			Tiempo Dedicado			Costo total (CUP)
	Profesor	Fondo salarial	Tarifa horaria	Horas al mes	Cantidad de meses	Total de horas	
Revisión teórica	1	\$ 1024,10	5,33	40	6	240	\$ 1279,20
	2	\$ 1100,10	5,72	25	6	150	\$ 858,00
Implementación y validación	1	\$ 1024,10	5,33	80	8	640	\$ 3411,20
	2	\$ 1100,10	5,72	60	8	480	\$ 2745,60
	3	\$ 1330,00	6,92	10	8	80	\$ 553,60
Total							\$ 8847,60

El impacto económico de la propuesta de solución está estrechamente relacionado con el resultado de su implementación y el bajo costo de su aplicación. Su puesta en práctica no conlleva la utilización de recursos económicos significativos, por cuanto su aplicación se reduce al tiempo que puede tomar ejecutar el aprendizaje, y que tras la adquisición de experiencia, disminuye con cada iteración. El tiempo necesario para la aplicación del sistema propuesto (a la hora de evaluar los proyectos) se encuentra en el orden de los milisegundos. El tiempo requerido para el aprendizaje del sistema aunque está en el orden de las horas o días, no afecta el funcionamiento de la herramienta de gestión de proyectos, debido a que dicho proceso sólo se realiza una vez al año, semestre o cuando los expertos de la organización lo consideren necesario. Mientras se realiza el aprendizaje del sistema se pueden seguir evaluando los proyectos con el sistema actual, lo cual implica que no se detenga la herramienta de gestión.

### 3.4 Conclusiones del capítulo

En el presente capítulo se muestra la síntesis de aplicación del sistema propuesto integrado a Xedro-GESPRO; se presentan los resultados experimentales de validación y una valoración social y económica, dando cumplimiento así al tercer objetivo específico de la investigación. Se arriba a las siguientes conclusiones:

- 1) Las variables propuestas para medir la calidad de la clasificación de proyectos, así como la aplicación de los pre-experimentos correspondientes, permitieron seleccionar la técnica ANFIS como la más idónea para la generación del sistema evaluador de proyectos.
- 2) El empleo de técnicas de *soft computing* para la evaluación de proyectos, incrementa la adaptabilidad de las herramientas de apoyo a la toma de decisiones ante los cambiantes estilos

de gestión provocados por la madurez alcanzada durante la mejora continua de las organizaciones.

- 3) La aplicación de técnicas de *soft computing* en herramientas informáticas para la gestión de proyectos como Xedro-GESPRO es un aporte novedoso, permitiendo elevar la calidad, competitividad y valor agregado de las mismas.
- 4) La propuesta de solución permitirá promover la gestión del conocimiento al ser puesto a disposición de una amplia gama de usuarios: un promedio de 4000 como clientes de la herramienta Xedro-GESPRO.

## **CONCLUSIONES**

La presente investigación arroja como principal resultado: el desarrollo de un sistema basado en técnicas de *soft computing* para la evaluación de la ejecución de proyectos, a través del cual se puede arribar a las siguientes conclusiones:

- 1) Las fuentes de datos relacionadas con la gestión de proyectos presentan imprecisión, vaguedad en los conceptos e incertidumbre en la información, situación donde el uso de técnicas de *soft computing* aporta buenos resultados para la clasificación.
- 2) El sistema propuesto posibilita preservar el conocimiento de los expertos en las organizaciones, teniendo en cuenta el conocimiento implícito almacenado y la capacidad de generalizarlo en un lenguaje lo suficientemente cercano al razonamiento humano.
- 3) Las ventajas que aporta el sistema propuesto están asociadas con la mejora continua de las herramientas para la toma de decisiones que se adapten al nivel de madurez alcanzado por la organización.
- 4) El empleo de técnicas de aprendizaje automático para la evaluación de proyectos, incrementa la adaptabilidad de las herramientas informáticas en las organizaciones con cambiantes estilos de gestión provocados por la madurez alcanza durante la mejora continua.
- 5) El desarrollo del sistema propuesto se logró con un costo mínimo de \$ 8847,60 pesos cubanos (CUP); este permite elevar la calidad, adaptabilidad y competitividad de la herramienta de gestión de proyectos, dotándola de un alto valor agregado.
- 6) A diferencia de las soluciones más cercanas consultadas en la literatura, el sistema propuesto efectúa el tratamiento de la incertidumbre en la información y la integración con herramientas informáticas para la gestión de proyectos, empleando tecnologías de *software* libre.

## **RECOMENDACIONES**

Los resultados obtenidos no agotan las posibilidades de investigación y mejora referente al campo de acción, por lo que se derivan las siguientes recomendaciones del trabajo:

- 1) Actualizar la base de casos de los proyectos y ejecutar el sistema propuesto cada seis meses para lograr la adaptación del mecanismo de evaluación según la mejora continua alcanzada por la organización.
- 2) Validar el sistema utilizando casos de entrenamiento que manejen la información referente a las áreas de conocimiento: logística y costo.
- 3) Adicionar al sistema otras técnicas de *soft computing* y comparar sus resultados con el objetivo de seleccionar la variante que mejor se ajuste a la organización.
- 4) Aplicar técnicas de aprendizaje no supervisado que permitan construir y ajustar el sistema de evaluación de proyectos sin requerir la intervención de expertos humanos en la clasificación de los casos de entrenamiento.
- 5) Extender la aplicación del sistema propuesto a otros escenarios orientados a la producción por proyectos ajenos a la producción de *software*, como pueden ser el caso de la construcción, medicina, educación, comercio y economía.

## REFERENCIAS BIBLIOGRÁFICAS

- ABELARDO, F. N., PIÑERO, P. Y., LUGO, J. A. & MENÉNDEZ, J. 2014. Cuadro de mando para la toma de decisiones en control y seguimiento de proyectos en Cuba. *I Conferencia Internacional de la Universidad de las Ciencias Informáticas, UCIENCIA 2014*. La Habana. ISBN: 978-959-286-026-1.
- ALBA, E., CHICANO, J. F., COTTA, C., DORRONSORO, B., LUNA, F., LUQUE, G. & NEBRO, A. J. 2004. Metaheurísticas Secuenciales y Paralelas para Optimización de Problemas Complejos. *In: JOYA, G., ATENCIA, M., OCHOA, A. & ALLENDE, S. (eds.) Optimización Inteligente*. Málaga: Servicio de Publicaciones de la Universidad de Málaga.
- ALCALÁ, R., ALCALÁ-FDEZ, J. & HERRERA, F. 2007. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. *IEEE Transactions on Fuzzy Systems*, 15, 616 - 635.
- ALCALÁ, R. & HERRERA, F. 2004. Genetic Tuning on Fuzzy Systems Based on the Linguistic 2-Tuples Representation. *XII Congreso Español sobre Tecnologías y Lógica Fuzzy*. Jaén: Universidad de Jaén.
- ALDANA, M. & ET.AL. 2013. Experiencias en la integración de sistemas de gestión de proyectos a nivel de datos con PostgreSQL. *COMPUMAT*. Universidad de las Ciencias Informáticas, La Habana. ISBN: 978-959-286-022-3.
- ARANGO, A. 2012. *Pronóstico del Índice General de la Bolsa de Valores de Colombia (IGBC) usando modelos de inferencia difusa*. Magister en Ingeniería Administrativa Magister, Universidad Nacional de Colombia, Medellín, Colombia.
- BACZYNSKI, M., BELIAKOV, G., SOLA, H. B. & PRADERA, A. 2013. Advances in Fuzzy Implication Functions. *In: HEIDELBERG, S. (ed.)*. Poland. ISBN: 978-3-642-35676-6.
- BATH, M. 2010. Project Classification Using Soft Computing. *International Conference on Advances in Computing, Control & Telecommunication Technologies*. IEEE Computer Society, Washington, USA. ISBN: 978-1-4244-5321-4.
- BELLO, R. & VERDEGAY, J. L. 2010. Los conjuntos aproximados en el contexto de la soft computing. *Revista Cubana de Ciencias Informáticas*, 4, 5-24. ISSN: 1994-1536.
- BERENJI, H. R. 2012. Learning and tuning of fuzzy rules. *In: NGUYEN, H. T. & SUGENO, M. (eds.) Fuzzy Systems: Modeling and Control*.
- BLANCO, E. L. J. 2011. La informática en la dirección de empresas. *In: DACOSTA, Z. (ed.)*. La Habana, Cuba: Editorial Félix Varela. ISBN: 978-959-07-1629-4.
- BONISSONE, P. P., CHEN, Y. U. T. O., GOEBEL, K. & KHEDKAR, P. S. 1999. Hybrid soft computing systems: Industrial and comercial applications. *Proceedings of the IEEE*, 87, 1641-1667.
- BOTÍA-BLAYA, J. A. 2003. *Sistemas Difusos* [Online]. Murcia: Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia. Available: [ants.dif.um.es/staff/juanbot/ml/files/20022003/fuzzy.pdf](http://ants.dif.um.es/staff/juanbot/ml/files/20022003/fuzzy.pdf).
- CASTELLANO, G., CASTIELLO, C., FANELLI, A. M. & MENCAR, C. 2005. Knowledge discovery by a neuro-fuzzy modeling framework. *Fuzzy Sets and Systems*, 149, 187-207. ISSN: 0165-0114.
- CASTRO DÍAZ-BALART, F. 2004. *Ciencia, Tecnología y Sociedad. Hacia un desarrollo sostenible en la Era de la Globalización*, La Habana, Instituto Cubano del Libro, Editorial Científico-Técnica. ISBN: 959-05-0342-X.
- CASTRO RUZ, R. 2011. Informe Central al VI Congreso del Partido Comunista de Cuba. *Consejo de Estado*. La Habana.
- CERTA, A., ENEA, M. & GIALLANZA, A. 2010. A synthetic measure for the assessment of the project performance. *Business Performance Measurement and Management*, 167-180.
- CONSEJO DE MINISTROS 2015. Decreto No.327 Reglamento del proceso inversionista. *In: MINISTERIO DE JUSTICIA (ed.) GO\_X\_5\_2015*. La Habana.
- CONWAY, J. E. 2015. *PL/R - R Procedural Language for PostgreSQL*. [En línea] [Online]. Available: [www.joeconway.com/plr/](http://www.joeconway.com/plr/).
- CÓRDON, O., DEL JESÚS, M. J., HERRERA, F. & LOZANO, M. 1999. MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach. *International Journal of Intelligence Systems*, 14, 1123-1153.
- COTTA, C. & MOSCATO, P. 2004. Evolutionary Computation: Challenges and Duties. *In: MENON, A. (ed.) Frontiers of Evolutionary Computation*. Boston MA: Kluwer Academic Publishers. ISBN: 1-4020-7782-3.
- COX, E., TABER, R., O'HAGAN, M. & O'HAGEN, M. 1998. *The Fuzzy Systems Handbook, Second Edition*, New York, AP Professional, Paperback.

- CUBAINDUSTRIA. 2014. *Normas Cubanas Online* [Online]. Available: [www.nconline.cubaindustria.cu/](http://www.nconline.cubaindustria.cu/).
- CHANG, J. L. & HSING, C. Y. 1999. A Heuristic Error-Feedback Learning Algorithm for Fuzzy Modeling. *IEEE Transaction Systems Man & Cybernetics*, 29, 686-691. ISSN: 0018-9472.
- DELGADO, R. 2011. La Dirección Integrada de Proyecto como Centro del Sistema de Control de Gestión en el Ministerio del Poder Popular para la Comunicación y la Información. In: CENDA (ed.). Caracas, Venezuela. ISBN: 9591602513.
- DWEIRI, F. & KABLAN, M. 2006. Using fuzzy decision making for the evaluation of the project management internal efficiency. 42. ISSN: 0167-9236.
- EDEKI, C. 2013. Agile Unified Process. *International Journal of Computer Science and Mobile Applications*, 1, 13-17. ISSN: 2321-8363.
- ESCOBAR, M. & ET.AL. 2013. Limpieza de datos: aplicación al sistema de gestión de proyectos GESPRO. *Serie Científica de la Universidad de las Ciencias Informáticas*, 6.
- FERNÁNDEZ, A., LÓPEZ, V., JESUS, M. J. D. & HERRERA, F. 2015. Revisiting Evolutionary Fuzzy Systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*.
- FREE SOFTWARE FOUNDATION. 1991. *GNU General Public License* [Online]. Boston. Available: [www.r-project.org/COPYING](http://www.r-project.org/COPYING).
- FREEMAN, J. A. & SKAPURA, D. M. 1991. *Neural Networks - Algorithms, Applications and Programming Techniques*. University of Houston at Clear Lake: Addison Wesley.
- GAO, H. 2010. A fuzzy-ANP approach to project management performance evaluation indices system. *International Conference on Logistics Systems and Intelligent Management*. IEEE. ISBN: 978-1-4244-7331-1.
- GOLBERG, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- GÓMEZ-SKARMETA, A. F. & JIMÉNEZ, F. 1999. Fuzzy modeling with hybrid systems. *Fuzzy Sets and Systems*, 104, 199–208. ISSN: 0165-0114.
- GONZÁLEZ, A. & PÉREZ, R. 1999. SLAVE: a genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems*, 7, 176-191.
- GOVINDARAJAN, A. 2014. A Novel Framework for Evaluating the Software Project Management Efficiency—An Artificial Intelligence Approach. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 12, 7054-7058.
- GRAUEL, A. & LUDWIG, L. 1999. Construction of Differentiable Membership Functions. *Fuzzy Sets and Systems*, 101, 219-225. ISSN: 0165-0114.
- HERACLES 2004. Modelado dinámico y aprendizaje automático aplicado a la gestión de proyectos software. *Revista de Procesos y Métricas de las Tecnologías de la Información*, 1, 21-28. ISSN: 1698-2029.
- HERRERA, F. 2004. Sistemas difusos evolutivos. *XII Congreso Español sobre Tecnologías y Lógica Fuzzy*. Jaén: Universidad de Jaén.
- HERRERA, F. & MARTÍNEZ, L. 2000. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Syst.*, 8, 746-752.
- ISHIBUCHI, H., NAKASHIMA, T. & MURATA, T. 2001. Three-objective genetics based machine learning for linguistic rule extraction. *Information Science*, 136, 109-133.
- ISHIBUCHI, H., NOZAKI, K., TANAKA, H., HOSAKA, Y. & MATSUDA, M. 1994. Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets and Systems*, 64, 129-144. ISSN: 0165-0114.
- ISO. 2003. *Quality Management systems - Guidelines for quality management in projects*.
- JACOBSON, I., BOOCH, G. & RUMBAUGH, J. 2000. *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- JAMEI, M., MAHFOUF, M. & LINKENS, D. A. 2004. Elicitation and fine-tuning of fuzzy control rules using symbiotic evolution. *Fuzzy Sets and Systems*, 147, 57–74. ISSN: 0165-0114.
- JANG, J.-S. R. 1993. ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. *IEEE Transaction Systems Man & Cybernetics*, 23, 665-685. ISSN: 0018-9472.
- JANG, J. S., SUN, C. & MIZUTANI, E. 1997. *Neuro fuzzy and Soft Computing*, New York, Prentice Hall.
- JIMÉNEZ, G. 2013. Extensiones para el control de la ejecución de proyectos basadas en el análisis de la dimensión geográfica. *COMPUMAT*. Universidad de las Ciencias Informáticas, La Habana. ISBN: 978-959-286-022-3.
- KDNUGGETS. 2014. What Analytics, Data Mining, Data Science software/tools you used in the past 12 months for a real project Poll. *Data Mining Community's Top Resource* [Online]. Available: [www.kdnuggets.com/polls/index.html](http://www.kdnuggets.com/polls/index.html).

- KELEMEN, A., LIANG, Y. & FRANKLIN, S. 2002. A comparative study of different machine learning approaches for decision making. *Recent Advances in Simulation, Computational Methods and Soft Computing*.
- KIM, J. & KASABOV, N. 1999. HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks*, 12, 1301 - 1319.
- KRUSE, R., BERTHOLD, M. R., MOEWES, C., GIL, M. Á., GRZEGORZEWSKI, P. & HRYNIEWICZ, O. (eds.) 2013. *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, ISBN: 978-3-642-33041-4.
- LARRAÑAGA, P. & LOZANO, J. A. 2001. Estimation of Distribution Algorithms EDA and probabilistic graphical models. *Estimation Distribution Algorithms. A new tool for evolutionary optimization*. Kluwer Academic Publishers.
- LARRAÑAGA, P., LOZANO, J. A. & MÜHLENBEIN, H. 2003. Estimation of Distribution Algorithms Applied To Combinatorial Optimization Problems. *Revista Iberoamericana de Inteligencia Artificial*, 19, 149-168.
- LIU, Y., ZHENG, D. & WANG, E. 2013. Research on Evaluation of Project Management Maturity Model Based on BP Neural Network. *Advances in information Sciences and Service Sciences(AISS)*, 5, 693-701.
- LÓPEZ, E. 2008. *Algoritmos para la Detección y Cuantificación de Defectos en Manzanas por Inspección Visual*. Maestría en Ciencias Computacionales MsC, Centro Nacional de Investigación y Desarrollo Tecnológico, México.
- LUGO, J. A., TORRES, S., Y., P. P., GARCÍA, A. M., DELGADO, R. & ARAGÓN, N. 2014. Control automatizado de proyectos en Cuba: un acercamiento. *I Conferencia Científica Internacional UCIENCIA 2014*. Universidad de las Ciencias Informáticas, La Habana, Cuba. ISBN: 978-959-286-026-1.
- LUGO, J. A. G. 2012. *Modelo para el control de la ejecución de proyectos basado en indicadores y lógica borrosa*. Máster MsC, Universidad de las Ciencias Informáticas, La Habana.
- MAHING, T. & MUHLENBEIN, H. 2001. Comparing the adaptive Boltzmann selection schedule SDS to truncation selection. *Third International Symposium on Adaptive Systems, Evolutionary Computation and Graphical Models ISAS 2001*.
- MAKREHCHI, M., BASIR, O. & KAMEL, M. 2003. Generation of Fuzzy Membership Functions Using Information Theory Measures and Genetic Algorithm. *International Fuzzy Systems Association World Congress IFSA 2003*.
- MAMDANI, E. H. 1977. Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26, 1182–1191.
- MARTÍN, B. D. B. 2001. *Redes Neuronales y Sistemas Difusos*. Zaragoza. España.
- MATHUR, Y. K. & NAND, A. 2014. Soft Computing Techniques and its Impact in Data Mining. *International Journal of Emerging Technology and Advanced Engineering*, 4, 658-662.
- MATHWORKS. 2014. MATLAB, Fuzzy Logic Toolbox, User's Guide. Available: [www.mathworks.com](http://www.mathworks.com).
- MEWADA, K. M., SINHAL, A. & VERMA, B. 2013. Adaptive Neuro-Fuzzy Inference System (ANFIS) Based Software Evaluation. *IJCSI International Journal of Computer Science*, 10, 244-250. ISSN: 1694-0784.
- MIC 2011. Estrategia de exportación de la industria informática. La Habana: Ministerio de la Informática y las Comunicaciones.
- MITCHELL, T. 1997. *Machine Learning*, Washington, New York, McGraw-Hill Science / Engineering / Math. ISBN: 0070428077.
- MORENO, A., ARMENGLO, E., BÉJAR, J., BELANCHE, L., CORTÉS, U., GAVALDÁ, R., JIMENO, J. M., LÓPEZ, B., MARTÍN, M. & SÁNCHEZ, M. 1994. *Aprendizaje Automatico*, Barcelona, UPC, Universitat Politècnica de Catalunya. ISBN: 84-7653-460-4.
- MUHLENBEIN, H., MAHING, T. & OCHOA, A. 1999. Schemata, distributions and graphical models in evolutionary optimization. *Heuristics*, 5, 213–247.
- NAUCK, D. 2000. *Data Analysis with Neuro - Fuzzy Methods*. Habilitationsschrift zur Erlangung des akademischen Grades doctor rerum naturalium habilitatus, Otto-von-Guericke, Magdeburg.
- NAUCK, D. 2005. Learning Algorithms for Neuro-Fuzzy Systems. *Do Smart Adaptive Systems Exist? : Springer Berlin Heidelberg*. ISBN: 978-3-540-32374-7.
- NAUCK, D., KLAWONN, F. & KRUSE, R. 1999. *Foundations of Neuro-Fuzzy systems*, New York, Wiley.
- NAUCK, D. & KRUSE, R. 1998. How the Learning of Rule Weights Affects the Interpretability of Fuzzy Systems. *In: SYSTEMS1998*, I. P. I. I. C. O. F. (ed.). Anchorage.
- NOMURA, H., HAYASHI, I. & WAKAMI, N. 1992. A Learning Method of Fuzzy Inference Rules by Descent Method. *IEEE International Conference on Fuzzy Systems*. San Diego, CA.
- OCHOA, A., MUHLENBEIN, H. & SOTO, M. 2000. A Factorized Distribution Algorithm using single connected Bayesian networks. *Lecture Notes in Computer Sciences*, 1917, 787–796.

- PAL, S. K., MITRA, S. & MITRA, P. 2001. Rough Fuzzy MLP: Modular Evolution, Rule Generation and Evaluation. *IEEE Trans. on Knowledge and Data Engineering*, 34, 56-69. ISSN: 10414347.
- PCC 2011. Lineamientos de la Política Económica y Social del Partido y la Revolución. La Habana.
- PEDRYCZA, W. & REFORMATATA, M. 2005. Genetically optimized logic models. *Fuzzy Sets and Systems*, 150, 351-371. ISSN: 0165-0114.
- PELIKAN, M., GOLBERG, D. & TSUTSUI, S. 2002. Combining the Strengths of the Bayesian Optimization Algorithm and Adaptive Evolution Strategies. ILLIGAL Technical Reports & Publications series on Genetic Algorithms and Evolutionary Computation.
- PÉREZ, I. & ET.AL. 2013. Aprendizaje de reglas borrosas para la toma de decisiones en la gestión de proyectos. *IV Congreso Iberoamericano de Ingeniería de Proyectos*. Chile: RIIPRO.
- PIÑERO, P. P. 2005. *Un modelo para el aprendizaje y la clasificación automática basado en técnicas de soft-computing*. Doctor en Ciencias Técnicas PhD, Universidad Central de las Villas, La Habana.
- PIÑERO, P. Y. 2013. GESPRO. Paquete para la gestión de proyectos. *Revista Nueva Empresa* [Online], 9. ISSN: 1682-2455.
- PMI 2013. A Guide to the Project Management Body of Knowledge (PMBOK). Pennsylvania: Project Management Institute. ISBN: 1-930699-73-5.
- PRINCE2 2009. Managing Successful Projects with Prince2. In: TSO (ed.). Londres.
- R-CRAN. 2015. *R-Cran Contributed Packages*. [En línea] [Online]. Available: [cran.r-project.org/web/packages/](http://cran.r-project.org/web/packages/).
- R-FOUNDATION. 2015. *The R Project for Statistical Computing* [Online]. Available: [www.r-project.org/](http://www.r-project.org/).
- R CORE TEAM. 2015. *The Comprehensive R Archive Network*. [En línea] [Online]. Available: [cran.r-project.org/](http://cran.r-project.org/).
- RAMOS, K. 2011. Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas*, 5, 1-16. ISSN: 1994-1536.
- RAPIDMINER INC. 2015. *Rapidminer*. [En línea] [Online]. Available: [rapidminer.com](http://rapidminer.com).
- RIZA, L. S., BERGMEIR, C., HERRERA, F. & BENITEZ, J. M. 2015. FRBS: Fuzzy Rule-based Systems for Classification and Regression Tasks.
- RSTUDIO. 2015. *About RStudio*. [Online]. Boston, Massachusetts. Available: [www.rstudio.com](http://www.rstudio.com).
- SÁNCHEZ, L. & OTERO, J. 2004. A fast genetic method for inducing descriptive fuzzy models. *Fuzzy Sets and Systems*, 141, 33-46. ISSN: 0165-0114.
- SANTANA, R., OCHOA, A. & SOTO, M. 2001. The mixture of trees Factorized Distribution Algorithms. *Genetic and Evolutionary Computation Conference GECCO 2001*.
- SANTIESTEBAN, A. M., LUGO, J. A., MARÍN, J., ROJAS, C., PIÑERO, P. Y. & DELGADO, R. 2014. Módulo de reportes de apoyo a la toma de decisiones en la gestión de proyectos. *I Conferencia Internacional de la Universidad de las Ciencias Informáticas, UCIENCIA 2014*. ISBN: 978-959-286-026-1.
- SEI 2010. CMMI for Dev. v1.3. Ed. Carnegie Mellon University. Pittsburgh.
- SEPTEN, L. 2014. *FRBS: Fuzzy rule-based systems* [Online]. Available: [sci2s.ugr.es/dicits/software/FRBS](http://sci2s.ugr.es/dicits/software/FRBS).
- SOTO, M. 2003. *Algoritmo con Distribución Factorizada basada en Poliárboles (PADA)*. PhD, Instituto de Cibernética Matemática y Física, Ciudad de la Habana.
- SPIEGELA, D. & SUDKAMPB, T. 2003. Sparse data in the evolutionary generation of fuzzy models. *Fuzzy Sets and Systems*, 138, 363-379. ISSN: 0165-0114.
- STANG, D. B. 2013. IT Project & Portfolio Management Magic Quadrant. *Gartner Inc.* [Online]. Available: [www.gartner.com/technology/reprints.do?id=1-1N9Q4HL&ct=131121&st=sb](http://www.gartner.com/technology/reprints.do?id=1-1N9Q4HL&ct=131121&st=sb).
- SUGENO, M. 1977. Fuzzy Measures and fuzzy integrals: A Survey, Fuzzy Automata and Decision Processes.
- TAKAGI, T. & SUGENO, M. 1985. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15, 116-132. ISSN: 0018-9472.
- THRIFT, P. 1991. Fuzzy logic synthesis with genetic algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA91)*. San Diego (United States of America).
- TORRES, K. M., TORRES, S. & PIÑERO, P. Y. 2014. Estrategia de formación integrada en Gestión de Proyectos Informáticos. *I Conferencia Científica Internacional UCIENCIA 2014*. Universidad de las Ciencias Informáticas, La Habana. ISBN: 978-959-286-026-1.
- TORRES, S., ESCOBAR, M. E., LUGO, J. A. & PIÑERO, P. Y. 2013. Limpieza de datos, su aplicación a un sistema de gestión de proyectos: GESPRO. *Congreso Internacional COMPUMAT 2013*. Universidad de las Ciencias Informáticas, La Habana. ISBN: 978-959-286-022-3.
- WAIKATO. 2015. *Weka 3: Data Mining Software in Java* [Online]. Available: [www.cs.waikato.ac.nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/).

- WANG, L.-X. & MENDEL, J. M. 1992. Generating fuzzy rule by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 1414 - 1427. ISSN: 0018-9472.
- WANG, S.-T., YUB, D.-J. & YANG, J.-Y. 2003. Integrating rough set theory and fuzzy neural network to discover fuzzy rules. *Intelligent Data Analysis*, 7, 59–73. ISSN: 1088-467.
- ZADEH, L. 1976. The concept of a linguistic variable and its applications to approximate reasoning. *In*: 9, I. S. (ed.).
- ZADEH, L. 1978. A fuzzy sets as a basis for the theory of possibility. *Fuzzy Sets and Systems*.
- ZADEH, L. 1994. Fuzzy Logic, Neural Networks and Soft Computing. *Communications of the ACM*, 37, 77-84.
- ZANDHUIS, A. & STELLINGWERF, R. 2013. ISO21500: Guidance on project management - A Pocket Guide. Van Haren Publishing.

## ANEXOS

Anexo 1: Resultados de la ejecución de cada algoritmo en las 20 particiones de datos, en cuanto a: cantidad de reglas borrosas generadas (CR); por ciento de clasificaciones correctas en entrenamiento (CC-E) y en prueba (CC-P); cantidad de falsos positivos en entrenamiento (FP-E) y en prueba (FP-P); cantidad de falsos negativos en entrenamiento (FN-E) y en prueba (FN-P); error cuadrático medio en entrenamiento (MSE-E) y en prueba (MSE-P); raíz del error cuadrático medio en entrenamiento (RMSE-E) y en prueba (RMSE-P); error porcentual de la media absoluta simétrica en entrenamiento (SMAPE-E) y en prueba (SMAPE-P).

#	Algoritmos	CR	CC-E	CC-P	FP-E	FP-P	FN-E	FN-P	MSE-E	MSE-P	RMSE-E	RMSE-P	SMAPE-E	SMAPE-P
1	ANFIS1	29	97,24	96,61	0	0	4	2	0,02	0,02	0,13	0,15	1,63	2,16
2	FIR_DM1	11	53,79	32,2	27	17	40	23	0,45	0,67	0,67	0,82	45,40	66,78
3	FS_HGD1	11	75,86	59,32	26	16	9	8	0,22	0,36	0,47	0,60	21,90	36,26
4	GFS_LT_RS1	5	78,62	69,49	27	17	4	1	0,19	0,20	0,44	0,44	19,15	19,50
5	GFS_THRIFT1	49	53,79	32,2	27	17	40	23	0,52	0,77	0,72	0,88	51,51	77,35
6	HYFIS1	35	91,72	86,44	12	7	0	1	0,10	0,13	0,31	0,37	9,68	13,41
7	WM1	31	95,86	91,53	6	5	0	0	0,12	0,17	0,35	0,41	11,97	16,95
8	ANFIS2	30	96,77	97,96	0	0	5	1	0,02	0,01	0,14	0,12	1,93	1,33
9	FIR_DM2	12	45,16	55,1	36	8	49	14	0,54	0,44	0,74	0,66	54,12	43,53
10	FS_HGD2	12	69,68	71,43	34	8	13	6	0,24	0,25	0,49	0,50	23,53	25,50
11	GFS_LT_RS2	8	76,77	81,63	36	8	0	1	0,19	0,17	0,43	0,42	18,71	17,49
12	GFS_THRIFT2	49	45,16	55,1	36	8	49	14	0,60	0,48	0,78	0,69	60,36	47,85
13	HYFIS2	38	89,68	87,76	16	5	0	1	0,12	0,14	0,34	0,37	11,75	13,51
14	WM2	30	94,84	89,8	8	4	0	1	0,14	0,17	0,37	0,42	13,58	17,29
15	ANFIS3	28	98,66	92,73	0	0	2	4	0,01	0,04	0,10	0,19	0,99	3,71
16	FIR_DM3	11	48,32	45,45	32	12	45	18	0,51	0,53	0,71	0,73	50,94	53,39
17	FS_HGD3	11	70,47	76,36	31	12	13	1	0,33	0,30	0,58	0,55	33,12	30,16
18	GFS_LT_RS3	7	75,84	78,18	32	12	4	0	0,21	0,20	0,46	0,45	20,92	20,01
19	GFS_THRIFT3	46	48,32	45,45	32	12	45	18	0,57	0,60	0,76	0,77	57,36	59,75
20	HYFIS3	35	90,6	89,09	14	6	0	0	0,10	0,13	0,31	0,35	9,71	12,55
21	WM3	34	95,3	94,55	7	3	0	0	0,13	0,14	0,36	0,38	13,04	14,22
22	ANFIS4	28	96,62	98,21	0	0	5	1	0,02	0,01	0,14	0,12	2,05	1,41
23	FIR_DM4	11	50	41,07	30	14	44	19	0,49	0,58	0,70	0,76	49,11	58,36
24	FS_HGD4	11	75	62,5	29	14	8	7	0,23	0,35	0,48	0,59	22,78	35,30
25	GFS_LT_RS4	5	75,68	62,5	34	19	2	2	0,18	0,34	0,43	0,58	18,14	33,68
26	GFS_THRIFT4	50	50	41,07	30	14	44	19	0,56	0,65	0,75	0,81	56,21	65,43
27	HYFIS4	33	93,24	89,29	8	6	2	0	0,04	0,06	0,20	0,25	3,86	6,25
28	WM4	29	94,59	91,07	8	4	0	1	0,13	0,18	0,36	0,43	12,88	18,22
29	ANFIS5	30	96,05	100	0	0	6	0	0,02	0,01	0,15	0,08	2,12	0,68
30	FIR_DM5	12	48,68	44,23	31	13	47	16	0,50	0,56	0,71	0,75	49,81	55,59
31	FS_HGD5	12	73,03	69,23	29	13	12	3	0,26	0,30	0,51	0,55	26,09	29,74
32	GFS_LT_RS5	7	78,95	75	31	13	1	0	0,19	0,17	0,44	0,41	19,03	16,94
33	GFS_THRIFT5	50	48,68	44,23	31	13	47	16	0,48	0,53	0,70	0,73	48,39	52,78

#	Algoritmos	CR	CC-E	CC-P	FP-E	FP-P	FN-E	FN-P	MSE-E	MSE-P	RMSE-E	RMSE-P	SMAPE-E	SMAPE-P
34	HYFIS5	38	90,13	88,46	15	5	0	1	0,09	0,10	0,30	0,32	8,78	10,50
35	WM5	33	95,39	92,31	7	4	0	0	0,13	0,16	0,36	0,40	13,00	15,93
36	ANFIS6	30	96,64	98,18	0	0	5	1	0,02	0,02	0,14	0,12	1,96	1,50
37	FIR_DM6	12	46,31	50,91	31	13	49	14	0,52	0,50	0,72	0,71	52,23	50,14
38	FS_HGD6	12	71,81	72,73	29	13	13	2	0,27	0,26	0,52	0,51	26,91	26,24
39	GFS_LT_RS6	10	77,18	74,55	31	13	3	1	0,19	0,18	0,43	0,43	18,82	18,10
40	GFS_THRIFT6	49	46,31	50,91	31	13	49	14	0,72	0,61	0,85	0,78	72,13	60,51
41	HYFIS6	36	90,6	85,45	14	8	0	0	0,11	0,17	0,32	0,42	10,52	17,23
42	WM6	31	95,3	89,09	7	6	0	0	0,13	0,17	0,36	0,41	13,15	16,90
43	ANFIS7	28	96,71	98,08	0	0	5	1	0,02	0,02	0,13	0,13	1,82	1,65
44	FIR_DM7	12	50	40,38	24	20	52	11	0,47	0,70	0,68	0,83	46,59	69,53
45	FS_HGD7	12	76,32	59,62	24	19	12	2	0,27	0,54	0,52	0,74	26,72	54,42
46	GFS_LT_RS7	4	82,24	59,62	24	20	3	1	0,21	0,31	0,46	0,55	20,94	30,69
47	GFS_THRIFT7	50	50	40,38	24	20	52	11	0,73	0,58	0,85	0,76	72,68	58,07
48	HYFIS7	34	57,89	57,69	14	11	50	11	0,38	0,47	0,62	0,69	38,48	47,45
49	WM7	32	97,37	78,85	4	11	0	0	0,12	0,25	0,34	0,50	11,75	25,08
50	ANFIS8	28	97,32	96,36	0	0	4	2	0,02	0,02	0,13	0,15	1,64	2,11
51	FIR_DM8	10	48,32	45,45	31	13	46	17	0,51	0,54	0,71	0,74	50,95	54,44
52	FS_HGD8	10	72,48	65,45	30	12	11	7	0,21	0,32	0,46	0,57	21,47	32,29
53	GFS_LT_RS8	5	79,19	74,55	31	13	0	1	0,15	0,26	0,39	0,51	15,07	26,36
54	GFS_THRIFT8	50	48,32	45,45	31	13	46	17	0,53	0,54	0,73	0,73	52,89	53,69
55	HYFIS8	35	90,6	85,45	14	6	0	2	0,10	0,15	0,32	0,39	10,41	15,13
56	WM8	32	95,97	87,27	6	6	0	1	0,13	0,19	0,36	0,43	13,13	18,63
57	ANFIS9	30	98,01	98,11	0	0	3	1	0,02	0,02	0,13	0,15	1,62	2,16
58	FIR_DM9	12	47,02	49,06	33	11	47	16	0,52	0,50	0,72	0,71	52,17	50,42
59	FS_HGD9	12	70,2	73,58	32	11	13	3	0,25	0,26	0,50	0,51	24,73	26,06
60	GFS_LT_RS9	4	77,48	75,47	33	12	1	1	0,16	0,29	0,40	0,54	16,18	28,97
61	GFS_THRIFT9	50	47,02	49,06	33	11	47	16	0,70	0,68	0,84	0,83	70,27	68,34
62	HYFIS9	35	90,6	85,45	14	6	0	2	0,10	0,15	0,32	0,39	10,41	15,13
63	WM9	33	96,69	90,57	5	5	0	0	0,11	0,13	0,33	0,37	11,01	13,47
64	ANFIS10	31	97,3	98,21	0	0	4	1	0,02	0,02	0,13	0,12	1,81	1,50
65	FIR_DM10	12	43,92	57,14	32	12	51	12	0,54	0,45	0,74	0,67	54,43	45,29
66	FS_HGD10	12	71,62	73,21	30	12	12	3	0,26	0,24	0,51	0,49	25,79	23,76
67	GFS_LT_RS10	5	77,03	75	32	12	2	2	0,19	0,26	0,43	0,51	18,75	26,34
68	GFS_THRIFT10	50	43,92	57,14	32	12	51	12	0,68	0,46	0,83	0,68	68,37	46,38
69	HYFIS10	38	21,62	21,43	0	0	116	44	1,82	1,43	1,35	1,20	181,76	142,8
70	WM10	34	95,27	92,86	7	4	0	0	0,13	0,13	0,37	0,36	13,45	12,99
71	ANFIS11	28	96,05	100	0	0	6	0	0,02	0,01	0,15	0,10	2,17	0,92
72	FIR_DM11	11	50,66	38,46	29	15	46	17	0,48	0,62	0,69	0,79	48,06	61,90
73	FS_HGD11	11	74,34	63,46	29	14	10	5	0,23	0,33	0,48	0,57	23,36	32,79
74	GFS_LT_RS11	7	78,29	69,23	30	15	3	1	0,16	0,23	0,41	0,48	16,42	23,36

#	Algoritmos	CR	CC-E	CC-P	FP-E	FP-P	FN-E	FN-P	MSE-E	MSE-P	RMSE-E	RMSE-P	SMAPE-E	SMAPE-P
75	GFS_THRIFT11	50	50,66	38,46	29	15	46	17	0,68	0,75	0,82	0,87	67,91	75,21
76	HYFIS11	34	94,74	98,08	0	1	8	0	0,05	0,05	0,23	0,21	5,42	4,50
77	WM11	32	94,74	90,38	8	5	0	0	0,13	0,17	0,36	0,41	13,06	17,01
78	ANFIS12	30	96,08	100	0	0	6	0	0,02	0,00	0,15	0,06	2,20	0,39
79	FIR_DM12	10	46,41	50,98	38	6	44	19	0,53	0,47	0,73	0,69	53,44	47,10
80	FS_HGD12	10	67,97	78,43	36	6	13	5	0,27	0,19	0,52	0,44	26,87	19,24
81	GFS_LT_RS12	4	67,97	76,47	45	12	4	0	0,25	0,16	0,50	0,41	25,02	16,49
82	GFS_THRIFT12	50	46,41	50,98	38	6	44	19	0,62	0,59	0,79	0,77	62,04	58,66
83	HYFIS12	37	89,54	92,16	16	3	0	1	0,12	0,08	0,35	0,28	11,95	7,70
84	WM12	34	94,77	98,04	8	1	0	0	0,13	0,12	0,37	0,35	13,39	12,21
85	ANFIS13	31	97,32	98,18	0	0	4	1	0,02	0,02	0,14	0,15	1,93	2,26
86	FIR_DM13	11	46,31	50,91	31	13	49	14	0,52	0,50	0,72	0,71	52,23	50,08
87	FS_HGD13	11	72,48	65,45	29	13	12	6	0,23	0,28	0,48	0,53	23,18	28,31
88	GFS_LT_RS13	8	77,85	74,55	31	13	2	1	0,19	0,22	0,44	0,47	19,38	22,32
89	GFS_THRIFT13	47	46,31	50,91	31	13	49	14	0,73	0,61	0,85	0,78	72,83	61,41
90	HYFIS13	37	93,29	80	10	9	0	2	0,09	0,24	0,29	0,49	8,62	24,23
91	WM13	34	97,32	87,27	4	7	0	0	0,13	0,16	0,36	0,40	12,93	16,28
92	ANFIS14	30	96,18	100	0	0	6	0	0,02	0,01	0,15	0,09	2,13	0,73
93	FIR_DM14	11	48,41	44,68	39	5	42	21	0,52	0,54	0,72	0,74	51,54	54,30
94	FS_HGD14	11	69,43	76,6	38	5	10	6	0,24	0,25	0,49	0,50	24,29	25,18
95	GFS_LT_RS14	7	75,16	82,98	39	5	0	3	0,18	0,20	0,42	0,45	17,83	20,21
96	GFS_THRIFT14	43	48,41	44,68	39	5	42	21	0,54	0,47	0,73	0,68	53,73	46,75
97	HYFIS14	37	89,81	82,98	16	4	0	4	0,11	0,14	0,33	0,37	10,82	14,02
98	WM14	33	95,54	85,11	7	4	0	3	0,13	0,22	0,36	0,47	12,95	22,08
99	ANFIS15	30	97,4	98	0	0	4	1	0,02	0,03	0,13	0,16	1,75	2,55
100	FIR_DM15	12	50	40	34	10	43	20	0,50	0,58	0,70	0,76	49,66	58,04
101	FS_HGD15	12	73,38	64	32	10	9	8	0,21	0,39	0,46	0,62	20,97	38,97
102	GFS_LT_RS15	4	75,32	66	34	10	4	7	0,18	0,35	0,43	0,60	18,26	35,49
103	GFS_THRIFT15	49	50	40	34	10	43	20	0,63	0,65	0,79	0,81	62,84	65,11
104	HYFIS15	37	22,08	20	0	0	120	40	1,60	2,00	1,27	1,41	160,27	200
105	WM15	33	97,4	86	4	7	0	0	0,10	0,15	0,31	0,39	9,89	15,30
106	ANFIS16	29	97,37	98,08	0	0	4	1	0,02	0,02	0,14	0,13	1,98	1,81
107	FIR_DM16	10	47,37	48,08	36	8	44	19	0,52	0,50	0,72	0,70	51,57	49,51
108	FS_HGD16	10	71,71	65,38	34	8	9	10	0,19	0,35	0,44	0,59	19,29	35,08
109	GFS_LT_RS16	5	69,74	78,85	45	8	1	3	0,20	0,22	0,45	0,47	19,85	22,11
110	GFS_THRIFT16	50	47,37	48,08	36	8	44	19	0,67	0,77	0,82	0,88	66,98	77,14
111	HYFIS16	36	90,79	86,54	14	5	0	2	0,10	0,13	0,32	0,36	10,27	12,79
112	WM16	33	96,05	90,38	6	4	0	1	0,13	0,16	0,36	0,41	12,89	16,42
113	ANFIS17	32	96,67	98,15	0	0	5	1	0,02	0,01	0,14	0,12	1,85	1,39
114	FIR_DM17	12	50	40,74	34	10	41	22	0,50	0,57	0,71	0,76	49,78	57,44
115	FS_HGD17	12	72	70,37	33	10	9	6	0,24	0,27	0,49	0,52	24,30	27,07

#	Algoritmos	CR	CC-E	CC-P	FP-E	FP-P	FN-E	FN-P	MSE-E	MSE-P	RMSE-E	RMSE-P	SMAPE-E	SMAPE-P
116	GFS_LT_RS17	5	76	77,78	34	10	2	2	0,19	0,22	0,44	0,47	19,17	22,22
117	GFS_THRIFT17	49	50	40,74	34	10	41	22	0,55	0,52	0,74	0,72	55,44	52,47
118	HYFIS17	40	90,67	88,89	14	6	0	0	0,10	0,12	0,31	0,35	9,73	12,26
119	WM17	33	96	90,74	6	5	0	0	0,13	0,15	0,36	0,39	12,99	15,38
120	ANFIS18	27	97,32	96,36	0	0	4	2	0,02	0,02	0,13	0,15	1,70	2,10
121	FIR_DM18	9	48,32	45,45	30	14	47	16	0,50	0,55	0,71	0,74	50,38	54,99
122	FS_HGD18	9	71,81	69,09	29	13	13	4	0,23	0,27	0,48	0,52	23,08	26,56
123	GFS_LT_RS18	4	77,85	70,91	30	14	3	2	0,16	0,20	0,40	0,45	16,28	20,05
124	GFS_THRIFT18	50	48,32	45,45	30	14	47	16	0,51	0,62	0,71	0,79	51,04	62,17
125	HYFIS18	32	91,28	85,45	13	7	0	1	0,11	0,21	0,34	0,46	11,38	20,99
126	WM18	31	96,64	89,09	5	6	0	0	0,12	0,17	0,35	0,41	12,20	16,97
127	ANFIS19	29	98,03	94,23	0	0	3	3	0,01	0,04	0,11	0,19	1,26	3,55
128	FIR_DM19	12	49,34	42,31	33	11	44	19	0,50	0,56	0,71	0,75	50,13	56,00
129	FS_HGD19	12	70,39	76,92	32	10	13	2	0,26	0,22	0,51	0,47	26,07	22,07
130	GFS_LT_RS19	7	69,08	75	33	11	14	2	0,24	0,19	0,49	0,43	23,99	18,59
131	GFS_THRIFT19	48	49,34	42,31	33	11	44	19	0,55	0,66	0,74	0,81	55,24	65,60
132	HYFIS19	37	88,82	90,38	17	4	0	1	0,12	0,10	0,35	0,31	12,15	9,76
133	WM19	35	95,39	92,31	7	3	0	1	0,13	0,17	0,36	0,41	12,92	16,93
134	ANFIS20	28	98,03	94,23	0	0	3	3	0,01	0,03	0,12	0,18	1,41	3,30
135	FIR_DM20	12	44,74	55,77	36	8	48	15	0,55	0,43	0,74	0,65	54,60	42,72
136	FS_HGD20	12	69,74	78,85	35	7	11	4	0,27	0,20	0,52	0,45	27,32	20,23
137	GFS_LT_RS20	5	71,71	76,92	36	8	7	4	0,21	0,19	0,46	0,43	20,78	18,60
138	GFS_THRIFT20	50	44,74	55,77	36	8	48	15	0,61	0,53	0,78	0,73	60,73	52,94
139	HYFIS20	36	89,47	88,46	16	5	0	1	0,09	0,11	0,30	0,33	9,07	11,16
140	WM20	34	94,74	92,31	8	3	0	1	0,14	0,16	0,37	0,40	13,58	16,02

Anexo 2: Estudio comparativo de herramientas de gestión de proyectos, incluida Xedro-GESPRO.

Herramientas Informáticas	Áreas de conocimiento abordada							Software de código abierto	Tratamiento de incertidumbre	Aprendizaje automático
	Alcance	Tiempo	Costo	Calidad	Recursos Humanos	Integración	Adquisición			
Collabtive	Si	Si	No	No	Si	No	No	Si	No	No
dotProject	Si	Si	No	Si	Si	No	No	Si	No	No
LibrePlan	Si	Si	Si	Si	Si	Si	No	Si	No	No
OnepointProject	Si	Si	Si	No	Si	No	No	Si	No	No
Plandora	Si	Si	Si	No	Si	No	No	Si	No	No
Project.net	Si	Si	Si	Si	Si	No	Si	Si	No	No
Project-Open	Si	Si	Si	Si	Si	No	No	Si	No	No
OpenERP	Si	Si	Si	Si	Si	Si	Si	Si	No	No
TeamLab	Si	Si	Si	No	Si	No	No	Si	No	No
Redmine	Si	Si	Si	Si	Si	Si	No	Si	No	No
web2project	Si	Si	Si	No	Si	No	No	Si	No	No
Achievo	Si	Si	Si	Si	No	Si	No	Si	No	No
Xedro-GESPRO	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si