



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**GENERACIÓN DE LAS REGLAS QUE REPRESENTAN
EL CONOCIMIENTO DE UN SISTEMA EXPERTO DE
CLASIFICACIÓN DE AUTOS POR MEDIO DE
PROGRAMACIÓN GENÉTICA**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
CARLOS EDUARDO TORRES CRUZ**

ASESOR:

M. EN C. CARLOS OLIVER MORALES

MÉXICO

2005

m. 344053



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recespional.

NOMBRE: Carlos Eduardo Torres

FECHA: 13/mayo/2005

FIRMA: Torres Eduardo E.

AGRADECIMIENTOS

A la UNAM, por darme la oportunidad de realizar mis estudios.

Al Departamento de Ingeniería de Sistemas Computacionales y Automatización del IIMAS-UNAM, por permitirme utilizar sus instalaciones para la elaboración de la tesis.

Al CONACYT, por la beca que me otorgó a través del proyecto No. 40602-A, titulado: "Algoritmos Evolutivos en Práctica : Evolución Natural Como Base de la Evolución Artificial".

A la Dra. Katya Rodríguez Vázquez, por su ayuda y comentarios durante la realización de la tesis.

A mi asesor, M. en C. Carlos Oliver Morales, por guiarme y apoyarme a lo largo este trabajo de tesis.

A mis padres, Anibal Torres Valdéz y Margarita Cruz Sandoval, por su amor y cariño.

A toda mi familia, por su apoyo y confianza.

A mis amigos, por los momentos que hemos compartido.

GRACIAS

Carlos Eduardo Torres Cruz

ÍNDICE GENERAL

	Página
AGRADECIMIENTOS	i
ÍNDICE GENERAL	ii
ÍNDICE DE FIGURAS	v
ÍNDICE DE TABLAS	vi
RESUMEN	viii
APORTACIONES	viii
CAPÍTULO 1	
1 INTRODUCCIÓN	
1.1 Hipótesis	2
1.2 Objetivos	2
1.3 Estructura de la tesis	2
CAPÍTULO 2	
2 SISTEMAS EXPERTOS	
2.1 Sistemas Expertos	4
2.2 Características	5
2.3 Elementos de un sistema experto	5
2.3.1 Base de hechos	5
2.3.2 Base de conocimientos	5
2.3.3 Método de inferencia	6
2.3.4 Módulo de comunicación	7
2.4 Tipos de sistemas expertos	7
CAPÍTULO 3	
3 CLASIFICACIÓN	
3.1 Clasificación	9
3.2 Algoritmos de clasificación	9
3.3 Proceso de clasificación	10
3.4 Creación del modelo de clasificación	10
3.4.1 Representación del clasificador	10
3.4.2 Evaluación del clasificador	10
3.5 Generalización	11

	Página
CAPÍTULO 4	
4 PROGRAMACIÓN GENÉTICA	
4.1 Computación Evolutiva	12
4.2 Programación Genética	12
4.3 Características	13
4.4 Proceso Evolutivo	14
4.4.1 Representación de los individuos	14
4.4.2 Generación de la población inicial	14
4.4.3 Evaluación	16
4.4.4 Asignación de aptitud	17
4.4.5 Selección	17
4.4.6 Cruza	19
4.4.7 Mutación	20
4.4.8 Elitismo	20
4.5 Aplicación de la <i>PG</i> en clasificación	20
CAPÍTULO 5	
5 EXPERIMENTO	
5.1 Caso de estudio	22
5.1.1 Base de datos de autos	22
5.1.2 Transformación de los datos	23
5.2 División de ejemplos	24
5.3 Características de la <i>PG</i> ejecutada	24
5.3.1 Parámetros de ejecución	25
5.3.2 Evaluación con los ejemplos de prueba	25
5.4 Análisis de la varianza	26
CAPÍTULO 6	
6 RESULTADOS	
6.1 Desempeño en <i>sensibilidad</i>	27
6.2 Desempeño en <i>especificidad</i>	31
6.3 Desempeño en <i>precisión</i>	34
6.4 Desempeño en <i>exactitud</i>	37
6.5 Mejores Individuos	40
CAPÍTULO 7	
7 DISCUSIÓN, CONCLUSIONES Y TRABAJOS FUTUROS	
7.1 Discusión	41
7.2 Conclusiones	43
7.3 Trabajos futuros	44
BIBLIOGRAFÍA	45

	Página
ANEXOS	
A. Desempeño de las 30 ejecuciones de la <i>PG</i> en la predicción de los ejemplos de prueba para cada clase en cuanto a <i>SENSIBILIDAD</i>	
A.1. Utilizando 20% de ejemplos en entrenamiento	49
A.2. Utilizando 40% de ejemplos en entrenamiento	50
A.3. Utilizando 60% de ejemplos en entrenamiento	51
A.4. Utilizando 80% de ejemplos en entrenamiento	52
A.5. Utilizando 100% de ejemplos en entrenamiento	53
B. Desempeño de las 30 ejecuciones de la <i>PG</i> en la predicción de los ejemplos de prueba para cada clase en cuanto a <i>ESPECIFICIDAD</i>	
B.1. Utilizando 20% de ejemplos en entrenamiento	54
B.2. Utilizando 40% de ejemplos en entrenamiento	55
B.3. Utilizando 60% de ejemplos en entrenamiento	56
B.4. Utilizando 80% de ejemplos en entrenamiento	57
B.5. Utilizando 100% de ejemplos en entrenamiento	58
C. Desempeño de las 30 ejecuciones de la <i>PG</i> en la predicción de los ejemplos de prueba para cada clase en cuanto a <i>PRECISIÓN</i>	
C.1. Utilizando 20% de ejemplos en entrenamiento	59
C.2. Utilizando 40% de ejemplos en entrenamiento	60
C.3. Utilizando 60% de ejemplos en entrenamiento	61
C.4. Utilizando 80% de ejemplos en entrenamiento	62
C.5. Utilizando 100% de ejemplos en entrenamiento	63
D. Desempeño de las 30 ejecuciones de la <i>PG</i> en la predicción de los ejemplos de prueba para cada clase en cuanto a <i>EXACTITUD</i>	
D.1. Utilizando 20% de ejemplos en entrenamiento	64
D.2. Utilizando 40% de ejemplos en entrenamiento	65
D.3. Utilizando 60% de ejemplos en entrenamiento	66
D.4. Utilizando 80% de ejemplos en entrenamiento	67
D.5. Utilizando 100% de ejemplos en entrenamiento	68
E. Cálculo del análisis de la varianza	
E.1. Fórmulas para el cálculo	69
E.2. Ejemplo	70

ÍNDICE DE FIGURAS

Figura	Página
1.1. Esquema descriptivo	3
2.1. Elementos de los sistemas expertos	6
2.2. Ciclo de inferencia de un sistema experto	7
2.3. Ejemplo de una consulta	8
3.1. Proceso de clasificación	10
4.1. Proceso de la Computación Evolutiva	13
4.2. Árbol de profundidad 3 (<i>Individuo: (B and (not C))</i>)	15
4.3. Generación por crecimiento para una profundidad de 3 (<i>Individuo: (B and (not C))</i>)	15
4.4. Generación en forma completa para una profundidad de 3 (<i>Individuo: (C and A) or (not B)</i>)	16
4.5. Generación Mitad-Mitad	16
4.6. Ruleta	18
4.7. Estocástico universal	18
4.8. Seleccionar punto de corte y subárbol de dos individuos	19
4.9. Intercambiar los subárboles seleccionados	19
4.10. Mutación	20
5.1. Evaluación del mejor individuo	25
6.1. Desempeño de los promedios en sensibilidad	28
6.2. Desempeño de las 30 ejecuciones en sensibilidad	29
6.3. Desempeño de las ejecuciones en sensibilidad	30
6.4. Desempeño de los promedios en especificidad	32
6.5. Desempeño de las ejecuciones en <i>especificidad</i>	33
6.6. Desempeño de los promedios en precisión	35
6.7. Desempeño de las ejecuciones en <i>precisión</i>	36
6.8. Desempeño de los promedios en exactitud	38
6.9. Desempeño de las ejecuciones en <i>exactitud</i>	39

ÍNDICE DE TABLAS

Tabla	Página
2.1. Característica de un sistema experto	5
5.1. Atributos y valores	22
5.2. Distribución de los ejemplos entre las clases	22
5.3. Ejemplos de cada clase	23
5.4. Transformación del Costo	23
5.5. Transformación del Mantenimiento	23
5.6. Transformación del # de Puertas	23
5.7. Transformación del # de Personas	23
5.8. Transformación del tamaño de la Cajuela	23
5.9. Transformación del nivel de Seguridad	23
5.10. Transformación a binario	24
5.11. Casos evaluados	24
5.12. Parámetros de ejecución para las clases	25
6.1. Resultados de <i>sensibilidad</i> en el Caso ₅ testigo, con 100% de ejemplos en entrenamiento	27
6.2. Resultados de <i>sensibilidad</i> en el Caso ₁ , con 20% de ejemplos en entrenamiento	27
6.3. Resultados de <i>sensibilidad</i> en el Caso ₂ , con 40% de ejemplos en entrenamiento	27
6.4. Resultados de <i>sensibilidad</i> en el Caso ₃ , con 60% de ejemplos en entrenamiento	28
6.5. Resultados de <i>sensibilidad</i> en el Caso ₄ , con 80% de ejemplos en entrenamiento	28
6.6. Resultados de <i>especificidad</i> en el Caso ₅ testigo, con 100% de ejemplos en entrenamiento	31
6.7. Resultados de <i>especificidad</i> en el Caso ₁ , con 20% de ejemplos en entrenamiento	31
6.8. Resultados de <i>especificidad</i> en el Caso ₂ , con 40% de ejemplos en entrenamiento	31
6.9. Resultados de <i>especificidad</i> en el Caso ₃ , con 60% de ejemplos en entrenamiento	31
6.10. Resultados de <i>especificidad</i> en el Caso ₄ , con 80% de ejemplos en entrenamiento	31
6.11. Resultados de <i>precisión</i> en el Caso ₅ testigo, con 100% de ejemplos en entrenamiento	34
6.12. Resultados de <i>precisión</i> en el Caso ₁ , con 20% de ejemplos en entrenamiento	34
6.13. Resultados de <i>precisión</i> en el Caso ₂ , con 40% de ejemplos en entrenamiento	34
6.14. Resultados de <i>precisión</i> en el Caso ₃ , con 60% de ejemplos en entrenamiento	34

	Página
6.15. Resultados de <i>precisión</i> en el Caso ₄ , con 80% de ejemplos en entrenamiento	34
6.16. Resultados de <i>exactitud</i> en el Caso ₅ testigo, con 100% de ejemplos en entrenamiento	37
6.17. Resultados de <i>exactitud</i> en el Caso ₁ , con 20% de ejemplos en entrenamiento	37
6.18. Resultados de <i>exactitud</i> en el Caso ₂ , con 40% de ejemplos en entrenamiento	37
6.19. Resultados de <i>exactitud</i> en el Caso ₃ , con 60% de ejemplos en entrenamiento	37
6.20. Resultados de <i>exactitud</i> en el Caso ₄ , con 80% de ejemplos en entrenamiento	37
6.21. Mejores individuos con 60% de ejemplos en entrenamiento	40
6.22. Mejores individuos con 100% de ejemplos en entrenamiento	40

RESUMEN

En este trabajo se obtuvieron las reglas de clasificación, como forma de inducir conocimiento, de una base de datos de autos tomada como caso de estudio. La creación de las reglas se realizó por medio de *Programación Genética*. Si se utiliza el total de ejemplos no hay problema de insuficiencia de información para crear las reglas, sin embargo, esto implica un costo computacional mayor. Se comparó el desempeño de la *Programación Genética* para crear las reglas utilizando el total de ejemplos y muestras menores. Se lograron obtener desempeños semejantes entrenando con una muestra aleatoria o con el total de ejemplos. Tomando una muestra fue posible predecir la clase a la que pertenecen ejemplos desconocidos. Utilizando una muestra aleatoria, se disminuye la cantidad de ejemplos que se tienen que evaluar, por lo que el costo computacional es menor.

APORTACIONES

- De la presente tesis se derivó la elaboración del artículo titulado: "*Evaluación del desempeño de la Programación Genética en la creación de reglas de clasificación con diferentes porcentajes de entrenamiento*", aceptado en el Segundo Congreso de Computación Evolutiva (CONCEV'05) que se llevará a cabo en la Universidad Autónoma de Aguascalientes del 25 al 27 de mayo del 2005.

CAPÍTULO 1

INTRODUCCIÓN

Los problemas los resolvemos a través de la aplicación del conocimiento que hemos aprendido con el tiempo. Utilizamos nuestro conocimiento por ejemplo, para decidir si una proposición es verdadera o falsa. El aprendizaje se manifiesta a diferentes niveles, tres de ellos son: la memorización, la deducción y la inducción. Cuando la cantidad de información es pequeña la memorización es útil y suficiente para resolver problemas particulares. Sin embargo, conforme la cantidad de información aumenta, su asimilación y memorización se hace más complicada. Se requiere entonces pasar a otro nivel de aprendizaje. En general hacemos relaciones y analogías entre situaciones conocidas para llegar a una solución. Si partimos de un caso general y llegamos a una solución particular habremos realizado una deducción. En el caso contrario, la inducción parte de casos particulares conocidos y generaliza la solución para casos desconocidos.

Los expertos, son personas que tienen la experiencia y el conocimiento para resolver problemas de una área específica. La computadora se puede utilizar como herramienta de memorización y consulta de los conocimientos de un experto. Se puede buscar también, utilizar la computadora para la extracción automática de conocimiento que sirva de apoyo para la persona que tomará la decisión final sobre su validez.

Las grandes cantidades de información hacen imposible su asociación de forma manual y aún con la computadora resulta costoso extraer el conocimiento de forma automática. La clasificación es una de las formas como se puede extraer conocimiento de los datos. Se crea un clasificador con datos conocidos y después se valida con datos nuevos. La validación indica si se extrajo conocimiento cuando el clasificador logra clasificar datos desconocidos. El clasificador entonces, habrá aprendido de los datos a un nivel mayor que una simple memorización.

La *Programación Genética* es uno de los métodos que ha sido utilizado para hacer clasificación de forma automática. El problema es la cantidad de datos necesarios para crear el clasificador, porque esto implica mayor costo computacional. En los trabajos consultados no encontramos un criterio que unifique la proporción del total de datos que debe utilizarse para crear el clasificador.

Para experimentar sobre el desempeño de la *Programación Genética* al variar la cantidad de datos utilizados en la creación del clasificador se tomó como caso de estudio una base de datos de autos. La base de datos describe las características de los autos mediante valores de costo, mantenimiento, número de puertas, número de personas, tamaño de la cajuela y nivel de seguridad. Dependiendo de los valores de cada característica un auto es clasificado como inaccesible, accesible, bueno o muy bueno.

1.1 Hipótesis

- La *Programación Genética* mantiene un mismo desempeño al crear las reglas de clasificación con el total de los ejemplos o con una muestra aleatoria.

1.2 Objetivos

Objetivo General

- Generar las reglas de clasificación.

Objetivos Específicos

- Aplicar *Programación Genética* para la generación de reglas de clasificación.
- Mejorar el desempeño de la *Programación Genética* con cantidades menores de ejemplos que se le proporcionan para crear reglas de clasificación.
- Disminuir el costo computacional de la *Programación Genética* en la creación de reglas de clasificación.

1.3 Estructura de la tesis

El esquema descriptivo de la tesis se muestra en la Figura 1.1. Los capítulos que componen este trabajo se describen a continuación. El *Capítulo 1*, plantea de forma general el contexto del trabajo, dando a conocer cual es el problema a resolver a través de la hipótesis y los objetivos. Los *Sistemas Expertos* son descritos mediante sus características y funcionamiento en el *Capítulo 2*, haciendo hincapié en la forma que representan el conocimiento para la solución de problemas. En el *Capítulo 3* se describe la *Clasificación*, su importancia como forma de obtener conocimiento y el proceso para realizarla. El método utilizado en la creación de las reglas de clasificación fue la *Programación Genética* que es presentada en el *Capítulo 4*. La descripción y desarrollo del *Experimento* sobre el caso de estudio, indicando los parámetros que se utilizaron y las variables que se midieron se presentan en el *Capítulo 5*. El *Capítulo 6* muestra los *Resultados* del experimento. La *Discusión*, *Conclusiones* y *Trabajos futuros* se presentan en el *Capítulo 7*.

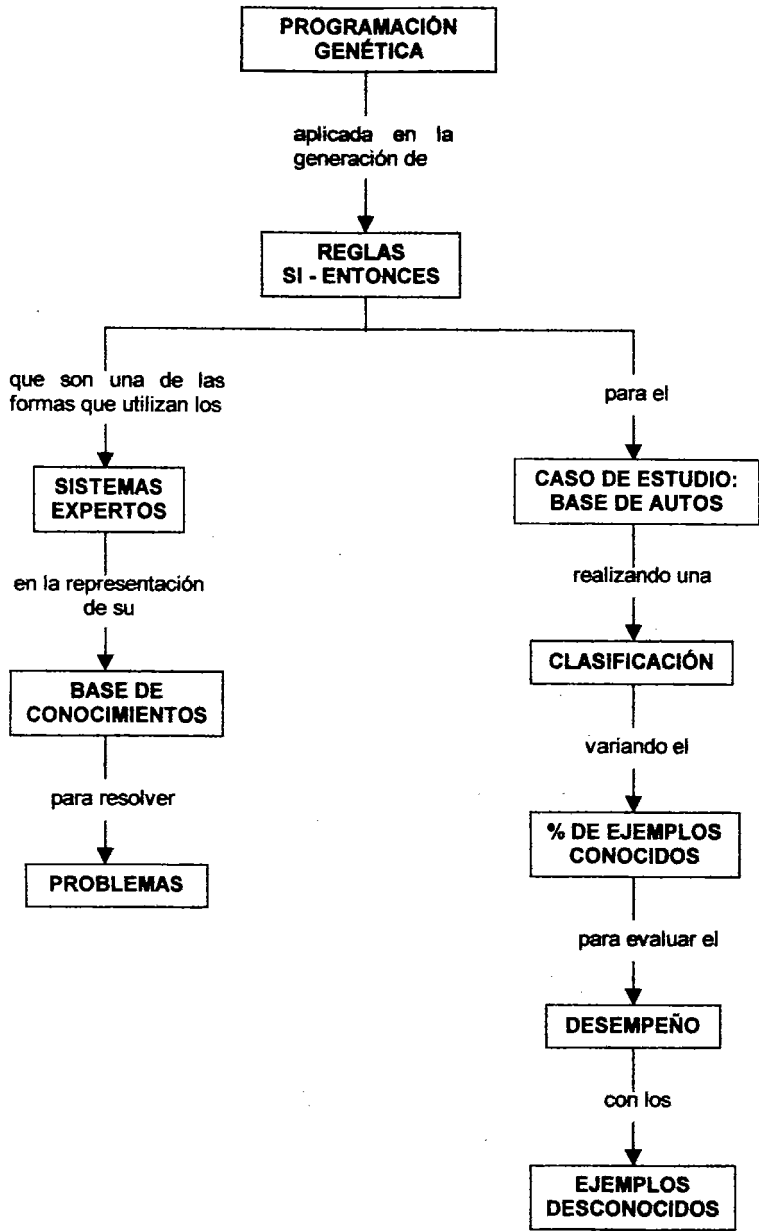


Figura 1.1. Esquema descriptivo.

CAPÍTULO 2

SISTEMAS EXPERTOS

2.1 Sistemas Expertos

La solución de un problema en la vida cotidiana la encontramos al evaluar las alternativas que conocemos y decidimos cual o cuales tomar. Nos preguntamos cuál es el resultado si tomo ésta o aquella alternativa. Si el resultado no es la solución, tomamos otras alternativas y encadenamos los resultados parciales para llegar a una solución final.

En problemas de un área de conocimiento específico (ciencia, medicina, ingeniería, negocios, etc.), consultamos a un experto. Un experto es una persona especializada por tener una larga experiencia, es decir, tiene tanto el conocimiento como las habilidades necesarias para resolver problemas [34]. Los expertos escogen las alternativas y toman decisiones de una manera más acertada que la mayoría de las personas [18].

La tecnología en general y la computación en particular auxilian al hombre en tareas repetitivas. El proceso que lleva a los expertos a solucionar un problema, se simula en una computadora obteniendo y representando el conocimiento del experto. Los sistemas expertos simulan la habilidad de tomar decisiones de un especialista humano, al imitar la forma en que las personas procesan la información, es decir, la manera en que piensa la gente cuando resuelve problemas [18].

Los sistemas expertos o también llamados sistemas basados en conocimiento, son programas de computadora especializados en un área de conocimiento [18][27][34]. Los problemas los resuelven utilizando conocimientos almacenados y procedimientos de inferencia. El sistema experto toma el papel del experto, aunque sirve sólo de apoyo, ya que la última decisión sobre el resultado la toma el experto [18].

El conocimiento primero debe ser extraído del experto, tarea considerada cuello de botella por la dificultad en la comunicación y entendimiento entre el experto y el encargado de crear el sistema. La ingeniería del conocimiento trabaja con personas especializadas en extraer y representar el conocimiento del experto [18][27][34]. Ejemplos de tres pioneros de los sistemas expertos son [34]: DENDRAL (1967), utilizado para la determinación de la estructura de un compuesto químico; PROSPECTOR (1974), utilizado en la búsqueda y exploración de yacimientos minerales; MYCIN (1977), sistema médico de apoyo en el diagnóstico y terapia de enfermedades.

2.2 Características

Los sistemas expertos se caracterizan por estar disponibles en cualquier momento, permanecer en el tiempo, ser fáciles para duplicar, pueden almacenar conocimientos de diferentes expertos y dar respuestas objetivas [18][34]. En su funcionamiento hay una independencia entre los datos y el método utilizado para encontrar la solución del problema. La descripción del programa se realiza de forma declarativa, es decir, se especifica ¿qué hacer?, pero no ¿cómo hacerlo? [34]. La solución se obtiene mediante la aplicación de inferencias lógicas [18][34]. En la Tabla 2.1 se presentan las características generales de los sistemas expertos [18].

Tabla 2.1. Característica de un sistema experto.

CARACTERÍSTICA	SISTEMA EXPERTO
Tipo de datos	Simbólico
Entrada de datos	Responder a entradas incompletas, incorrectas, imprecisas
Control	Débil. Maneja un mecanismo de inferencia (reglas de inferencia)
Control-Datos	Separación explícita, independencia
Búsqueda de solución	Grande
Solución	Varía con el problema. Es satisfactoria
Explicación	Habitualmente
Aplicaciones	Razonamiento simbólico

2.3 Elementos de un sistema experto

La Figura 2.1 muestra los elementos de los sistemas expertos: base de hechos, base de conocimiento, método de inferencia y módulos de comunicación. Algunos sistemas contienen una parte de inducción de reglas para aprender nuevo conocimiento mediante ejemplos [18].

2.3.1 Base de hechos

La base de hechos, memoria activa o universo de conocimiento, es la información incondicionalmente correcta [18][34].

2.3.2 Base de conocimientos

Existen tres formas de expresar el conocimiento: por procedimientos para describir los pasos de alguna tarea (algoritmo), en forma declarativa para determinar si algo

es verdadero o falso, y tácitamente cuando el concepto queda sobreentendido [34].

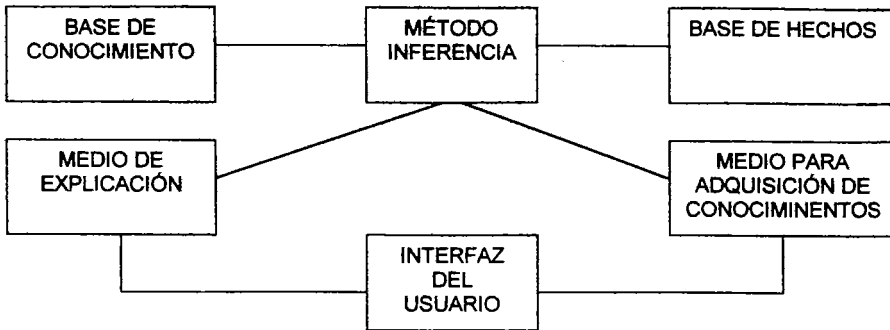


Figura 2.1. Elementos de los sistemas expertos.

Las expresiones declarativas se utilizan para relacionar hechos y acciones que representan el conocimiento extraído del experto. Este conocimiento es de tipo superficial, empírico y no de explicación científica.

La base de conocimientos contiene expresiones declarativas representadas por alguna de las siguientes formas: reglas de producción, redes semánticas, marcos, objetos, guiones, lógica de predicado de primer orden o lógica borrosa [18].

Las reglas de producción son la forma en que se representará el conocimiento en el caso de estudio de este trabajo. Las reglas de producción establecen la relación *antecedente-consecuente* mediante la siguiente estructura [18][34]:

Si (antecedente) ENTONCES (consecuente)

donde el antecedente son los hechos que se evalúan para llegar a una conclusión (consecuente). Esta representación es la herramienta principal para la simulación de la forma en que las personas resuelven problemas [18].

La evaluación de una regla es determinista si las condiciones afectan totalmente el resultado, pero es probable si el resultado depende de las probabilidades de los hechos [34].

Las ventajas de las reglas de producción son su carácter declarativo, sencillez, uniformidad, independencia y modularidad. Sus desventajas son su rápido crecimiento y la introducción de repeticiones y contradicciones [18][34].

2.3.3 Método de inferencia

El método de inferencia simula el proceso que sigue el experto para buscar y ejecutar las alternativas adecuadas. Las soluciones se construyen dinámicamente seleccionando e interpretando la información de la base de conocimientos y de la base de hechos [18][34]. En la Figura 2.2 se muestra el ciclo de inferencia de los sistemas expertos.

La estrategia de búsqueda que el mecanismo de inferencia utiliza puede ser ordenada o aleatoria [34]. Una búsqueda ordenada se hace con el encadenamiento de reglas cuando el consecuente de una regla se convierte en el antecedente de la siguiente. Los tipos de encadenamiento son [27]:

- a) Encadenamiento hacia delante: se van evaluando los hechos hasta llegar a una conclusión.
- b) Encadenamiento hacia atrás: se parte de una conclusión y se buscan los hechos que la sustentan.

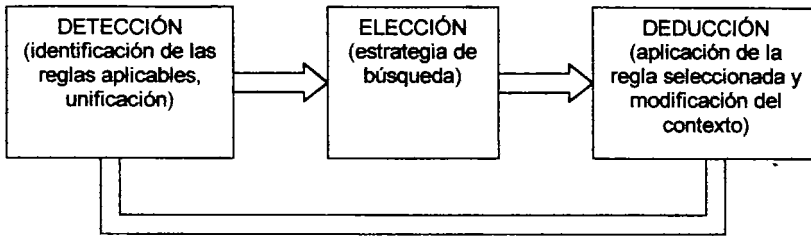


Figura 2.2. Ciclo de inferencia de un sistema experto.

2.3.4 Módulo de comunicación

Los módulos de comunicación permiten el intercambio de información del sistema con el exterior para [18][34]:

- Consulta: el usuario introduce los hechos esperando recibir una confirmación o negación del hecho.
- Explicación: el sistema explica y justifica el razonamiento que siguió para llegar a una conclusión.
- Adquisición: el sistema obtiene nuevo conocimiento para aumentar su capacidad de respuesta. Se obtienen más conocimientos del experto o mediante el aprendizaje con ejemplos.

La Figura 2.3 presenta un ejemplo donde el programa responde a la consulta del usuario. Las reglas de producción están representadas como cláusulas Horn [27].

2.4 Tipos de sistemas expertos

Los sistemas expertos pueden ser aplicados para resolver alguna de las siguientes tareas: configuración, interpretación, diagnóstico, reparación, planeación, enseñanza, control, supervisión o predicción [18][27][34]. De acuerdo con el nivel de confianza que se les otorgue son de comprobación, consulta o decisión [18]. Se han desarrollado distintos sistemas expertos según el área del conocimiento donde son aplicados, el objetivo de cada uno de ellos, su estructura y funcionamiento [34].

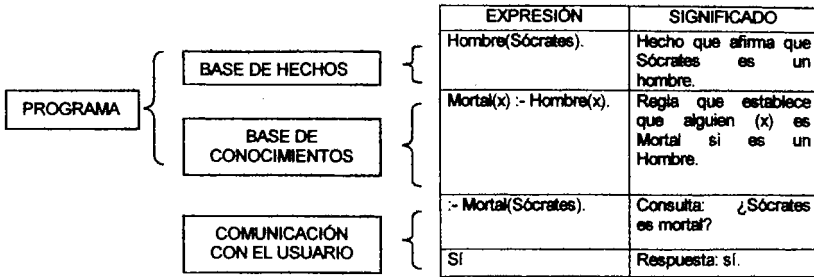


Figura 2.3. Ejemplo de una consulta.

Se ha hablado de la importancia del conocimiento para resolver problemas y como mediante los Sistemas Expertos la computadora simula el proceso que lleva a un experto a encontrar la solución. El capítulo siguiente explica la clasificación como forma automática de obtener conocimiento de los datos.

CAPÍTULO 3

CLASIFICACIÓN

3.1 Clasificación

Las grandes organizaciones generan una gran cantidad de datos, por ejemplo: médicos, científicos, demográficos, financieros, etc. [21]. En estas bases de datos la relación entre los datos se traduce en conocimiento. Por ejemplo, saber como se tomaron las decisiones en el pasado para basarnos en la toma de decisiones del futuro [17][21][39]. Sin embargo, la cantidad de datos que se llegan a tener hace difícil encontrar conocimiento por inspección manual. Se recurre entonces, a formas automáticas de análisis que transforman los datos en conocimiento. El descubrimiento de conocimiento y la minería de datos se utilizan para la difícil tarea de extraer válidos, novedosos, potencialmente útiles y entendibles modelos de los datos [17][21][32][39].

El descubrimiento de conocimiento es la última etapa del proceso que siguen los datos de acuerdo con su funcionalidad, ya que antes son recolectados y administrados (almacenamiento, procesamiento, transferencia) [21]. En el descubrimiento de conocimiento las etapas generales son [39]: selección de datos, filtración de errores, identificación de variables relevantes, creación del modelo, validación e interpretación del modelo.

La clasificación es una de las formas en que la minería de datos logra extraer conocimiento de las bases de datos. Primero se crea el modelo conociendo la clase de un conjunto de ejemplos de entrenamiento. Después se evalúa la exactitud del modelo para clasificar los ejemplos de prueba. Este procedimiento es también conocido como aprendizaje supervisado [15][17][21][30][39].

El conocimiento extraído de la base de datos sirve de apoyo en la toma de decisiones. Por ejemplo, un clasificador en la venta de artículos se utiliza para saber qué clientes pueden comprar un determinado artículo [21]. En el caso de datos médicos, el clasificador es útil para hacer diagnósticos [17].

3.2 Algoritmos de clasificación

Cualquier algoritmo de clasificación busca crear un modelo que clasifique correctamente la clase de los casos de prueba y no la memorización de los casos de entrenamiento [17][21]. Algunos de los algoritmos utilizados para clasificación son [30]: generadores de árboles de decisión, inducción de reglas, algoritmos evolutivos, redes neuronales, aprendizaje bayesiano y programación lógica inductiva. En este trabajo se utiliza un algoritmo evolutivo para la creación del modelo de clasificación, la *Programación Genética*.

3.3 Proceso de clasificación

Las bases de datos están compuestas de ejemplos, llamados también objetos, instancias de datos, casos, registros o tuplas. Las características que describen cada ejemplo son denominadas atributos. Los atributos son de predicción (ej. costo, tamaño, etc.), o de clase (ej. accesible, no accesible) [17].

El modelo que establece la relación entre los datos y más específicamente, entre los atributos de predicción y las clases, se obtiene de la siguiente forma [17][21]:

1. Dividir aleatoriamente el total de ejemplos en dos grupos independientes: ejemplos de entrenamiento y ejemplos de prueba.
2. Crear el modelo. El algoritmo trabaja con los ejemplos de entrenamiento para crear el modelo.
3. Evaluar el modelo. Se mide la capacidad del modelo para clasificar correctamente los ejemplos de prueba (examen de aprendizaje).

La Figura 3.1 muestra la secuencia del proceso de clasificación.

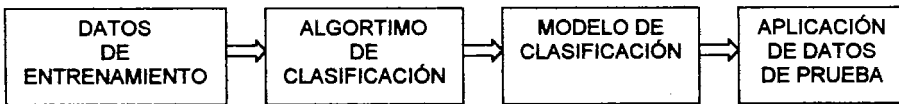


Figura 3.1. Proceso de clasificación.

3.4 Creación del modelo de clasificación

El número de modelos candidatos a soluciones tiende a ser infinito por la cantidad de ejemplos con que se entrena el algoritmo. Los algoritmos de clasificación utilizan dos tipos de alternativas para crear el modelo. En la representación del modelo usan formas y tipos de funciones determinados, mientras para la búsqueda aplican métodos para discriminar entre modelos y modificarlos [17].

3.4.1 Representación del clasificador

Algunas de las diferentes formas de representación de modelos de clasificación son por ejemplo [21]: reglas, árbol de decisión, fórmula matemática y red neuronal. El clasificador para el caso de estudio será representado por reglas que están definidas por medio de la siguiente estructura [17][21]:

Si (antecedente) ENTONCES (consecuente)

3.4.2 Evaluación del clasificador

Los algoritmos de clasificación son evaluados de acuerdo con alguna de las características siguientes: exactitud, comprensión, interés, velocidad, robustez o

capacidad de ser escalable [17][21][30]. La exactitud es el porcentaje de aciertos del modelo con los ejemplos de prueba. La comprensión considera la parte sintáctica (estructura y elementos) y la semántica (entendimiento y lógica), siendo una medida subjetiva que depende del conocimiento sobre el problema y la interpretación del modelo. El interés es también una percepción subjetiva, ya que valora lo novedoso, sorprendente o útil del modelo. El costo computacional para la generación del modelo nos indica la velocidad. Si el modelo tiene la habilidad para clasificar datos con ruido, entonces se califica como robusto. Se dice que el algoritmo de clasificación es escalable cuando construye modelos con bases de datos grandes.

3.5 Generalización

La generalización la podemos ver como la agrupación de elementos en un solo concepto. Se realiza una generalización, por ejemplo, cuando se transforman valores continuos en discretos. Los valores continuos son agrupados en categorías estableciendo rangos de valores [17].

El valor real de predicción del modelo, también denominada, habilidad de generalización, se evalúa con los ejemplos de prueba. Un alto grado de exactitud con los ejemplos de entrenamiento no garantiza el mismo éxito en los ejemplos de prueba, es decir, no se tiene la certeza de predecir correctamente datos futuros [17].

En las reglas de clasificación *SI-ENTONCES*, el antecedente está compuesto por condiciones. La generalización en una regla se logra si tiene menos condiciones e igual o mayor valor de exactitud que otra [17]. La especialización es el caso contrario. Los casos extremos en el aprendizaje son el sobre-entrenamiento y el bajo-entrenamiento. Si la regla se especializa sólo en los ejemplos de entrenamiento, es el caso de sobre-entrenamiento, no habiendo en tal caso generalización para la predicción de los casos de prueba. Una generalización alta de los ejemplos de entrenamiento se convierte en un bajo-entrenamiento al no considerar casos particulares o excepciones [17].

La clasificación se ha descrito mediante un proceso general para la extracción automática de conocimiento. Un algoritmo utilizado para clasificar es la *Programación Genética* que se explicará en el siguiente capítulo.

CAPÍTULO 4

PROGRAMACIÓN GENÉTICA

4.1 Computación Evolutiva

La evolución de las especies en la naturaleza muestra un proceso donde los individuos más adaptados al ambiente sobrevivirán. El proceso evolutivo muestra el aprendizaje de las especies respecto del medio que les rodea [24].

El término Computación Evolutiva se refiere a todos los sistemas computacionales de búsqueda, aprendizaje u optimización de soluciones, que basan su funcionamiento en los principios de evolución y herencia de las especies naturales. Algunos ejemplos de tales sistemas son [29]: Estrategias Evolutivas, la Programación Evolutiva, los Algoritmos Genéticos y la Programación Genética.

La Figura 4.1 muestra el proceso evolutivo [29], que consiste en crear una primera generación de individuos (representación computacional de las soluciones). Los individuos son evaluados con el problema a resolver. Después en un ciclo generacional, cada nueva generación es creada por la selección de individuos de la generación anterior. Los individuos seleccionados serán probablemente modificados por operadores genéticos y nuevamente evaluados. El ciclo generacional terminará cuando se cumpla una condición, como encontrar una solución aceptable o llegar a una generación especificada.

La idea de intentar que las computadoras aprendan a resolver problemas por sí mismas introdujo el concepto de aprendizaje de máquina. El aprendizaje de máquina es el estudio de los algoritmos que mejoran automáticamente a través de su experiencia. Entre los métodos de aprendizaje de máquina están incluidos los de Computación Evolutiva [2].

4.2 Programación Genética

En la *Programación Genética (PG)* [24], el interés es que las computadoras se programen automáticamente, por lo tanto los individuos son programas de computadora. La *PG* se aplica a diferentes problemas, los cuales se solucionan mediante la inducción de un programa. El objetivo es buscar el programa que produzca una salida deseada para ciertos datos de entrada. Dependiendo del problema, el programa de computadora buscado puede ser una fórmula, una estrategia de control, un procedimiento, un modelo, un árbol de decisión, una estrategia de juego, etc.. Para realizar la búsqueda, la *PG* mantiene una población de individuos que siguen un proceso de aprendizaje guiado por los datos. El proceso consiste en la aplicación cíclica de algún método de selección basado en la aptitud de cada individuo y algunos operadores genéticos para crear nuevas generaciones.

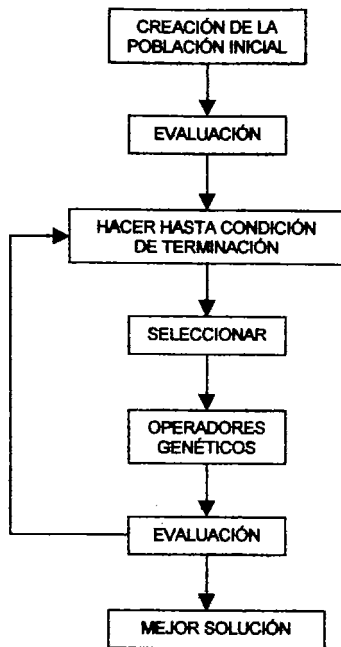


Figura 4.1. Proceso de la Computación Evolutiva.

4.3 Características

En la propuesta inicial de PG [24], la forma de representar a los individuos es por medio de estructuras jerárquicas (árboles). Sin embargo, han surgido variantes que utilizan estructuras lineales o grafos, mismas que modifican también la forma de aplicar la cruce y la mutación. Las características comunes de las diferentes variantes son [2]:

- Aplicación de decisiones estocásticas: se utilizan números pseudo-aleatorios para imitar la evolución natural.
- Estructura de los individuos: se generan estructuras de longitud variable que sean expresiones válidas en algún lenguaje de programación.
- Simulación de la evolución: trabajan con una población de individuos que sobreviven de acuerdo con su grado de adaptación.
- Operadores genéticos: se modifican los individuos de la generación actual con operadores genéticos como la cruce, la mutación, y otras que benefician la búsqueda y generación de nuevos individuos.

4.4 Proceso Evolutivo

La *PG* para un problema dado se pone en práctica al realizar varias ejecuciones en las que se definen: el conjunto de ejemplos, las funciones, los terminales, el tamaño de la población, el número de generaciones, el porcentaje de cruce, el porcentaje de mutación, el porcentaje de elitismo, la profundidad inicial y la profundidad máxima [24]. Los parámetros anteriores son utilizados durante el proceso evolutivo.

4.4.1 Representación de los individuos

La representación de los individuos se hace con una estructura de árbol, una lista lineal o un grafo, en cualquier caso habrá dos conjuntos de elementos [2][24]:

- Conjunto de funciones: lo conforman los comandos, instrucciones, operadores o funciones definidas. Por ejemplo: funciones Booleanas (*AND*, *OR*, *NOT*), condicionales (*Si-Entonces-De lo contrario*), aritméticas (*suma*, *resta*, *multiplicación*, *división*), ciclos (*Mientras(condición)-Hacer*, *Hacer-Hasta(condición)*), etc.
- Conjunto de terminales: son las variables y constantes que utilizan las funciones como entradas. Por ejemplo: variables Booleanas, números reales, funciones que no reciben valores de entrada (*Mover-Derecha*, *Mover-Izquierda*).

En la definición de funciones y terminales se deben tener presente las siguientes propiedades [24]:

- Cerradura: el valor de un terminal o el resultado de una función debe ser válido como entrada para cualquiera de las funciones definidas. Se deben crear alternativas para evitar errores al evaluar una función que tenga restricciones en los valores que recibe.
- Suficiencia: el conjunto de funciones y terminales deben tener los mínimos elementos para expresar una solución del problema.
- Universalidad: se deben utilizar las funciones y terminales básicos para resolver el problema y no deben agregarse funciones o terminales muy complejos si estos no benefician el desempeño de las soluciones.

4.4.2 Generación de la población inicial

En la creación de los individuos como estructuras de árbol, no se define su tamaño y forma final, ya que es mediante un proceso aleatorio como se selecciona una función o un terminal. El máximo crecimiento sólo se limita por el parámetro de profundidad y por la cantidad de parámetros que reciba una función. Cualquier elemento del individuo es denominado nodo y su profundidad está dada por el número de nodos que hay entre él y el nodo raíz. La Figura 4.2 muestra un árbol de profundidad 3.

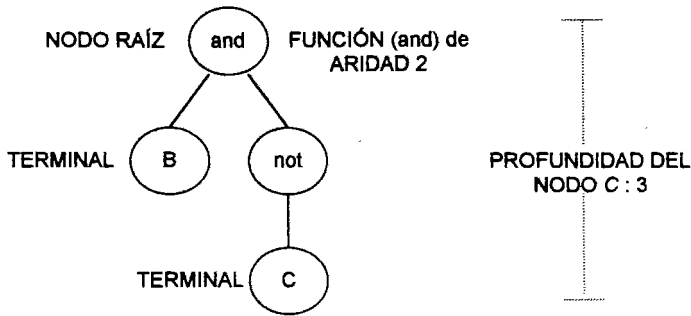


Figura 4.2. Árbol de profundidad 3 (Individuo: (B and (not C))).

Los métodos para crear individuos son: por crecimiento, en forma completa y mitad-mitad [24]. Todos los nodos son seleccionados aleatoriamente y el primero es el nodo raíz, seleccionado del conjunto de funciones.

- Crecimiento: los nodos siguientes a la raíz se seleccionan entre el conjunto de funciones y terminales. Cuando se llega a la profundidad de creación establecida, sólo se seleccionan nodos del conjunto de terminales. Este método permite la creación de una población de soluciones de diferente forma y probablemente sin la máximo profundidad posibles. Un ejemplo de un individuo creado por este método es mostrado en la Figura 4.3.

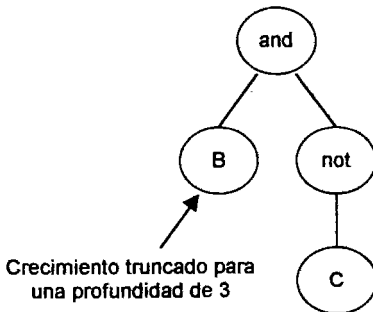


Figura 4.3. Generación por crecimiento para una profundidad de 3 (Individuo: (B and (not C))).

- Completo: los nodos siguientes a la raíz se seleccionan del conjunto de funciones hasta llegar a la profundidad de creación establecida, a partir de la cual, se seleccionan sólo terminales. Como lo muestra el ejemplo de la Figura 4.4, las soluciones tendrán el máximo de nodos posibles para la profundidad establecida.

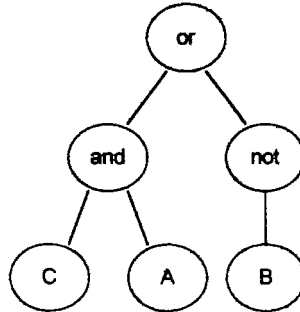


Figura 4.4. Generación en forma completa para una profundidad de 3
(Individuo: (C and A) or (not B)).

- Mitad-mitad: la población se divide equitativamente en grupos de profundidades, desde la profundidad de 2 niveles que es la mínima, hasta el parámetro de profundidad de creación inicial establecido. Para cada grupo, la mitad de individuos se crean por el método de crecimiento y la otra mitad por el completo. Los individuos creados varían en forma y tamaño. La Figura 4.5 muestra la representación gráfica de este método.

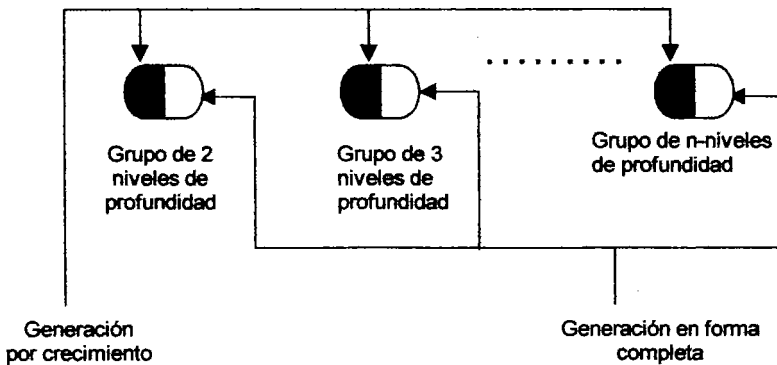


Figura 4.5. Generación Mitad-Mitad.

4.4.3 Evaluación

El dominio del problema es el conjunto de ejemplos con que serán evaluados los individuos. En el caso del aprendizaje supervisado se obtienen dos subconjuntos del dominio del problema, el conjunto de entrenamiento y el conjunto de prueba. Los ejemplos de entrenamiento serán utilizados para la evaluación de los

individuos. La función objetivo es la que define la forma en que serán evaluados los individuos. En muchos problemas la función objetivo directa es el conteo de los errores respecto de la salida esperada. En otros casos la medida es una combinación del error, la eficiencia u otros factores que ayuden a diferenciar las soluciones [24].

4.4.4 Asignación de aptitud

La asignación de aptitud se hace con el fin de escalar las evaluaciones de los individuos. El *Ranking Lineal* [38] y el *Ranking Exponencial* [1], son métodos para asignación de aptitud. Para el experimento se utilizará el *Ranking Lineal*, por lo que primero son ordenados los individuos del peor al mejor de acuerdo con su evaluación. Después se les asigna una aptitud mediante la expresión (4.1).

$$p + \left(\frac{i}{n-1} \right) (m - p) \quad (4.1)$$

donde: n es el número de individuos de la población actual, p es la cantidad de copias esperadas del peor individuo, el índice i indica la posición que ocupa el individuo dentro de la población ordenada y m es la cantidad de copias esperadas del mejor individuo.

4.4.5 Selección

Los individuos están sujetos al principio de reproducción y supervivencia. Los mejor calificados tendrán más posibilidades de ser seleccionados y pasar a la siguiente generación. Algunos de los métodos para realizar la selección son: la Ruleta [19], el Estocástico Universal [1] y el Torneo [20].

- Ruleta: se representa mediante un círculo dividido con las probabilidades de selección de los individuos. Cada probabilidad es el cociente de la evaluación del individuo y la suma de evaluaciones de la población. La Figura 4.6 muestra el ejemplo de la distribución de probabilidades de una población con 7 individuos. La suma acumulada de probabilidades partiendo del primer individuo define los rangos. La selección se hace generando un número aleatorio entre 0 y 1. Se ubica este valor en alguno de los rangos para saber que individuo pasa a la siguiente población. La población se completa generando todos los números aleatorios necesarios. En la siguiente población pueden aparecer varias copias de un mismo individuo.
- Estocástico Universal. Teniendo la distribución de probabilidades se genera un número aleatorio entre 0 y un valor llamado intervalo de selección obtenido mediante la expresión (4.2).

$$i = \frac{n}{m} \quad (4.2)$$

siendo i el intervalo de selección, n la suma de probabilidades de los individuos y m el total de individuos a seleccionar. El primer individuo seleccionado se obtiene al ubicar el número aleatorio en los rangos de probabilidades. Los individuos restantes se obtienen sumando sucesivamente el intervalo de selección al número aleatorio. Gráficamente se observa el método en la Figura 4.7.

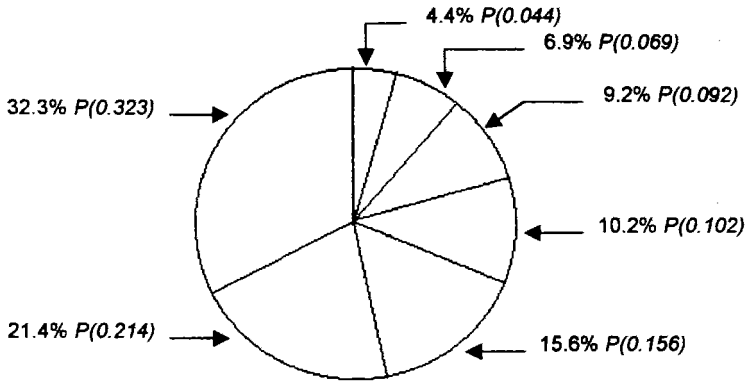


Figura 4.6. Ruleta.

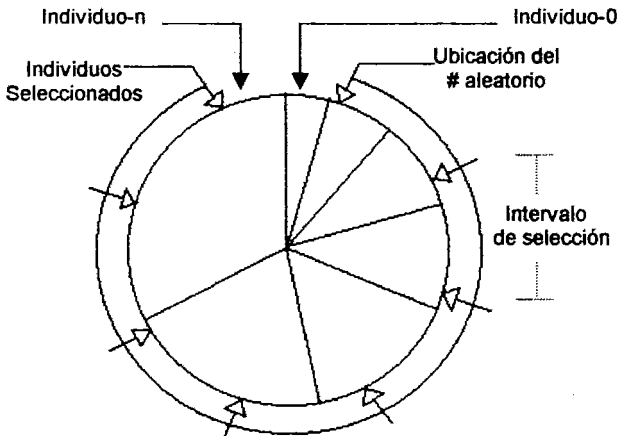


Figura 4.7. Estocástico universal.

- Torneo: en este método se agrupa una cantidad k de individuos para competir, siendo el ganador del torneo el individuo con la mayor aptitud. El

torneo se repite hasta completar el tamaño de la población, siendo posible la participación de todos los individuos en cada torneo.

4.4.6 Cruza

La cruza se utiliza para crear nuevos individuos mediante el intercambio de información de dos individuos de la población [2][24].

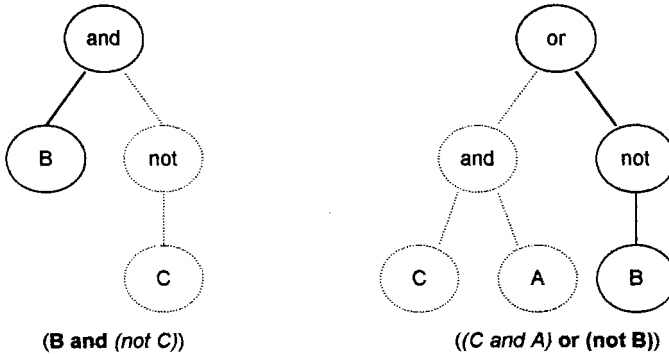


Figura 4.8. Seleccionar punto de corte y subárbol de dos individuos.

El porcentaje de cruza indica la probabilidad de los individuos para participar en la cruza. Para realizar la cruza se seleccionan dos individuos y en cada uno se selecciona aleatoriamente un punto de corte. A partir de los puntos de corte, en cada individuo se identifican los subárboles como lo muestra la Figura 4.8. En la Figura 4.9 se hace el intercambio de los subárboles para crear dos nuevos individuos. Después de hacer el intercambio no se debe rebasar la profundidad máxima establecida.

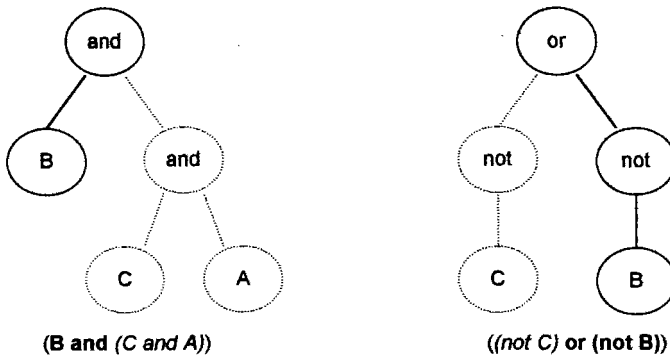


Figura 4.9 Intercambiar los subárboles seleccionados.

4.4.7 Mutación

El operador de mutación busca agregar variedad en la población con nuevos individuos al modificar sus funciones y/o terminales. El porcentaje de mutación determina la probabilidad de los individuos de ser mutados. La mutación en un individuo se realiza al seleccionar aleatoriamente un punto de corte para eliminar el subárbol correspondiente y sustituirlo por otro generado aleatoriamente [2]. No se debe rebasar la profundidad máxima establecida al aplicar la mutación. El proceso se observa en la Figura 4.10.

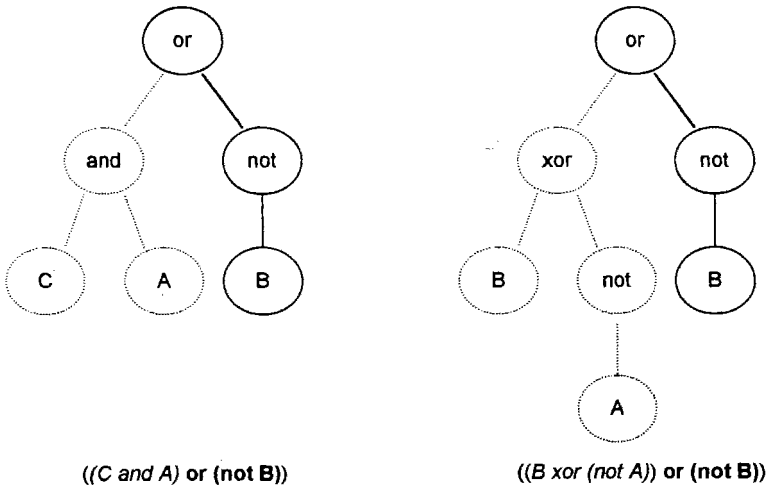


Figura 4.10 Mutación.

4.4.8 Elitismo

El espacio de soluciones en *PG* está conformado por la generación actual de individuos después de un ciclo de ejecución. El elitismo es una estrategia para pasar de una generación a la siguiente una copia de cierto porcentaje de individuos sin ser modificados. El objetivo es conservar una copia de los mejores individuos mientras no aparezcan otros que los superen [7].

4.5 Aplicación de la *PG* en clasificación

La *PG* obtiene el antecedente de la regla de clasificación para una clase en cada ejecución. Los individuos representan el antecedente de la regla, mientras el consecuente queda definido por la clase que se quiera predecir [5][16].

El total de ejemplos se divide aleatoriamente en dos conjuntos disjuntos, el conjunto de entrenamiento para crear el clasificador y el conjunto de prueba para evaluar su desempeño. No se busca un individuo que memorice las salidas

correctas para el conjunto de entrenamiento, sino que el mejor individuo sea capaz de predecir las salidas de los datos de prueba. La capacidad de predicción del clasificador indica aprendizaje y generalización de los ejemplos [16].

Los problemas de clasificación donde los ejemplos sólo pertenecen a una de dos clases se pueden resolver como una tabla de verdad. La *PG* ha sido aplicada para obtener funciones booleanas de diferentes tablas de verdad [12][26][31][35]. La cantidad de datos de la tabla afecta directamente el costo computacional, razón por la que si se cuenta con una gran cantidad de datos es necesario limitar los ejemplos de entrenamiento. El lado opuesto es la insuficiencia de datos, que no permite hacer un buen entrenamiento y validación. Si se toman pocos ejemplos para entrenamiento puede suceder un sobreentrenamiento, por lo que la evaluación de los individuos y la selección de los calificados como mejores sería incorrecta. Por el contrario, ocupar muchos ejemplos es costoso y además es imposible evaluar el total de ejemplos que pueden existir. Los métodos de partición y duplicidad de ejemplos son aplicados para resolver estos problemas [9][36]. La distribución de ejemplos entre las clases es un factor que afecta la cantidad y proporción de ejemplos que deben utilizarse para entrenamiento [37].

Se han resuelto problemas de clasificación aplicando *PG*. Algunos utilizan la validación cruzada de *k-partes* para validar sus resultados, donde *k* toma el valor de 10 [14][13]. Otros utilizan la misma cantidad de ejemplos para entrenamiento y prueba [10][25] u otra proporción [5][6][16][23].

Se observa que no se tiene un criterio para especificar por qué utilizar cierta proporción de datos para entrenamiento. A través del experimento del siguiente capítulo, se busca una proporción que mantenga el mismo desempeño que con el total de los datos.

CAPÍTULO 5

EXPERIMENTO

5.1 Caso de Estudio

Se tiene una base de datos de autos donde la combinación de distintos atributos determinan la clase a la que pertenece cada auto [4]. Para cada clase se buscó el modelo de clasificación a través de la regla:

SI <antecedente> ENTONCES <consecuente>

Con este tipo de regla se responde si un auto pertenece o no a una clase, validando el antecedente con los atributos y el consecuente con la clase.

5.1.1 Base de datos de autos

Los atributos y valores que describen a cada auto son mostrados en la Tabla 5.1.

Tabla 5.1. Atributos y valores.

ATRIBUTO	VALORES				# VALORES
COSTO	<i>muy alto</i>	<i>alto</i>	<i>medio</i>	<i>bajo</i>	4
MANTENIMIENTO	<i>muy alto</i>	<i>alto</i>	<i>medio</i>	<i>bajo</i>	4
PUERTAS	2	3	4	5 <i>más</i>	4
PERSONAS	2	4	<i>más</i>		3
CAJUELA	<i>chica</i>	<i>mediana</i>	<i>grande</i>		3
SEGURIDAD	<i>baja</i>	<i>media</i>	<i>alta</i>		3

El total de ejemplos son 1728, que se obtienen al multiplicar el número de valores de los atributos (4x4x4x3x3x3). La clase a que pertenece cada ejemplo es una de las cuatro presentadas en la Tabla 5.2, donde se muestra también la distribución de los ejemplos entre las clases.

Tabla 5.2. Distribución de los ejemplos entre las clases.

CLASE	# DE EJEMPLOS	% DE EJEMPLOS
<i>Inaccesible</i>	1210	70.02
<i>Accesible</i>	384	22.22
<i>Bueno</i>	69	3.99
<i>Muy bueno</i>	65	3.76

La Tabla 5.3 muestra un ejemplo de cada clase:

Tabla 5.3. Ejemplos de cada clase.

COSTO	MANTENIMIENTO	PUERTAS	PERSONAS	CAJUELA	SEGURIDAD	CLASE
<i>muy alto</i>	<i>muy alto</i>	2	2	<i>pequeña</i>	<i>baja</i>	<i>Inaccesible</i>
<i>muy alto</i>	<i>medio</i>	2	4	<i>pequeña</i>	<i>alta</i>	<i>Accesible</i>
<i>medio</i>	<i>bajo</i>	2	4	<i>pequeña</i>	<i>alta</i>	<i>Bueno</i>
<i>medio</i>	<i>medio</i>	2	4	<i>grande</i>	<i>alta</i>	<i>Muy bueno</i>

5.1.2 Transformación de los datos

La base de datos se transformó de valores nominales a valores binarios para buscar la función booleana que clasifique los ejemplos de cada clase. En la transformación se utilizaron dos bits para codificar cada valor de los atributos. La transformación del *Costo*, *Mantenimiento*, *Puertas*, *Personas*, *Cajuela* y *Seguridad* se muestra en la Tabla 5.4, Tabla 5.5, Tabla 5.6, Tabla 5.7, Tabla 5.8 y Tabla 5.9, respectivamente.

Tabla 5.4.
Transformación del *Costo*.

ATRIBUTO	VARIABLES BOOLEANAS
<i>Costo</i>	A B
<i>muy alto</i>	0 0
<i>alto</i>	0 1
<i>medio</i>	1 0
<i>bajo</i>	1 1

Tabla 5.5.
Transformación del *Mantenimiento*.

ATRIBUTO	VARIABLES BOOLEANAS
<i>Mantenimiento</i>	C D
<i>muy alto</i>	0 0
<i>alto</i>	0 1
<i>medio</i>	1 0
<i>bajo</i>	1 1

Tabla 5.6.
Transformación del # de *Puertas*.

ATRIBUTO	VARIABLES BOOLEANAS
<i>Puertas</i>	E F
2	0 0
3	0 1
4	1 0
<i>5 más</i>	1 1

Tabla 5.7.
Transformación del # de *Personas*.

ATRIBUTO	VARIABLES BOOLEANAS
<i>Personas</i>	G H
2	0 0
4	0 1
<i>más</i>	1 0

Tabla 5.8. Transformación del tamaño de la *Cajuela*. Tabla 5.9. Transformación del nivel de *Seguridad*.

ATRIBUTO	VARIABLES BOOLEANAS
<i>Cajuela</i>	I J
<i>Pequeña</i>	0 0
<i>Mediana</i>	0 1
<i>Grande</i>	1 0

ATRIBUTO	VARIABLES BOOLEANAS
<i>Seguridad</i>	K L
<i>baja</i>	0 0
<i>media</i>	0 1
<i>alta</i>	1 0

El consecuente de la regla que representa la clase a la que pertenece cada ejemplo, se codificó con un bit, donde un 1 indica pertenencia a la clase y un 0 lo

contrario. La transformación binaria de algunos ejemplos se muestra en la Tabla 5.10.

Tabla 5.10. Transformación a binario.

VARIABLES BINARIAS												CLASE			
A	B	C	D	E	F	G	H	I	J	K	L	Inaccesible	Accesible	Bueno	Muy bueno
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
.
.
.
1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	1

5.2 División de ejemplos

Se evaluaron cinco casos diferentes de porcentajes de entrenamiento de acuerdo con la Tabla 5.11.

Tabla 5.11. Casos evaluados.

CASO	% DE ENTRENAMIENTO
Caso ₁	20%
Caso ₂	40%
Caso ₃	60%
Caso ₄	80%
Caso ₅	100%

El Caso₅ se tomó como testigo, ya que no tiene limitaciones de información para crear el clasificador. En los otros cuatro casos, la PG es entrenada con un tamaño menor de ejemplos y evaluada con los ejemplos restantes.

5.3 Características de la PG ejecutada

Para validar los resultados, la PG se ejecutó 30 veces para cada clase y caso de entrenamiento. La generación de la población inicial se hizo por el método de mitad-mitad. Se evaluaron los individuos mediante la función objetivo (5.1), que mide la correlación estandarizada [10][25], más un cociente que mide la simplicidad de la función en términos del número de elementos [6]:

$$f = \left[\left(\frac{1-r}{2} \right) (100) + \left(\frac{n}{m} \right) \right] \tag{5.1}$$

donde r es la correlación, 100 es una constante que da más peso al primer sumando, n es el número de elementos del individuo y m es el número de elementos máximos posibles de acuerdo con la profundidad máxima establecida. Se aplicó asignación de aptitud mediante *ranking lineal*, para después hacer la selección utilizando el método estocástico universal. Se aplicó cruza, mutación y elitismo.

5.3.1 Parámetros de ejecución

En todas las ejecuciones, el conjunto de funciones quedó conformado por las funciones booleanas (*AND*, *OR*, *XOR*, *NOT*, *NAND*, *NOR*, *NXOR*) y la función (*IF-THEN-ELSE*). La aridad de las funciones es 2, excepto el *NOT* de aridad 1 y el *IF* de aridad 3. El conjunto de terminales quedó determinado por las variables booleanas obtenidas de la transformación de la base de datos (A,B,...,L). Se aplicó un porcentaje de 90% en cruza [6][14] y 10% en mutación [6]. El elitismo se aplicó al guardar una copia del mejor individuo después de cada generación. Los parámetros de número de individuos, número de generaciones, profundidad inicial y profundidad máxima se establecieron para cada clase como se muestra en la Tabla 5.12.

Tabla 5.12. Parámetros de ejecución para las clases.

CLASE	# INDIVIDUOS	# GENERACIONES	PROFUNDIDAD INICIAL	PROFUNDIDAD MÁXIMA
<i>Inaccesible</i>	200	800	5	6
<i>Accesible</i>	200	600	6	7
<i>Bueno</i>	200	500	5	6
<i>Muy Bueno</i>	200	500	5	6

5.3.2 Evaluación con los ejemplos de prueba

El mejor individuo de cada ejecución se evalúa con los ejemplos de prueba en aciertos y errores mediante las siguientes variables [5][8][10]: *TP* (*True-Positive*), *TN* (*True-Negative*), *FP* (*False-Positive*), *FN* (*False Negative*), que se observan en la Figura 5.1.

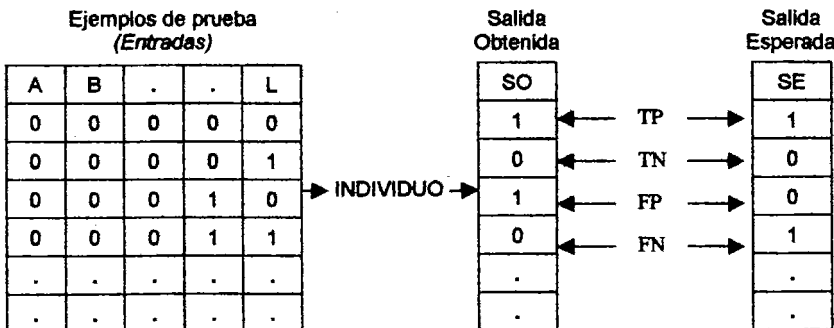


Figura 5.1. Evaluación del mejor individuo.

Las variables anteriores se utilizan para obtener el desempeño del clasificador en términos de las siguientes medidas [21]:

- *Sensibilidad*: porcentaje de ejemplos positivos (1's) bien clasificados.
- *Especificidad*: porcentaje de ejemplos negativos (0's) bien clasificados.
- *Precisión*: porcentaje de ejemplos clasificados como positivos que son realmente positivos.
- *Exactitud*: porcentaje de clasificaciones correctas.

5.4 Análisis de la varianza

El promedio de desempeño en *sensibilidad*, *especificidad*, *precisión* y *exactitud* de las 30 ejecuciones para cada clase se utilizó para realizar el análisis de la varianza entre cada caso evaluado (Caso₁ 20%, Caso₂ 40%, Caso₃ 60%, Caso₄ 80%) y el caso testigo (Caso₅ 100%).

Para el análisis de la varianza se establece la hipótesis nula (H_0) y su hipótesis alternativa (H_1) que se indican como [11]:

- *Hipótesis nula*: afirma la igualdad entre las medias estadísticas de distintos grupos de datos, como se observa en la expresión (5.2).

$$H_0 : m_1 = m_2 \text{ ó bien } H_0 : m_1 - m_2 = 0 \quad (5.2)$$

- *Hipótesis alternativa*: está indica que existe diferencia estadística entre las medias de distintos grupos de datos, se indica mediante la expresión (5.3).

$$H_1 : m_1 \neq m_2 \quad (5.3)$$

Se hace el análisis para determinar si se acepta o se rechaza la hipótesis nula. Para decidir sobre la hipótesis nula se calcula el contraste de la *F de Snedecor* como se ejemplifica en el Anexo E. Se compara el valor calculado con el de la tabla de la distribución de *Snedecor* a un nivel de significancia (probabilidad de equivocación en la decisión). El resultado de la comparación se interpreta con las siguientes reglas:

- SI (el valor calculado es MAYOR al obtenido por la tabla) ENTONCES (se rechaza la hipótesis nula) y POR LO TANTO (hay diferencia, es decir se acepta la hipótesis alternativa).
- SI (el valor calculado es MENOR al obtenido por la tabla) ENTONCES (se acepta la hipótesis nula) y POR LO TANTO (no hay diferencia, rechazándose la hipótesis alternativa).

CAPÍTULO 6

RESULTADOS

Los resultados de las 30 ejecuciones se presentan como el individuo Peor, Mejor, Promedio y la Desviación Estándar respecto de la variable medida. Se presentan los resultados del caso testigo. En los casos evaluados se muestra además el cálculo y la interpretación del análisis de la varianza mediante el contraste de la *F* de *Snedecor* contra el caso testigo. La comparación e interpretación del valor calculado se hizo con la tabla de distribución de *Snedecor* a un nivel de significancia del 1%, cuyo valor es 7.09.

6.1. Desempeño en sensibilidad

Los valores de *sensibilidad* para los casos testigos de las diferentes clases se muestran en la Tabla 6.1. El resultado del análisis de la varianza en *sensibilidad* para los casos evaluados se presentan en la Tabla 6.2, Tabla 6.3, Tabla 6.4 y Tabla 6.5.

Tabla 6.1. Resultados de *sensibilidad* en el Caso₅ testigo, con 100% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR
<i>Inaccesible</i>	0.7933	0.9867	0.9278	0.0575
<i>Accesible</i>	0.8359	1	0.969	0.0418
<i>Bueno</i>	0.5217	1	0.784	0.1517
<i>Muy bueno</i>	0.7384	1	0.926	0.0910

Tabla 6.2. Resultados de *sensibilidad* en el Caso₁, con 20% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8689	0.9885	0.9333	0.0318	0.214	No
<i>Accesible</i>	0.7292	1	0.861	0.0909	34.9623	<i>Si</i>
<i>Bueno</i>	0.109	0.9464	0.4654	0.2446	36.741	<i>Si</i>
<i>Muy bueno</i>	0.1803	1	0.6716	0.2425	28.9569	<i>Si</i>

Tabla 6.3. Resultados de *sensibilidad* en el Caso₂, con 40% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.795	0.9748	0.9119	0.0399	1.5487	No
<i>Accesible</i>	0.7901	1	0.9195	0.0650	12.3011	<i>Si</i>
<i>Bueno</i>	0.3488	1	0.6514	0.1725	9.991	<i>Si</i>
<i>Muy bueno</i>	0.7173	1	0.8916	0.0872	2.2489	No

Tabla 6.4. Resultados de *sensibilidad* en el Caso₃, con 60% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8455	0.985	0.9341	0.0343	0.2714	No
<i>Accesible</i>	0.4691	1	0.9343	0.1039	2.8886	No
<i>Bueno</i>	0.2121	1	0.6765	0.2306	4.5512	No
<i>Muy bueno</i>	0.7419	1	0.8779	0.0720	5.1701	No

Tabla 6.5. Resultados de *sensibilidad* en el Caso₄, con 80% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8888	0.9844	0.9419	0.0276	1.4632	No
<i>Accesible</i>	0.4342	1	0.894	0.1502	6.9339	No
<i>Bueno</i>	0.3571	1	0.7293	0.2045	1.3843	No
<i>Muy bueno</i>	0.7857	1	0.939	0.0688	0.3806	No

En la Figura 6.1 se muestra el desempeño gráficamente de los promedios y se resaltan los casos que establecieron diferencia con el caso testigo en cuanto a *sensibilidad*.

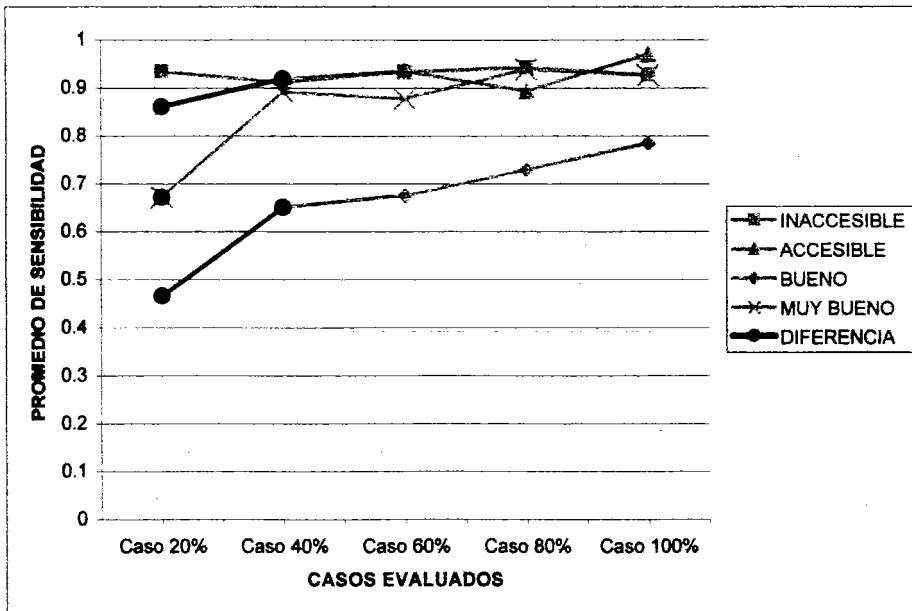


Figura 6.1. Desempeño de los promedios en *sensibilidad*.

Se observa que para todas las clases utilizando un porcentaje entre el 40% y el 60% de ejemplos seleccionados al azar se obtienen desempeños iguales en *sensibilidad* que con el 100%. Incluso para la clase *Inaccesible* se puede utilizar menos del 20% de ejemplos.

Se debe señalar que en la clase *Accesible* la media del Caso₂ es mayor que la del Caso₄ y sin embargo sólo el primero establece diferencia de desempeño con el caso de referencia. Se observaron las ejecuciones para tratar de explicar este resultado y la respuesta posible es el número de valores de 1 (máximo valor *sensibilidad*) en los diferentes casos. En la Figura 6.2 se muestra que el Caso₂ obtuvo 6 valores de 1 en las 30 ejecuciones, en tanto que el Caso₄ obtuvo 13, más cercano este último con los 12 que se obtuvieron en el caso de testigo (Casos₅).

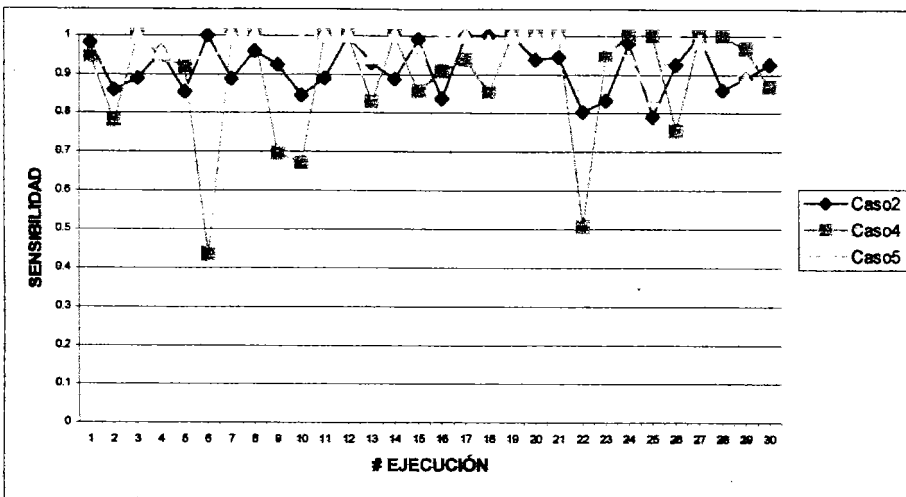


Figura 6.2. Desempeño de las 30 ejecuciones en *sensibilidad*.

Los resultados de las 30 ejecuciones en cuanto a *sensibilidad* para las diferentes clases y porcentajes evaluados son presentados en el Anexo A.1, Anexo A.2, Anexo A.3, Anexo A.4 y Anexo A.5. En la Figura 6.3 se muestra la distribución de estos resultados mediante un histograma.

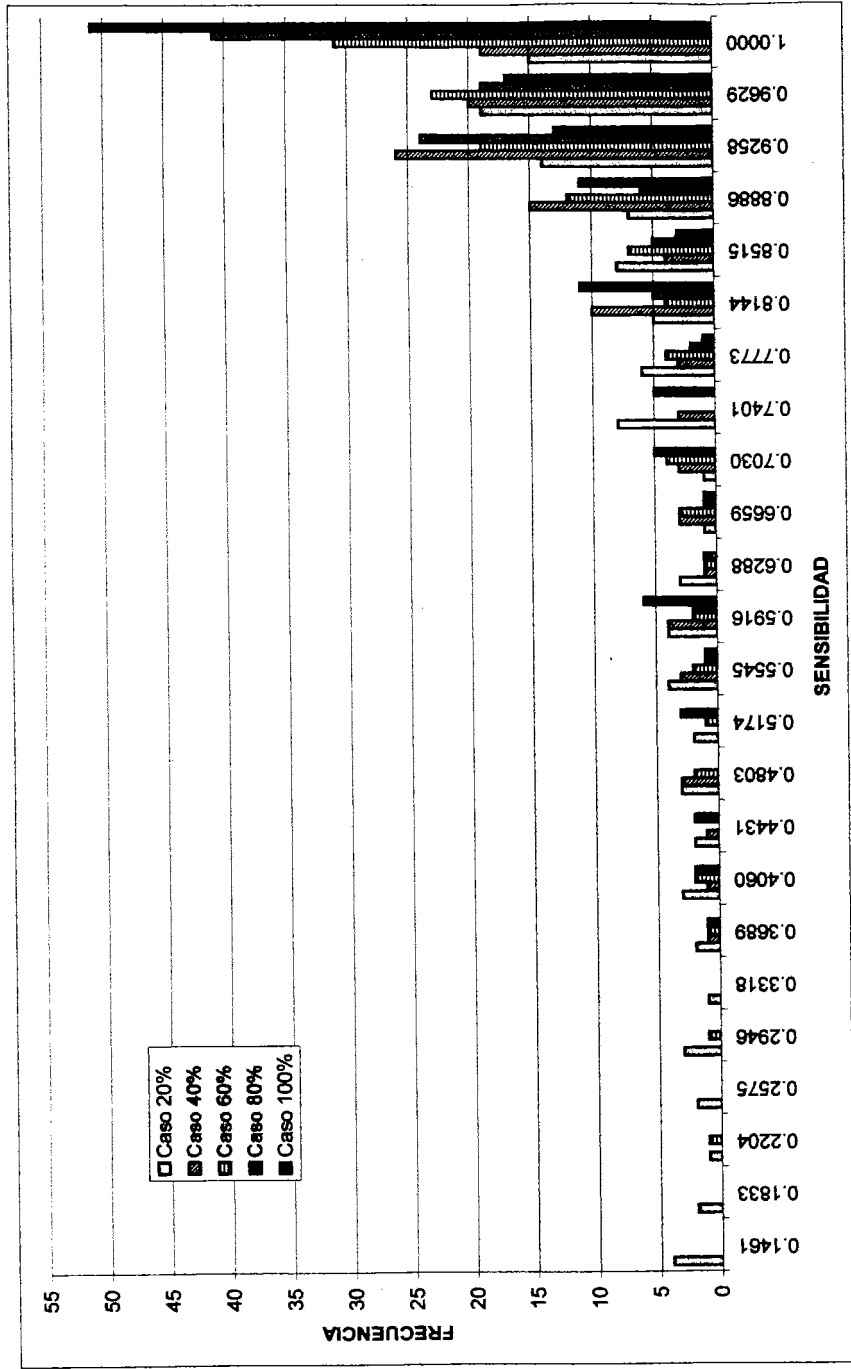


Figura 6.3 Desempeño de las ejecuciones en sensibilidad.

6.2 Desempeño en especificidad

Para la *especificidad*, los resultados de los casos testigos los presenta la Tabla 6.6. Se realizó el análisis de la varianza en *especificidad* obteniéndose los resultados de la Tabla 6.7, Tabla 6.8, Tabla 6.9 y Tabla 6.10 para los diferentes porcentajes de los casos evaluados.

Tabla 6.6. Resultados de *especificidad* en el Caso₅ testigo, con 100% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR
<i>Inaccesible</i>	0.8339	1	0.9491	0.0453
<i>Accesible</i>	0.7142	0.965	0.8411	0.0653
<i>Bueno</i>	0.9535	1	0.9884	0.0125
<i>Muy bueno</i>	0.9759	1	0.9966	0.0059

Tabla 6.7. Resultados de *especificidad* en el Caso₁, con 20% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.5576	1	0.92	0.0843	2.767	No
<i>Accesible</i>	0.7037	0.9442	0.8456	0.0634	0.0747	No
<i>Bueno</i>	0.9592	0.9977	0.9844	0.0091	2.0057	No
<i>Muy bueno</i>	0.9781	1	0.9915	0.0066	9.8434	Si

Tabla 6.8. Resultados de *especificidad* en el Caso₂, con 40% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8407	1	0.9498	0.0416	0.0037	No
<i>Accesible</i>	0.4289	0.9551	0.8384	0.1024	0.0147	No
<i>Bueno</i>	0.9696	1	0.9893	0.0088	0.106	No
<i>Muy bueno</i>	0.98	1	0.9971	0.0044	0.1256	No

Tabla 6.9. Resultados de *especificidad* en el Caso₃, con 60% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.5911	1	0.9151	0.0779	4.2704	No
<i>Accesible</i>	0.5724	0.983	0.8366	0.0890	0.049	No
<i>Bueno</i>	0.9171	1	0.9882	0.0168	0.0027	No
<i>Muy bueno</i>	0.9754	1	0.9977	0.0049	0.5417	No

Tabla 6.10. Resultados de *especificidad* en el Caso₄, con 80% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8301	1	0.9466	0.0447	0.0485	No
<i>Accesible</i>	0.6851	0.9822	0.8516	0.0803	0.3125	No
<i>Bueno</i>	0.9642	1	0.9943	0.0073	4.9411	No
<i>Muy bueno</i>	0.9887	1	0.9984	0.0032	1.9352	No

El resultado del análisis de la varianza para *especificidad* se observa en la Figura 6.4, donde los casos que tuvieron diferencia con el caso testigo son resaltados.

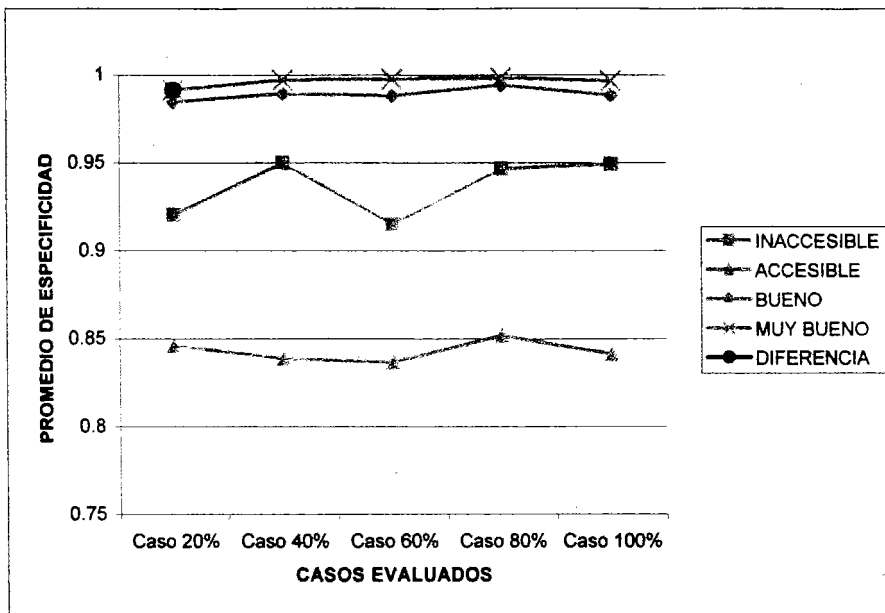


Figura 6.4. Desempeño de los promedios en *especificidad*.

En cuanto a *especificidad*, el rango común entre las cuatro clases esta entre el 20% y el 40%, donde mantienen un mismo desempeño que con el 100% de los ejemplos. La clase *Muy Bueno* es la que establece ese rango, ya que las otras tres clases pueden utilizar menos del 20%.

En las 30 ejecuciones en cuanto a *especificidad* se obtuvieron los resultados del Anexo B.1, Anexo B.2, Anexo B.3, Anexo B.4 y Anexo B.5, para las diferentes clases y porcentajes evaluados. La distribución de estos resultados se presenta en el histograma de la Figura 6.5.

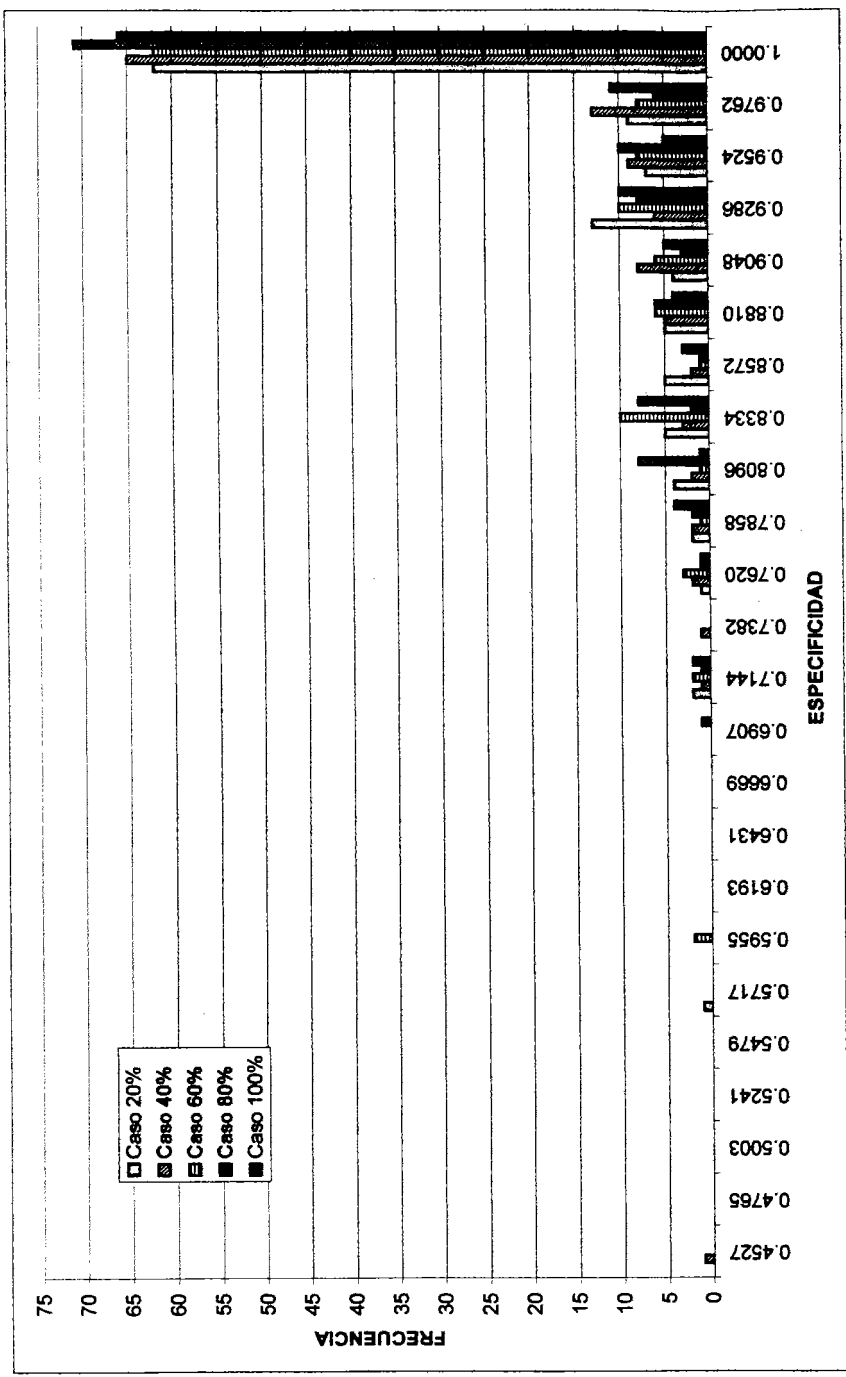


Figura 6.5. Desempeño de las ejecuciones en especificidad.

6.3 Desempeño en *precisión*

La Tabla 6.11 muestra los resultados de *precisión* de los casos testigos en cada clase. Los resultados del análisis de la varianza realizado sobre los promedios de los valores de *precisión* para los casos evaluados son presentados en la Tabla 6.12, Tabla 6.13, Tabla 6.14 y Tabla 6.15.

Tabla 6.11. Resultados de *precisión* en el Caso₅ testigo, con 100% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR
<i>Inaccesible</i>	0.9328	1	0.9779	0.0188
<i>Accesible</i>	0.5	0.8722	0.6507	0.0949
<i>Bueno</i>	0.4652	1	0.7938	0.1526
<i>Muy Bueno</i>	0.5833	1	0.933	0.1097

Tabla 6.12. Resultados de *precisión* en el Caso₁, con 20% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8343	1	0.9661	0.0322	3.0108	No
<i>Accesible</i>	0.4789	0.8032	0.6284	0.0837	0.9302	No
<i>Bueno</i>	0.25	0.9062	0.5569	0.1653	33.2448	<i>Si</i>
<i>Muy bueno</i>	0.3333	1	0.7634	0.1675	21.5081	<i>Si</i>

Tabla 6.13. Resultados de *precisión* en el Caso₂, con 40% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.9306	1	0.9774	0.0183	0.0139	No
<i>Accesible</i>	0.3216	0.8487	0.6415	0.1108	0.1207	No
<i>Bueno</i>	0.4166	1	0.7564	0.1639	0.8364	No
<i>Muy bueno</i>	0.6825	1	0.9363	0.0856	0.0171	No

Tabla 6.14. Resultados de *precisión* en el Caso₃, con 60% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8333	1	0.9635	0.0318	4.5745	No
<i>Accesible</i>	0.3868	0.8941	0.6425	0.1183	0.0877	No
<i>Bueno</i>	0.3373	1	0.7868	0.1728	0.0273	No
<i>Muy bueno</i>	0.5142	1	0.9496	0.1010	0.372	No

Tabla 6.15. Resultados de *precisión* en el Caso₄, con 80% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.9274	1	0.9748	0.0198	0.3941	No
<i>Accesible</i>	0.472	0.9074	0.6565	0.1201	0.0427	No
<i>Bueno</i>	0.5862	1	0.8791	0.1135	6.0396	No
<i>Muy bueno</i>	0.6923	1	0.968	0.0675	2.2037	No

En la Figura 6.6 se resaltan los casos que de acuerdo con el análisis de la varianza tuvieron diferencia con el caso testigo respecto a la *precisión*.

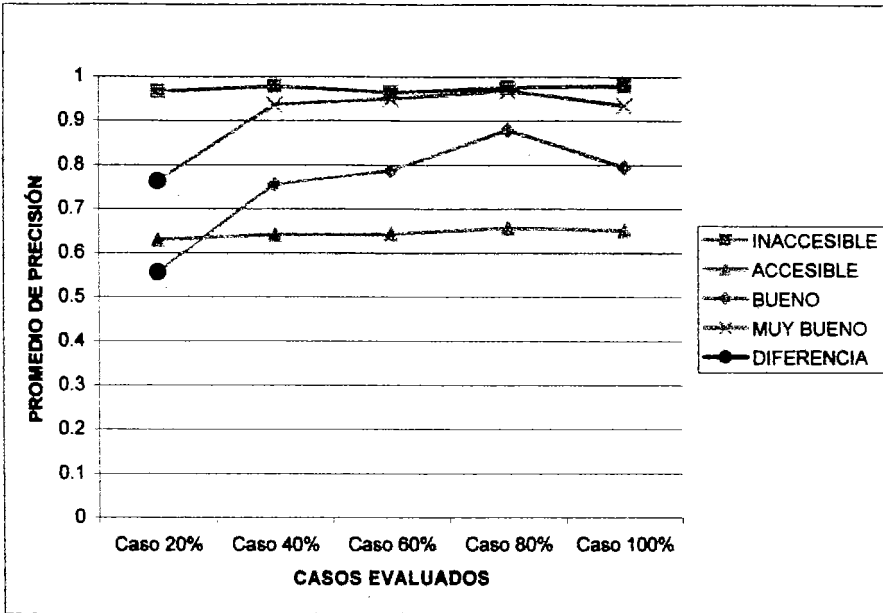


Figura 6.6. Desempeño de los promedios en *precisión*.

Utilizando una cantidad de ejemplos entre el 20% y el 40% se obtiene el mismo resultado en *precisión* que si se utiliza el 100%. Las clases *Inaccesible* y *Accesible* trabajan bien aún con menos del 20% de ejemplos.

En el Anexo C.1, Anexo C.2, Anexo C.3, Anexo C.4 y Anexo C.5 se muestran los resultados de las 30 ejecuciones en cuanto a *precisión* para las diferentes clases y porcentajes evaluados. El histograma de la Figura 6.7 muestra la distribución de estos resultados.

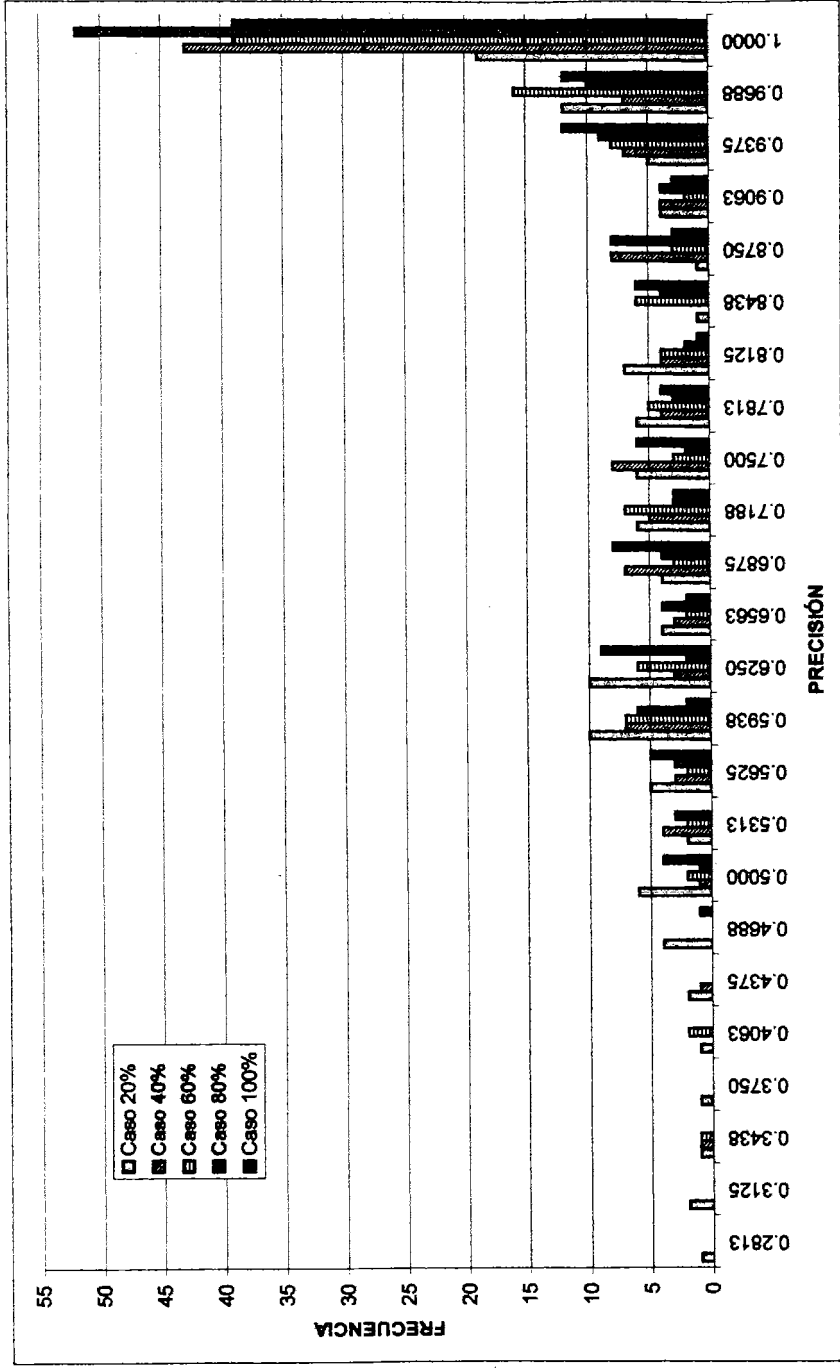


Figura 6.7. Desempeño de las ejecuciones en precisión.

6.4 Desempeño en exactitud

Los casos testigos en cuanto a *exactitud* los muestra la Tabla 6.16. El análisis de la varianza para los casos evaluados en cuanto a *exactitud* se presentan en la Tabla 6.17, Tabla 6.18, Tabla 6.19 y Tabla 6.20.

Tabla 6.16. Resultados de *exactitud* en el Caso₅ testigo, con 100% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR
<i>Inaccesible</i>	0.8136	0.9733	0.9342	0.0355
<i>Accesible</i>	0.7777	0.9467	0.8695	0.0451
<i>Bueno</i>	0.9542	0.9913	0.9803	0.0092
<i>Muy Bueno</i>	0.9716	1	0.994	0.0065

Tabla 6.17. Resultados de *exactitud* en el Caso₁, con 20% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8561	0.967	0.9291	0.02379	0.4222	No
<i>Accesible</i>	0.7656	0.9117	0.8487	0.0357	3.9343	No
<i>Bueno</i>	0.9421	0.99	0.963	0.0118	39.3861	<i>Sí</i>
<i>Muy bueno</i>	0.9479	0.9964	0.9788	0.0124	34.9928	<i>Sí</i>

Tabla 6.18. Resultados de *exactitud* en el Caso₂, con 40% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8563	0.971	0.9231	0.0274	1.8241	No
<i>Accesible</i>	0.5506	0.9334	0.8559	0.0714	0.776	No
<i>Bueno</i>	0.9527	0.9951	0.9754	0.0092	4.0798	No
<i>Muy bueno</i>	0.979	1	0.9929	0.0053	0.4603	No

Tabla 6.19. Resultados de *exactitud* en el Caso₃, con 60% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.8569	0.9725	0.9283	0.0240	0.5588	No
<i>Accesible</i>	0.6632	0.947	0.8574	0.0588	0.7958	No
<i>Bueno</i>	0.9205	0.9927	0.9738	0.0138	4.5328	No
<i>Muy bueno</i>	0.9719	0.9985	0.9931	0.0057	0.3193	No

Tabla 6.20. Resultados de *exactitud* en el Caso₄, con 80% de ejemplos en entrenamiento.

CLASE	PEOR	MEJOR	PROMEDIO	DESVIACIÓN ESTÁNDAR	F CALCULADA	DIFERENCIA
<i>Inaccesible</i>	0.9034	0.9855	0.9447	0.0201	1.9979	No
<i>Accesible</i>	0.7543	0.9235	0.861	0.0448	0.5419	No
<i>Bueno</i>	0.9605	0.9972	0.9825	0.0105	0.7901	No
<i>Muy bueno</i>	0.9826	1	0.9959	0.0044	1.6704	No

El desempeño en *exactitud* y los casos que establecieron diferencia con el caso testigo se muestra en la Figura 6.8.

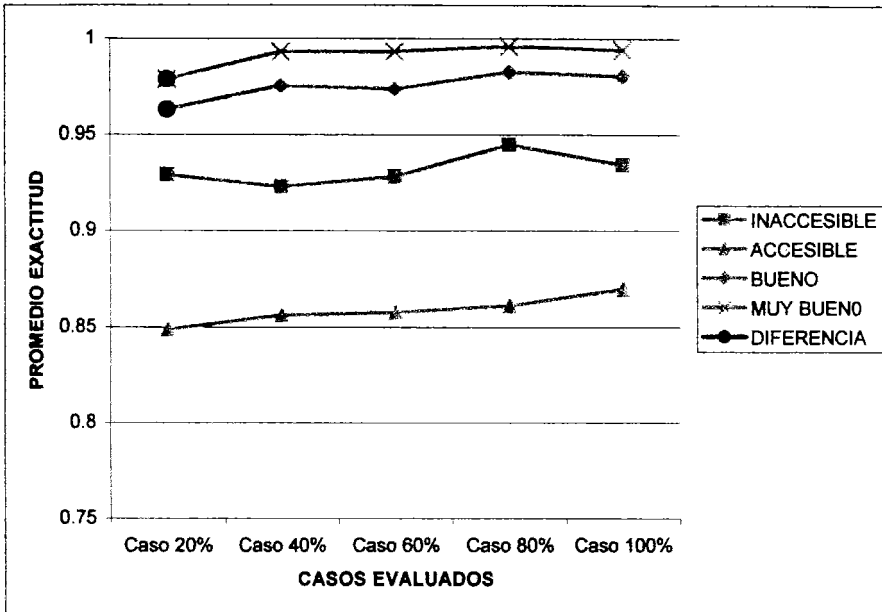


Figura 6.8. Desempeño de los promedios en *exactitud*.

Al seleccionar entre 20% y 40% de ejemplos el desempeño en *exactitud* se mantiene igual que con el 100%. Aquí las clases *Inaccesible* y *Accesible* mantienen un mismo desempeño por debajo del 20%.

Para la *exactitud*, los resultados de las 30 ejecuciones para las diferentes clases y porcentajes evaluados son presentados en el Anexo D.1, Anexo D.2, Anexo D.3, Anexo D.4 y Anexo D.5. La distribución de estos resultados se muestra en la Figura 6.9 mediante un histograma.

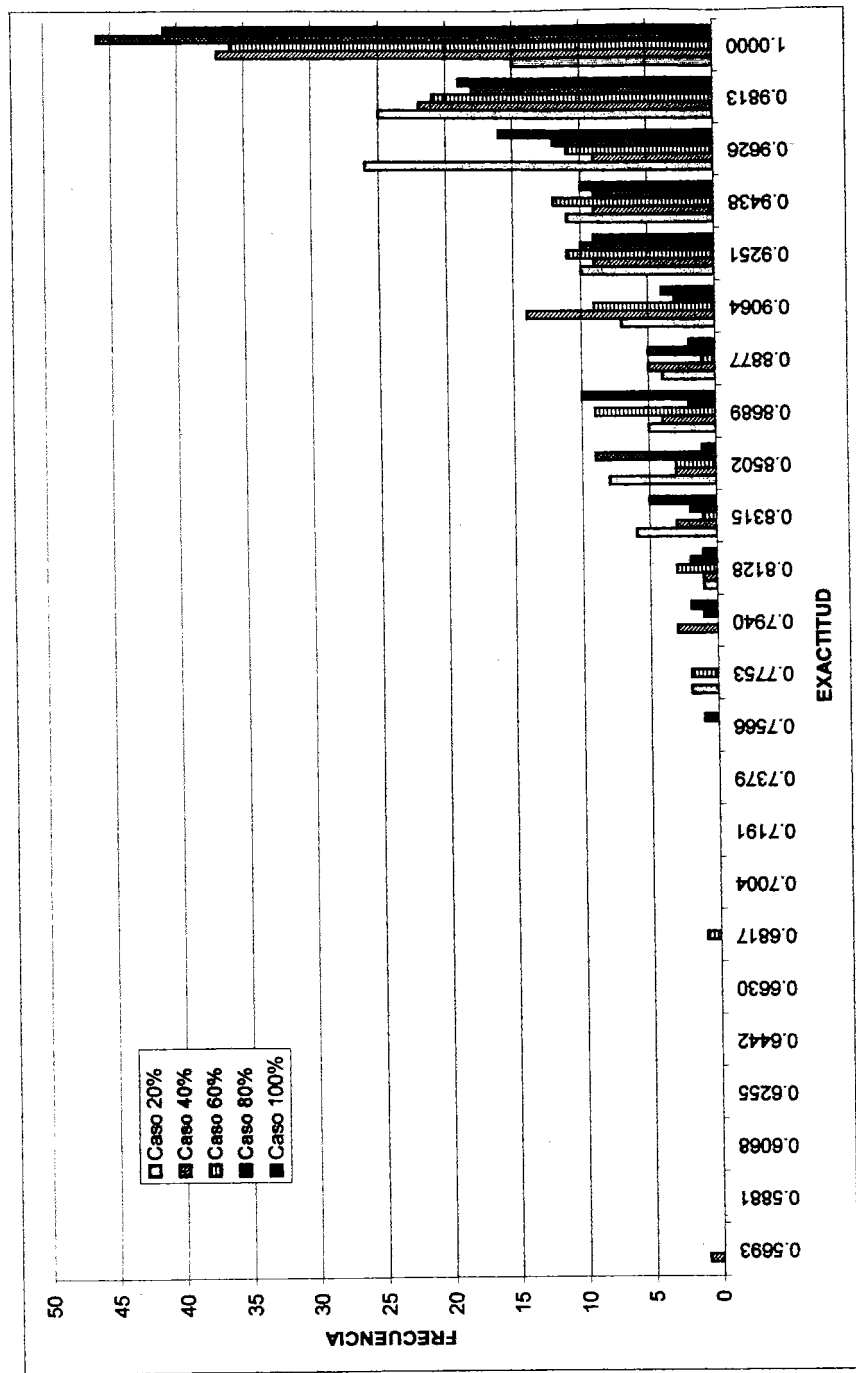


Figura 6.9. Desempeño de las ejecuciones en exactitud.

6.5 Mejores Individuos

El 60%, fue el porcentaje de entrenamiento en donde el desempeño de las cuatro variables medidas (*sensibilidad, especificidad, precisión y exactitud*) se mantuvo igual que con el caso testigo para todas las clases. Por lo anterior se muestra en la Tabla 6.21 el individuo que tiene el mejor promedio obtenido con las cuatro medidas para cada clase con un 60% de entrenamiento y en la Tabla 6.22 los individuos del 100%.

Tabla 6.21. Mejores individuos con 60% de ejemplos en entrenamiento.

CLASE	PROMEDIO	REPRESENTACIÓN
<i>Inaccesible</i>	0.9724	nand-nor-nor-H-G-nor-nor-nor-I-A-nor-K-L-K-or-or-and-J-nor-I-K-nor-nor-C-B-not-D-or-C-nor-nor-I-K-nor-A-C-
<i>Accesible</i>	0.8917	xor-if-if-K-nand-xor-H-G-xor-A-not-C-L-xor-H-G-and-and-if-B-and-D-K-B-nand-C-C-xor-H-G-and-and-xor-H-G-if-xor-and-A-C-and-I-L-xor-C-not-K-L-and-and-not-K-xor-H-G-xor-xor-and-A-A-and-A-C-xor-H-G-
<i>Bueno</i>	0.9677	and-and-if-and-and-E-A-or-L-K-and-C-G-and-or-L-E-or-D-B-and-and-and-C-A-or-D-B-and-or-L-K-H-if-and-xor-A-C-and-C-C-nxor-nxor-K-J-J-if-and-C-nand-J-E-and-or-I-K-nand-I-K-and-and-L-A-L-
<i>Muy bueno</i>	0.9877	and-and-and-A-and-K-or-G-H-and-A-nand-nand-E-J-nand-I-K-and-and-and-or-C-D-or-C-B-A-A-

Tabla 6.22. Mejores individuos con 100% de ejemplos en entrenamiento.

CLASE	PROMEDIO	REPRESENTACIÓN
<i>Inaccesible</i>	0.9818	nand-or-or-and-or-J-C-or-C-A-xor-or-J-A-nand-D-E-and-or-or-J-A-or-D-K-or-or-I-K-C-and-and-or-or-J-A-or-I-K-or-or-A-B-C-and-or-or-D-D-or-C-A-and-or-H-G-xor-K-L-
<i>Accesible</i>	0.9199	and-and-nor-and-nxor-L-L-nxor-K-L-nor-G-H-nor-nor-or-C-or-J-I-or-xor-B-C-or-J-K-nor-or-xor-I-B-nxor-K-I-or-xor-I-C-xor-I-A-and-nor-nor-B-or-A-or-C-B-nor-C-or-A-D-nor-nor-G-H-nor-if-L-xor-I-J-xor-C-A-if-L-A-xor-K-C-
<i>Bueno</i>	0.9415	and-and-and-if-B-C-D-if-G-A-if-J-E-A-and-C-if-A-if-K-A-L-A-and-and-nxor-I-xor-L-J-if-C-if-G-C-H-I-C-
<i>Muy bueno</i>	1	and-and-K-and-or-or-E-I-and-G-F-and-or-B-C-A-and-A-and-or-C-D-and-or-H-G-or-J-I-

CAPÍTULO 7

DISCUSIÓN, CONCLUSIONES Y TRABAJOS FUTUROS

7.1 Discusión

- La *Programación Genética* ha sido aplicada en problemas de clasificación, donde el reto principal es lograr la generalización al predecir la clase de los ejemplos no conocidos [6][16]. En este trabajo se buscó disminuir la cantidad de ejemplos necesarios para construir clasificadores semejantes en desempeño. El desempeño será semejante entre clasificadores creados con diferentes porcentajes de entrenamiento, si su capacidad de generalización no establece diferencia estadística. Los resultados indican, que para todas las clases se logró la generalización seleccionando al azar un porcentaje de ejemplos para entrenamiento menor al 100%. En las figuras que comparan los promedios de las variables, se observa que la gráfica de la variable en cuestión tiene una inclinación marcada en el porcentaje donde se establece diferencia, haciendo notar que el comportamiento es diferente a los porcentajes superiores. En los histogramas, se observa que las ejecuciones de los porcentajes menores, tienden a distribuirse a lo largo de todos los rangos, mientras que las ejecuciones de los porcentajes más cercanos al 100%, tienden a concentrarse en los rangos mayores del histograma. El comportamiento observado en los histogramas indica que la *PG* tiene un desempeño más estable en todas las ejecuciones si el porcentaje de entrenamiento es mayor. El método de partición de ejemplos *cross validation*, es utilizado en otros trabajos, donde toman un porcentaje de ejemplos de entrenamiento de 90% y 10% para validación [14][13]. En el caso de estudio utilizado, el porcentaje menor de ejemplos de entrenamiento que estableció diferencia estadística fue 40%, mientras que a un 60% el desempeño de las variables medidas (*sensibilidad, especificidad, precisión, exactitud*) no estableció diferencia en ninguna de las clases. El rango general para tomar la muestra de ejemplos de entrenamiento se estableció entre el 40% y el 60%. La evaluación en la creación del clasificador en la *PG*, es la etapa que consume mayor tiempo computacional, porque evalúa durante N generaciones cada individuo de la población con el total de ejemplos de entrenamiento [36]. Por lo anterior, con la disminución de ejemplos de entrenamiento sin pérdidas en la generalización, se obtiene una ganancia en el costo computacional necesario para crear el modelo de clasificación.

- En particular, para cada variable medida en la validación de los clasificadores, la diferencia estadística de desempeño se establece en diferentes porcentajes. Uno de los factores que afecta esta variación es la distribución de los ejemplos entre las clases, principalmente respecto de la predicción de la clase que tiene menos ejemplos, que en la mayoría de los problemas es la más importante [37]. De acuerdo con los resultados obtenidos, se corrobora que la distribución afectó el costo y desempeño de clasificar las clases que tienen menos cantidad de ejemplos. La clase que tiene mayor cantidad de ejemplos, *Inaccesible* con un 70.02%, mantuvo un mismo desempeño en todas las medidas desde el 20% hasta el 100%. La clase con la menor cantidad de ejemplos, *Muy bueno* con 3.76%, requiere entre un 20% y un 40% de ejemplos de entrenamiento para todas las medidas.
- Los resultados de un clasificador no se deben validar solamente con la medida de *exactitud*. Si se busca predecir la clase que tiene la menor cantidad de ejemplos y se evalúa la exactitud, el resultado puede ser alto con sólo predecir la mayoría de los ejemplos que no pertenecen a la clase [37]. Observando el desempeño de los clasificadores con la clase que tiene menos ejemplos, *Muy bueno*, vemos que en *exactitud* y *especificidad* sus valores están por arriba de 0.95, no siendo sorprendente porque estas dos medidas están valorando demasiado a los ejemplos que no pertenecen a la clase. En cambio, el desempeño de los clasificadores en *sensibilidad* y *precisión* para la misma clase, *Muy bueno*, disminuyen sus valores, siendo 0.1803 el peor resultado obtenido en *sensibilidad* con un entrenamiento del 20%, donde establece diferencia de desempeño en comparación con el 100%. Por su parte la clase *Inaccesible*, no tiene problemas con las medidas porque predomina en cantidad de ejemplos. El clasificador para la clase *Inaccesible*, si se crea con un 20% no disminuye en su desempeño de predicción. De las cuatro medidas utilizadas, la *sensibilidad* fue la que influyó para establecer el rango general de entrenamiento entre 40% y 60%, ya que esta valora los ejemplos que sí pertenecen a la clase.
- La semejanza entre las características de ejemplos de diferentes clases es otro factor que afecta el desempeño de la clasificación [3]. Los ejemplos de las clases *Accesible* y *Bueno*, fueron difíciles de clasificar, siendo estas las que influyeron en el establecimiento del rango general de entrenamiento entre el 40% y el 60%. Los valores de las características de los ejemplos de las clases *Accesible* y *Bueno*, no son tan opuestos y diferentes como los de las clases *Inaccesible* y *Muy bueno*.
- La función booleana para la salida de una tabla de verdad se obtiene mediante métodos como mapas de *Karnaugh* [22] o *Quine-McCluskey* [33][28]. En la obtención de las reglas de clasificación, se utilizaron funciones booleanas para predecir la clase. Como alternativa para resolver tablas de verdad se utiliza la *Programación Genética* [12][26][31][35]. Para

disminuir el costo computacional de utilizar toda la tabla se podría realizar una ejecución en dos etapas. En la primera etapa, se utilizaría una muestra de la tabla hasta obtener individuos con una aptitud alta, sabiendo que se obtienen desempeños iguales que con toda la tabla. La segunda etapa, continuaría mejorando la población respecto a la tabla completa, hasta obtener el individuo que resuelva exactamente la salida esperada.

- La agrupación y orden entre funciones y terminales utilizados para crear los individuos no se especifica durante el proceso evolutivo de la *PG* ejecutada. Los individuos pueden tener redundancias al evaluar una función con los mismos parámetros de entrada ($(A)OR(A)$), o una doble negación ($NOT(NOT(A))$). La función objetivo utilizada para evaluar y diferenciar a los mejores individuos durante el proceso evolutivo, compara la salida obtenida por el individuo con la salida esperada. Al observar los individuos obtenidos con el 60% y con el 100%, no se les compara por la cantidad de elementos que tienen, sino por el promedio de las cuatro medidas que se midieron. Sus valores promedios son muy semejantes y no existe diferencia estadística en desempeño entre los casos a los que pertenecen. Utilizar cualquiera de los dos individuos presentados para cada clase no tendrá una diferencia significativa en la predicción, pero sí existe diferencia en el esfuerzo requerido para crearlos, siendo menor para los individuos del 60%.

7.2 Conclusiones

- Se pueden obtener reglas de clasificación sin pérdida de generalización o desempeño tomando una muestra aleatoria del total de ejemplos.
- El rango general para tomar la muestra de ejemplos de entrenamiento se estableció entre el 40% y el 60%.
- Tomar sólo una muestra de ejemplos disminuye el costo computacional necesario para crear las soluciones.
- La distribución de los ejemplos entre las clases influye en el desempeño de la clase con menos ejemplos, obligando a aumentar la cantidad de ejemplos de entrenamiento.
- Es importante utilizar otras medidas para evaluar el desempeño de un clasificador a parte de la *exactitud*. La medida que estableció la necesidad de aumentar el porcentaje de entrenamiento fue la *sensibilidad*, porque predice los ejemplos que sí pertenecen a la clase.
- Los ejemplos que son diferentes en pocas características y que pertenecen a diferentes clases, complican la creación y el desempeño del clasificador, requiriéndose más ejemplos de entrenamiento para poder diferenciarlos.
- En la solución de tablas de verdad, se puede comenzar por buscar el mejor individuo utilizando una muestra de la tabla. Después de aproximarse a la solución, se utiliza la tabla completa para buscar la función booleana que obtiene la salida exacta. La ganancia es la reducción del costo computacional.

7.3 Trabajos futuros

- Buscar dentro del rango general de entrenamiento establecido un punto mínimo.
- Cuantificar el tiempo computacional entre los diferentes porcentajes de entrenamiento.
- Modificar los operadores de la *Programación Genética* para buscar mejores resultados.
- Utilizar otras funciones objetivo para evaluar a los individuos durante el proceso evolutivo.
- Evaluar el desempeño de las soluciones con otras medidas como el análisis de la curva ROC (*Receiver Operating Characteristic*) y el área bajo la curva AUC (*Area Under the ROC Curve*) [3][37].
- Utilizar otras bases de datos con mayor cantidad de datos, atributos y con diferente distribución de los ejemplos entre las clases.

BIBLIOGRAFÍA

- [1] Baker, J. E. "Reducing bias and inefficiency in the selection algorithm", *Proc. ICGA 2*, pp. 14-21, Lawrence Erlbaum Associates, Publishers. 1987.
- [2] Banzhaf, W., Nordin, P., Keller, R. E. y Francone, F. D. *Genetic Programming An Introduction on the Automatic Evolution of Computer Programs and its Application*. Morgan Kaufmann Publishers, Inc., San Francisco California, 1998.
- [3] Batista G. E. A. P. A., Prati, R. C. y Monard, M. C. "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data", *SIGKDD Explorations Newsletter*, vol. 6, number 1, pp. 20-29, (eds.) Nitesh Chawla, Nathalie Japkowicz, Aleksander Kolcz. 2004.
- [4] Blake, C.L. y Merz, C.J. UCI repository of machine learning databases. Technical report, University of California, Irvine, Dept. of Information and Computer Sciences, 1998. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [5] Bojarczuk, C. C., Lopes, H. S. y Freitas, A. A. "Discovering comprehensible classification rules by using Genetic Programming: a case study in a medical domain", *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, Florida, USA, pp. 953-958. 1999.
- [6] Bojarczuk, C. C., Lopes, H. S. y Freitas, A. A. "Genetic programming for knowledge discovery in chest-pain diagnosis", *IEEE Engineering in Medicine and Biology Magazine*, pp. 38-44. 2000.
- [7] De Jong, K. A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. thesis, University of Michigan, Ann Arbor. 1975.
- [8] Díaz, B. I., Ortega, L. O., y Restrepo A. "Comparación de dos técnicas de minería de datos para la clasificación de liptinitas", *COMCEV-03, Primer Congreso Mexicano de Computación Evolutiva*, Centro de Investigación en Matemáticas A. C. CIMAT, Guanajuato, México, pp. 283-293. 2003.
- [9] Dietterich, T. G. "Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms", *Neural Computation*, vol. 10, pp. 1895-1923. 1998.
- [10] Doherty, G. "Fundamental analysis using genetic programming for classification rule induction", *Genetic Algorithms and Genetic Programming at Stanford 2003*, USA, Stanford, California, (eds.) R. J. Koza, pp. 45-51. 2003.

-
- [11] Downie, N. M. y Heath, R.W. *Métodos Estadísticos Aplicados*. HARLA Harper & Row Latinoamericana. 1973.
- [12] Droste, S. "Efficient Genetic Programming for Finding Good Generalizing Boolean Functions", *Proceedings of the Second Annual Conference on Genetic Programming*, month 13-16. Morgan Kaufmann Publishers, San Francisco CA. pp. 82-87. 1997.
- [13] Eggermont J., Eiben, A. E., y van Hemert, J. I. "A comparison of genetic programming variants for data classification", *Proc. Intelligent Data Analysis (IDA-99)*. 1999.
- [14] Eggermont, J. "Evolving Fuzzy Decision Trees with Genetic Programming and Clustering", *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2002*, vol. 2278, month 3-5, Springer-Verlag, Kinsale, Ireland, (eds). Lutton, E., Foster, J. A., Miller, J., Ryan, C., Tettamanzi, A. G. B. pp. 71-82. 2002.
- [15] Esparza García, J. A. *Clasificadores Genéticos Usados para la Predicción de los Niveles de Ozono en la Ciudad de México*. 1999.
- [16] Freitas, A. A. "A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction", *Genetic Programming 1997: Proc. 2nd Annual Conf.*, (eds). J. R. Koza and K. Deb et al., Stanford University, CA, USA, publisher Morgan Kaufmann. pp. 96-101. 1997.
- [17] Freitas, A. A. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, Berlin Heidelberg. 2002.
- [18] Giarratano, J. *Sistemas Expertos Principios y Programación*. International Thomson Editores, 2001.
- [19] Golberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Alabama, USA, 1989.
- [20] Golberg, D. E., Deb, K., y Korb, B. "Do not worry, be messy", *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, 1991.
- [21] Han, J. y Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers. San Francisco, California USA. 2001.
- [22] Karnaugh, M. "A map method for synthesis of combinational logic circuits", *Transactions of the AIEE, Communications and Electronics*, 72(1):593-599, 1953.

- [23] Koza, J. R. "Concept formation and decision tree induction using the genetic programming paradigm", *Parallel Problem Solving from Nature - Proceedings of 1st Workshop, {PPSN} 1*, vol. 496, month 1-3. Springer-Verlag, Berlin, Germany, Dortmund, Germany. (eds.) Schwefel, H. P. and Manner, R., pp. 124-128. 1991.
- [24] Koza, J. R. *Genetic Programming: On the Programming of Computer by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts, 1992.
- [25] Koza, J. R. "Evolution of a computer program for classifying protein segments as transmembrane domains using genetic programming", *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, pp. 244-252. 1994.
- [26] Langdon, W. B. y Poli, R. "Boolean Functions Fitness Spaces", *Late Breaking Papers at the Genetic Programming 1998 Conference*. month 22-25. Stanford University Bookstore. University of Wisconsin, Madison, Wisconsin, USA. 1998.
- [27] McAllister, J. *Inteligencia artificial y PROLOG en microordenadores*. Marcombo. Trad. Silvia Verneti-Blina, Ángel Toribio González. Barcelona, 1991.
- [28] McCluskey, E. J. "Minimization of boolean functions", *Bell Systems Technical Journal*, 1956.
- [29] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.
- [30] Michie, D., Spiegelhalter, D. J. y Taylor, C. C. *Machine Learning, Neural and Statistical Classification*. 1994.
- [31] Poli, R y Page, J. "Solving High-Order Boolean Parity Problems with Smooth Uniform Crossover, Sub-Machine Code GP and Demes", *Genetic Programming and Evolvable Machines*, vol. 1, num. 1/2, pp. 37-56.
- [32] Pyle, D. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, San Francisco, CA, USA. 1999.
- [33] Quine, W. V. "A way to simplify truth functions", *American Mathematical Monthly*, 1955.
- [34] Sánchez y Beltrán, J. P. *Sistemas Expertos Una Metodología de Programación*, RA-MA, 1988.

- [35] Serna, P. E. "Diseño de Circuitos Lógicos Combinatorios utilizando Programación Genética" *Tesis de Maestría*, Maestría en Inteligencia Artificial UV/LANIA, Xalapa Veracruz, México, 2001.
- [36] Teller, A. y Andre, D. "Automatically Choosing the Number of Fitness Cases: The Rational Allocation of Trials", *Genetic Programming 1997: Proceedings of the Second Annual Conference*. month 13-16, Morgan Kaufmann Publishers, Stanford University, CA, USA, pp. 321-328. 1997.
- [37] Weiss, G. y Provost, F. "The effect of class distribution on classifier learning", *Technical Report ML-TR 43*, Department of Computer Science, Rutgers University. 2001.
- [38] Whitley, D. "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best", *Proc. ICGA 3*, Morgan Kaufmann Publishers, pp. 116-121. 1989.
- [39] Wong, M. L, Leung, K. S. *Data Mining Using Grammar Based Genetic Programming and Applications*. Klumer Academic Publishers, Massachusetts, USA. 2000.

ANEXO A. Desempeño de las 30 ejecuciones de la *PG* en la predicción de los ejemplos de prueba para cada clase en cuanto a *SENSIBILIDAD*.

A.1. Utilizando 20% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9356	0.7308	0.7222	1
1	0.884	0.9801	0.9464	0.3389
2	0.9622	0.9405	0.8301	0.7115
3	0.9645	0.75	0.4655	0.923
4	0.972	0.7389	0.109	1
5	0.9139	0.798	0.25	1
6	0.9086	0.9739	0.3518	0.9019
7	0.9885	0.7292	0.1333	0.7346
8	0.9327	0.9833	0.3793	0.6415
9	0.9225	0.83	0.4406	0.5357
10	0.9444	0.9766	0.5535	0.8392
11	0.9053	0.8514	0.2456	0.5918
12	0.8943	0.8657	0.72	0.5283
13	0.9147	0.8263	0.1186	0.4814
14	0.9443	0.8571	0.1406	0.875
15	0.8689	0.8782	0.8163	0.7547
16	0.9603	0.9935	0.2711	0.1803
17	0.9303	0.9339	0.5833	0.9622
18	0.9733	0.8436	0.4285	0.1964
19	0.9233	0.7831	0.4482	0.8113
20	0.8709	1	0.2931	0.8111
21	0.9599	0.7566	0.1551	0.6938
22	0.9146	0.8233	0.9375	0.3965
23	0.9506	0.7935	0.7037	0.9122
24	0.954	0.937	0.5593	0.7636
25	0.9651	0.7647	0.2881	0.9107
26	0.9362	0.9536	0.5576	0.3035
27	0.9107	0.8132	0.3965	0.4745
28	0.9822	0.9583	0.5283	0.5925
29	0.9123	0.765	0.5882	0.4827

A.2. Utilizando 40% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.8876	0.9819	0.5909	0.909
1	0.9748	0.859	0.6666	0.8837
2	0.9172	0.8898	0.3809	0.8979
3	0.8895	0.9587	0.7435	1
4	0.8955	0.8539	0.5581	0.7714
5	0.928	1	0.7826	0.7317
6	0.9672	0.8884	0.6363	0.7857
7	0.9508	0.9594	0.4634	0.9
8	0.8845	0.9241	0.5434	0.7894
9	0.8945	0.8448	0.6382	0.909
10	0.8736	0.8888	0.4347	1
11	0.9097	1	0.6842	0.9318
12	0.9439	0.9292	0.4615	0.9117
13	0.9553	0.8878	0.8648	0.8684
14	0.9232	0.9912	0.6744	0.9148
15	0.795	0.8369	0.8095	0.8787
16	0.9026	1	0.5476	0.9722
17	0.9319	1	0.5581	0.9069
18	0.822	1	0.8863	0.9512
19	0.8584	0.9386	0.9545	1
20	0.9114	0.9446	0.74	0.9111
21	0.9253	0.8036	0.5555	1
22	0.9084	0.8319	0.7804	0.7777
23	0.9507	0.9819	0.9487	0.7173
24	0.8834	0.7901	0.4634	0.9555
25	0.9439	0.9257	0.3488	1
26	0.9203	1	0.5416	0.8139
27	0.944	0.8595	0.6585	0.909
28	0.9339	0.8907	1	1
29	0.9291	0.9256	0.625	0.75

A.3. Utilizando 60% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.8788	0.8553	0.387	0.9473
1	0.9326	1	0.9655	0.7894
2	0.9491	1	0.3333	0.826
3	0.904	1	0.2121	0.9166
4	0.9377	1	1	0.8888
5	0.959	0.8857	0.6875	1
6	0.909	0.8445	0.6857	0.8928
7	0.8929	1	1	0.8275
8	0.8844	0.9567	0.5853	0.9142
9	0.9471	1	0.65	0.9285
10	0.9081	0.9407	0.6785	0.8965
11	0.9751	0.95	0.64	0.9047
12	0.9354	0.8961	0.8928	0.9333
13	0.9548	1	0.5517	0.8928
14	0.9536	0.923	0.5142	0.8571
15	0.9148	0.858	0.7027	0.9523
16	0.9626	0.9675	0.5862	0.862
17	0.9768	0.8536	0.75	0.7741
18	0.9203	0.9724	0.4516	0.8148
19	0.9325	0.9642	1	0.931
20	0.9185	0.9753	1	0.8846
21	0.9478	0.8333	0.8064	1
22	0.8455	0.4691	0.9696	0.7419
23	0.9459	0.9872	0.5937	0.7857
24	0.9454	1	0.2941	0.75
25	0.9714	0.9529	0.913	0.8
26	0.985	1	0.5384	0.826
27	0.9699	1	0.6451	1
28	0.9793	0.9503	0.878	0.88
29	0.8872	0.9933	0.3823	0.92

A.4. Utilizando 80% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9595	0.9459	0.8947	0.875
1	0.9401	0.7792	0.9166	1
2	0.9083	1	0.647	0.9333
3	0.9606	0.9459	0.3571	1
4	0.97	0.92	0.3846	1
5	0.9094	0.4342	1	1
6	0.9718	1	0.8235	1
7	0.9277	1	0.6	0.8333
8	0.9552	0.6944	0.7857	1
9	0.9626	0.6712	0.9166	0.8888
10	0.9	1	0.4375	0.8888
11	0.9057	1	0.6875	0.909
12	0.9215	0.8292	0.3888	0.923
13	0.9033	1	0.5294	0.9
14	0.9617	0.8571	0.8	1
15	0.931	0.909	0.6842	1
16	0.9709	0.9382	0.8666	1
17	0.9672	0.8539	0.5	0.8333
18	0.9844	1	0.5882	0.8947
19	0.9349	1	0.8333	0.913
20	0.9459	1	0.5625	1
21	0.9686	0.5058	0.9166	1
22	0.9264	0.9506	0.5	1
23	0.931	1	1	1
24	0.926	1	0.7	0.8095
25	0.9551	0.7536	1	1
26	0.9744	1	0.909	1
27	0.9112	1	1	0.7857
28	0.9829	0.9662	0.75	0.9
29	0.8888	0.8676	0.9	0.8823

A.5. Utilizando 100% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9586	0.8802	0.5217	0.8615
1	0.9561	0.9635	0.8695	1
2	0.9123	1	0.7826	0.8
3	0.9636	0.9557	0.826	0.7384
4	0.9694	0.9713	0.7246	0.7384
5	0.8785	0.9218	0.9565	1
6	0.9115	1	0.7826	0.923
7	0.9487	1	0.971	1
8	0.7504	0.9687	0.8695	1
9	0.9223	0.9921	0.9565	1
10	0.9537	1	0.8695	0.9846
11	0.9694	1	0.7391	0.9384
12	0.9768	0.9453	0.5652	1
13	0.7933	1	0.5652	0.8
14	0.9867	0.9583	0.8405	0.8
15	0.7933	0.9947	0.913	0.8
16	0.9289	1	1	1
17	0.9694	0.9843	0.5652	1
18	0.9479	1	0.9565	0.923
19	0.9661	1	0.5652	1
20	0.9123	1	0.5652	0.9692
21	0.971	0.9739	0.6376	0.8615
22	0.9082	0.9817	0.7826	1
23	0.9289	0.9427	0.8695	1
24	0.9016	0.8359	0.7826	0.923
25	0.9694	1	0.7391	0.923
26	0.9123	0.9843	1	1
27	0.9652	0.9296	0.5652	0.8
28	0.9694	0.888	0.8695	1
29	0.938	1	0.8695	1

ANEXO B. Desempeño de las 30 ejecuciones de la *PG* en la predicción de los ejemplos de prueba para cada clase en cuanto a *ESPECIFICIDAD*.

B.1. Utilizando 20% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.8912	0.9081	0.9888	0.9962
1	0.9359	0.8629	0.9871	0.9954
2	0.9783	0.8185	0.9721	0.9924
3	0.8345	0.916	0.9766	0.9797
4	0.9066	0.8483	0.9866	0.9903
5	0.9884	0.9442	0.9766	0.9963
6	0.9903	0.7064	0.9826	1
7	0.5576	0.8668	0.9864	0.997
8	0.9423	0.7839	0.9774	0.9969
9	0.9705	0.8605	0.9954	0.9933
10	1	0.7608	0.9823	0.9824
11	0.9745	0.8956	0.9925	0.9868
12	0.946	0.8281	0.9883	0.9879
13	0.9168	0.8451	0.997	0.9864
14	0.9053	0.8213	0.9969	0.9842
15	0.975	0.7986	0.967	0.9969
16	0.8799	0.7917	0.9788	0.9833
17	0.941	0.7861	0.9909	0.9977
18	0.924	0.9126	0.9849	0.9902
19	0.9184	0.8769	0.9788	0.9962
20	0.9361	0.7037	0.9879	0.9924
21	0.959	0.9168	0.9917	0.9955
22	1	0.8563	0.9919	0.9954
23	0.9857	0.9232	0.9826	0.9856
24	0.887	0.7983	0.9592	0.9791
25	0.9166	0.9215	0.9939	1
26	0.8951	0.8255	0.9977	0.9969
27	0.9532	0.8462	0.975	1
28	0.7793	0.8118	0.9783	0.9939
29	0.9123	0.9327	0.9864	0.9781

B.2. Utilizando 40% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9697	0.7828	0.998	1
1	0.9591	0.8873	0.9949	0.9909
2	0.9746	0.9265	0.994	1
3	0.9417	0.8107	0.9931	0.995
4	0.9404	0.9371	0.9909	0.999
5	0.8407	0.7487	0.994	0.9949
6	0.9791	0.8671	0.996	0.9959
7	0.9344	0.8024	0.997	1
8	0.9689	0.7941	0.9858	0.9929
9	0.9651	0.9055	1	1
10	0.9369	0.8977	1	0.9961
11	0.9329	0.7127	0.9809	0.9979
12	0.9477	0.8263	1	1
13	0.926	0.8587	0.9912	1
14	0.9839	0.8601	0.9818	1
15	1	0.896	0.9959	1
16	0.974	0.7224	0.9892	0.989
17	0.9116	0.7797	0.9891	1
18	0.9653	0.4289	0.9768	1
19	0.9671	0.8824	0.9969	0.997
20	1	0.9039	0.9696	1
21	0.8562	0.9303	0.9969	0.9989
22	1	0.8939	0.9769	1
23	0.8639	0.7406	0.987	0.9959
24	0.9335	0.9188	1	0.98
25	0.9037	0.8827	0.9788	1
26	1	0.8189	0.9909	0.9939
27	0.9551	0.9551	0.9745	1
28	0.9871	0.907	0.9781	0.9969
29	0.9757	0.8729	0.9819	1

B.3. Utilizando 60% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9142	0.923	0.9969	1
1	0.9049	0.7477	0.9939	1
2	0.8	0.6923	1	0.9985
3	0.9417	0.5724	0.9954	0.991
4	0.9601	0.826	0.9787	1
5	0.8728	0.9021	0.9863	0.9969
6	0.887	0.9503	0.9832	0.997
7	0.9688	0.8102	0.9171	1
8	0.9537	0.8255	1	1
9	0.8767	0.702	0.9984	0.9984
10	1	0.8277	0.9807	1
11	0.8857	0.8952	0.9925	1
12	0.9132	0.8959	0.9835	1
13	0.9139	0.8253	0.9954	1
14	0.8599	0.9074	1	0.9754
15	1	0.9273	0.9969	1
16	0.9523	0.8494	0.994	1
17	0.8894	0.9488	0.9924	1
18	0.972	0.8226	1	1
19	0.9772	0.7608	0.9762	1
20	0.953	0.7735	0.9864	1
21	0.9305	0.9084	0.9896	0.9939
22	1	0.983	0.9893	0.9925
23	0.9241	0.7606	0.9853	1
24	0.8602	0.8655	0.9924	0.9939
25	0.8805	0.8113	0.9511	0.9969
26	0.5911	0.818	0.9941	1
27	0.9359	0.8098	0.9969	0.9969
28	0.9565	0.821	1	1
29	0.9774	0.9351	1	1

B.4. Utilizando 80% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	1	0.7788	0.9642	1
1	0.9821	0.9637	0.997	1
2	0.9716	0.7749	0.9848	1
3	0.9401	0.8639	1	1
4	0.9375	0.9151	1	1
5	0.994	0.9667	0.9915	0.9908
6	0.9587	0.8039	0.9969	1
7	0.9385	0.7072	1	0.997
8	0.94	0.9795	0.9974	1
9	0.9619	0.9822	1	1
10	0.9827	0.7508	1	0.9969
11	0.9509	0.7913	0.9942	1
12	0.9	0.9507	0.9971	1
13	1	0.8084	0.9939	1
14	0.8918	0.8959	0.9828	0.9969
15	0.9298	0.9267	0.9847	1
16	0.9813	0.8125	1	1
17	0.9117	0.9221	1	1
18	0.8636	0.855	0.9969	1
19	0.8301	0.6851	0.9914	1
20	0.94	0.7907	1	1
21	0.9417	0.9166	0.9943	0.9969
22	1	0.8614	0.9969	0.9907
23	0.921	0.7977	0.9939	1
24	0.9482	0.8716	0.9938	0.9938
25	0.8699	0.9061	0.9942	1
26	0.9189	0.7985	0.994	1
27	1	0.8057	0.997	1
28	0.9909	0.798	1	0.9887
29	1	0.8687	0.994	1

B.5. Utilizando 100% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.8918	0.9181	1	0.9855
1	0.9633	0.8943	0.9927	0.9921
2	1	0.8214	0.9981	1
3	0.8725	0.8266	0.9885	1
4	0.9169	0.7775	0.9891	1
5	0.9517	0.9211	0.9915	1
6	0.9749	0.8392	0.9963	1
7	0.9613	0.8214	0.9535	0.9909
8	0.9613	0.8928	0.9927	1
9	0.9305	0.8459	0.9819	0.9969
10	1	0.8214	0.9963	0.9975
11	0.9169	0.8214	0.9945	0.9813
12	0.888	0.8802	1	1
13	1	0.8586	0.9981	1
14	0.8339	0.9166	0.9819	1
15	1	0.7604	0.9608	1
16	1	0.7678	0.9837	1
17	0.9169	0.8229	0.9981	0.9969
18	0.944	0.8214	0.9819	1
19	0.9864	0.8273	1	0.9957
20	1	0.7142	0.9981	0.9993
21	0.8899	0.9389	0.9879	0.9759
22	0.9826	0.7686	0.9891	0.9969
23	1	0.805	0.9783	1
24	0.9749	0.965	0.9963	1
25	0.9169	0.7142	0.9945	1
26	0.9536	0.7723	0.9547	0.9969
27	0.9922	0.9084	0.9981	1
28	0.9169	0.9203	0.9831	0.9969
29	0.9362	0.869	0.9927	0.9969

ANEXO C. Desempeño de las 30 ejecuciones de la *PG* en la predicción de los ejemplos de prueba para cada clase en cuanto a *PRECISIÓN*.

C.1. Utilizando 20% de ejemplos de entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9528	0.7113	0.7222	0.9122
1	0.9714	0.6674	0.7571	0.7692
2	0.9905	0.5925	0.5432	0.7872
3	0.9297	0.7147	0.4655	0.64
4	0.96	0.5919	0.25	0.7636
5	0.9944	0.8032	0.3111	0.9107
6	0.9955	0.4853	0.4523	1
7	0.8343	0.6122	0.3076	0.9
8	0.974	0.5576	0.423	0.8947
9	0.9869	0.6225	0.8125	0.7692
10	1	0.5307	0.5636	0.6619
11	0.9873	0.6954	0.5833	0.617
12	0.9753	0.5771	0.6923	0.6363
13	0.9632	0.6075	0.6363	0.5909
14	0.9592	0.5789	0.6923	0.6666
15	0.9884	0.5591	0.4761	0.909
16	0.9507	0.5764	0.3636	0.3333
17	0.9742	0.5505	0.7446	0.9444
18	0.9683	0.7337	0.5454	0.4583
19	0.9632	0.6453	0.4814	0.8958
20	0.9703	0.4789	0.5151	0.7674
21	0.9821	0.716	0.45	0.85
22	1	0.6129	0.8035	0.7931
23	0.9935	0.7454	0.6229	0.7323
24	0.9501	0.5648	0.3793	0.6
25	0.9651	0.7482	0.68	1
26	0.9549	0.6025	0.9062	0.8095
27	0.9784	0.6104	0.4107	1
28	0.909	0.592	0.4912	0.8
29	0.9598	0.7699	0.625	0.4912

C.2. Utilizando 40% de ejemplos de entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9864	0.5518	0.9285	1
1	0.9839	0.6725	0.8484	0.8085
2	0.9881	0.7709	0.7272	1
3	0.9713	0.5741	0.8055	0.875
4	0.9712	0.7909	0.7272	0.9642
5	0.9306	0.5244	0.8571	0.8571
6	0.9897	0.6697	0.875	0.8918
7	0.972	0.5695	0.8636	1
8	0.985	0.5505	0.641	0.8108
9	0.9832	0.7205	1	1
10	0.9704	0.7067	1	0.9111
11	0.967	0.5031	0.5777	0.9534
12	0.9773	0.5898	1	1
13	0.9693	0.6325	0.7804	1
14	0.9926	0.6676	0.617	1
15	1	0.6939	0.8947	1
16	0.9879	0.4956	0.6764	0.7608
17	0.9599	0.5721	0.6857	1
18	0.9841	0.3216	0.629	1
19	0.985	0.6722	0.9333	0.923
20	1	0.7424	0.5522	1
21	0.9332	0.7553	0.8928	0.9772
22	1	0.6975	0.5818	1
23	0.9445	0.5034	0.74	0.8918
24	0.968	0.7254	1	0.6825
25	0.957	0.6905	0.4166	1
26	1	0.6048	0.7428	0.8536
27	0.9801	0.8487	0.5094	1
28	0.9942	0.7361	0.6071	0.9302
29	0.9888	0.6892	0.5813	1

C.3. Utilizando 60% de ejemplos de entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.9682	0.7683	0.8571	1
1	0.956	0.5369	0.875	1
2	0.9211	0.47	1	0.95
3	0.9702	0.3868	0.7	0.7857
4	0.9831	0.625	0.7083	1
5	0.9388	0.6966	0.7096	0.9354
6	0.9563	0.8223	0.6857	0.9259
7	0.9834	0.5806	0.3373	1
8	0.9767	0.615	1	1
9	0.9431	0.4707	0.9629	0.9629
10	1	0.6059	0.5937	1
11	0.9514	0.7203	0.7619	1
12	0.9646	0.7113	0.6944	1
13	0.9681	0.5793	0.8421	1
14	0.9422	0.7346	1	0.5142
15	1	0.7732	0.9285	1
16	0.9789	0.6478	0.8095	1
17	0.9508	0.8383	0.8076	1
18	0.9865	0.5924	1	1
19	0.9894	0.5056	0.5897	1
20	0.9777	0.5683	0.7428	1
21	0.968	0.7017	0.7812	0.8857
22	1	0.8941	0.8205	0.8214
23	0.966	0.5457	0.6551	1
24	0.9337	0.6711	0.6666	0.8571
25	0.952	0.6183	0.3888	0.923
26	0.8333	0.5991	0.7777	1
27	0.9754	0.5823	0.909	0.9285
28	0.9814	0.6169	1	1
29	0.9881	0.7967	1	1

C.4. Utilizando 80% de ejemplos de entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	1	0.5109	0.5862	1
1	0.9909	0.8571	0.9166	1
2	0.9864	0.5763	0.6875	1
3	0.9691	0.6542	1	1
4	0.97	0.75	1	1
5	0.9735	0.7857	0.7692	0.85
6	0.9837	0.5563	0.9333	1
7	0.9721	0.5187	1	0.909
8	0.9751	0.8928	0.9166	1
9	0.983	0.9074	1	1
10	0.9904	0.5253	1	0.9411
11	0.9778	0.5396	0.8461	1
12	0.9591	0.8395	0.875	1
13	1	0.6323	0.8181	1
14	0.9495	0.7021	0.6666	0.9444
15	0.9642	0.7777	0.7222	1
16	0.9915	0.5846	1	1
17	0.9632	0.7916	1	1
18	0.9442	0.6363	0.909	1
19	0.9274	0.472	0.7692	1
20	0.976	0.5693	1	1
21	0.9762	0.6515	0.8461	0.9333
22	1	0.6754	0.9	0.875
23	0.96	0.5939	0.8823	1
24	0.9726	0.7007	0.875	0.8947
25	0.9301	0.6666	0.8461	1
26	0.9621	0.5703	0.8333	1
27	1	0.5573	0.9583	1
28	0.9956	0.585	1	0.6923
29	1	0.6145	0.8181	1

C.5. Utilizando 100% de ejemplos de entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9539	0.7544	1	0.7
1	0.9838	0.7226	0.8333	0.8333
2	1	0.6153	0.9473	1
3	0.9464	0.6116	0.75	1
4	0.9646	0.555	0.7352	1
5	0.977	0.7695	0.825	1
6	0.9883	0.64	0.9	1
7	0.9828	0.6153	0.4652	0.8125
8	0.9784	0.7209	0.8333	1
9	0.9687	0.6479	0.6875	0.9285
10	1	0.6153	0.909	0.9411
11	0.9646	0.6153	0.85	0.663
12	0.9532	0.6927	1	1
13	1	0.6689	0.9285	1
14	0.9328	0.7666	0.659	1
15	1	0.5426	0.4921	1
16	1	0.5517	0.7187	1
17	0.9646	0.6136	0.9285	0.9285
18	0.9753	0.6153	0.6875	1
19	0.994	0.6233	1	0.9027
20	1	0.5	0.9285	0.9843
21	0.9537	0.8201	0.6875	0.5833
22	0.9918	0.5479	0.75	0.9285
23	1	0.5801	0.625	1
24	0.9882	0.8722	0.9	1
25	0.9646	0.5	0.85	1
26	0.9787	0.5526	0.4791	0.9285
27	0.9965	0.7437	0.9285	1
28	0.9646	0.7611	0.6818	0.9285
29	0.9717	0.6857	0.8333	0.9285

ANEXO D. Desempeño de las 30 ejecuciones de la *PG* en la predicción de los ejemplos de prueba para cada clase en cuanto a *EXACTITUD*.

D.1. Utilizando 20% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.9223	0.8662	0.9785	0.9963
1	0.899	0.8886	0.9855	0.9674
2	0.967	0.8452	0.9667	0.9819
3	0.9248	0.8796	0.9551	0.9775
4	0.9522	0.8232	0.9523	0.9906
5	0.9368	0.9117	0.9472	0.9964
6	0.9329	0.7656	0.958	0.9963
7	0.8561	0.836	0.9494	0.9877
8	0.9356	0.8271	0.9524	0.9833
9	0.9365	0.8539	0.9718	0.975
10	0.9606	0.8076	0.9653	0.9767
11	0.9269	0.886	0.962	0.9731
12	0.9096	0.8361	0.9788	0.9703
13	0.9154	0.8409	0.9605	0.9667
14	0.9327	0.8293	0.9573	0.9804
15	0.8996	0.8166	0.9617	0.9877
16	0.9367	0.8365	0.9486	0.9479
17	0.9334	0.8185	0.9732	0.9963
18	0.9587	0.8973	0.9624	0.958
19	0.9219	0.8561	0.9566	0.9892
20	0.89	0.7671	0.9587	0.9775
21	0.9596	0.8821	0.9566	0.9849
22	0.9399	0.8492	0.99	0.9703
23	0.9612	0.8945	0.9718	0.9826
24	0.9334	0.8286	0.9421	0.9706
25	0.9508	0.8852	0.9638	0.9963
26	0.924	0.8533	0.9812	0.9689
27	0.9235	0.8387	0.9508	0.9779
28	0.9197	0.8443	0.9611	0.9783
29	0.9123	0.8946	0.9718	0.9573

D.2. Utilizando 40% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.9112	0.8254	0.9808	0.9961
1	0.9704	0.8813	0.9816	0.9864
2	0.9346	0.9185	0.9694	0.9951
3	0.9057	0.8418	0.9838	0.9951
4	0.9093	0.9189	0.9729	0.9913
5	0.9016	0.8032	0.9848	0.9846
6	0.971	0.8721	0.9809	0.9874
7	0.9459	0.836	0.9765	0.9961
8	0.91	0.822	0.9662	0.9855
9	0.9161	0.8919	0.9836	0.9961
10	0.8924	0.8958	0.9749	0.9963
11	0.917	0.7774	0.9701	0.9951
12	0.945	0.8481	0.9798	0.9971
13	0.9468	0.8649	0.9868	0.9952
14	0.9414	0.8891	0.9691	0.9961
15	0.8563	0.883	0.9884	0.9963
16	0.9238	0.7819	0.9718	0.9884
17	0.9257	0.8298	0.9716	0.9961
18	0.8618	0.5506	0.9729	0.998
19	0.8891	0.8939	0.9951	0.9971
20	0.9382	0.9132	0.9585	0.9961
21	0.9035	0.9035	0.9779	0.999
22	0.9355	0.8798	0.9691	0.9922
23	0.9254	0.7916	0.9855	0.9837
24	0.8987	0.8914	0.9789	0.979
25	0.9316	0.8922	0.9527	1
26	0.945	0.8582	0.9701	0.9865
27	0.9473	0.9334	0.9622	0.9961
28	0.9497	0.9034	0.9788	0.9971
29	0.9432	0.8852	0.9682	0.9893

D.3. Utilizando 60% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Buena	Muy buena
0	0.8877	0.9075	0.9696	0.9985
1	0.924	0.8048	0.9927	0.9942
2	0.906	0.7582	0.971	0.9928
3	0.9161	0.6632	0.958	0.9884
4	0.8442	0.8651	0.9797	0.9971
5	0.9306	0.8988	0.9725	0.9971
6	0.9031	0.9277	0.9682	0.9929
7	0.9176	0.8497	0.9205	0.9929
8	0.906	0.8551	0.9754	0.9956
9	0.9248	0.7644	0.9783	0.9956
10	0.9364	0.8526	0.9687	0.9956
11	0.9479	0.9073	0.9797	0.9971
12	0.9291	0.8959	0.9799	0.9971
13	0.9438	0.8591	0.9768	0.9957
14	0.926	0.9108	0.9754	0.9719
15	0.9421	0.9118	0.9812	0.9985
16	0.9595	0.8757	0.9771	0.9944
17	0.9494	0.9263	0.9826	0.9898
18	0.9364	0.854	0.9757	0.9928
19	0.9461	0.802	0.977	0.9971
20	0.9291	0.8208	0.9869	0.9958
21	0.9424	0.893	0.9816	0.9942
22	0.893	0.8627	0.9884	0.9814
23	0.9393	0.8117	0.9678	0.9913
24	0.9178	0.8945	0.9581	0.9826
25	0.945	0.8457	0.9498	0.9884
26	0.8569	0.8569	0.9774	0.9944
27	0.9605	0.8497	0.9813	0.9971
28	0.9725	0.8511	0.9927	0.9956
29	0.9161	0.947	0.9696	0.9971

D.4. Utilizando 80% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.971	0.8116	0.9605	0.9945
1	0.9537	0.9235	0.9942	1
2	0.9277	0.8276	0.9682	0.9971
3	0.9537	0.8815	0.9739	1
4	0.9595	0.9161	0.9779	1
5	0.9775	0.8501	0.9917	0.9913
6	0.9682	0.8426	0.9884	1
7	0.931	0.7774	0.9826	0.9914
8	0.9508	0.9234	0.99	1
9	0.9624	0.9183	0.9971	0.9971
10	0.9277	0.8046	0.9748	0.9913
11	0.919	0.8323	0.9806	0.9971
12	0.9154	0.9219	0.9676	0.9971
13	0.9335	0.8559	0.971	0.9971
14	0.9393	0.8872	0.9752	0.9971
15	0.9306	0.9228	0.9682	1
16	0.9741	0.8401	0.9944	1
17	0.9508	0.9046	0.9768	0.9943
18	0.9483	0.8843	0.9769	0.9942
19	0.9034	0.7543	0.9861	0.9942
20	0.9442	0.8361	0.9797	1
21	0.9608	0.8199	0.9917	0.9971
22	0.9508	0.8821	0.971	0.9913
23	0.9277	0.8439	0.9942	1
24	0.9335	0.9012	0.9768	0.9826
25	0.9248	0.8757	0.9944	1
26	0.9566	0.841	0.9913	1
27	0.9378	0.8439	0.9972	0.9913
28	0.9855	0.8363	0.9919	0.9863
29	0.9219	0.8685	0.9913	0.9942

D.5. Utilizando 100% de ejemplos en entrenamiento.

#Ejecución	Inaccesible	Accesible	Bueno	Muy bueno
0	0.9386	0.9097	0.9809	0.9809
1	0.9583	0.9097	0.9878	0.9924
2	0.9386	0.8611	0.9895	0.9924
3	0.9363	0.8553	0.982	0.9901
4	0.9537	0.8206	0.9785	0.9901
5	0.9004	0.9212	0.9901	1
6	0.9305	0.875	0.9878	0.9971
7	0.9525	0.8811	0.9542	0.9913
8	0.8136	0.9097	0.9878	1
9	0.9247	0.8784	0.9809	0.9971
10	0.9875	0.8611	0.9913	0.9971
11	0.9537	0.8611	0.9843	0.9797
12	0.9502	0.8946	0.9826	1
13	0.8553	0.89	0.9809	0.9924
14	0.9409	0.9259	0.9762	0.9924
15	0.8553	0.8125	0.9589	0.9924
16	0.9502	0.8194	0.9843	1
17	0.9537	0.8587	0.9809	0.9971
18	0.9467	0.8611	0.9809	0.9971
19	0.9722	0.8657	0.9826	0.9959
20	0.9386	0.7777	0.9809	0.9982
21	0.9467	0.9467	0.9739	0.9716
22	0.9305	0.8159	0.9809	0.9971
23	0.9502	0.8356	0.9739	1
24	0.9236	0.9363	0.9878	0.9971
25	0.9537	0.7777	0.9843	0.9971
26	0.9247	0.8194	0.9565	0.9971
27	0.9733	0.9131	0.9809	0.9924
28	0.9537	0.9131	0.9785	0.9971
29	0.9375	0.8981	0.9878	0.9971

Anexo E. Cálculo del análisis de la varianza.

E.1. Fórmulas para el cálculo:

- Suma total de cuadrados:

$$\sum x_i^2 = \sum X^2 - \frac{(\sum X)^2}{N}$$

- Suma de cuadrados entre grupos:

$$\sum x_e^2 = \left[\sum \frac{(\sum X)^2}{n} \right] - \frac{(\sum X_r)^2}{N}$$

- Suma de cuadrados dentro de los grupos:

$$\sum x_d^2 = \sum x_i^2 - \sum x_e^2$$

- Grados de libertad:

<i>Grados de libertad (gl)</i>	<i>Valor</i>
Del total de grupos (gl_t)	Número total de casos (N) menos 1
Entre grupos (gl_e)	Número de grupos (k) menos 1
Dentro de los grupos (gl_d)	Suma del número de casos dentro de cada subgrupo (n) menos 1, (n_1-1) + (n_2-1) + ... + (n_k-1)

- Cuadrado medio entre grupos:

$$\frac{\sum x_e^2}{gl_e}$$

- Cuadrado medio dentro de los grupos:

$$\frac{\sum x_d^2}{gl_d}$$

- El contraste de la *F* de *Snedecor* resulta del cociente:

$$F = \frac{\text{cuadrado medio entre grupos}}{\text{cuadrado medio dentro de los grupos}}$$

E.2. Ejemplo. Análisis de la varianza entre el Caso₁ 20% y el Caso₅ 100% de ejemplos de entrenamiento para la clase *Inaccesible* en cuanto a *Sensibilidad*.

Caso ₁ 20%	Caso ₅ 100%	Caso ₁ 20%	Caso ₅ 100%
X	X	X ²	X ²
0.9356	0.9586	0.8753	0.9189
0.884	0.9581	0.7815	0.9141
0.9622	0.9123	0.9258	0.8323
0.9645	0.9636	0.9303	0.9285
0.972	0.9694	0.9448	0.9397
0.9139	0.8785	0.8352	0.7718
0.9086	0.9115	0.8256	0.8308
0.9885	0.9487	0.9771	0.9000
0.9327	0.7504	0.8699	0.5631
0.9225	0.9223	0.8510	0.8506
0.9444	0.9537	0.8919	0.9095
0.9053	0.9694	0.8196	0.9397
0.8943	0.9768	0.7998	0.9541
0.9147	0.7933	0.8367	0.6293
0.9443	0.9867	0.8917	0.9736
0.8689	0.7933	0.7550	0.6293
0.9603	0.9289	0.9222	0.8629
0.9303	0.9694	0.8655	0.9397
0.9733	0.9479	0.9473	0.8985
0.9233	0.9661	0.8525	0.9333
0.8709	0.9123	0.7585	0.8323
0.9599	0.971	0.9214	0.9428
0.9146	0.9082	0.8365	0.8248
0.9506	0.9289	0.9036	0.8629
0.954	0.9016	0.9101	0.8129
0.9651	0.9694	0.9314	0.9397
0.9362	0.9123	0.8765	0.8323
0.9107	0.9652	0.8294	0.9316
0.9822	0.9694	0.9647	0.9397
0.9123	0.938	0.8323	0.8798

$$\sum X = 28.0001$$

$$27.8332$$

$$\sum X^2 = 26.1629$$

$$25.9190$$

$$\bar{X} = 0.9333$$

$$0.9278$$

- Promedio total $\bar{X}_i = 27.9167$
- Suma de cuadrados total $\sum X_i^2 = 0.1260$
- Suma de cuadrados entre grupos $\sum X_e^2 = 0.0005$
- Suma de cuadrados dentro de los grupos $\sum X_d^2 = 0.1255$
- Cálculos de cuadrados medios

VARIACION	gl	Suma de cuadrados	Cuadrados Medios
Entre grupos	1	0.0005	0.0005
Dentro de los grupos	58	0.1255	0.0022
Total	59	0.1260	

- Contraste de la F (calculado) de *Snedecor* = 0.214
- Valor de la F (tabla) al 1% de significancia = 7.09
- El valor calculado es MENOR al obtenido por la tabla, por lo que se acepta la hipótesis nula y por lo tanto, NO HAY DIFERENCIA ESTADÍSTICA SIGNIFICATIVA entre las medias de los grupos analizados.