



# UNIVERSIDAD DE CUENCA

## FACULTAD DE INGENIERÍA

### ESCUELA DE INFORMÁTICA

Trabajo de graduación previo a la obtención del título de  
Ingeniero de Sistemas

SISTEMA EXPERTO PARA EL APOYO EN EL DIAGNÓSTICO DE GLAUCOMA  
CRÓNICO DE ÁNGULO ABIERTO

**Autores**

Andrés Sebastián Molina Ortega  
Jonattan Oswaldo Ochoa Mora

**Director**

Ing. Ángel Espinoza V.

Cuenca – Ecuador

2011



## **Dedicatoria**

Quiero dedicar este trabajo a mis padres y familia, por todo lo que me han dado en esta vida, especialmente por sus sabios consejos y por estar a mi lado en los momentos difíciles.

A mis amigos, por acompañarme en cada una de las locuras que he emprendido, por soportarme y convertirse en la ayuda idónea en momentos en los que todo creía perdido.

De manera muy especial a ti “Papi Diego”, quien desde el cielo me guía y estoy seguro que en estos momentos está orgullosa de mí.

Wiki

2



## **Dedicatoria**

Agradezco infinitamente a Dios y a mis amados padres, Ayda y Oswaldo, ya que sin su apoyo, cariño y comprensión, mi esfuerzo y entrega se verían nublados en la incertidumbre, dedico este nuevo logro personal a ellos y a todas aquellas personas que de una u otra forma me ayudaron a crecer y convertirme en quien soy ahora, brindándome su apoyo y enseñanzas cada día. Agradezco también a una persona muy especial, mi querido tío Oswaldo, que con sus consejos y afecto ha llegado a ser como un segundo padre para mí, a mis hermanos y sobrinos que con su alegría me dan una razón más para vivir y superarme cada día.

Joom



## **Agradecimientos**

A nuestros padres y familiares por el apoyo incansable, desinteresado e incondicional, que nos han brindado durante estos años.

A nuestro director Ing. Ángel Espinoza por la guía que nos brindó durante el desarrollo de nuestra tesis.

Al Dr. Pablo Zeas Sacoto quien nos brindó la ayuda, estuvo siempre dispuesto a transmitir sus conocimientos y experiencia para el adecuado desarrollo de este proyecto.



## Resumen

El presente trabajo de titulación tuvo como objetivo el desarrollar un sistema experto para el apoyo en el diagnóstico de Glaucoma Crónico de Ángulo Abierto (GCAA), el mismo fue realizado bajo la coordinación del Oftalmólogo Pedro Zeas Sacoto, experto en el tema.

El proyecto “*SISTEMA EXPERTO PARA EL APOYO EN EL DIAGNÓSTICO DE GLAUCOMA CRÓNICO DE ÁNGULO ABIERTO*” se desarrolló en base a un sistema de tres capas, utilizando los lenguajes de programación Visual Basic .Net y Matlab y como motor de base de datos relacional (RDBMS) SQLServer 2005.

Finalmente como resultado se ha obtenido un sistema amigable e intuitivo que permite el manejo automatizado de todo lo concerniente a un diagnóstico y posterior tratamiento para un paciente que sufre la enfermedad del Glaucoma Crónico de Ángulo Abierto.



# ÍNDICE

<b>CAPÍTULO 1.....</b>	<b>12</b>
1.1 Título del Proyecto .....	13
1.2 Introducción .....	13
1.3 Antecedentes .....	15
1.4 Alcance .....	15
1.5 Justificación .....	16
1.6 Auspicio Docente .....	16
1.7 Objetivos .....	17
1.7.1 Objetivo General .....	17
1.7.2 Objetivos Específicos.....	17
1.8 Recursos del Proyecto.....	17
1.8.1 Recursos de Hardware .....	17
1.8.2 Recursos de Software .....	18
1.8.3 Recursos Humanos.....	18
1.9 Metodología.....	18
1.10 Entregables .....	21
<b>CAPÍTULO 2.....</b>	<b>24</b>
2.1 Ingeniería del Conocimiento .....	25
2.2 Inteligencia Artificial .....	27



2.2.1	Definición de Inteligencia Artificial .....	28
2.2.2	Clasificación de la IA.....	30
2.3	Sistemas Expertos .....	31
2.3.1	Introducción .....	31
2.3.2	Historia de los Sistemas Expertos .....	32
2.3.3	Definición de Sistema Experto .....	34
2.3.4	Características de los Sistemas Expertos .....	35
2.3.5	Clasificación de los Sistemas Expertos .....	36
2.3.6	Ventajas y Limitaciones de los Sistemas Expertos .....	38
2.3.7	Componentes de un Sistema Experto.....	41
2.3.8	Ciclo de Desarrollo de un Sistema Experto .....	47
2.4	Sistemas Expertos Basados en Reglas.....	49
2.4.1	Introducción .....	49
2.4.2	La Base de Conocimiento .....	50
2.4.3	El Motor de Inferencia .....	57
2.4.4	Control de Coherencia .....	76
2.4.5	Explicando Conclusiones.....	82
2.5	Especificación y Descripción de las Herramientas y Tecnologías de Desarrollo.....	83
2.5.1	Matlab.....	83
2.5.2	Microsoft Visual Studio .....	85



2.5.3	Microsoft SQL Server .....	91
<b>CAPÍTULO 3.....</b>		<b>99</b>
3.1	Captura de Requerimientos .....	100
3.2	Análisis y Diseño de la Base de Datos .....	101
3.2.1	Diccionario de datos .....	103
3.3	Análisis y Diseño de la Aplicación .....	116
3.4	Análisis y Diseño de la Aplicación .....	117
3.4.1	Flujo de Proceso de Diagnostico .....	117
3.4.2	Modelo de Casos de Uso.....	118
3.4.3	Diagramas de Secuencia .....	128
	Diagnostico GCAA .....	128
	Evolución Paciente .....	129
	Consultar Diagnostico .....	129
	Evaluación STAR.....	130
	Simulación del Nervio Óptico.....	130
	Manejo Videos .....	131
	Configuración Examen .....	131
	Configuración Parámetros .....	132
	Modificar Historia Clínica .....	133
	Eliminar Historia Clínica .....	134
3.4.4	Diagramas de Actividad .....	135



Diagnostico GCAA ..... 135

Evolución Paciente ..... 136

Consulta Diagnósticos ..... 136

Evaluación STAR ..... 137

Simulación Nervio Óptico ..... 137

Manejo de Videos ..... 138

Configuración Examen ..... 138

Configuración Parámetros ..... 139

Modificar Historia Clínica ..... 139

Eliminar Historia Clínica ..... 140

3.4.5 Diagramas de Estado ..... 141

Diagnostico GCAA ..... 141

Evolución Paciente ..... 142

Consulta Diagnostico ..... 142

Evaluación STAR ..... 143

Simulación Nervio Óptico ..... 143

Manejo de Videos ..... 144

Configuración Examen ..... 144

Configuración Parámetros ..... 145

Modificar Historia Clínica ..... 145



Eliminar Historia Clínica .....	146
<b>CAPITULO 4.....</b>	<b>147</b>
4.1 Implementación de la Aplicación.....	148
4.2 Implementación de Procesamiento de Imágenes .....	151
4.2.1 Tipos de imágenes en Matlab .....	151
4.2.2 Lectura de imágenes en Matlab.....	151
4.2.3 Conversiones entre tipos de imágenes.....	151
4.2.4 Selección de una sección de una imagen en Matlab .....	152
4.2.5 Determinar el tamaño de una imagen.....	152
4.2.6 Transformaciones Geométricas .....	153
4.3 Detección de Contornos .....	154
4.4 Elementos Estructurantes.....	155
4.5 Operadores Morfológicos .....	155
4.6 SEGMENTACIÓN.....	156
3.11 MÉTODOS DE CLASIFICACIÓN DE PATRONES .....	157
3.12 Desarrollo de la aplicación para reconocimiento de caracteres.....	158
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>161</b>
Conclusiones .....	161
Recomendaciones.....	162
<b>ANEXOS .....</b>	<b>163</b>
Anexo 1: Análisis de Parámetros .....	164



Anexo 2: Manual de Usuario..... 166

**GLOSARIO ..... 187**

**BIBLIOGRAFÍA ..... 188**



# CAPÍTULO 1



## 1.1 Título del Proyecto

Sistema Experto para el Apoyo en el Diagnóstico de Glaucoma Crónico de Ángulo Abierto.

## 1.2 Introducción

La oftalmología es la especialidad médica que estudia las enfermedades de los ojos y sus tratamientos, las principales enfermedades que afectan a los ojos son infecciones, inflamaciones, degeneraciones y enfermedades sistémicas (diabetes, presión alta, reumatismo, etc.)

Entre estas enfermedades una de las más devastadoras es el glaucoma, que por ser una enfermedad silenciosa (sus síntomas aparecen cuando la enfermedad se encuentra en un estado avanzado) no se detecta fácilmente.

Una de las ventajas de la oftalmología es ser una de las pocas ciencias médicas objetivas ya que en la mayoría de enfermedades no da lugar a especulaciones o malas interpretaciones en el diagnóstico, en este contexto los sistemas expertos pueden ayudar para optimizar los procesos de diagnóstico de los trastornos oculares [ofta2].

Los sistemas expertos son llamados así porque emulan el comportamiento de un experto en un dominio concreto y en ocasiones son utilizados por estos para optimizar la rapidez de respuesta dando lugar a una mejora significativa en la productividad del experto. Son aplicaciones informáticas capaces de solucionar un conjunto de problemas que exigen un gran conocimiento sobre un determinado tema. Un sistema experto es un conjunto de programas que, sobre una base de conocimientos posee información de uno o más expertos en un área específica. Estos sistemas imitan las actividades de un humano para resolver problemas de distinta índole (no necesariamente tiene que ser de inteligencia artificial), se



basan en el conocimiento declarativo (hechos sobre objetos y situaciones) y el conocimiento de control (información sobre el seguimiento de una acción) [SE1].

Estos sistemas expertos se clasifican en:

- **Sistemas expertos basados en reglas:** Trabajan mediante la aplicación de reglas preestablecidas y comparación de resultados. También pueden trabajar por inferencia lógica dirigida, bien empezando con una evidencia inicial en una determinada situación y dirigiéndose hacia la obtención de una solución, o bien con hipótesis sobre las posibles soluciones y volviendo hacia atrás para encontrar una evidencia existente (o una deducción de una evidencia existente) que apoye una hipótesis en particular.
- **Sistemas expertos basados en redes neuronales:** Las cuales consisten en una simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez [SE2].

Los sistemas expertos orientados al diagnóstico pueden estar encaminados a:

- **Sistemas expertos para la toma de decisiones:** permiten a una persona no calificada tomar una decisión que va más allá de su nivel de conocimientos o experiencia (uso más común en sistemas expertos orientados a la industria).
- **Sistemas expertos para el apoyo a las decisiones:** Permiten recordar, a una persona que toma decisiones con experiencia, opciones a considerar que alguna vez conoció (sistemas expertos para el apoyo en el diagnóstico médico) [SE1].



### 1.3 Antecedentes

El Oftalmólogo Pablo Zeas provee un servicio a la comunidad, colaborando en la solución de los problemas oftalmológicos, ofreciendo a la colectividad del Austro y del país sus servicios para el diagnóstico de un importante número de trastornos oculares, así como valoraciones funcionales.

Entre los exámenes que el Doctor realiza se encuentran: pruebas para la evaluación de trastornos oculares como son el Glaucoma y su valoración en cada una de sus variantes y clasificaciones.

### 1.4 Alcance

El alcance del proyecto de tesis planteado, comprende:

- Diseño e implementación de la base de conocimientos para el diagnóstico de glaucoma crónico de ángulo abierto (GCAA).
- Diseño e implementación del módulo de inferencia para el apoyo en la toma de decisiones.
- Desarrollo de un módulo para el procesamiento de imágenes en la prueba de campimetría. Que será capaz de:
  - En una primera visita del paciente comparar los resultados con un patrón de visión normal con la finalidad de apoyar al diagnóstico.
  - En visitas posteriores, comparará con los resultados obtenidos anteriormente y de esta manera evaluar el progreso de la enfermedad.
- Con la finalidad de hacer un seguimiento de la enfermedad del paciente se implementará un módulo de manejo de información acerca de los mismos.
- Implementación del método S.T.A.R (Scoring Tool for Assessing Risk, Método para la Valoración de Riesgo de Glaucoma a 5 años).



- Implementación de un módulo de ilustración de la anatomía del ojo humano mediante el uso de imágenes y animaciones 2D. El módulo estará en la capacidad de mostrar:
  - Vista de la anatomía de ojo humano con parámetros normales con fines ilustrativos.
  - Imágenes del ojo humano con parámetros normales de visión.
  - Imágenes del ojo humano con diferentes anomalías en el campo visual.
- Elaboración de un módulo de ayuda y manual de usuario como recurso local.

El tiempo estimado para el desarrollo del proyecto es de 10 meses a partir de la aprobación del tema de tesis.

## **1.5 Justificación**

En la actualidad dentro de nuestro país se presenta un elevado número de casos de Glaucoma Crónico de Ángulo Abierto, que no son detectados tempranamente por diversas razones y que llegan sin un diagnóstico hacia la ceguera.

La implementación de un sistema experto será de gran ayuda para los Oftalmólogos, ya que contarán con una herramienta que les permitirá optimizar el tiempo de respuesta en el diagnóstico dentro de una consulta, debido a que el experto no requerirá procesar un sinnúmero de parámetros necesarios para diagnosticar la enfermedad, además contará con un registro de la evolución de grado de enfermedad del paciente en cada visita. Lo que conlleva a un mejor diagnóstico de la enfermedad.

## **1.6 Auspicio Docente**

El presente proyecto de Tesis cuenta con el auspicio docente del Ing. Ángel Espinoza, catedrático de la Facultad de Ingeniería de la Universidad de Cuenca.



## **1.7 Objetivos**

### **1.7.1 Objetivo General**

Desarrollar e Implementar un Sistema Experto Para el Apoyo en el Diagnóstico de Glaucoma Crónico de Ángulo Abierto.

### **1.7.2 Objetivos Específicos**

1. Definir el tipo de sistema experto (basado en reglas o en redes neuronales) a ser implementado en la aplicación.
2. Crear una base de conocimientos que esté en la capacidad de asistir con el diagnóstico del Glaucoma Crónico de Ángulo Abierto (GCAA por sus siglas en español) y el método STAR.
3. Implementar una base de datos con la información del paciente y la capacidad de almacenar los resultados de la campimetría en forma gráfica.
4. Proveer al especialista un módulo de procesamiento de imágenes que este en la capacidad de comparar los resultados del examen de campimetría realizado al paciente con uno de parámetros normales o con exámenes llevados a cabo anteriormente.
5. Implementar un módulo de ilustración de la anatomía del ojo humano mediante el uso de imágenes y animaciones 2D.
6. Permitir al especialista modificar los parámetros ya existentes y sus rangos establecidos para el GCAA, incluyendo los valores normales.

## **1.8 Recursos del Proyecto**

### **1.8.1 Recursos de Hardware**

Los recursos de hardware que se utilizarán en este proyecto se especifican a continuación:

- Dos computadores portátiles para el desarrollo del proyecto, propiedad de los desarrolladores de tesis.



### 1.8.2 Recursos de Software

Los recursos de software que se utilizarán en este proyecto son software propietario.

- Sistema Operativo Windows Xp y superior.
- Visual Studio .Net 2008
- SQL server 2005
- MATLAB 2009
- Adobe Flash CS4

### 1.8.3 Recursos Humanos

- Andrés Molina Ortega, desarrollador de tesis.
- Jonattan Ochoa Mora, desarrollador de tesis.
- Ing. Ángel Espinoza, Director de tesis.
- Dr. Pablo Zeas, Médico especialista.
- Especialistas en diferentes áreas de la Oftalmología.

## 1.9 Metodología

Para cumplir con cada uno de los objetivos planteados se llevará a cabo la siguiente metodología:

**Definir el tipo de sistema experto (basado en reglas o en redes neuronales) a ser implementado en la aplicación.**

Analizar y catalogar todas las variables a ser utilizadas en la base de conocimientos, así como el método en el cual se basará el motor de inferencia para el apoyo a la toma de decisiones, realizando una investigación a fondo de las ventajas y desventajas de cada tipo de sistema experto (basado en reglas o en redes neuronales), con el objetivo de tomar una decisión del tipo de sistema experto a ser implementado.

**Crear una base de conocimientos que este en la capacidad de diagnosticar el Glaucoma Crónico de Ángulo Abierto (GCAA) y el método STAR.**



Para cumplir con este objetivo usaremos las fases de adquisición del conocimiento donde se deberá transferir o transformar los conocimientos del experto al programa, para lo que se tendrá que cumplir con las siguientes fases:

- Fase inicial de Conceptualización y Análisis: Se determinarán las áreas a desarrollar para lo cual se hará uso de:
  - Entrevistas tanto escritas como orales a las personas involucradas en el proyecto.
  - Lluvia de ideas con el objetivo de determinar la información que se encontrará en la base de conocimientos.
  - Elaboración de diagramas entidad relación, diccionario de datos.
- Fase de desarrollo:
  - Se procede a detallar las cadenas de razonamiento y a desarrollar el conjunto de las reglas principales.
  - Se codifican reglas de producción individuales y el experto puede examinarlas y criticarlas.
  - Se realiza un prototipo a ser probado por el experto.
- Fase final: La mayor parte de la estructura de la Base de Conocimientos está en correctamente diseñada y las interacciones con el experto producen cambios más bien pequeños que pueden ser modificados oportunamente.

**Implementar una base de datos con la información del paciente y la capacidad de almacenar los resultados de la campimetría en forma gráfica.**

- Normalización de las tablas.
- Elaboración de diagramas entidad relación.
- Elaboración del diccionario de datos.

**Proveer al especialista un módulo de procesamiento de imágenes que este en la capacidad de comparar los resultados del examen de campimetría realizado al paciente con uno de parámetros normales o con exámenes llevados a cabo anteriormente.**

Se utilizará la herramienta MATLAB para el procesamiento de las imágenes de los resultados de los exámenes, dicha aplicación será integrada a .Net.

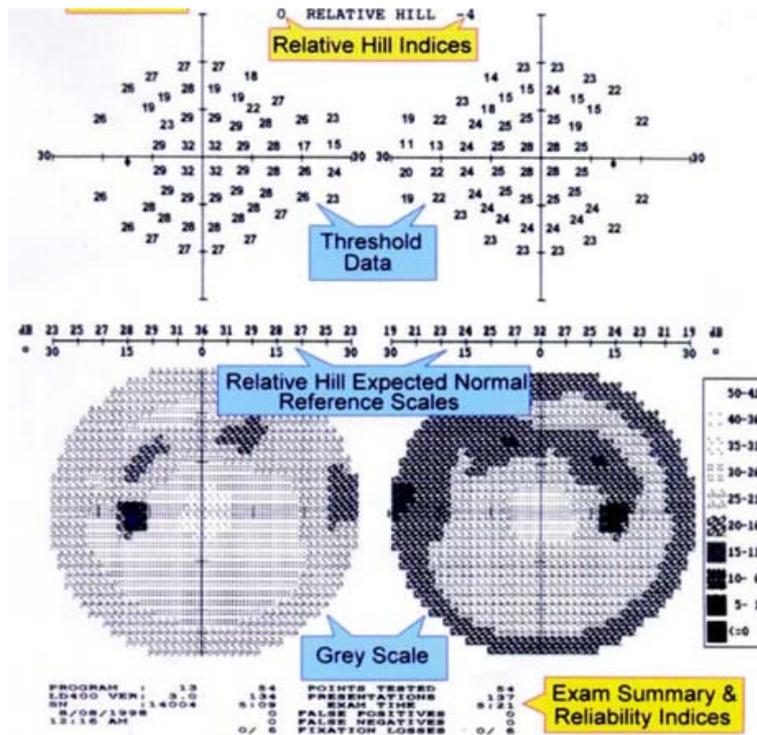


Figura: 1.1 Resultado de una Campimetría

En la Figura: 1.1 podemos observar el resultado de un examen del campo visual (Campimetría o Perimetría), donde las zonas más oscuras (escala de grises) representan una mayor pérdida de la visión a diferencia de las zonas blancas que indican visión normal.

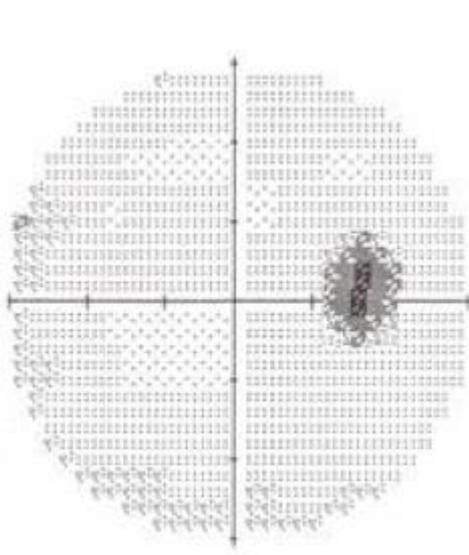


Figura: 3 Resultados Normales

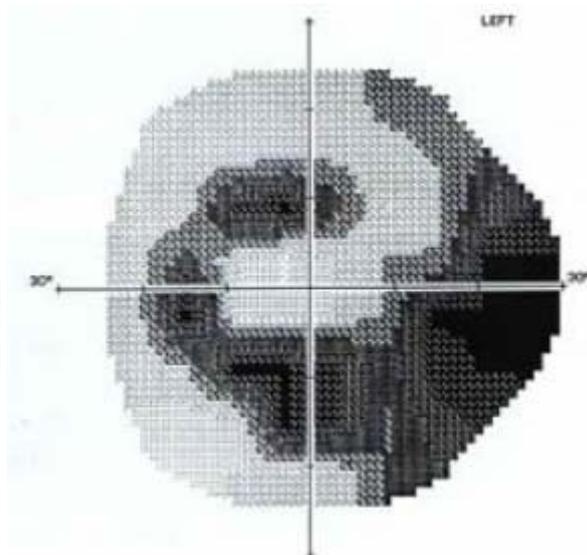


Figura: 4 Alteraciones por Glaucoma

En las Figuras 3 y 4, podemos observar los resultados de la campimetría de una manera más detallada, en la figura 3 (campimetría normal) se puede apreciar una pequeña



mancha (llamada punto ciego) y es el punto donde se conecta el nervio óptico al ojo. En la figura 4, podemos apreciar la presencia de tonalidades de gris hasta llegar al negro absoluto, que nos indica que en esas porciones la ausencia de visión es absoluta.

### **Implementar un módulo de ilustración de la anatomía del ojo humano mediante el uso de imágenes y animaciones 2D.**

Dentro de la aplicación de .Net junto con SQL Server se almacenarán y cargarán las imágenes a ser utilizadas por el especialista.

### **Permitir al especialista modificar los parámetros ya existentes y sus rangos establecidos para el GCAA, incluyendo los valores normales.**

En el caso de ser necesaria alguna modificación en los parámetros del GCAA serán modificados dichos datos desde la aplicación en la base de datos SQL Server.

## **1.10 Entregables**

Analizando cada uno de los Objetivos del Tema de Tesis planteado, en consenso con las partes interesadas, se ha contemplado el siguiente Plan de Entregables.

<b>Nombre de Entregable</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Modelo de Datos y Módulo de Inferencia</b>	Documento Escrito	Especificaciones de la Base de Conocimientos a ser Implementada, incluyendo Diccionario de Datos, criterios de validación para los campos. De manera que sea el punto de partida para el diseño de los módulos posteriores. Además del Modelo de datos para el manejo de pacientes.  Análisis y Diseño del conjunto de reglas necesarias para el diagnóstico del GCAA y Método S.T.A.R



<b>Prototipo V 0.1</b>	Aplicación	<p>Una versión preliminar que deberá estar en la capacidad de:</p> <ul style="list-style-type: none"><li>• Diagnosticar el tipo de GCAA, basándose en la aplicación de las reglas y parámetros establecidos en el entregable anterior.</li><li>• Determinación del porcentaje de riesgo de glaucoma a 5 años, basándose en las reglas establecidas por el método S.T.A.R</li></ul>
<b>Prototipo V 0.2</b>	Aplicación	<p>Aprobado el entregable (Prototipo V0.1), se añadirá las siguientes funcionalidades a la aplicación. Este entregable contará con los siguientes módulos:</p> <ul style="list-style-type: none"><li>• Módulo de Procesamiento de Imagen con las siguientes funcionalidades:<ul style="list-style-type: none"><li>○ Comparación de resultados entre una Campimetría con parámetros normales y la del paciente.</li><li>○ Seguimiento y comparación entre resultados previos del mismo paciente, y exámenes recientes.</li></ul></li><li>• Módulo de Ilustración de la Anatomía del Ojo humano.<ul style="list-style-type: none"><li>○ Módulo de Modificación de parámetros existentes en el Modelo de Datos.</li></ul></li></ul>
<b>Versión Final</b>	Aplicación	<p>Entrega de la versión final, que comprende todas las funcionalidades descritas en los Objetivos.</p>



		Ajustes a los prototipos anteriores.
<b>Manual de Usuario</b>	Documento residente en la aplicación y en Formato Físico.	Documento de soporte para el uso del Sistema, que contiene la descripción y pasos a seguir para completar cada una de las tareas disponibles en la aplicación.



## CAPÍTULO 2



## 2.1 Ingeniería del Conocimiento

El término de Ingeniería del Conocimiento fue acuñado en la década de los 70 orientado hacia la construcción de Sistemas Expertos (Sistemas Basados en el Conocimiento, SBC), lo cual fue el centro de investigación de Inteligencia Artificial.

En el año 1977 fue presentado el primer sistema experto conocido como MYCIN, a partir de este proyecto se han presentado diferentes sistemas con aplicación en múltiples áreas del conocimiento. En la actualidad estos artefactos “inteligentes” funcionan muy bien en tareas específicas y sus juicios rivalizan con los de los humanos en el nivel de destreza.

En la actualidad la Ingeniería del conocimiento se apoya en metodologías de instrucciones y en las ciencias de la computación y de la Información, intentando representar el conocimiento y razonamiento humanos en un determinado dominio, dentro de un sistema artificial.

El punto clave del desarrollo de un Sistema Basado en el Conocimiento es el momento de traspasar el conocimiento que posee el experto a un sistema real. En este proceso no sólo se han de captar los elementos que componen el dominio del experto, sino que también se han de adquirir las metodologías de resolución que utilizan éstos.

Este trabajo de extracción del conocimiento (Knowledge elicitation) se realiza durante la interacción entre dos personajes, el ingeniero del conocimiento (IC) (persona que conoce el formalismo de representación que utilizará el SBC) y el experto (persona que posee el conocimiento, pero que no tiene por qué usar un formalismo para representarlo).

Durante las entrevistas entre el IC y el experto, el primero ha de ayudar a sistematizar el conocimiento del experto, consiguiendo que vaya explicitando las diferentes técnicas que utiliza para resolver los problemas de su dominio, de manera que se puedan representar en un formalismo computable. Esta metodología de extracción del conocimiento es bastante lenta (se cita que se suele extraer información equivalente a de dos a cinco reglas de producción por día).



Varias son las dificultades que dan una producción tan baja a esta metodología:

La naturaleza especializada del dominio hace que el IC deba aprender unas nociones básicas para que pueda establecerse una comunicación (Vocabulario básico, elementos que intervienen en el dominio, formalismos que utilizan los expertos, etc.).

Los expertos se encuentran más cómodos pensando en términos de ejemplos típicos que razonando en términos generales, que son de los que realmente se podría hacer una mejor abstracción.

La búsqueda de un formalismo de representación que se adapte adecuadamente al problema y que sea fácil de interpretar y adoptar por el experto. Este formalismo ha de ser susceptible de ser transformado en algo computable.

Por lo general, a los expertos les es muy difícil explicitar los pasos que utilizan para resolver los problemas. Es la que se ha denominado paradoja del experto. Cuanta más experiencia, menos explícitos son los razonamientos del experto y más ocultos los métodos de resolución.

Si observamos cómo un experto resuelve un problema, éste omite muchas cadenas de razonamiento e información que da por supuesta, y a la que no asigna importancia dentro de la resolución, pero que si se quiere abordar de manera sistemática sí es necesaria.

Con todas estas circunstancias, podemos observar que la auténtica dificultad de la extracción del conocimiento estriba en descubrir los métodos mediante los que se usa el conocimiento en la resolución y no tanto en la adquisición del conocimiento estático del problema (elementos del problema y relaciones).

Sobre la adquisición de los elementos básicos del dominio, existen bastantes herramientas automáticas, encuadradas dentro del área del aprendizaje automático, que permiten reducir el esfuerzo. Sobre la adquisición automática de conocimiento de resolución de problemas, también hay algunas técnicas automáticas, pero que por lo general son a niveles bastante básicos.



## 2.2 Inteligencia Artificial

Se denomina inteligencia artificial (IA) a la rama de las Ciencias de la Computación dedicada al desarrollo y construcción de agentes (entidades) inteligentes (racionalidad).

Entiéndase a un agente como cualquier cosa capaz de percibir su entorno (recibir entradas), procesar tales percepciones y actuar en su entorno (proporcionar salidas). Además se define a la racionalidad como la característica que posee para maximizar un resultado esperado.

Por lo tanto, y de manera más específica la inteligencia artificial es la disciplina que se encarga de construir procesos que al ser ejecutados sobre un diseño físico producen acciones o resultados que maximizan una medida de rendimiento determinada, basándose en la secuencia de entradas percibidas y en el conocimiento almacenado en tal diseño.

Existen distintos tipos de conocimiento y medios de representación del conocimiento, el cual puede ser cargado en el agente por su diseñador o puede ser aprendido por el mismo agente utilizando técnicas de aprendizaje.

También se distinguen varios tipos de procesos válidos para obtener resultados racionales, que determinan el tipo de agente inteligente:

- Ejecución de una respuesta predeterminada por cada entrada (análogas a actos reflejos en seres vivos).
- Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.
- Algoritmos genéticos (análogo al proceso de evolución de las cadenas de ADN).
- Redes neuronales artificiales (análogo al funcionamiento físico del cerebro de animales y humanos).
- Razonamiento mediante una lógica formal (análogo al pensamiento abstracto humano).

También existen distintos tipos de percepciones y acciones, pueden ser obtenidas y producidas, respectivamente por sensores físicos y sensores mecánicos en máquinas, pulsos



eléctricos u ópticos en computadoras, tanto como por entradas y salidas de bits de un software y su entorno software.

Varios ejemplos se encuentran en el área de control de sistemas, planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, reconocimiento de escritura, reconocimiento del habla y reconocimiento de patrones. Los sistemas de IA actualmente son parte de la rutina en campos como economía, medicina, ingeniería y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia como ajedrez de computador y otros videojuegos.

### **2.2.1 Definición de Inteligencia Artificial**

La inteligencia artificial a lo largo de la historia se ha definido dentro de cuatro áreas:

- Sistemas que piensan como humanos.
- Sistemas que piensan racionalmente.
- Sistemas que actúan como humanos.
- Sistemas que actúan racionalmente.

Cada uno de estos enfoques presenta una definición acorde a los autores que las defienden, aquí analizaremos cada uno de ellas con un mayor detalle.

### **Comportamiento Humano: Prueba de Turing**

Esta prueba fue diseñada por Alan Turing en el año de 1950 para proporcionar una definición operacional y satisfactoria de la inteligencia. Dicha prueba está basada en la incapacidad que tiene un evaluador humano de distinguir si las respuestas a una serie de preguntas planteadas pertenecen o no a una persona, en este caso el computador superaría la prueba. En la actualidad para que un computador supere la prueba necesita cumplir con las siguientes capacidades:

- Procesamiento de lenguaje natural que le permita comunicarse satisfactoriamente.
- Representación del conocimiento para almacenar lo que conoce o siente.
- Razonamiento automático para utilizar la información almacenada para responder a preguntas y extraer nuevas conclusiones.
- Aprendizaje automático para adaptarse a nuevas circunstancias.



La Prueba de Turing evita la interacción entre el evaluador y el computador, aunque existe la llamada Prueba Global de Turing en la cual se evalúa la capacidad de percepción del computador para lo cual este debería estar dotado de:

- Visión Computacional para percibir objetos.
- Robótica para manipular y mover objetos.

### **Pensar como Humano: El Enfoque del modelo Cognitivo**

En este enfoque para poder evaluar al programa primeramente es necesario determinar cómo piensan los humanos. Se utilizan dos métodos:

- Mediante introspección “Intentado atrapar nuestros propios pensamientos conforme éstos van pareciendo”. (Russel & Norvig, 2004).
- Mediante experimentos psicológicos.

Cuando se cuenta con una teoría lo suficientemente precisa de cómo trabaja la mente, el programa de computador puede ser implementado, a continuación se deberá evaluar los datos de entrada/salida del programa junto con los tiempos de reacción, si estos son similares a los de un humano entonces se puede decir que algunos mecanismos del programa son comparables con los que utiliza un ser humano.

### **Pensamiento racional: el enfoque de las leyes del pensamiento**

En 1965 existían programas que resolvían cualquier problema descrito en notación lógica (el programa no se detenía hasta encontrar una solución). La tradición logística construye sistemas inteligentes a partir de estos programas.

Existen dos problemas para la elaboración de estos programas. La dificultad al momento de transformar el conocimiento informal a términos formales que requieren de notación lógica. Y el segundo problema es la cantidad de datos necesarios para resolver un problema, los cuales pueden agotar los recursos de las computadoras.



## Actuar de forma racional: el enfoque del agente racional

Agente viene del latín “agere” que quiere decir agente. Los agentes informáticos a más de razonar se encuentran dotados de controles autónomos para que perciban su entorno, se adapten a los cambios y sean capaces de alcanzar objetivos. Un agente racional es aquel que actúa buscando el mejor resultado esperado.

La inteligencia artificial busca realizar una acción dada para llegar a una conclusión lógica mediante el uso de un agente racional. Sin embargo existen acciones que no llevan a una situación lógica y en las cuales se debe tomar una decisión. Existen también situaciones en las cuales se actúa racionalmente y no es necesario realiza inferencias.

Este enfoque nos presenta diversos problemas. El primero de ellos es que efectuar inferencias lógicas es solo uno de los pasos para llegar a la racionalidad, el segundo problema es que la conducta humana obedece a un complejo proceso evolutivo que en una gran medida es desconocido.

### 2.2.2 Clasificación de la IA

La IA se divide en dos escuelas de pensamiento:

- La inteligencia artificial convencional
- La inteligencia computacional

#### Inteligencia artificial convencional

Se conoce también como IA simbólico-deductiva. Está basada en el análisis formal y estadístico del comportamiento humano ante diferentes problemas:

- **Razonamiento basado en casos:** Ayuda a tomar decisiones mientras se resuelven ciertos problemas concretos y aparte que son muy importantes requieren de un buen funcionamiento.
- **Sistemas expertos:** Infieren una solución a través del conocimiento previo del contexto en que se aplica y ocupa de ciertas reglas o relaciones.
- **Redes bayesianas:** Propone soluciones mediante inferencia estadística.



- **Inteligencia artificial basada en comportamientos:** que tienen autonomía y pueden autoregularse y controlarse para mejorar.
- **Smart process management:** facilita la toma de decisiones complejas, proponiendo una solución a un determinado problema al igual que lo haría un especialista en la actividad.

## **Inteligencia artificial computacional**

La Inteligencia Computacional también conocida como IA subsimbólica-inductiva implica desarrollo o aprendizaje interactivo por ejemplo, modificaciones interactivas de los parámetros en sistemas conexionistas. El aprendizaje se realiza basándose en datos empíricos.

Se centra en el estudio de mecanismos adaptativos para permitir el comportamiento inteligente de sistemas complejos y cambiantes. Se presenta como una alternativa a la GOFAI (Good Old-Fashioned Artificial Intelligence), tratando de no confiar en algoritmos heurísticos tan habituales en la Inteligencia Artificial más tradicional. Dentro de la Inteligencia Computacional podemos encontrar técnicas como las Redes Neuronales, Computación Evolutiva, Swarm Intelligence, Sistemas Inmunes Artificiales o Sistemas difusos. También se relaciona con técnicas como los Fractales, Teoría del Caos, Wavelets, etc.

La Inteligencia Computacional combina elementos de aprendizaje, adaptación, evolución y Lógica difusa para crear programas que son, en cierta manera, inteligentes. La investigación en Inteligencia Computacional no rechaza los métodos estadísticos, pero muy a menudo aporta una vista complementaria. Las Redes Neuronales son una rama de la inteligencia computacional muy relacionada con el aprendizaje automático.

## **2.3 Sistemas Expertos**

### **2.3.1 Introducción**

Los Sistemas Expertos han sido producto de la evolución de la Inteligencia Artificial. Empezaron a emerger como Sistemas desarrollados como investigación en las Universidades



durante la década de 1970. Y desde entonces se han convertido en una de las innovaciones más importantes de la Inteligencia Artificial, ya que han demostrado ser productos de un éxito comercial ascendente así como también interesantes herramientas de desarrollo e investigación. [2]

Los Sistemas Expertos han probado ser de gran efectividad en un sinnúmero de problemas, cuyo dominio normalmente requeriría la clase de inteligencia que posee un experto humano. Su campo de aplicación abarca muchas áreas, dondequiera que los expertos humanos sean necesarios para resolver problemas, los Sistemas Expertos son aptos para su aplicación. [2]

Consideramos a alguien un experto en un área o en un problema cuando este individuo tiene un conocimiento especializado sobre este. En los Sistemas Expertos a este tipo de conocimiento se lo llama conocimiento sobre el dominio. La palabra dominio se usa para enfatizar que el conocimiento pertenece a un problema específico.

### **2.3.2 Historia de los Sistemas Expertos**

Los primeros Sistemas Expertos surgieron desde Laboratorios de Investigación en algunas de las mejores universidades de Estados Unidos durante la década de los 1960s y 1970s. Estos Sistemas fueron desarrollados para resolver problemas específicos enfatizando el uso del conocimiento antes que el uso de Algoritmos y Métodos generales de búsqueda. Este alcance marcó un importante punto de partida para el desarrollo de las Arquitecturas de los Sistemas de Inteligencia Artificial de la época [2]. A mediados de la década de los 60, los Investigadores Alan Newel y Herbert Simon desarrollaron un programa llamada G.P.S (General Problem Solver, Solucionador General de Problemas). Podía trabajar con Criptoaritmética, con las Torres de Hanoi y con otros problemas similares. Lo que no podía el G.P.S era resolver problemas del mundo real como un diagnóstico médico [4].

Algunos Investigadores decidieron entonces cambiar por completo el enfoque del problema restringiendo su ambición a un dominio específico e intentando simular el



razonamiento de un experto humano, entonces en lugar de dedicarse a computarizar la inteligencia general, se centraron en dominios del conocimiento muy concretos. De esta manera nacieron los Sistemas Expertos [4].

El primer Sistema Experto en ser desarrollado fue DENDRAL, desarrollado en la Universidad de Stanford a finales de los años 60. Este sistema era capaz de identificar estructuras químicas moleculares a partir de su análisis espectrográfico. DENDRAL usa conocimiento Heurístico obtenido de expertos Químicos para ayudar a construir el problema, de ese modo se reduce el rango de búsqueda de la solución. Durante las pruebas, DENDRAL descubrió un número de estructuras previamente desconocidas aún por los expertos [2].

Como los investigadores ganaron más experiencia en el desarrollo de DENDRAL, se dieron cuenta sobre la dificultad de obtener el conocimiento del experto. Esto llevó al desarrollo de Meta-DENDRAL, un componente para DENDRAL que estaba en la capacidad de aprender nuevas reglas.

Poco después de que DENDRAL fue terminado, el desarrollo de MYCIN empezó en la Universidad de Stanford, MYCIN es un Sistema Experto que diagnostica enfermedades infecciosas de la sangre determina una lista recomendada de terapias para el paciente. Como parte del proyecto de Programación Heurística de la Universidad de Stanford, muchos proyectos se derivaron de MYCIN, y fueron completados incluyendo un componente de adquisición de conocimiento llamado THEIRESIUS, un componente tutorial llamado GUIDON, y un componente tipo Shell llamado EMYCIN (por Essential MYCIN). EMYCIN fue utilizado para construir otros sistemas de diagnóstico incluyendo PUFF, un Sistema Experto para el diagnóstico de enfermedades pulmonares. EMYCIN se convirtió en un modelo de diseño para muchas Sistemas Expertos comerciales [2].

El rendimiento de MYCIN aumentó significativamente con el pasar de los años y la adquisición de nuevo conocimiento. Pruebas indican que el desempeño de este sistema iguala o excede al desempeño de un Físico humano experimentado. La base de



conocimientos inicial de MYCIN contenía solo 200 reglas, este número fue gradualmente incrementándose hasta más de 600 reglas a finales de los años 1980 [2].

Otros Sistemas Expertos más recientes como PROSPECTOR, asisten a los Geólogos en el descubrimiento de depósitos de minerales. R1 es un sistema experto usado por DIGITAL EQUIPMENT CORP. Para seleccionar y configurar componentes para Sistemas Complejos de Computadores. Desde la introducción de estos sistemas más recientes, se han completado numerosas versiones comerciales y militares con un alto grado de éxito [2].

Actualmente el duro, difícil y cambiante mercado competitivo se vuelve más duro por la gran diversidad de información que se ven obligados a almacenar y analizar, razones por la cual las organizaciones se ven en la necesidad de recurrir a poderosas y robustas herramientas o Sistemas que les sirvan de soporte a la hora de tomar decisiones. De esta forma, estos Inteligentes, precisos y eficientes Sistemas son adoptados por más organizaciones, en los cuales se convierten en una importante estrategia de negocio. Por otra parte es importante mencionar que los SE seguirán siendo usados en todas y cada una de las áreas donde los expertos humanos son escasos. Por consecuencia de lo anterior estos sistemas son usados por personas no especializadas, y en algunos casos por especialistas con poca experiencia, por lo cual el uso frecuente de los Sistemas Expertos, produce conocimiento en los usuarios del mismo [4].

### **2.3.3 Definición de Sistema Experto**

Es una aplicación informática capaz de solucionar un conjunto de problemas que exigen un gran conocimiento sobre un determinado tema. Un Sistema Experto es un conjunto de programas que, sobre una base de conocimientos, posee información de uno o más expertos en un área específica. Se puede entender como una rama de la Inteligencia Artificial, donde el poder de resolución de un problema en un programa de computadora viene del conocimiento de un dominio específico. Estos Sistemas imitan las actividades de un humano para la resolución de problemas de distinta índole (no necesariamente tiene que ser de inteligencia artificial). También podemos mencionar que un Sistema Experto se basa



en conocimiento declarativo (hechos sobre objetos, situaciones) y conocimiento de control (información sobre el seguimiento de una acción).

Para que un Sistema Experto sea una herramienta efectiva, los usuarios deben interactuar de una forma fácil, reuniendo dos capacidades para poder cumplirlo:

1. Explicar su Razonamiento o Base de Conocimientos: Los Sistemas Expertos se deben desarrollar siguiendo ciertas Reglas o pasos comprensibles de manera que se pueda generar la explicación para cada una de las reglas, que a la vez se basan en hechos.
2. Adquisición de nuevos conocimientos o Integrador del Sistema: Son mecanismos de razonamiento que nos sirven para modificar conocimientos anteriores. Sobre la base de lo anterior podemos decir que los Sistemas Expertos son el producto de investigaciones en el campo de la Inteligencia Artificial, ya que ésta no pretende sustituir a los expertos humano, sino que desea ayudarlos a realizar con más rapidez y eficacia todas las tareas que realiza.

Debido a esto en la actualidad se está mezclando diferentes técnicas o aplicaciones aprovechando las ventajas que cada una de estas ofrece para poder tener empresas más seguras. Un ejemplo de estas técnicas sería los agentes que tienen la capacidad de negociar y navegar a través de recursos en línea; por esa razón en la actualidad juega un papel preponderante en los Sistemas Expertos. [5]

#### **2.3.4 Características de los Sistemas Expertos**

Los Sistemas Expertos se diferencian de los Sistemas de Computo convencionales en algunas formas:

1. Los Sistemas Expertos usan el Conocimiento antes que los Datos para controlar el proceso de solución. Mucho del conocimiento usado es Heurístico antes que Algorítmico.
2. El Conocimiento es codificado y mantenido como una Entidad separada del programa de control. Como tal, este no es compilado junto con el programa de control, esto permite el incremento y refinamiento de la Base de Conocimientos sin recompilar el programa de control. Además es posible en algunos casos usar diferentes Bases de Conocimiento con el mismo programa de Control, lo que nos permitiría producir diferentes tipos de Sistemas Expertos. Estos sistemas son conocidos como "SE Shell" y pueden ser cargados con múltiples Bases de Conocimiento.



3. Los Sistemas Expertos están en la capacidad de explicar el ¿Cómo? Llegaron a determinada conclusión o resultado, y ¿Porqué? Se solicita determinada información en la consulta. Esto es muy importante ya que le da al usuario la oportunidad de entender la habilidad de razonamiento del Sistema. De ese modo se mejora la confiabilidad de estos Sistemas por parte del usuario.
4. Los Sistemas Expertos son Representaciones Simbólicas para el Conocimiento (reglas, redes o tramas) y realiza su tarea de inferencia a través de cálculos simbólicos que asemejan representaciones del lenguaje natural. (Una excepción de esto son los Sistemas Expertos basados en Arquitecturas de Redes Neuronales)
5. Los Sistemas Expertos a menudo razonan usando Meta-Conocimiento, esto es que razonan con Conocimiento acerca de ellos mismos, además de sus limitaciones de su propio Conocimiento y Capacidades.[2]

### 2.3.5 Clasificación de los Sistemas Expertos

Para poder establecer una clasificación de los Sistemas Expertos, debemos enfocarnos en la naturaleza del problema para el que se lo desarrolla. Los problemas con los que pueden tratar los sistemas expertos pueden clasificarse en dos tipos:

- Problemas esencialmente deterministas
- Problemas esencialmente estocásticos

Por ejemplo, en el caso de un sistema experto para Transacciones Bancarias o un Sistema de Control de Tráfico pueden contener algunos elementos de incertidumbre, pero son esencialmente problemas deterministas. Por otra parte, en el campo médico, las relaciones entre síntomas y enfermedades se conocen sólo en un cierto grado de certeza (la presencia de un conjunto de síntomas no siempre implica la presencia de una enfermedad). Estos tipos de problemas pueden también incluir algunos elementos deterministas, pero se trata fundamentalmente de problemas estocásticos.

Por esta razón, los sistemas expertos pueden clasificarse en dos tipos principales según la naturaleza de los problemas para los que están diseñados: deterministas y estocásticos.



Los problemas de tipo determinista pueden ser formulados usando un conjunto de reglas que relacionan varios objetos bien definidos.

Los sistemas expertos que tratan problemas deterministas son conocidos como **Sistemas Expertos basados en Reglas**, porque sacan sus conclusiones basándose en un conjunto de reglas preestablecidas y comparación de resultados utilizando un mecanismo de razonamiento lógico (inferencia lógica dirigida), bien empezando con una evidencia inicial en una determinada situación y dirigiéndose hacia la obtención de una solución, o bien con hipótesis sobre las posibles soluciones y volviendo hacia atrás para encontrar una evidencia existente (o una deducción de una evidencia existente) que apoye una hipótesis en particular [1].

En situaciones inciertas, es necesario introducir algunos medios para tratar la incertidumbre. Por ejemplo, algunos sistemas expertos usan la misma estructura de los sistemas basados en reglas, pero introducen una medida asociada a la incertidumbre de la reglas y a la de sus premisas. En este caso se pueden utilizar algunas fórmulas de propagación para calcular la incertidumbre asociada a las conclusiones. Durante las últimas décadas han sido propuestas algunas medidas de incertidumbre. Algunos ejemplos de estas medidas son los *factores de certeza*, usados en Shells para generar sistemas expertos tales como el sistema experto MYCIN; *la lógica difusa* y *la teoría de la evidencia* de Dempster y Shafer.

Otra medida intuitiva de incertidumbre es la **probabilidad**, en la que la distribución conjunta de un conjunto de variables se usa para describir las relaciones de dependencia entre ellas, y se sacan conclusiones usando fórmulas muy conocidas de la teoría de la probabilidad. Un caso conocido de esto es el sistema experto PROSPECTOR, que utiliza el teorema de Bayes para la exploración de minerales.

Los Sistemas expertos que utilizan la probabilidad como medida de incertidumbre se conocen como **Sistemas Expertos Probabilísticos** y la estrategia de razonamiento que usan es conocida como **Razonamiento probabilístico** o Inferencia probabilística.



En un principio los sistemas expertos de tipo probabilístico tuvieron que enfrentar varios obstáculos, debido a las dificultades en la definición de la distribución de probabilidad conjunta de las variables, lo que ciertamente atrasó su desarrollo. Pero con la introducción de los **modelos de redes probabilísticas**, estos problemas se han superado y estos sistemas expertos han regresado de forma espectacular durante los últimos años. Estos modelos, que incluyen las redes de Markov y la Bayesianas, se basan en una representación gráfica de las relaciones entre las variables, que conducen a formas más eficientes de definición de la distribución conjunta de probabilidad y una propagación de incertidumbre más eficiente, que nos permite sacar conclusiones.

En los últimos años han surgido los **Sistemas Expertos basados en Redes Neuronales**, que consisten en la simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo de estos sistemas es conseguir que las máquinas arrojen respuestas similares a las que es capaz de dar el cerebro. [2]

Ahora si tomamos en cuenta el propósito y el usuario final al que está destinado el uso del sistema, podemos clasificar a los Sistemas Expertos en:

**Sistemas Expertos para la toma de decisiones:** Estos Sistemas permiten a una persona no calificada, tomar una decisión que va más allá de su nivel de conocimientos o experiencia. Este tipo de Sistemas son más usados o están orientados a la Industria.

**Sistemas Expertos para el apoyo en las decisiones:** Este tipo de sistemas permiten recordar a una persona con experiencia, decisiones u opciones a considerar que alguna vez conoció, por ejemplo los Sistemas Expertos para el apoyo en el diagnóstico médico.

### **2.3.6 Ventajas y Limitaciones de los Sistemas Expertos**

El Desarrollo o la adquisición de un Sistema Experto es generalmente costoso, pero el mantenimiento y el coste marginal de su uso repetitivo es relativamente bajo. Por otro lado, la ganancia en términos monetarios, tiempo y precisión resultantes del uso de los



sistemas expertos son muy altas, y la amortización de la inversión es muy rápida. Sin embargo, antes de desarrollar o comprar un sistema experto debe realizarse un análisis coste-beneficio.

Hay varias ventajas en el uso de los Sistemas Expertos. Las más importantes son:

- Con el uso de sistemas expertos, el personal con poca experiencia puede resolver problemas que requieren los conocimientos de un experto. Esto también es importante en el caso de que existan pocos expertos humanos en determinada área. Además, el número de personas con acceso al conocimiento aumenta con el uso de los sistemas expertos.
- El Conocimiento de varios expertos humanos pueden combinarse, lo que conlleva a sistemas expertos más fiables, ya que se obtiene un Sistema Experto que combina la sabiduría colectiva de varios expertos humanos en lugar del conocimiento de uno solo.
- Los sistemas expertos pueden responder a preguntas y resolver problemas mucho más rápidamente que un experto humano. Por ello, los sistemas son muy valiosos en casos en los que el tiempo de respuesta es crítico.
- En algunos casos, la complejidad del problema impide al experto humano resolverlo. En otros casos la solución propuesta por los expertos humanos no es fiable. Debido a la capacidad de los ordenadores de procesar un elevadísimo número de operaciones complejas de forma rápida y aproximada, los sistemas expertos suministran respuestas rápidas y fiables en situaciones en la que los expertos humanos no pueden.
- Los sistemas expertos pueden ser utilizados para realizar operaciones peligrosas, monótonas, aburridas e incómodas para los humanos, los sistemas expertos pueden ser la única solución viable en una situación en la que la tarea a realizar desborda al ser humano (por ejemplo, un avión o una cápsula espacial dirigida por un sistema experto)
- A diferencia de un experto humano, un Sistema Experto no envejece, y por tanto no sufre pérdida de facultades con el paso del tiempo, además los sistemas expertos no se ven afectados por condiciones externas, un experto humano sí (cansancio, presión, etc.).

El uso de los Sistemas Expertos se recomienda especialmente en las siguientes situaciones:

- Cuando el conocimiento es difícil de adquirir o se basa en reglas que solo pueden ser aprendidas de la experiencia.



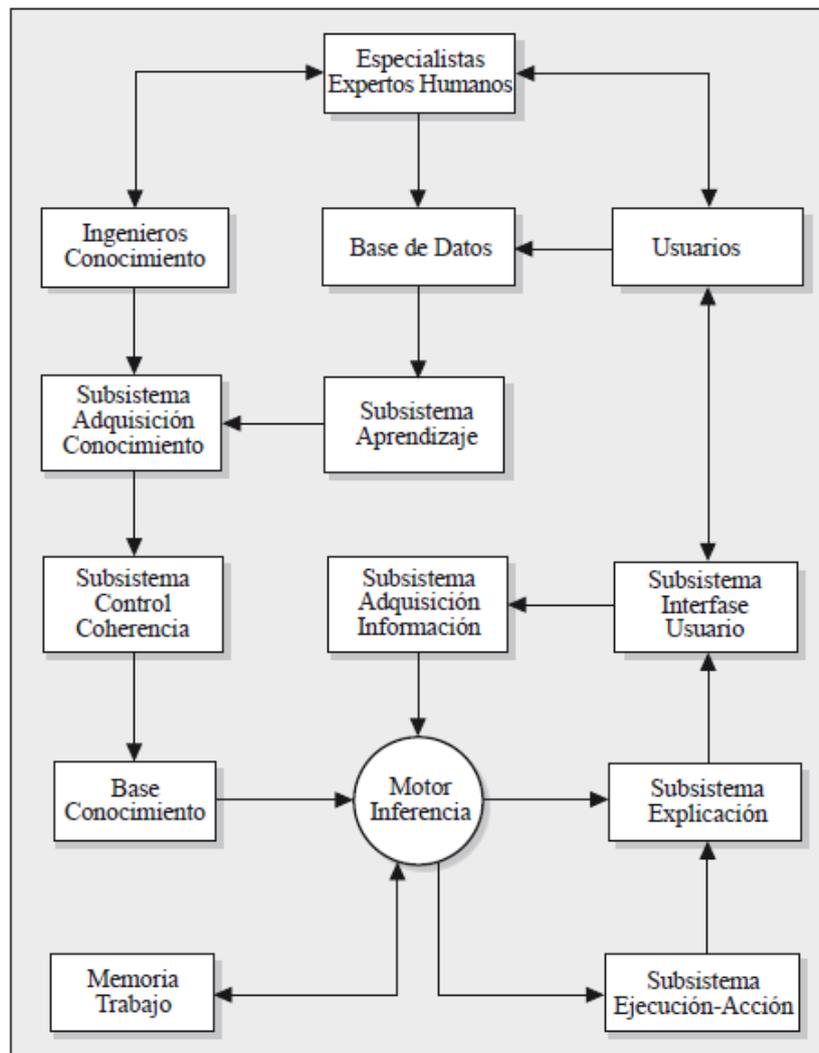
- Cuando la mejora continua del conocimiento es esencial y/o cuando el problema está sujeto a reglas o códigos cambiantes.
- Cuando los expertos humanos son caros o difíciles de encontrar.
- Cuando el conocimiento de los usuarios sobre el tema es limitado.

### **Limitaciones de los Sistemas Expertos (Desventajas)**

- **Sentido común:** Para un Sistema Experto no hay nada obvio. Por ejemplo, un sistema experto sobre medicina podría admitir que un hombre lleva 40 meses de embarazo, a no ser que se especifique que esto no es posible ya que un hombre no puede procrear hijos.
- **Lenguaje natural:** Con un experto humano podemos mantener una conversación informal mientras que con un sistema experto no podremos.
- **Capacidad de aprendizaje:** Cualquier persona aprende con relativa facilidad de sus errores y de errores ajeno, que un Sistema experto realice esto es muy complicado.
- **Perspectiva global:** Un experto humano es capaz de distinguir cuales son las cuestiones relevantes de un problema y separarlas de cuestiones secundarias.
- **Capacidad Sensorial:** Un Sistema experto carece de sentidos.
- **Flexibilidad:** Un humano es sumamente flexible a la hora de aceptar datos para la resolución de un problema.
- **Conocimiento no estructurado:** Un sistema experto no es capaz de manejar conocimientos poco estructurados.

### 2.3.7 Componentes de un Sistema Experto

Un Sistema Experto, generalmente se encuentra conformado por los siguientes componentes:



**Figura 2.1:** Componentes típicos de un sistema experto. Las flechas representan el flujo de la información [1].

A continuación, tenemos la descripción de cada uno de estos componentes.



## La Componente Humana

Un Sistema Experto generalmente es el resultado de la colaboración de uno o varios expertos humanos especialistas en un determinado tema de estudio y los Ingenieros del Conocimiento, con los usuarios en mente. Aquí, los expertos humanos son los encargados de suministrar los conocimientos básicos sobre el tema de interés, y los ingenieros del Conocimiento son los encargados de trasladar este conocimiento a un lenguaje, que el Sistema Experto sea capaz de interpretar. El éxito de la interacción entre los Expertos e Ingenieros es determinante en el ciclo de desarrollo de un Sistema experto, por esta razón esta etapa requiere una enorme dedicación y esfuerzo debido a los diferentes lenguajes en que se expresan las diferentes partes y las diferentes experiencias que tienen.

## La Base de Conocimiento

Los expertos o especialistas son los encargados de suministrar a los ingenieros de conocimiento una Base de Conocimiento ordenada y estructurada, además de un conjunto de relaciones bien definidas y explicadas. Esta forma estructurada de pensar requiere que los expertos humanos repiensen, reorganicen y reestructuren la base de conocimiento y, como resultado el experto o especialista se transforma en un mejor conocedor de su propio campo de especialidad.

Es importante saber diferenciar entre Datos y Conocimiento, El conocimiento se refiere a afirmaciones de validez general tales como las reglas, distribuciones de probabilidad, etc. Cuando nos referimos a los datos, estamos hablando acerca de la información relacionada con una aplicación particular.

**Ejemplo:** En el Diagnóstico médico, los síntomas, las enfermedades y las relaciones entre ellos, forman parte del conocimiento, mientras que los síntomas particulares de un determinado paciente forman parte de los datos.

Mientras el conocimiento es permanente, los datos son efímeros, o sea no forman parte de la componente permanente de un sistema y son destruidos luego de ser utilizados.



El conocimiento es almacenado en la Base de Conocimientos y los Datos se almacenan en la memoria de Trabajo. Además todos los procedimientos de los diferentes Sistemas y subsistemas que son de carácter transitorio se almacenan en la memoria de trabajo.

### **Subsistema de Adquisición de Conocimiento**

Controla el flujo del nuevo conocimiento que fluye del experto humano a la base de datos. El Sistema determina qué nuevo conocimiento se necesita, o si el conocimiento recibido es en realidad nuevo, o sea, si este debe incluirse en la base de datos y, de ser necesario, incorpora estos conocimientos a la misma.

### **Control de Coherencia**

Este subsistema ha aparecido recientemente en los sistemas expertos. Sin embargo es un componente esencial de los mismos. Este subsistema controla la consistencia de la base de datos y evita que unidades de conocimiento inconsistentes entren en la misma. En ciertas situaciones complejas, incluso un experto humano puede formular afirmaciones inconsistentes. Por esta razón, si un subsistema de control de la coherencia, unidades de conocimiento contradictorio pueden formar parte de la base de conocimiento, dando lugar a un comportamiento insatisfactorio del sistema. Es también bastante común, especialmente en sistemas con mecanismos de propagación de incertidumbre, que lleguemos a conclusiones absurdas o en conflicto como, por ejemplo, situaciones en las que el sistema genera probabilidades mayores que la unidad o negativas. Por ello, el subsistema de control de la coherencia comprueba e informa a los expertos de las inconsistencias. Por otra parte, cuando se solicita información de los expertos humanos, el subsistema de control de coherencia informa sobre las restricciones que ésta debe cumplir para ser coherente con la existente en la base de conocimiento, ayudando a los expertos a dar información fiable.



## **Motor de Inferencia**

Es el corazón del sistema experto, su objetivo principal es el sacar conclusiones aplicando el conocimiento a los datos. Por ejemplo, en diagnóstico médico, los síntomas de un paciente (datos) son analizados a la luz de los síntomas y las enfermedades y sus relaciones (conocimiento). Las conclusiones del Motor de Inferencia pueden estar basadas en Conocimiento Determinista o Conocimiento Probabilístico.

Como es lógico, el tratamiento de situaciones de incertidumbre (probabilísticas) puede ser considerablemente más difícil que el tratamiento de situaciones ciertas (deterministas). En muchos casos, algunos hechos (casos) no se conocen con mucha certeza. Por ejemplo en un paciente que no está seguro de sus síntomas, puede darse el caso que tengamos que trabajar con conocimiento de tipo no determinista, o sea de casos en lo que solo se dispone de información aleatoria o difusa. El motor de inferencia es también responsable de la propagación de este conocimiento incierto. En los Sistemas Expertos basados en probabilidad, la propagación de incertidumbre es la tarea principal del motor de inferencia, este nos permite sacar conclusiones bajo incertidumbre. Esta tarea tiene tal complejidad que se considera la componente más crítica y a la vez la más débil de casi todos los sistemas expertos existentes.

## **El Subsistema de Adquisición de Conocimiento**

Cuando el conocimiento inicial es muy limitado y no se pueden sacar conclusiones, el Motor de Inferencia utiliza el subsistema de adquisición de conocimiento para obtener el conocimiento necesario y continuar con el proceso de inferencia hasta que se hayan sacado conclusiones. En algunos casos, el usuario del sistema puede suministrar la información requerida para este y otros objetivos. Por esta razón es muy necesaria la inclusión en la estructura del sistema experto de una Interfaz de usuario y de una comprobación de la consistencia de la información suministrada por el usuario antes de ser introducida en la memoria de trabajo.



## La Interfaz de Usuario

Es el enlace entre el Sistema Experto y el Usuario, entonces para que el Sistema Experto sea una herramienta efectiva, debemos implementar mecanismos eficientes para la presentación y la captura de información (debe ser de manera fácil y agradable al usuario). Un ejemplo de la información que tiene que ser mostrada tras el trabajo del motor de inferencia, es de las conclusiones, las razones que expliquen tales conclusiones y una explicación de las acciones efectuadas por el sistema experto. Por otro lado, cuando el Motor de Inferencia no puede concluir sus operaciones debido a errores o ausencia de información, la Interfaz de Usuario es el vehículo para obtener la información necesaria del usuario. Consecuentemente, cuando no se implementa adecuadamente este componente, disminuye notablemente la calidad de un Sistema Experto. Otra razón de la importancia de la interfaz de usuario es que los usuarios evalúan comúnmente los Sistemas Expertos y otros sistemas por la calidad de dicha Interfaz, más que por la efectividad del Sistema experto mismo, aunque no se debería juzgar la calidad de un Sistema experto por su interfaz.

## El Subsistema de Ejecución de Órdenes

Es la componente que permite al Sistema experto iniciar acciones. Estas acciones se basan en las conclusiones sacadas por el motor de inferencia. Como ejemplo podemos pensar en un Sistema Experto diseñado para analizar el tráfico ferroviario que puede decidir retrasar o parar ciertos trenes para optimizar el tráfico global, o un sistema para controlar una central nuclear que puede abrir o cerrar ciertas válvulas, mover barras, etc., para evitar accidentes. La explicación de las razones por las que se inician estas acciones puede darse al usuario mediante el Subsistema de Explicación.

## El Subsistema de Explicación

El usuario puede pedir una explicación de las conclusiones sacadas o de la acciones iniciadas por un determinado Sistema Experto. Debido a esto, es necesario un subsistema



que explique el proceso seguido por el motor de inferencia o por el subsistema de ejecución. Por ejemplo, si un cajero automático decide rechazar la palabra clave (una acción realizada por el usuario), la máquina puede mostrar un mensaje (una explicación) como la siguiente:

“Lo Sentimos, su palabra clave es todavía incorrecta tras tres intentos. Su tarjeta será retenida, para garantizar su seguridad, por favor póngase en contacto con su banco en horas de oficina.”

En muchas aplicaciones, se necesita la explicación de las conclusiones debido a los riesgos asociados con las acciones a ejecutar. Por ejemplo en un Sistema de Diagnóstico Médico, los médicos son los responsables del diagnóstico, independientemente de las herramientas técnicas utilizadas para sacar conclusiones. En estas situaciones, al carecer de un sistema de explicación, los doctores podrían no tener la capacidad de explicar a sus pacientes las razones de su diagnóstico.

### **El Subsistema de Aprendizaje**

Una de las principales características de un Sistema Experto es su capacidad para aprender. Diferenciaremos entre Aprendizaje Estructural y Aprendizaje Paramétrico.

El Aprendizaje Estructural se refiere a algunos aspectos relacionados con la estructura del conocimiento (reglas, distribuciones de probabilidad, etc.). Por esta razón, el descubrimiento de nuevos síntomas relevantes para una enfermedad o la inclusión de una nueva regla en la base de conocimiento son ejemplos de Aprendizaje estructural.

El Aprendizaje Paramétrico se refiere a la estimación de los parámetros necesarios para construir la base de conocimiento. Por ello, la estimación de frecuencias o probabilidades asociadas a síntomas o enfermedades es un ejemplo de aprendizaje paramétrico.



### 2.3.8 Ciclo de Desarrollo de un Sistema Experto

Generalmente, las etapas de Diseño e Implementación de un sistema experto comprenden las siguientes fases:

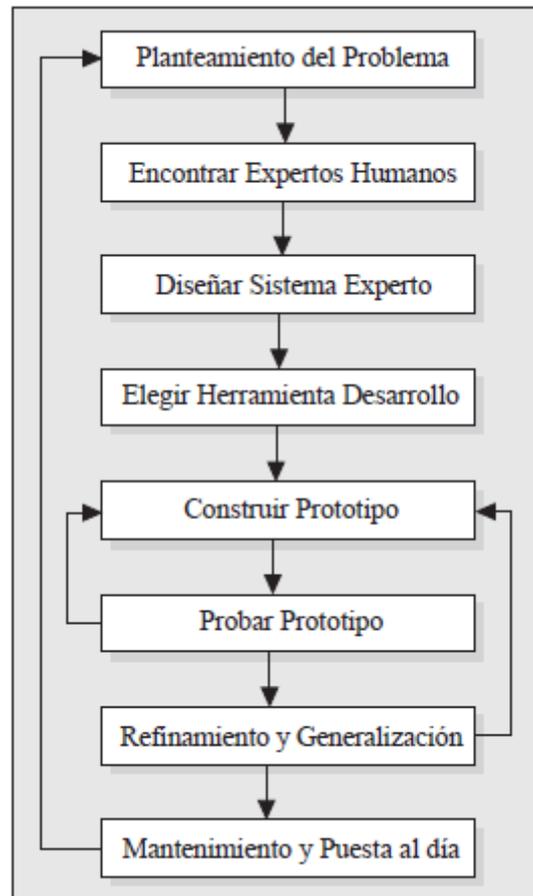


Figura 2.2: Etapas en el Desarrollo de un Sistema Experto

A continuación daremos una explicación de cada una de la Etapas que comprenden el ciclo de desarrollo de un Sistema Experto.



## **Planteamiento del Problema**

La primera etapa de cualquier proyecto de software es normalmente la definición del problema a resolver. Puesto que el objetivo principal de un sistema experto es responder a preguntas y resolver problemas, esta etapa del ciclo de desarrollo es quizás la más importante en el desarrollo de un sistema experto. Si el sistema está mal definido, se espera que el sistema suministre resultados incorrectos o incoherentes.

## **Encontrar Expertos Humanos que puedan resolver el problema**

En algunos casos, sin embargo, las bases de datos pueden jugar el papel del experto humano.

## **Diseño de un Sistema Experto**

Esta etapa incluye el diseño de estructuras para almacenar el Conocimiento, el Motor de Inferencia, el Subsistema de Explicación, la Interfaz de usuario, etc.

## **Elección de la Herramienta de Desarrollo, Shell, o Lenguaje de Programación**

Debemos decidir si realizaremos un Sistema Experto a medida, o usar un Shell, una Herramienta, o un lenguaje de programación. Si existiera un Shell que satisfaga todos los requerimientos de diseño, ésta debería ser la elección, no solo por razones de tipo financiero sino por razones de fiabilidad. Los Shell y Herramientas comerciales están sujetas a rigurosos controles de calidad, a los que otros programas no lo están.

## **Desarrollo y Prueba de un prototipo**

Si el prototipo no pasa las pruebas requeridas, las etapas anteriores (con las modificaciones apropiadas) deben ser repetidas hasta que se obtenga un prototipo que satisfaga correctamente los requerimientos.



## **Refinamiento y Generalización**

Es esta etapa se corrigen los fallos y se incluyen nuevas funcionalidades no incorporadas en el diseño inicial.

## **Mantenimiento y puesta al día**

En esta etapa, el usuario plantea problemas o defectos del prototipo, corrige errores, actualiza el producto con nuevos avances, etc.

Todas estas etapas influyen en la calidad del Sistema experto Resultante, que siempre debe ser evaluado en función de las aportaciones de los usuarios.

## **2.4 Sistemas Expertos Basados en Reglas**

### **2.4.1 Introducción**

En nuestras actividades cotidianas podemos encontrar muchas situaciones complejas gobernadas por reglas deterministas, por ejemplo: Sistemas de Control de Tráfico, Sistemas de Seguridad, Transacciones Bancarias, etc. Los Sistemas Basados en Reglas son una Herramienta eficiente para tratar estos problemas. Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en Sistemas Expertos. La Base de conocimiento contiene el conjunto de reglas que definen el problema, y el Motor de inferencia saca las conclusiones aplicando la lógica clásica a estas reglas.

En el transcurso de este capítulo describiremos la Base de Conocimientos de los Sistemas Expertos basados en Reglas y daremos una definición y ejemplos de reglas, que constituyen el corazón de la Base de Conocimientos. Seguidamente, se describe cómo opera el Motor de Inferencia, como trabaja el Subsistema de Control de Coherencia, y cómo se explican las conclusiones sacadas por el Motor de inferencia. Además veremos un ejemplo de aplicación y finalmente analizaremos algunas de las limitaciones de los Sistemas expertos basados en Reglas



## 2.4.2 La Base de Conocimiento

En los Sistemas Expertos Basados en Reglas intervienen dos elementos importantes:

- La Base de Conocimientos
- Los Datos

Los Datos están formados por la evidencia o los hechos conocidos en una situación particular. Este elemento es dinámico, es decir puede cambiar de una aplicación a otra. Por esta razón, no es de naturaleza permanente y se almacena en la Memoria de Trabajo

En situaciones de tipo Deterministas, las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas. El Conocimiento se almacena en la Base de Conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos.

La información almacenada en la base de Conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje.

Para darnos una mejor idea de lo que es una regla, supongamos que tenemos un conjunto de objetos y, por simplicidad, que cada objeto puede tener uno y sólo uno de un conjunto de valores posibles. Como lo ilustramos en la siguiente tabla de ejemplo.

Objeto	Conjunto de valores posibles
Nota	{0,1,...,10}
Calificación	{suspenso, notable, aprobado, suspenso}
Puesto	{0,1,...,100}
Admitir	{si, pendiente, no}
Notificar	{si, no}

**Tabla 2.1:** Tabla de ejemplo de objetos con sus posibles valores.

Seguidamente formularemos unos ejemplos de reglas aplicadas a los objetos.



**Regla1:** Si nota > 9, entonces calificación = sobresaliente.

**Regla2:** Si puesto < 20 o nota > 7, entonces Admitir = sí y Notificar = sí.

Cada una de las reglas anteriores relaciona dos o más objetos y está formada por las partes siguientes:

- **La premisa de la regla:** Es la expresión lógica entre las palabras clave “**si**” y “**entonces**”, la premisa puede contener una o más afirmaciones **objeto-valor** conectadas con operadores lógicos “**y**”, “**o**”, ó “**no**”. Por ejemplo:
  - La premisa de la regla 1 consta de una única afirmación objeto-valor, mientras que las premisas de la Regla 2 constan de dos afirmaciones objeto-valor conectadas por un operador lógico.
- **La conclusión de la Regla:** Es la expresión lógica tras la palabra clave “**entonces**”.

El ejemplo anterior nos facilita dar la siguiente definición de Regla.

**Regla:** “Una Regla es una afirmación lógica que relaciona dos o más objetos e incluye dos partes, la Premisa y la Conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones **objeto-valor** conectadas mediante los operadores lógicos **y**, **o**, ó **no**.”

Una regla se describe normalmente como “*Si premisa, entonces conclusión*”. En general, ambas, la premisa y la conclusión de una regla, pueden contener afirmaciones múltiples objeto-valor. Una expresión lógica que contiene sólo una afirmación objeto-valor se denomina “**Expresión Lógica Simple**”, por otro lado se la denomina “**Expresión Lógica Compuesta**”. Por ejemplo, las expresiones lógicas en ambas, premisa y conclusión de la Regla2 es compuesta. La Regla1 es simple.

### **Ejemplo 1: Cajero Automático.**

Como un ejemplo de problema determinista que puede ser formulado usando un conjunto de reglas, consideremos una situación en la que el usuario (por ejemplo, un cliente) desea sacar dinero de su cuenta corriente mediante un cajero automático (CA). En el momento en que el usuario introduce la tarjeta en el CA, la máquina la lee y la verifica. Si



la tarjeta no es verificada con éxito (por ejemplo cuando no es legible), el CA devuelve la tarjeta al usuario con el mensaje de error correspondiente. En otro caso, el CA pide al usuario su número de identificación personal (NIP). Si el número es incorrecto, se dan tres oportunidades de corregirlo. Si el NIP es correcto, el CA pregunta al usuario cuánto dinero desea sacar. Para que el pago se autorice, la cantidad solicitada no debe exceder de una cierta cantidad límite diaria, además de haber suficiente dinero en su cuenta.

En este caso se tienen siete objetos, y cada objeto puede tomar uno y sólo un valor de entre sus posibles valores.

La siguiente tabla muestra estos objetos y sus posibles valores.

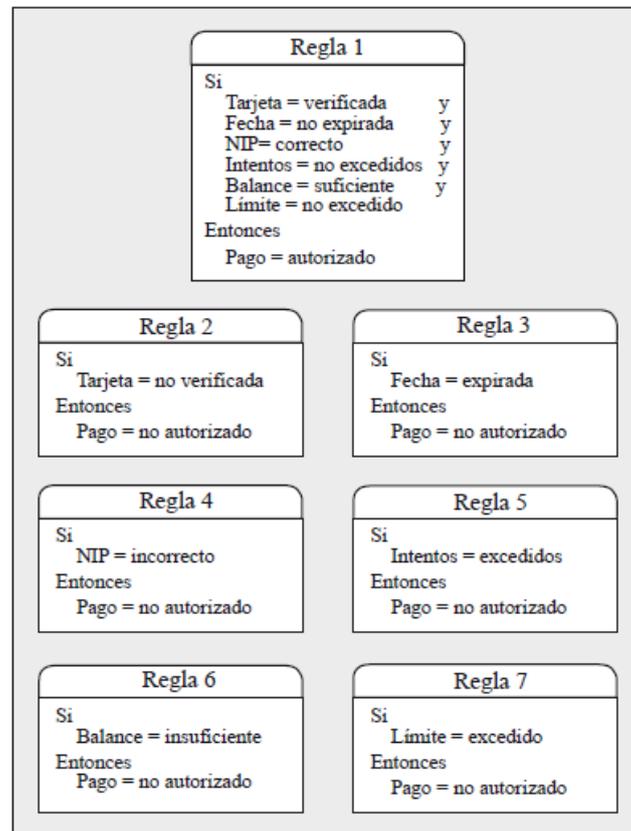
Objeto	Conjunto de valores posibles
Tarjeta	{verificada, no verificada}
Fecha	{expirada, no expirada}
NIP	{correcto, incorrecto}
Intentos	{excedidos, no excedidos}
Balance	{suficiente, insuficiente}
Límite	{excedido, no excedido}
Pago	{autorizado, no autorizado}

**Tabla 2.2:** Objetos y posibles Valores para el ejemplo de Cajero Automático

La siguiente figura muestra siete reglas que gobiernan la estrategia que el CA debe seguir cuando un usuario intenta sacar dinero de su cuenta. En la Regla1, por ejemplo, la premisa consiste en seis afirmaciones objeto-valor conectadas mediante un operador lógico “y” lo que indica que la premisa es cierta si las seis afirmaciones lo son. Por ello, la Regla 1 relaciona el objeto Pago (en la conclusión) con los demás objetos. Según la Regla 1, la acción que debe iniciar el CA es dar el dinero al usuarios si la tarjeta se ha verificado correctamente, la fecha no ha expirado, el NIP es correcto, el número de intentos para dar el NIP correcto no se ha excedido y la cantidad solicitada no excede ni la cantidad disponible



ni el límite máximo diario. Las expresiones lógicas en cada una de las reglas restantes de la Figura constan de una sola afirmación. Debemos tener en cuenta que la Regla 1 indica cuándo debe permitirse el pago, y las restantes cuándo debe rechazarse.



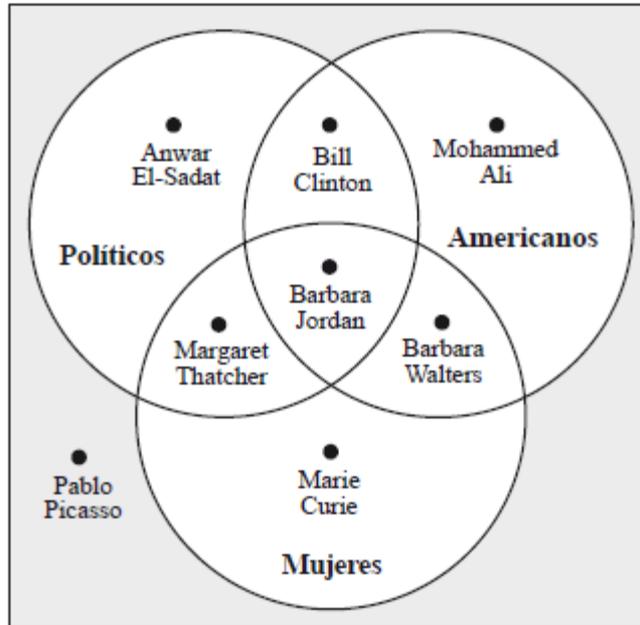
**Figura 2.4:** Ejemplos de Reglas para sacar dinero de un cajero automático

### Ejemplo 2: Gente Famosa

Para ilustrar este ejemplo suponemos que tenemos una Base de datos consistente en  $N$  individuos. Para cada individuo, la base de datos contiene cuatro atributos: nombre, sexo, nacionalidad y profesión. Supóngase que la base de datos muestra sólo si una persona es americana, política y/o si es mujer. Cada uno de estos atributos es binario (puede tomar solo dos valores posibles). En este caso, la base de datos puede contener, como mucho,  $2^3 = 8$  conjuntos disjuntos. Estos conjuntos se muestran en la Figura 2.2. En esta figura se muestra también el nombre de una persona en cada subconjunto. La Tabla 2.3 da un ejemplo de una base de datos que contiene  $N=8$  personas famosas. En este caso se tienen



cuatro objetos: Nombre, Americano, Político y Mujer. El primer objeto puede tomar uno de N posibles valores (los nombres de cada persona) y cada uno de los tres últimos objetos puede tomar el valor de “sí” o el valor “no”.



**Figura 2.2:** Un ejemplo de una base de datos con tres atributos binarios que dividen la población en ocho conjuntos disjuntos.

Partiendo de la Tabla 2.3 se pueden construir reglas para identificar a cada persona, resultando un total de ocho reglas.

Nombre	Americano	Político	Mujer
Barbara Jordan	sí	sí	sí
Bill Clinton	sí	sí	no
Barbara Walters	sí	no	sí
Mohammed Ali	sí	no	no
Margaret Thatcher	no	sí	sí
Anwar El-Sadat	no	sí	no



Marie Curie	no	no	sí
Pablo Picasso	no	no	no

**Tabla 2.3:** Una Base de Datos mostrando cuatro objetos y sus valores correspondientes para el ejemplo de las personas famosas.

Por ejemplo, la regla siguiente corresponde al presidente Clinton:

Regla 1: **Si** Nombre = Clinton, **entonces** Americano = sí **y** Político = si **y** Mujer = no.

Las Restantes siete reglas pueden formularse de forma análoga.

Los dos ejemplos ilustrados anteriormente serán utilizados a lo largo de este capítulo con el objetivo de ilustrar varios conceptos relacionados con los sistemas expertos basados en reglas.

Algunos sistemas imponen ciertas restricciones a las reglas. Por ejemplo:

- No permitir en la premisa el operador lógico *o*, *y*
- Limitar las conclusiones a expresiones lógicas simples.

Hay buenas razones para imponer estas restricciones. En primer lugar, las reglas que satisfacen estas restricciones son fáciles de tratar a la hora de escribir un programa software. En segundo lugar, las dos restricciones anteriores no dan lugar a una pérdida de generalidad, puesto que reglas mucho más generales pueden ser reemplazadas por conjuntos de reglas de esta forma. Esto es llamado como “**Sustitución de Reglas**”. Por tanto, el conjunto de reglas especificado inicialmente por el experto humano puede requerir una sustitución posterior por un conjunto de reglas equivalente para satisfacer estas restricciones.

En la Tabla 2.4 tenemos ejemplos de sustitución de reglas. Debemos notar que cada una de las reglas de la primera columna puede ser sustituida por el correspondiente



conjunto de reglas de la segunda columna y que todas las reglas de ésta satisfacen las condiciones anteriores. Por ejemplo:

Regla	Reglas Equivalentes
Si A o B, entonces C	Si A, entonces C Si B, entonces C
Si $\overline{A \circ B}$ , entonces C	Si $\overline{A}$ y $\overline{B}$ , entonces C
Si $\overline{A} \vee \overline{B}$ , entonces C	Si $\overline{A}$ , entonces C Si $\overline{B}$ , entonces C
Si (A o B) y C, entonces D	Si A y C, entonces D Si B y C, entonces D
Si $\overline{(A \circ B)}$ y C, entonces D	Si $\overline{A}$ y B y C, entonces D
Si $\overline{A} \vee \overline{B}$ y C, entonces D	Si $\overline{A}$ y C, entonces D Si $\overline{B}$ y C, entonces D
Si A, entonces B y C	Si A, entonces B Si A, entonces C
Si A, entonces B o C	Si A y $\overline{B}$ , entonces C Si A y $\overline{C}$ , entonces B
Si A, entonces $\overline{B} \vee \overline{C}$	Si A y B, entonces $\overline{C}$ Si A y C, entonces $\overline{B}$
Si A, entonces $\overline{B \circ C}$	Si A, entonces $\overline{B}$ Si A, entonces $\overline{C}$

**Tabla 2.4**

La primera regla compuesta de la Tabla 2.4:

- Regla 1: Si A o B, entonces C,

Puede ser reemplazada por las dos reglas simples:

- Regla 1a: Si A, entonces C.
- Regla 1b: Si B, entonces C.



### 2.4.3 El Motor de Inferencia

Como lo hemos indicado anteriormente, hay dos tipos de elementos: los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento).

El Motor de inferencia usa ambos para obtener nuevas conclusiones o hechos. Por Ejemplo si la premisa de una regla es cierta, entonces la conclusión de la regla debe ser también cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones. Por ello, los hechos iniciales o datos de partida y las conclusiones derivadas de ellos forman parte de los hechos o datos que se dispone en un momento dado.

Las conclusiones pueden clasificarse en dos tipos: simples y compuestas.

Las conclusiones Simples son las resultan de una regla simple. Las conclusiones compuestas son las que resultan de más de una regla. Con la finalidad de obtener conclusiones, los expertos usan diferentes tipos de reglas y estrategias de inferencia y control que son usadas por el Motor de Inferencia para obtener conclusiones simples y compuestas.

Entre las principales Reglas de inferencia tenemos:

- Modus Ponens
- Modus Tollens
- Resolución

Entre las Estrategias de Inferencia tenemos:

- Encadenamiento de Reglas
- Encaminamiento de Reglas orientado a un objetivo
- Compilación de Reglas

Las dos primeras reglas de inferencia son usadas para obtener conclusiones simples y el resto de reglas y estrategias se usan para obtener conclusiones compuestas. Sin embargo debemos tener en cuenta, que ninguna de las estrategias mencionadas anteriormente, si



son implementadas solas, conduce a todas las soluciones posibles. Por ello, deben implementarse varias reglas y estrategias en el desarrollo del sistema experto para que el Motor de Inferencia sea capaz de obtener tantas conclusiones como sea posible.

### Modus Ponens y Modus Tollens

El Modus Ponens es talves la regla de inferencia que más se usa comúnmente, es usada para obtener conclusiones simples. En esta se examina la premisa de la regla, y si es cierta, la conclusión pasa a ser parte del conocimiento.

Por ejemplo, si tenemos la siguiente regla “Si A es cierto, entonces B es cierto” y que sabemos además que “A es cierto.” Entonces , tal como se muestra en la figura 2.3, el resultado o conclusión de la regla Modus Ponens es que “B es cierto.” Esta regla de inferencia, que parece trivial, debido a que nos es muy familiar, es la base de un gran número de Sistemas Expertos.

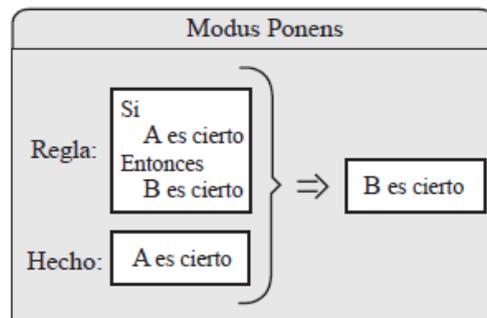


Figura 2.3: Ilustración de la regla de inferencia Modus Ponens

La regla de Inferencia Modus Tollens se utiliza también para la obtención de conclusiones simples. En este caso se examina la conclusión y si esta es falsa, se concluye que la premisa también es falsa.

Por ejemplo, supongamos nuevamente que tenemos la regla “Si A es cierto, entonces B es cierto”, pero se sabe que “B es falso”. Entonces usando la regla Modus Ponens, no se puede obtener ninguna conclusión, pero tal como lo ilustramos en la figura



2.4, la regla Modus Tollens llega a la conclusión que “A es falso”. Aunque es muy simple y con muchas aplicaciones útiles, esta regla es menos usada que Modus Ponens.

Por ello, la regla Modus Ponens se mueve hacia adelante, o sea, de la premisa a la conclusión de una regla, mientras que la regla Modus Tollens se mueve hacia atrás, o sea desde la conclusión hasta la premisa. Estas dos reglas de inferencia no deben ser vistas como alternativas sino como complementarias.

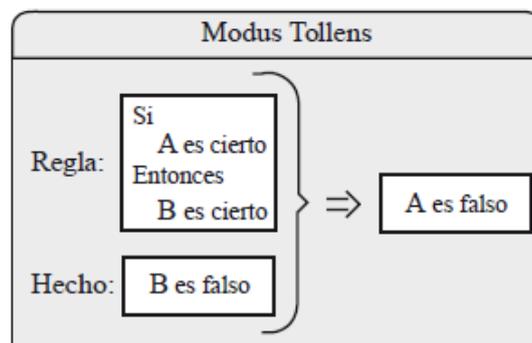


Figura 2.4 Ilustración de la regla Modus Tollens

La regla Modus Ponens necesita información de los objetos de la premisa para concluir, mientras que la regla Modus Tollens necesita información sobre los objetos de la conclusión. De hecho, para un Motor de Inferencia que solamente usa Modus Ponens, la incorporación de la regla de inferencia Modus Tollens puede ser considerada como una expansión de la base de Conocimientos mediante la adición de reglas, tal como lo podemos comprobar en el siguiente ejemplo:

### **Ejemplo 2.3 La Regla Modus Tollens equivale a una expansión de la Base de Conocimientos**

Supongamos que la Base de Conocimientos consiste solamente en la regla 1, mostrada en la Figura 2.5. Podemos usar la regla de inferencia Modus Tollens para “invertir” la Regla 1 y obtener alguna conclusión cuando se tiene información sobre los objetos de su conclusión.



Regla 1	
Si	
Tarjeta = verificada	y
Fecha = no expirada	y
NIP = correcto	y
Intentos = no excedidos	y
Balance = suficiente	y
Límite = no excedido	
Entonces	
Pago = autorizado	

Figura 2.5 Regla 1

Entonces, aplicar la regla Modus Tollens a la regla “Si A, entonces B” es equivalente a aplicar la regla Modus Ponens a la regla “Si  $\bar{B}$ , entonces  $\bar{A}$ ”. En este caso de Regla 1, usando la equivalencia

$$\bar{A} = C \text{ y } B = C \Leftrightarrow \bar{A} = F \text{ o } B = F,$$

Obtenemos la regla 1b, mostrada en la figura 2.6. Por ello, utilizar ambas, las reglas Modus Ponens y Modus Tollens cuando la base de conocimiento contiene sólo la regla 1, es equivalente a usar la regla Modus Ponens cuando la Base de Conocimiento contiene ambas, la Regla 1 y la Regla 1b.

Regla 1b	
Si	
Pago = no autorizado	
Entonces	
Tarjeta = no verificada	o
Fecha = expirada	o
NIP = incorrecto	o
Intentos = excedidos	o
Balance = insuficiente	o
Límite = excedido	

Figura 2.6 La Regla 1b puede obtenerse de la Regla 1 usando la regla de inferencia Modus Tollens.



Por otro lado, el rendimiento del motor de inferencia depende del conjunto de reglas en su base de conocimiento. Hay situaciones en las que el motor de inferencia puede concluir utilizando un conjunto de reglas, pero no puede, utilizando otro (aunque estos sean lógicamente equivalentes). A continuación se da un ejemplo ilustrativo.

#### **Ejemplo 2.4 Inferencia con dos conjuntos equivalentes de reglas**

Supongamos de nuevo que tenemos dos motores de inferencia: El Motor *E1*, cuya base de conocimiento contiene las siete reglas de la figura 2.1, y el motor *E2*, cuya base de conocimiento contiene las siete reglas de la figura 2.7. Debemos notar que los dos conjuntos de reglas son lógicamente equivalentes.

Supongamos además que se sabe que el valor del NIP es incorrecto. Si ambos *E1* y *E2* utilizan sólo la regla de inferencia Modus Ponens, entonces *E1* será capaz de concluir que **Pago = no autorizado** (por la regla 4), pero *E2* no concluirá. Por ello, algunas de las conclusiones lógicamente derivables pueden no ser obtenidas usando sólo la regla de inferencia Modus Ponens.

Por otra parte, si ambos motores usan la regla Modus Tollens, entonces ambos concluirán.

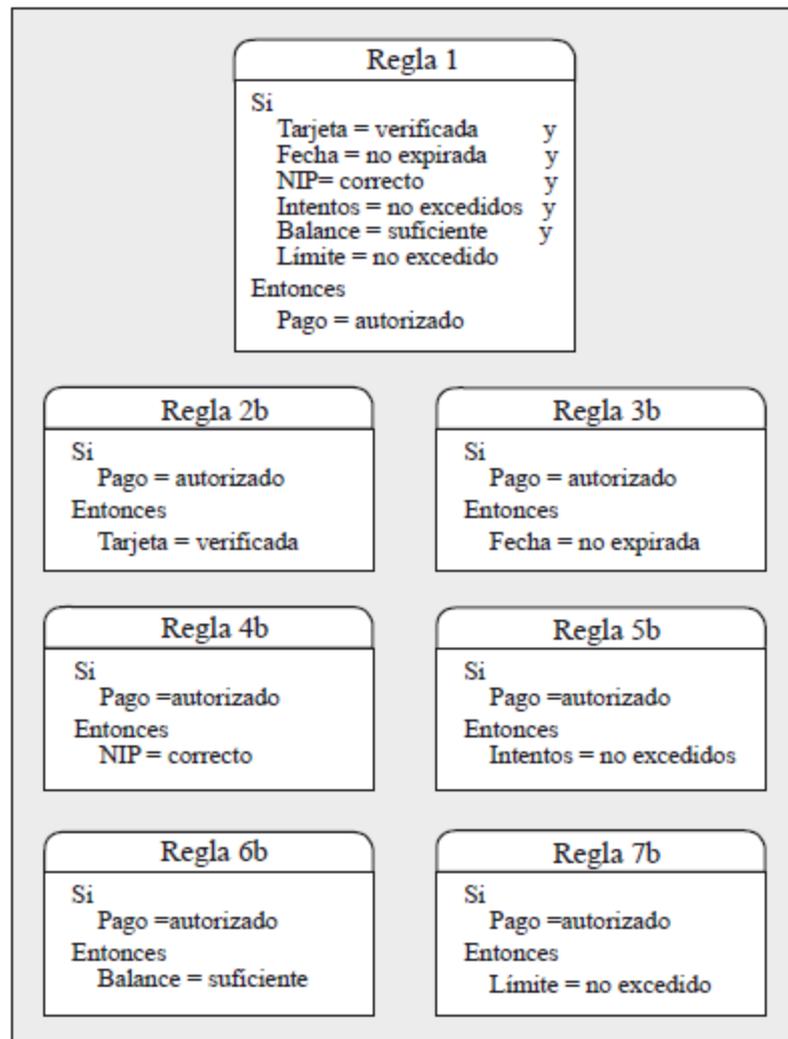


Figura 2.7. Conjunto de Reglas lógicamente equivalentes al conjunto de reglas de la figura 2.1

### El Mecanismo de Resolución

Las Reglas de Inferencia Modus Ponens y Modus Tollens pueden ser utilizadas para obtener conclusiones simples. Por otra parte, las conclusiones compuestas, que se basan en dos o más reglas, se obtienen usando el llamado Mecanismo de Resolución. Esta regla de Inferencia consiste en las siguientes etapas:

1. Las Reglas son sustituidas por expresiones lógicas equivalentes.
2. Estas expresiones lógicas se combinan en otra expresión lógica.
3. Esta última expresión se utiliza para obtener la Conclusión.



Estas etapas involucran conceptos tales como la combinación y simplificación de expresiones lógicas, que se ilustran de un modo intuitivo en los ejemplos que siguen.

### Ejemplo 2.5 Mecanismo de Resolución 1.

Supongamos que tenemos las dos reglas:

- Regla 1: Si A es cierto, entonces B es cierto.
- Regla 2: Si B es cierto, entonces C es cierto.

La primera etapa en el mecanismo de resolución consiste en sustituir cada una de las dos reglas por expresiones lógicas equivalentes. Esto lo hacemos de la siguiente manera: (véase la figura 2.8)

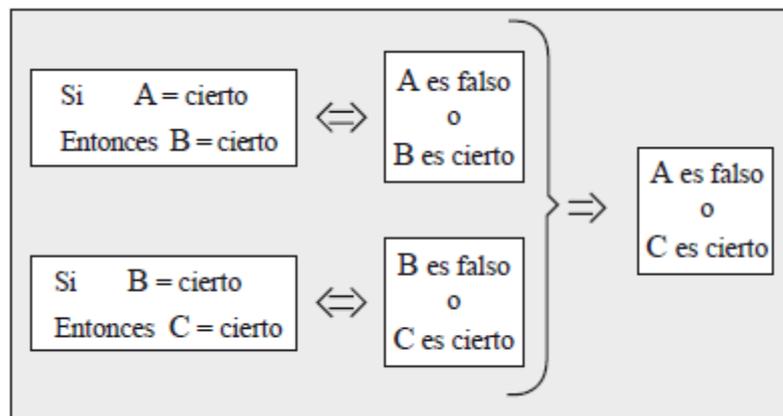


Figura 2.8. Un ejemplo ilustrando la regla de inferencia correspondiente al mecanismo de resolución.

- La regla 1 es equivalente a la expresión lógica “A es falso o B es cierto”. Una muestra de esta equivalencia se muestra en la tabla de verdad que se muestra en la tabla 2.6.
- Similarmente, la regla 2 es equivalente a la expresión lógica: “B es falso o C es cierto”

A	B	$\bar{A}$	Si A, entonces B	$\bar{A} \vee B$
C	C	F	C	C
C	F	F	F	F
F	C	C	C	C
F	F	C	C	C



Tabla 2.6. Tabla de verdad mostrando que la regla “Si A es cierto, entonces B es cierto” es equivalente a la expresión lógica “A es falso o B es cierto”

La segunda etapa consiste en combinar las dos expresiones anteriores en una, tal como sigue: Las expresiones lógicas “A es falso o B es cierto” y “B es falso o C es cierto” implican la expresión “A es falso o C es cierto”. Una prueba de esta equivalencia se muestra en la tabla 2.7. Esta última expresión se utiliza seguidamente en la tercera etapa para obtener la conclusión. Las etapas anteriores se ilustran en la Figura 2.8.

A	B	C	$\bar{A} \vee B$	$\bar{B} \vee C$	$(\bar{A} \vee B) \wedge (\bar{B} \vee C)$	$\bar{A} \vee C$
C	C	C	C	C	C	C
C	C	F	C	F	F	F
C	F	C	F	C	F	C
C	F	F	F	C	F	F
F	C	C	C	C	C	C
F	C	F	C	F	F	C
F	F	C	C	C	C	C
F	F	F	C	C	C	C

Tabla 2.7. Una Tabla de verdad que muestra que las expresiones lógicas “A es falso o B es cierto” y “B es falso o C es cierto” implican la expresión lógica “A es falso o C es cierto”

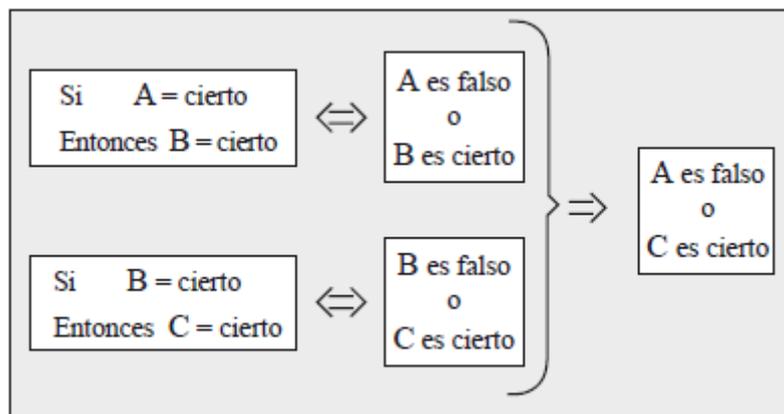




Figura 2.8 Un ejemplo ilustrando la regla de inferencia correspondiente al mecanismo de resolución.

### Ejemplo 2.6 Mecanismo de Resolución 2.

Consideremos nuevamente el ejemplo del Cajero Automático con el objeto añadido “Explicar”, que puede tomar los valores {si, no}, indicando si se necesita explicar las acciones del CA. Apliquemos ahora el mecanismo de resolución a la evidencia “NIP = incorrecto” y las reglas siguientes:

- Si NIP = incorrecto entonces Pago = no autorizado.
- Si Pago = no autorizado entonces Explicar = sí.

Como lo ilustramos en la figura 2.9, la regla de inferencia correspondiente al mecanismo de resolución conduce a la conclusión Explicar = sí.

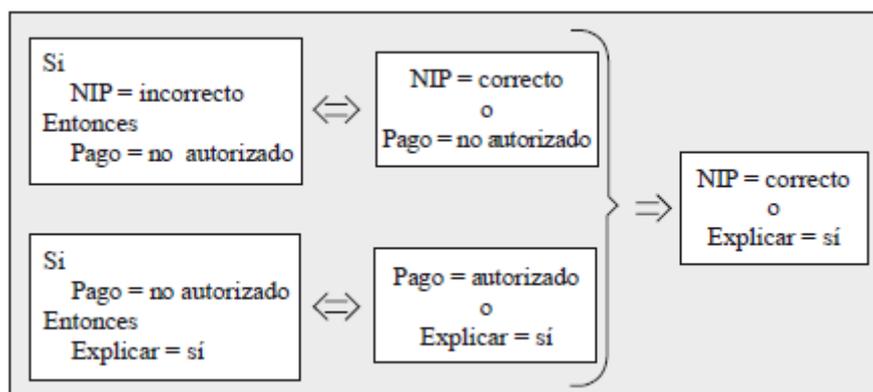


Figura 2.9 La regla de inferencia del mecanismo de resolución aplicada al ejemplo del CA.

En efecto, siguiendo los pasos indicados, tenemos:

1. Las dos reglas se sustituyen por las expresiones equivalentes:
  - NIP = correcto o Pago = no autorizado
  - Pago = autorizado o Explicar = sí
2. Las dos expresiones anteriores se combinan de la forma indicada para dar la expresión NIP = correcto o Explicar = sí.
3. Esta última expresión se combina con la evidencia NIP = incorrecto, y se obtiene la conclusión compuesta, explicar = Sí.



Es muy importante señalar que la regla de inferencia correspondiente al mecanismo de resolución no siempre conduce a conclusiones, pues, de hecho, puede no conocerse la verdad o falsedad de ciertas expresiones, si esto sucede, el sistema experto, o para ser más precisos, su motor de inferencia debe decidir entre:

- Abandonar la regla, dada la imposibilidad de obtener conclusiones, o
- Preguntar al usuario, mediante el subsistema de demanda de información, sobre la verdad o falsedad de una o varias expresiones para poder continuar el proceso de inferencia hasta que se obtenga una conclusión.

### Encadenamiento de Reglas

Una de las estrategias de inferencia más usada para la obtención de conclusiones compuestas es el llamado “Encadenamiento de Reglas”, podemos usar este método cuando las premisas de ciertas reglas coinciden con las conclusiones de otras. Cuando se encadenan las reglas, los hechos pueden utilizarse para dar lugar a nuevos hechos. Esto se repite sucesivamente hasta que no pueden obtenerse más conclusiones. El tiempo que utiliza este proceso hasta que se termina depende, tanto de los hechos conocidos, como de las reglas que se activan. La estrategia de encadenamiento de reglas se da en el algoritmo siguiente:

#### Algoritmo 2.1 Encadenamiento de Reglas.

- **Datos:** Una base de conocimiento (objetos y reglas) y algunos hechos iniciales.
  - **Resultado:** El conjunto de hechos derivados lógicamente de ellos.
1. Asignar a los objetos sus valores conocidos tales como los dan los hechos conocidos o la evidencia.
  2. Ejecutar cada regla de la base de conocimiento y concluir nuevos hechos si es posible.
  3. Repetir la Etapa 2 hasta que no puedan ser obtenidos nuevos hechos.

Este algoritmo puede ser implementado de muchas maneras. Una de ellas comienza con las reglas cuyas premisas tiene valores conocidos. Estas reglas deben concluir y sus conclusiones dan lugar a nuevos hechos. Estos nuevos hechos se añaden al conjunto de hechos conocidos, y el proceso continúa hasta que no pueden obtenerse nuevos hechos. Este proceso se ilustra, a continuación, aplicándolo en ejemplos:



### Ejemplo 2.7 Encadenamiento de Reglas 1

La figura 2.10 muestra un ejemplo de seis reglas que relacionan a 13 objetos, del A al M. Las relaciones entre estos objetos implicadas por las seis reglas puede ser representada gráficamente, tal como se muestra en la figura 2.11, donde cada objeto se representa por un nodo. Las Aristas representan la conexión entre los objetos de la premisa de la regla y el objeto de su conclusión. Debemos notar que las premisas de algunas reglas coinciden con las conclusiones de otras reglas. Por ejemplo, las conclusiones de las reglas 1 y 2 (los objetos C y G) son las premisas de la Regla 4.

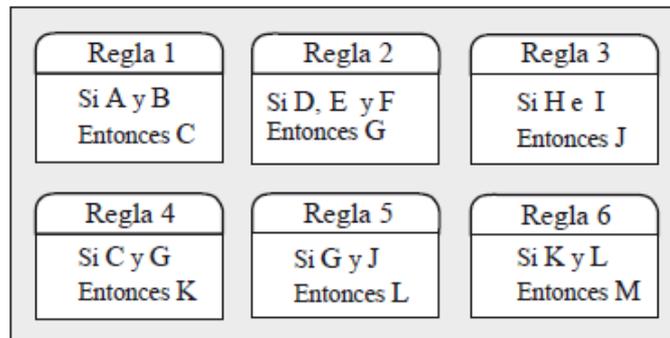


Figura 2.10 Un ejemplo de conjunto de seis reglas relacionando 13 objetos.

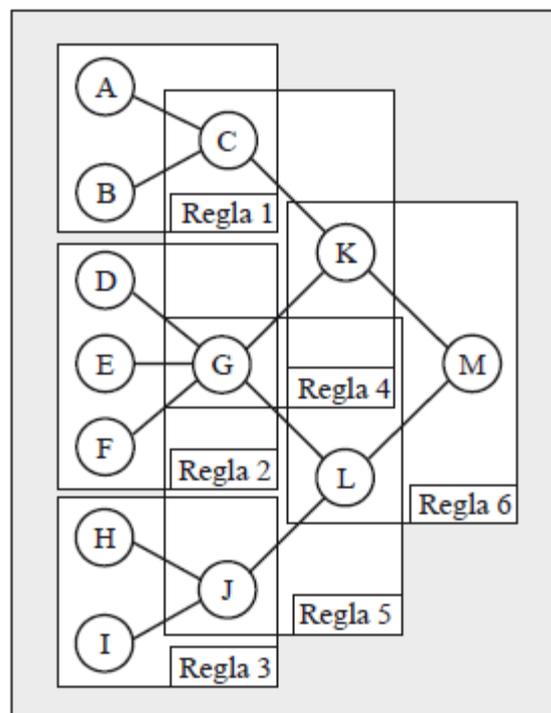




Figura 2.11 Una representación gráfica de las relaciones entre las seis reglas de la figura 2.10

Supóngase que sabemos que los objetos A, B, D, E, F, H e I son ciertos (verdaderos) y los restantes objetos son de valor desconocido. La figura 2.12 distingue entre objetos con valor conocido (los hechos) y objetos con valores desconocidos. En este caso, el algoritmo de encadenamiento de reglas procederá de la siguiente manera.

- La Regla 1 concluye que C = cierto
- La Regla 2 concluye que G = cierto
- La Regla 3 concluye que J = cierto
- La Regla 4 concluye que K = cierto
- La Regla 5 concluye que L = cierto
- La Regla 6 concluye que M = cierto

Puesto que no pueden obtenerse más conclusiones, el proceso se detiene. Este proceso se ilustra en la figura 2.12, donde los números en el interior de los nodos indican el orden en el que concluyen los hechos.

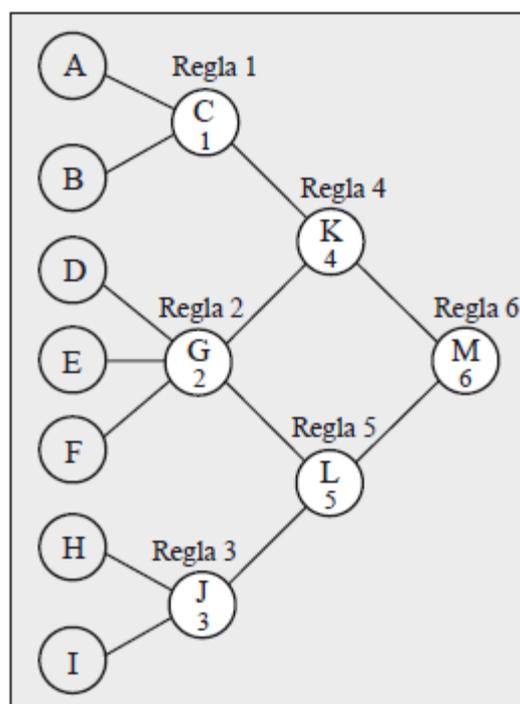


Figura 2.12 Un ejemplo que ilustra la estrategia de encadenamiento de reglas.



Los nodos con valores conocidos aparecen sombreados y los números en su interior indican el orden en el que concluyen los hechos.

### **Ejemplo 2.8 Encadenamiento de Reglas 2.**

Considérese de nuevo las seis reglas de la figura 2.10 y supóngase ahora que se dan los hechos  $H = \text{cierto}$ ,  $I = \text{cierto}$ ,  $K = \text{cierto}$  y  $M = \text{falso}$ . Esto lo ilustramos en la Figura 2.13, donde los objetos con valore conocidos (los hechos) aparecen como sombreados y la variable objetivo se muestra redondeada por una circunferencia. Supongamos, en primer lugar, que el motor de inferencia usa las dos reglas de Inferencia Modus Ponens y Modus Tollens. Entonces al aplicar el Algoritmo 2.1, obtenemos:

1. La Regla 3 concluye que  $J = \text{cierto}$  (Modus Ponens)
2. La Regla 6 concluye (Modus Tollens) que  $K = \text{falso}$  o  $L = \text{falso}$ , pero, puesto que  $K = \text{cierto}$ , deberá ser  $L = \text{falso}$ .
3. La Regla 5 concluye (Modus Tollens) que  $G = \text{falso}$  o  $J = \text{falso}$ , pero, puesto que  $J = \text{cierto}$ , deberá ser  $G = \text{falso}$ .

En consecuencia, se obtiene que la conclusión  $G = \text{falso}$ . Sin embargo, si el motor de inferencia sólo utiliza la regla de inferencia Modus Ponens, el algoritmo se detendrá en la Etapa 1, y no se concluirá nada para el objeto  $G$ . Este es otro ejemplo que ilustra la utilidad de la regla de inferencia Modus Tollens.

Nótese que la estrategia de encadenamiento de reglas diferencia claramente entre la memoria de trabajo y la base de conocimiento. La memoria de trabajo contiene datos que surgen mediante el periodo de consulta. Las premisas de la reglas se comparan con los contenidos de la memoria de trabajo y cuando se obtienen nuevas conclusiones son pasadas también a la base de conocimiento.

### **Encadenamiento de Reglas Orientado a un Objetivo**

El Algoritmo de encadenamiento de reglas orientado a un objetivo requiere del usuario seleccionar, en primer lugar, una variable o nodo objetivo; entonces el algoritmo navega a través de la reglas en búsqueda de una conclusión para el nodo objetivo. Si no obtenemos ninguna conclusión con la información existente, entonces el algoritmo fuerza a



preguntar al usuario en busca de nueva información sobre los elementos que son relevantes para obtener la información necesaria para llegar al objetivo.

Algunos autores llaman a los Algoritmos de Encadenamiento y de encadenamiento orientado a un objetivo “**Encadenamiento hacia adelante y encadenamiento hacia atrás**”, respectivamente. Pero esta terminología puede ser confusa, puesto que ambos algoritmos pueden, en realidad, utilizar las dos reglas de inferencia Modus Ponens (hacia adelante) y Modus Tollens (hacia atrás).

El Algoritmo de encadenamiento de reglas orientado a un objetivo es descrito a continuación.

### **Algoritmo 2.2 Encadenamiento de reglas Orientado a un Objetivo**

- **Datos:** Una base de conocimiento (objetos y reglas), algunos hechos iniciales, y un nodo variable objetivo.
  - **Resultado:** El Valor del nodo o variable objetivo.
1. Asigna a los Objetivos sus Valores conocidos tales como están dados en los hechos de partida, si es que existe alguno. Marcar todos los objetos cuyo valor ha sido asignado. Si el nodo objetivo está marcado, ir a la Etapa 7; en otro caso:
    - a. Designar como objetivo inicial el objetivo en curso.
    - b. Marcar el objetivo en curso.
    - c. Sea  $\text{ObjetivosPrevios} = \emptyset$ , donde  $\emptyset$  es el conjunto vacío.
    - d. Designar todas las reglas como activas (ejecutables)
    - e. Ir a la Etapa 2
  2. Encontrar una regla activa que incluya el objetivo en curso y ninguno de los objetos en  $\text{ObjetivoPrevios}$ . Si se encuentra una Regla, ir a la Etapa 3; en otro caso, ir a la Etapa 5.
  3. Ejecutar la regla referente al objetivo en curso. Si concluye, asignar el valor concluido al objetivo en curso, e ir a la Etapa 6; en otro caso, ir a la Etapa 4.
  4. Si todos los Objetos de la regla están marcados, declarar la regla como inactiva e ir a la Etapa 2; en otro caso:
    - a. Añadir el objetivo en curso a *ObjetivosPrevios*.
    - b. Designar uno de los objetos no marcados en la regla como el objetivo en curso.
    - c. Marcar el Objetivo en curso.
    - d. Ir a la Etapa 2.



5. Si el objetivo en curso es el mismo que el objetivo inicial, ir a la Etapa 7; en otro caso, preguntar al usuario por el valor del objetivo en curso. Si no se da un valor, ir a la Etapa 6; en otro caso asignar al objeto el valor dado e ir a la Etapa 6.
6. Si el objetivo en curso es el mismo que el objetivo inicial, ir a la Etapa 7; en otro caso, designar el objetivo previo como objetivo en curso, eliminarlo de *ObjetivoPrevios*, e ir a la Etapa 2.
7. Devolver el valor del objetivo en curso si es conocido.

A continuación ilustramos a través de un ejemplo el encadenamiento de reglas orientado a un objetivo.

### Ejemplo 2.9 Encadenamiento de Reglas orientado a un Objetivo.

Consideremos las seis reglas de las Figuras 2.10 y 2.11. Supongamos que seleccionemos el nodo M como nodo objetivo y que sabemos que los objetos D, E, F y L son ciertos. Estos nodos están sombreados en la figura 2.14. Las Etapas del Algoritmo de encadenamiento de reglas orientado a un objetivo se ilustran en la figura 2.14, donde el número en el interior de un nodo indica el orden en el que se visita cada nodo.

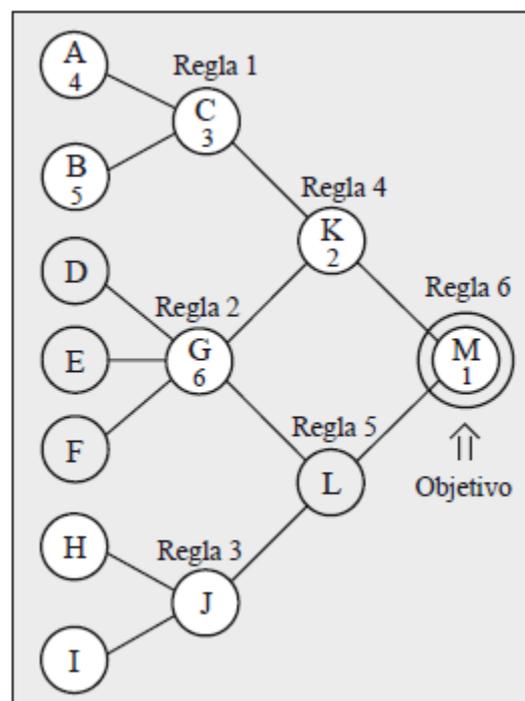


Figura 2.14. Un ejemplo que ilustra el algoritmo de encaminamiento de reglas orientado a un objetivo. Los nodos cuyo valor es conocido están sombreados, el nodo objetivo se ha



rodeado por una circunferencia, y el número en el interior de un nodo indica el orden en el que visita cada nodo.

Estas etapas son:

- **Etapas 1:** Se asigna el valor **cierto** (verdadero) a los objetos D, E, F, L y se marcan. Puesto que el nodo objetivo M no está marcado, entonces
  - Se designa el objeto M como objeto en curso.
  - Se marca el objeto M. Por lo tanto tenemos: **ObjetosMarcados** = {D, E, F, L, M}
  - **ObjetosPrevios** =  $\phi$ .
  - Las seis reglas están activas. Por tanto, se tiene ReglasActivas = {1,2,3,4,5,6}.
  - Nos dirigimos a la Etapa 2.
- **Etapas 2:** Buscamos una regla que incluya el objetivo en curso M. Se encuentra la Regla 6, por lo que se va a la etapa 3.
- **Etapas 3:** La regla 6 no puede concluir puesto que el valor del objeto K es desconocido. Así que se va a la etapa 4.
- **Etapas 4:** El objeto K no está marcado. Entonces
  - **ObjetivosPrevios** = {M}.
  - Se elige el objeto K como objetivo en curso.
  - El objeto K está marcado. Por tanto se tiene, **ObjetosMarcados** = {D, E, F, L, M, K}
  - Se va a la Etapa 2
- **Etapas 2:** Se busca una regla que incluya el Objetivo en curso K pero no el anterior M. Se encuentra la Regla 4, y se continúa con la Etapa 3.
- **Etapas 3:** La regla 4 no puede concluir puesto que se desconocen los valores de los objetos C y G. Por ello, se continúa con la Etapa 4.
- **Etapas 4:** Los objetos C y G no están marcados, entonces
  - **ObjetivosPrevios** = {M, K}
  - Se elige uno de los objetos no marcados C o G como el nuevo objetivo en curso. Supongamos que elegimos C.
  - Marcamos el objeto C.
  - Por tanto, se tiene **ObjetosMarcados** = {D, E, F, L, M, K, C}
  - Continuamos con la Etapa 2.
- **Etapas 2:** Se busca una regla activa que incluya el objetivo en curso A pero no los objetivos previos {M, K, C}. No se encuentra ninguna regla que satisfaga estas condiciones, así que pasamos a la Etapa 5.
- **Etapas 5:** Puesto que el objetivo en curso A es diferente del inicial M, se pregunta al usuario por el valor del objeto A. Supóngase que A toma el valor **cierto** (verdadero), entonces se hace A = cierto y seguimos con la Etapa 6.



- **Etapa 6:** El objetivo en curso A no coincide con el previo M. Por tanto, el objetivo C se designa como objetivo en curso y se elimina de la lista de *ObjetivosPrevios*. Por esto:  $\text{ObjetivosPrevios} = \{M, K\}$  y se continua con la Etapa 2.
- **Etapa 2:** Se busca la Regla activa que concluya el objetivo C pero no los anteriores  $\{M, K\}$ . Se encuentra la Regla 1, por lo que vamos a la Etapa 3.
- **Etapa 3:** La Regla 1 no puede concluir porque el valor del objetivo B es desconocido. Así que se va a la Etapa 4.
- **Etapa 4:** El Objeto B no está marcado. Entonces
  - **ObjetivosPrevios** =  $\{M, K, C\}$
  - Se elige como objetivo en curso el único objeto no marcado, B.
  - Marcamos el objeto B. Debido a esto,  $\text{ObjetosMarcados} = \{D, E, F, L, M, K, C, A, B\}$ .
  - Se va a la Etapa 2.
- **Etapa 2:** Se busca una regla activa que incluya el objetivo B, pero no los previos  $\{M, K, C\}$ . Como no encontramos ninguna regla, se va a la Etapa 5.
- **Etapa 5:** Puesto que el objetivo en curso B no coincide con el inicial M, se pregunta al usuario el valor del objetivo en curso B. Supóngase que se da un valor cierto a B, entonces se hace:  $B = \text{cierto}$  y se va a la Etapa 6.
- **Etapa 6:** Como el objetivo en curso B no coincide con el inicial M, se designa el objetivo previo C como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello,  $\text{ObjetivosPrevios} = \{M, K\}$  y se va a la Etapa 2.
- **Etapa 2:** Buscamos una regla activa que incluya el objetivo en curso C pero no los anteriores  $\{M, k\}$ . Se encuentra la Regla 1, por lo que se va a la Etapa 3.
- **Etapa 3:** Puesto que  $A = \text{cierto}$  y  $B = \text{cierto}$ , entonces  $C = \text{cierto}$  por la regla 1. Ahora se va a la Etapa 6.
- **Etapa 6:** EL Objetivo en Curso C no coincide con el inicial M. Entonces, se designa el objetivo previo K como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello,  $\text{ObjetivosPrevios} = \{M\}$  y se va a la Etapa 2.
- **Etapa 2:** Se busca una Regla activa que incluya el objetivo en curso K pero no los anteriores  $\{M\}$ . Se encuentra la Regla 4, por lo que se va a la Etapa 3.
- **Etapa 3:** La Regla 4 no puede concluir puesto que el valor del objeto G es desconocido. Por tanto, se va a la Etapa 4.
- **Etapa 4:** El Objeto G no está marcado. Entonces
  - **ObjetivosPrevios** =  $\{M, K\}$ .
  - El único objeto no marcado G se elige como objetivo en curso.
  - Se marca el Objeto G. Por esto  $\text{Objetos Marcados} = \{D, E, F, L, M, K, C, A, B, G\}$ .
  - Se va a la Etapa 2.
- **Etapa 2:** Se busca una regla activa que incluya el objetivo en curso G pero no los anteriores  $\{M, K\}$ . Se encuentra la Regla 2, por lo que se va a la Etapa 3.
- **Etapa 3:** Puesto que  $D = \text{cierto}$ ,  $E = \text{cierto}$  y  $F = \text{cierto}$ , entonces  $G = \text{cierto}$  por la Regla 2. Ahora se va a la Etapa 6.



- **Etapa 6:** El objetivo en curso G no coincide con el inicial M. Entonces se designa el Objetivo Previo K como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello,  $ObjetivosPrevios = \{M\}$  y se va a la Etapa 2.
- **Etapa 2:** Se busca una Regla activa que incluya el objetivo en curso K pero no los anteriores  $\{M\}$ . Se encuentra la Regla 4, por lo que se va a la Etapa 3.
- **Etapa 3:** Puesto que C = cierto y G = cierto, entonces K = cierto por la regla 4. Seguidamente se va a la etapa 6.
- **Etapa 6:** El Objetivo en curso K no coincide con el inicial M. Entonces, se designa el objetivo previo M como objetivo en curso y se elimina de *ObjetivosPrevios*. Por ello  $ObjetivosPrevios = \emptyset$  y se va a la etapa 2.
- **Etapa 2:** Se busca una regla activa que incluya el objetivo en curso M. Se encuentra la Regla 6, por lo que se va a la Etapa 3.
- **Etapa 3:** Puesto que K= cierto y L = cierto, entonces M = cierto por la regla 6. Ahora se va a la etapa 6.
- **Etapa 6:** El Objetivo en curso M coincide con el inicial. En consecuencia, se va a la Etapa 7.
- **Etapa 7:** El Algoritmo devuelve el valor **M = cierto**.

Debemos notar que a pesar de que los objetos H, I y J tienen valores desconocidos, el algoritmo orientado a un objetivo ha sido capaz de concluir el valor del objetivo M. La razón de este resultado está en que el conocimiento de objeto L convierte al conocimiento de los objetos H, I y J es irrelevante para el conocimiento del objeto M.

Las estrategias de encadenamiento de reglas se utilizan en problemas en los que algunos hechos (por ejemplo, síntomas) se dan por conocidos y se buscan algunas conclusiones (por ejemplo: enfermedades). Por el contrario las estrategias de encadenamiento de reglas orientadas a un objetivo se utilizan en problemas en los que se dan algunos objetivos (enfermedades) y se buscan los hechos (síntomas) para que éstas sean posibles.



## Compilación de reglas

Otra forma de tratar con reglas encadenadas consiste en comenzar con un conjunto de datos (información) y tratar de alcanzar algunos objetivos. Esto se conoce con el nombre de **compilación de reglas**. Cuando ambos, datos y objetivos, se han determinado previamente, las reglas pueden ser compiladas, es decir, pueden escribirse los objetivos en función de los datos para obtener las llamadas **ecuaciones objetivo**. Este método es explicado mejor en un ejemplo práctico.

### Ejemplo 2.12 Compilación de Reglas

Considérese el conjunto de seis reglas de la figura 2.11 y supóngase que son conocidos los valores de los objetos A,B,D,E,F,H e I y que los restantes objetos, C,G,J,K,L y M, son objetivos. Detonemos por “ $\wedge$ ” y el operador “ $\vee$ ”; entonces, utilizando las seis reglas, pueden obtenerse las siguientes ecuaciones objetivo:

- La Regla 1 implica  $C = A \wedge B$ .
- La Regla 2 implica  $G = D \wedge E \wedge F$ .
- La Regla 3 implica  $J = H \wedge I$ .
- La Regla 4 implica  $K = C \wedge G = (A \wedge B) \wedge (D \wedge E \wedge F)$
- La Regla 5 implica  $L = G \wedge J = (D \wedge E \wedge F) \wedge (H \wedge I)$
- La Regla 6 implica  $M = K \wedge L = A \wedge B \wedge D \wedge E \wedge F \wedge H \wedge I$

Las tres primeras ecuaciones son equivalentes a las tres primeras reglas. Las tres ecuaciones objetivo son, respectivamente, equivalentes a las reglas siguientes:

- Regla 4a: Si A y B y D y E y F, entonces K.
- Regla 5a: Si D y E y F y H e I, entonces L
- Regla 6a: Si A y B y D y E y F y H e I, entonces M.

Por ello, si, por ejemplo, cada uno de los objetos {A,B,D,E,F,H,I} toma el valor cierto, entonces se obtiene de forma inmediata, a partir de las Reglas 4a, 5a y 6a, que los objetos {K,L,M} deben ser ciertos.



#### 2.4.4 Control de Coherencia

En situaciones complejas, incluso verdaderos expertos pueden dar información inconsistente (por ejemplo, reglas inconsistentes y/o combinaciones de hechos no factibles). Por ello, es muy importante controlar a coherencia del conocimiento tanto durante la construcción de la Base de Conocimiento como durante los procesos de adquisición de datos y razonamiento. Si la base de conocimiento contiene información inconsistente (por ejemplo, reglas y/o hechos), es muy probable que el sistema experto se comporte de forma poco satisfactoria y obtenga conclusiones absurdas.

El Objetivo del Control de la Coherencia consiste en

1. Ayudar al usuario a no dar hechos inconsistentes, por ejemplo, dándole al usuario las restricciones que debe satisfacer la información demandada.
2. Evitar que entre en la base de conocimiento cualquier tipo de conocimiento inconsistente o contradictorio.

El Control de la Coherencia debe hacerse controlando la coherencia de las reglas y la de los hechos.

Objetos		Conclusiones		Conclusiones Contradictorias
A	B	Regla 1	Regla 2	
C	C	$B = C$	$B = F$	Sí
C	F	$B = C$	$B = F$	Sí
F	C	-	-	No
F	F	-	-	No

Tabla 2.8 Una tabla de verdad que muestra que las Reglas 1 y 2 son coherentes.



## Coherencia de Reglas

### Definición 2.2 Reglas Coherentes.

Un Conjunto de Reglas se denomina coherente si existe, al menos, un conjunto de valores de todos los objetos que producen conclusiones no contradictorias.

En consecuencia, un conjunto coherente de reglas no tiene por qué producir conclusiones no contradictorias para todos los posibles conjuntos de valores de los Objetos. Es decir, es suficiente que exista un conjunto de valores que conduzcan a conclusiones no contradictorias.

### Ejemplo 2.13 Conjunto de Reglas Incoherentes.

Considérese las cuatro reglas siguientes, que relacionan dos Objetos A y B binarios {C,F}:

- **Regla 1:** Si  $A = C$ , entonces  $B = C$ .
- **Regla 2:** Si  $A = C$ , entonces  $B = F$ .
- **Regla 3:** Si  $A = F$ , entonces  $B = C$ .
- **Regla 4:** Si  $A = F$ , entonces  $B = F$ .

Entonces, pueden obtenerse las siguientes conclusiones:

1. Las Reglas 1 -2 son coherentes puesto que, tal como se muestra en la tabla 2.8, para  $A = F$ , no producen conclusiones.
2. Las Reglas 1-3 son coherentes puesto que para  $A = F$  y  $B = C$ , producen una conclusión ( $B = C$ ) (Véase la Tabla 2.9)
3. Las Reglas 1-4 son incoherentes porque producen conclusiones contradictorias para todos los posibles valores de A y B, tal como se ve en la Tabla 2.10

Nótese que un conjunto de reglas puede ser coherente, aunque algunos conjuntos de valores puedan producir conclusiones inconsistentes. Estos conjuntos de valores se llaman *valores no factibles*. Por ejemplo, las Reglas 1-2 son coherentes, aunque producen conclusiones inconsistentes en todos los casos en que  $A = C$ . En consecuencia, el subsistema de control de coherencia eliminará automáticamente el valor C de la lista de posibles



valores del Objeto A, permitiendo de esta forma al usuario seleccionar sólo valores factibles de los objetos.

Objetos		Conclusiones			Conclusiones Contradictorias
A	B	Regla 1	Regla 2	Regla 3	
C	C	B = C	B = F	-	Sí
C	F	B = C	B = F	-	Sí
F	C	-	-	B = C	No
F	F	-	-	B = C	Sí

Tabla 2.9 Una Tabla de Verdad que muestra que las reglas 1-3 son coherentes

Objetos		Conclusiones				Conclusiones Contradictorias
A	B	Regla 1	Regla 2	Regla 3	Regla 4	
C	C	B = C	B = F	-	-	Sí
C	F	B = C	B = F	-	-	Sí
F	C	-	-	B = C	B = F	Sí
F	F	-	-	B = C	B = F	Sí

Tabla 2.10 Una Tabla de verdad que muestra que las Reglas 1-4 son incoherentes.



**Definición 2.3 Valores no factibles:** Se dice que un valor “a” para el Objeto “A” no es factible si las conclusiones obtenidas al hacer  $A=a$  contradicen cualquier combinación de valores del resto de los objetos.

Por ello, cualquier valor no factible debe ser eliminado de la lista de valores posibles de su correspondiente objeto para eliminar la posibilidad de que el motor de inferencia pueda obtener conclusiones inconsistentes.

### Ejemplo 2.15 Coherencia de Reglas

Supongamos que tenemos los cuatro objetos:  $A \in \{0,1\}$ ,  $B \in \{0,1\}$ ,  $C \in \{0,1,2\}$  y  $D \in \{0,1\}$ . Considérese las cuatro reglas:

- **Regla 1:** Si  $A = 0$  y  $B = 0$ , entonces  $C = 0$ .
- **Regla 2:** Si  $A = 0$  y  $D = 0$ , entonces  $C = 1$ .
- **Regla 3:** Si  $A = 0$  y  $B = 0$ , entonces  $C = 1$ .
- **Regla 4:** Si  $A = 0$ , entonces  $B = 0$ .
- **Regla 5:** Si  $B = 0$ , entonces  $A = 1$ .

Supóngase ahora que se desea añadir las tres últimas reglas a una Base de Conocimientos que contiene las dos primeras reglas. Entonces, las Reglas 1 y 3 son inconsistentes, puesto que tienen la misma premisa pero diferentes conclusiones. Por tanto la Regla 3 debe ser rechazada y el experto humano informado de la razón del rechazo. El experto humano corregirá la regla en cuestión y/o reglas existentes si estas fueren incorrectas. La regla 4 entrará en la base de conocimiento, puesto que es consistente con las reglas 1 y 2. La regla 5 es inconsistente con la Regla 4. Por ello, la consistencia de ambas reglas debe ser comprobada antes de pasar a formar parte de la Base de Conocimiento.



## Coherencia de Hechos

Los Datos o evidencias suministrados por los usuarios deben ser también consistentes en sí y con el conjunto de reglas de la base de Datos. Por ello, el sistema no debe aceptar hechos que contradigan el conjunto de reglas y/o el conjunto de hechos existente en cada instante del proceso.

Por ejemplo con una base de Conocimiento que contenga las dos primeras reglas del Ejemplo 2.15, El Sistema no debe aceptar el conjunto de hechos  $A = 0$ ,  $B = 0$  y  $C = 1$  puesto que contradicen la Regla 1.

El Sistema debe también comprobar si existe o no, una solución factible e informar al usuario en consecuencia. Si en el ejemplo anterior se trata de dar la información  $A = 0$ ,  $B = 0$  y  $D = 0$ , el sistema debe detectar que no existe ningún valor de  $C$  que sea consistente con la Base de Conocimiento, debemos notar que antes de conocer los valores de los objetos, existe una solución factible. Por ejemplo:

$A = 0$ ,  $B = 0$ ,  $C = 0$  y  $D = 1$  (estos hechos no contradicen la base de conocimiento). Por ello, la inconsistencia surge de que los hechos y las reglas sean inconsistentes.

La Coherencia de los hechos puede lograrse mediante las estrategias siguientes:

1. Eliminar todos los valores no factibles (los que contradicen el conjunto de reglas y/o hechos) de los objetos una vez detectados. Cuando se pregunte al usuario por información sobre los valores de un conjunto de objetos, el sistema experto deberá aceptar sólo los valores de cada objeto que sean consistentes con las reglas y con los conocimientos previos. Consideremos, por ejemplo la base de conocimientos del Ejemplo 2.15 y supóngase que al sistema experto se le ha dado la información  $A=0$  y  $C=1$ ; entonces el sistema debe saber que  $B \neq 0$ . Por ello, este valor debe ser eliminado de la lista de posibles valores del Objeto B.
2. El Motor de Inferencia debe comprobar que los hechos conocidos no contradicen el conjunto de reglas. En la Situación anterior, por ejemplo, el sistema no debe aceptar el conjunto de hechos  $A=1, B=1$  y  $C=2$ . Si el sistema no elimina los valores no factibles, entonces el usuario podrá dar evidencias contradictorias tales como Pago = Autorizado y NIP = incorrecto, que vimos en el ejemplo 2.1 del Cajero Automático.



Por lo tanto tan pronto como se da la evidencia, Pago = Autorizado, el sistema debe seleccionar sólo los valores del NIP que no conduzcan a conclusiones que sean contradictorias.

3. Suministrar al usuario una lista de objetos a los que no se ha asignado valores previamente.
4. Para cada uno de los objetos, mostrar y aceptar sólo sus valores factibles
5. Actualizar continuamente la base de conocimiento, o sea, tan pronto como se dé un hecho o se obtenga una conclusión, y la correcta eliminación de los valores no factibles. El motor de inferencia obtiene todas las conclusiones posibles examinando, y posiblemente concluyendo, las reglas tan pronto como una simple unidad de información llega al sistema.

Nótese que dar varias unidades de información simultáneamente puede conducir a inconsistencias en la base de datos. Por ejemplo, dado  $A=0$ , no se puede dar la información combinada  $B=0$  y  $C=1$ .

En este caso, el orden de la información afecta a los posibles valores futuros de los objetos que conducen a compatibilidad, es decir tras dar  $A =0$  se puede dar  $B=0$  ó  $C = 1$ , pero estas dos opciones imponen restricciones diferentes a los posibles futuros valores de los restantes objetos.

La actualización continua de la base de conocimiento es muy importante puesto que no actualizar implica la posibilidad de que evidencias contradictorias puedan convivir en la base de conocimiento. Por ello el conocimiento debe ser actualizado inmediatamente tras la incorporación de cada hecho.

Por ello, tanto la eliminación automática de valores no factibles como la actualización continua del conocimiento aseguran la coherencia de la base de conocimiento.



### 2.4.5 Explicando Conclusiones

Muchas veces las conclusiones no bastan para satisfacer al usuario de un sistema experto, Normalmente los usuarios esperan que el sistema les de algún tipo de explicación que indique el por qué de las conclusiones. Durante el proceso realizado por el motor de inferencia, las reglas activas (las que han concluido) forman la base del mecanismo de explicación, que es regulado por el subsistema de explicación.

En los sistemas expertos basados en reglas, es fácil dar explicaciones de las conclusiones obtenidas. El Motor de Inferencia obtiene conclusiones basándose en un conjunto de reglas y, por tanto, conoce de que regla precede cada conclusión. Por ello, el sistema puede dar al usuario la lista de hechos concluidos junto con las reglas usadas para la obtención de los mismos.

#### Ejemplo 2.17 Explicando Conclusiones

Considérese las seis reglas de la Figuras 2.10 y 2.11. Como en el Ejemplo 2.7, supongamos que se sabe que los objetos A, B, D, E, F, H e I son *ciertos* y que los restantes objetos toman valores desconocidos. Entonces, Aplicando el Algoritmo 2.1 y al examinar las reglas que han sido ejecutadas, el sistema experto puede suministrar la explicación siguiente a las conclusiones obtenidas:

1. Hechos dados:

A = cierto,    B = cierto,    D = cierto,    E = cierto,  
F = cierto,    H = cierto,    I = cierto

2. Conclusiones y Explicaciones:

- C = cierto, basada en la regla 1.
- G = cierto, basada en la regla 2.
- J = cierto, basada en la regla 3.
- K = cierto, basada en la regla 4.
- L = cierto, basada en la regla 5.
- M = cierto, basada en la regla 6.



## 2.5 Especificación y Descripción de las Herramientas y Tecnologías de Desarrollo.

### 2.5.1 Matlab

Matlab (abreviatura de *MATrix LABORatory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

MATLAB, escrito en C, se comunica con el usuario a través de una ventana con un prompt, desde donde se definen variables y se utilizan subprogramas o funciones tanto predefinidas como definidas por el mismo usuario. Permite el uso de diversos tipos de variables como matrices de reales, complejos o caracteres sin la necesidad de hacer una declaración previa lo que lo hace muy versátil a la hora de hacer programas en que el número de variables a utilizar es incierto. Posee altas capacidades de manejo gráfico en ya sea en dos o en tres dimensiones lo que permite a un usuario relativamente diestro un despliegue claro de la información. También incluye herramientas de interfaz gráfica con el usuario. Además es posible comunicarse con el sistema operativo para leer o escribir archivos, entre otras.

MATLAB se complementa con paquetes de subprogramas y funciones orientados a aplicaciones específicas llamados *Toolboxes*. Existe una gran cantidad de ellos, especialmente en áreas de análisis matemático y de modelación, simulación y control de procesos, análisis de imágenes, entre otros.

Entre los *toolboxes* más comunes tenemos:

- Database Toolbox
- MATLAB Compiler
- Symbolic Math Toolbox
- NAG Foundation Toolbox - Numerical & Statistical Library
- Mapping Toolbox
- Wavelet Toolbox
- Partial Differential Equation Toolbox
- Financial Toolbox
- LMI Control Toolbox
- Fuzzy Logic Toolbox
- Model Predictive Control Toolbox
- Frequency Domain Identification Toolbox
- Statistics Toolbox



- Image Processing Toolbox
- Neural Network Toolbox
- Signal Processing Toolbox
- Optimization Toolbox
- Robust Control Toolbox
- System Identification Toolbox
- Control System Toolbox

Existen algunos que son más básicos y que son prerequisites de otros, por ejemplo, el *toolbox* financiero utiliza funciones que están definidas en los *toolboxes* de estadísticas y de optimización (*Statistics and Optimization toolboxes*).

Muchos *toolboxes* definen todo un protocolo de cómo representar elementos de interés para ellos, por ejemplo el de control de sistemas define una manera de representar sistemas dinámicos que es usado como base para otros *toolboxes* más específicos. Actualmente existen *toolboxes* que permiten la comunicación con otros programas o elementos, como por ejemplo, el *Excel Link Toolbox* para interactuar con datos de planillas de cálculo Excel.

Una mención especial merece un *toolbox* que permite una representación en diagrama de bloques para sistemas dinámicos orientado a la simulación, este es *SIMULINK*. Este permite la simulación de complejos sistemas donde la interconexión de bloques básicos o no tan básicos se hace a través de la conexión de líneas que llevan las señales del tiempo. Simulink se utiliza en general para el diseño y prueba de diversos sistemas, así como para el diseño, ajuste y prueba de controladores para procesos. Existen librerías para Simulink (llamadas también *toolboxes*), entre estos se puede mencionar:

- Stateflow
- Power System Blockset
- Communications Toolbox
- DSP Blockset
- Fixed-Point Blockset
- Nonlinear Control Design Blockset
- Real-Time Workshop

Una gran cantidad de áreas de investigación desarrollan *toolboxes* con bloques donde se encuentran modelos de operaciones unitarias que se utilizan en simulaciones, como es el caso de Power System Blockset donde se encuentran máquinas sincrónicas, elementos para modelar transformadores y líneas de alta tensión, etc. También hay empresas privadas de componentes electrónicos (Motorola) construyen *toolboxes* para el diseño de circuitos mediante simulink.



## Image Processing Toolbox

Este Toolbox proporciona a MATLAB un conjunto de funciones que amplían las capacidades del producto para realizar desarrollo de aplicaciones y de nuevos algoritmos en el campo del procesamiento y análisis de imágenes. Algunas de las funciones más importantes son:

- Análisis de imágenes y estadística.
- Diseño de filtros y recuperación de imágenes.
- Mejora de imágenes.
- Operaciones morfológicas.
- Definición de mapas de colores y modificación gráfica.
- Operaciones geométricas.
- Transformación de imágenes.

### 2.5.2 Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

## Historia

### Visual Studio 6.0

Se lanzó en 1998 y fue la última versión en ejecutarse en la plataforma Win9x. Los números de versión de todas las partes constituyentes pasaron a 6.0, incluyendo Visual J++ y Visual InterDev que se encontraban en las versiones 1.1 y 1.0 respectivamente. Esta versión fue la base para el sistema de desarrollo de Microsoft para los siguientes 4 años, en los que Microsoft migró su estrategia de desarrollo al .NET Framework.

Visual Studio 6.0 fue la última versión en que Visual Basic se incluía de la forma en que se conocía hasta entonces; versiones posteriores incorporarían una versión muy diferente del lenguaje



con muchas mejoras, fruto de la plataforma .NET. También supuso la última versión en incluir Visual J++, que proporcionaba extensiones de la plataforma Java, lo que lo hacía incompatible con la versión de Sun Microsystems. Esto acarreó problemas legales a Microsoft, y se llegó a un acuerdo en el que Microsoft dejaba de comercializar herramientas de programación que utilizaran la máquina virtual de Java.

Aunque el objetivo a largo plazo de Microsoft era unificar todas las herramientas en un único entorno, esta versión en realidad añadía un entorno más a Visual Studio 5.0: Visual J++ y Visual Interdev se separaban del entorno de Visual C++, al tiempo que Visual FoxPro y Visual Basic seguían manteniendo su entorno específico.

### **Visual Studio .NET 2002**

En esta versión se produjo un cambio sustancial, puesto que supuso la introducción de la plataforma .NET de Microsoft. .NET es una plataforma de ejecución intermedia multilenguaje, de forma que los programas desarrollados en .NET no se compilan en lenguaje máquina, sino en un lenguaje intermedio (CIL - Common Intermediate Language) denominado Microsoft Intermediate Language (MSIL). En una aplicación MSIL, el código no se convierte a lenguaje máquina hasta que ésta se ejecuta, de manera que el código puede ser independiente de plataforma (al menos de las soportadas actualmente por .NET). Las plataformas han de tener una implementación de Infraestructura de Lenguaje Común (CLI) para poder ejecutar programas MSIL. Actualmente se pueden ejecutar programas MSIL en Linux y Mac OS X usando implementaciones de .NET que no son de Microsoft, tales como Mono y DotGNU.

Visual Studio .NET 2002 supuso también la introducción del lenguaje C#, un lenguaje nuevo diseñado específicamente para la plataforma .NET, basado en C++ y Java. Se presentó también el lenguaje J# sucesor de J++ el cual, en lugar de ejecutarse en una máquina virtual de Java, se ejecuta únicamente en el framework .NET. El lenguaje Visual Basic fue remodelado completamente y evolucionó para adaptarse a las nuevas características de la plataforma .NET, haciéndolo mucho más versátil y dotándolo con muchas características de las que carecía. Algo similar se llevó a cabo con C++, añadiendo extensiones al lenguaje llamadas Managed Extensions for C++ con el fin de que los programadores pudieran crear programas en .NET. Por otra parte, Visual FoxPro pasa a comercializarse por separado.



Todos los lenguajes se unifican en un único entorno. La interfaz se mejora notablemente en esta versión, siendo más limpia y personalizable.

Visual Studio .NET puede usarse para crear programas basados en Windows (usando Windows Forms en vez de COM), aplicaciones y sitios web (ASP.NET y servicios web), y dispositivos móviles (usando el .NET Compact Framework).

Esta versión requiere un sistema operativo basado en NT. La versión interna de Visual Studio .NET es la 7.0.

### **Visual Studio .NET 2003**

Visual Studio .NET 2003 supone una actualización menor de Visual Studio .NET. Se actualiza el .NET Framework a la versión 1.1. También se añade soporte con el fin de escribir aplicaciones para determinados dispositivos móviles, ya sea con ASP.NET o con el .NET Compact Framework. Además el compilador de Visual C++ se mejora para cumplir con más estándares, el Visual C++ Toolkit 2003.

Visual Studio 2003 se lanza en 4 ediciones: Academic, Professional, Enterprise Developer, y Enterprise Architect. La edición Enterprise Architect incluía una implantación de la tecnología de modelado Microsoft Visio, que se centraba en la creación de representaciones visuales de la arquitectura de la aplicación basadas en UML. También se introdujo Enterprise Templates, para ayudar a grandes equipos de trabajo a estandarizar estilos de programación e impulsar políticas de uso de componentes y asignación de propiedades.

Microsoft lanzó el Service Pack 1 para Visual Studio 2003 el 13 de Septiembre de 2006.

La versión interna de Visual Studio .NET 2003 es la 7.1 aunque el formato del archivo es 8.0.

### **Visual Studio 2005**

Visual Studio 2005 se empezó a comercializar a través de Internet a partir del 4 de Octubre de 2005 y llegó a los comercios a finales del mes de Octubre en inglés. En castellano no salió hasta el 4 de Febrero de 2006. Microsoft eliminó .NET, pero eso no indica que se alejara de la plataforma .NET, de la cual se incluyó la versión 2.0.

La actualización más importante que recibieron los lenguajes de programación fue la inclusión de tipos genéricos, similares en muchos aspectos a las plantillas de C#. Con esto se



consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en áreas donde antes no era posible. C++ tiene una actualización similar con la adición de C++/CLI como sustituto de C# manejado.

Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Visual Studio 2005 también añade soporte de 64 bits. Aunque el entorno de desarrollo sigue siendo una aplicación de 32 bits Visual C++ 2005 soporta compilación para x86 - 64 (AMD64 e Intel 64) e IA-64 (Itanium). El SDK incluye compiladores de 64 bits así como versiones de 64 bits de las librerías.

Visual Studio 2005 tiene varias ediciones radicalmente distintas entre sí: Express, Standard, Professional, Tools for Office, y 5 ediciones Visual Studio Team System. Éstas últimas se proporcionaban conjuntamente con suscripciones a MSDN cubriendo los 4 principales roles de la programación: Architects, Software Developers, Testers, y Database Professionals. La funcionalidad combinada de las 4 ediciones Team System se ofrecía como la edición Team Suite.

Tools for the Microsoft Office System está diseñada para extender la funcionalidad a Microsoft Office.

Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas disponibles gratuitamente a través de la página de Microsoft se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows, y Visual Web Developer para la creación de sitios web ASP.NET. Las ediciones express carecen de algunas herramientas avanzadas de programación así como de opciones de extensibilidad.

Se lanzó el service Pack 1 para Visual Studio 2005 el 14 de Diciembre de 2006.

La versión interna de Visual Studio 2005 es la 8.0, mientras que el formato del archivo es la 9.0.



## Visual Studio 2008

Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.

El nuevo framework (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

Cuando se inicia un proyecto, seleccionando el lenguaje, se pueden crear cuatro tipos de proyecto WPF:

- WinFX Windows Application
- WinFX Web Browser Application
- WinFX Custom Library
- WinFX Service Library

Una de las mejores características que se agregaron a Visual Studio 2008 es la capacidad de especificar el Framework sobre el cual se desea compilar. En las cajas de dialogo Advanced Compiler Settings (VB) y Advanced Build Settings (C#), ahora existe un nuevo campo denominado Target Framework que permite seleccionar lo siguiente:

- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades.

La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.



Con Visual Studio Tools for Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.

Visual Studio 2008 permite incorporar características del nuevo Windows Presentation Foundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft Foundation Class Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.

LINQ (Language Integrated Query) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a Base de Datos, a través de extensiones para C++ y Visual Basic así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando la misma sintaxis, prescindiendo del uso de lenguajes especializados como SQL o XPath.

Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).

.NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

ClickOnce ha sido mejorado para brindar soporte a Windows Vista:

- Soporta la implementación de proyectos WPF Web Browser Application. Como estos proyectos requieren una implementación especial y una configuración adicional de seguridad, ahora Visual Studio provee de los asistentes y pasos requeridos para configurar dichos elementos.
- Las empresas que desarrollan software comercial (ISVs), toman ventaja de una característica interesante ya que pueden firmar nuevamente el manifiesto de la aplicación, colocando elementos dentro del manifiesto que permiten identificar a la empresa.



Soporte para publicación a través del Publicador de Proyectos o del Asistente de Publicación.

## Visual Studio 2010

Visual Studio 2010 es la versión más reciente de esta herramienta, acompañada por .NET Framework 4.0. La fecha prevista para el lanzamiento de la versión final fue el 12 de abril de 2010.

Hasta ahora, uno de los mayores logros de la versión 2010 de Visual Studio ha sido el de incluir las herramientas para desarrollo de aplicaciones para Windows 7, tales como herramientas para el desarrollo de la Taskbar (System.Windows.Shell) y la Ribbon Preview para WPF.

### 2.5.3 Microsoft SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL, Interbase, Firebird o MySQL.

#### Características de Microsoft SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
  
- Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.
- Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.
- En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD



- Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

## **Programación**

### **T-SQL**

T-SQL (Transact-SQL) es el principal medio de programación y administración de SQL Server. Expone las palabras clave para las operaciones que pueden realizarse en SQL Server, incluyendo creación y modificación de esquemas de la base de datos, introducir y editar datos en la base de datos, así como supervisión y gestión del propio servidor. Las aplicaciones cliente, ya sea que consuman datos o administren el servidor, aprovechan la funcionalidad de SQL Server mediante el envío de consultas de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente. SQL Server permite que sean administrados mediante T-SQL. Para esto, expone tablas de sólo lectura con estadísticas del servidor. La funcionalidad para la administración se expone a través de procedimientos almacenados definidos por el sistema que se pueden invocar desde las consultas de T-SQL para realizar la operación de administración. También es posible crear servidores enlazados (Linked Servers) mediante T-SQL. Los servidores enlazados permiten el funcionamiento entre múltiples servidores con una consulta.

### **Cliente Nativo de SQL**

Cliente Nativo de SQL es la librería de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server Plugins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos genéricos.



## **Desventajas**

MSSQL usa Address Windowing Extensión (AWE) para hacer el direccionamiento de 64-bit. Esto le impide usar la administración dinámica de memoria y sólo le permite alojar un máximo de 64GB de memoria compartida.

MSSQL no maneja compresión de datos (en SQL Server 2005 y 2000, solamente la versión 2008 Enterprise Edition incluye esta característica), por lo que ocupa mucho espacio en disco.

MSSQL está atado a la plataforma del sistema operativo sobre la cual se instala.

## **SQL Server 2005**

SQL Server 2005 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

El motor de datos SQL Server 2005 constituye el núcleo de esta solución de administración de datos empresariales. Asimismo, SQL Server 2005 combina lo mejor en análisis, información, integración y notificación. Esto permite que su negocio cree y despliegue soluciones de BI rentables que ayuden a su equipo a incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles.

La integración directa con Microsoft Visual Studio, el Microsoft Office System y un conjunto de nuevas herramientas de desarrollo, incluido el Business Intelligence Development Studio, distingue al SQL Server 2005. Ya sea que usted se desempeñe como encargado de desarrollo, administrador de base de datos, trabajador de la industria de la información o dirija una empresa, SQL Server 2005 ofrece soluciones innovadoras que le ayudan a obtener más valor de sus datos.

## **Características SQL Server 2005**

Hoy en día las organizaciones enfrentan numerosos desafíos de datos, tales como la necesidad de tomar decisiones más rápidas y más orientadas a datos, la necesidad de aumentar la productividad y flexibilidad del personal de desarrollo y presionan para reducir los presupuestos



generales de informática (TI) a la vez que escalan la infraestructura para satisfacer las exigencias cada vez mayores.

SQL Server 2005 está diseñado para ayudar a las empresas a enfrentar estos desafíos. Esta solución de administración y análisis de datos de próxima generación ofrece seguridad, escalabilidad y disponibilidad mayores a las aplicaciones de datos empresariales y analíticas, a la vez que las hace más fáciles de crear, desplegar y administrar.

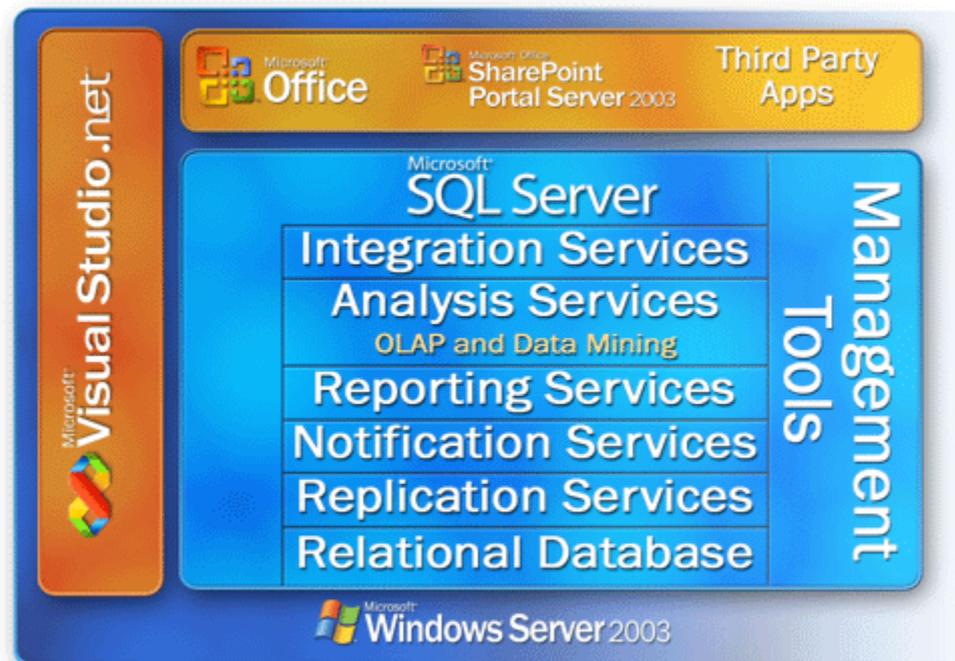
Con la ampliación de las ventajas de SQL Server 2000, SQL Server 2005 ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar lo siguiente:

- Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.
- Maximizar la productividad de TI mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.
- Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

SQL Server 2005 potencia su infraestructura de datos en tres áreas clave: administración de datos empresariales, productividad del encargado del desarrollo e inteligencia empresarial (BI). También abre nuevos caminos en precios y licencias accesibles, rutas de actualización a SQL Server 2005 y el sistema Microsoft Windows Server.

### **Plataforma de datos de SQL Server**

SQL Server es una solución de datos: global, integrada y de extremo a extremo que habilita a los usuarios en toda su organización mediante una plataforma más segura, confiable y productiva para datos empresariales y aplicaciones de BI. SQL Server 2005 provee herramientas sólidas y conocidas a los profesionales de TI, así como también a trabajadores de la información, reduciendo la complejidad de la creación, despliegue, administración y uso de aplicaciones analíticas y de datos empresariales en plataformas que van desde los dispositivos móviles hasta los sistemas de datos empresariales. A través de un conjunto global de características, la interoperabilidad con sistemas existentes y la automatización de tareas rutinarias, SQL Server 2005 ofrece una solución completa de datos para empresas de todos los tamaños. La Figura muestra el diseño de la plataforma de datos SQL Server 2005.



La plataforma de datos SQL Server incluye las siguientes herramientas:

- **Base de datos relacional:** Un motor de base de datos relacional más segura, confiable, escalable y altamente disponible con mejor rendimiento y compatible para datos estructurados y sin estructura (XML).
- **Servicios de réplica:** Replica de datos para aplicaciones de procesamiento de datos distribuidas o móviles, alta disponibilidad de los sistemas, concurrencia escalable con almacenes de datos secundarios para soluciones de información empresarial e integración con sistemas heterogéneos, incluidas las bases de datos Oracle existentes.
- **Notification Services:** Capacidades avanzadas de notificación para el desarrollo y el despliegue de aplicaciones escalables que pueden entregar actualizaciones de información personalizadas y oportunas a una diversidad de dispositivos conectados y móviles.
- **Integration Services:** Capacidades de extracción, transformación y carga (ELT) de datos para almacenamiento e integración de datos en toda la empresa.
- **Analysis Services:** Capacidades de procesamiento analítico en línea (OLAP) para el análisis rápido y sofisticado de conjuntos de datos grandes y complejos, utilizando almacenamiento multidimensional.
- **Reporting Services:** Una solución global para crear, administrar y proporcionar tanto informes tradicionales orientados al papel como informes interactivos basados en la Web.
- **Herramientas de administración:** SQL Server incluye herramientas integradas de administración para administración y optimización avanzadas de bases de datos, así como también integración directa con herramientas tales como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS). Los protocolos de acceso de datos estándar reducen drásticamente el tiempo que demanda integrar los datos en SQL Server



con los sistemas existentes. Asimismo, el soporte del servicio Web nativo está incorporado en SQL Server para garantizar la interoperabilidad con otras aplicaciones y plataformas.

- **Herramientas de desarrollo:** SQL Server ofrece herramientas integradas de desarrollo para el motor de base de datos, extracción, transformación y carga de datos, minería de datos, OLAP e informes que están directamente integrados con Microsoft Visual Studio para ofrecer capacidades de desarrollo de aplicación de extremo a extremo. Cada subsistema principal en SQL Server se entrega con su propio modelo de objeto y conjunto de interfaces del programa de aplicación (API) para ampliar el sistema de datos en cualquier dirección que sea específica de su negocio.
- La plataforma de datos SQL Server 2005 ofrece los siguientes beneficios a las organizaciones de todas las magnitudes:
  - Aprovechamiento de activos de datos. Además de brindar una base de datos segura y confiable para aplicaciones analíticas y del rubro, SQL Server 2005 permite que los clientes obtengan más valor de sus datos al incluir una funcionalidad incorporada tal como informe, análisis y minería de datos. Puede aprovechar esta potencia y flexibilidad para entregar datos a cada rincón de su organización a una fracción del coste de algunos otros sistemas.
  - Aumento de la productividad. A través de las capacidades globales de BI y la integración con herramientas conocidas como Microsoft Office System, SQL Server 2005 brinda a los trabajadores de la información en toda su organización información empresarial crítica y oportuna adaptada a sus necesidades específicas. El objetivo es ampliar la BI a todos los usuarios en una organización y, en última instancia, ayudar a los usuarios en todos los niveles de la organización a tomar mejores decisiones empresariales según uno de sus activos más valiosos: sus datos.
  - Reducción de la complejidad de TI. SQL Server 2005 simplifica el desarrollo, el despliegue y la administración de aplicaciones de unidad de negocios y analíticas al ofrecer un entorno de desarrollo flexible para los encargados del desarrollo y herramientas integradas y automatizadas de administración para los administradores de bases de datos.
  - Menor coste total de propiedad (TCO). El enfoque y la atención integrados sobre la facilidad de uso y despliegue en SQL Server 2005 ofrece los costes directos, de implementación y mantenimiento más bajos de la industria para obtener un rápido rendimiento de su inversión en la base de datos.
  - SQL Server 2005 brinda la tecnología y las capacidades con las que puede contar su organización. Con avances significativos en las áreas clave de administración de datos empresariales, productividad del encargado del desarrollo y BI, son considerables los beneficios de actualización o migración a SQL Server 2005.

## SQL server 2008

SQL Server 2008 es un elemento fundamental de la Plataforma de Datos de Microsoft, capaz de gestionar cualquier tipo de datos, en cualquier sitio y en cualquier momento. Le permite almacenar datos de documentos estructurados, semiestructurados o no estructurados como son las



imágenes, música y archivos directamente dentro de la base de datos. SQL Server 2008 le ayuda a obtener más rendimiento de los datos, poniendo a su disposición una amplia gama de servicios integrados como son consultas, búsquedas, sincronizaciones, informes y análisis. Sus datos pueden almacenarse y recuperarse desde sus servidores más potentes del Data Center hasta los desktops y dispositivos móviles, permitiéndole tener un mayor control sobre la información sin importar dónde se almacena físicamente.

SQL Server 2008 le permite utilizar sus datos en aplicaciones a medida desarrolladas con Microsoft® .NET y Visual Studio y también desde su propia Arquitectura Orientada a Servicio (SOA) y los procesos empresariales empleando Microsoft® BizTalk® Server.

Además, las personas que gestionan la información pueden acceder directamente a los datos con las herramientas que utilizan habitualmente como Microsoft® Office 2007. SQL Server 2008 le ofrece una plataforma de datos, fiable, productiva e inteligente para cubrir todas sus

## **SOLUCIONES**

### **Business Intelligence**

SQL Server 2008 es una plataforma escalable de Business Intelligence optimizada para la integración de datos, elaboración de informes y análisis que hace posible poner al alcance de todos usuarios la inteligencia empresarial.

### **Virtualización y consolidación de servidores**

La virtualización de servidor, también conocida como virtualización de hardware, es un tema de plena actualidad en el mundo de TI debido a que permite reducir de manera drástica los costes y mejorar la agilidad de las organizaciones.

### **Consolidación de servidores**

SQL Server 2008 puede contribuir a reducir los costes de hardware y mantenimiento mediante una solución de consolidación de servidores flexible que aporta un rendimiento y una manejabilidad extraordinarios a las organizaciones.



## **OLTP**

SQL Server 2008 es el motor de base de datos escalable y de alto rendimiento que necesitan las aplicaciones de misión crítica con las mayores exigencias de disponibilidad y seguridad. SQL Server reduce el coste total de propiedad gracias a su mayor manejabilidad en entornos corporativos.

## **Data Warehouse**

SQL Server le ofrece una plataforma de data warehouse completa y escalable que le permite integrar datos dentro del DW más rápidamente, escalar y gestionar volúmenes de datos y usuarios cada vez mayores facilitando a todos las vistas de síntesis que necesitan.

## **Desarrollo de aplicaciones**

SQL Server 2008 constituye el eje central de una plataforma completa de programación de datos que le permite acceder y manipular datos críticos de negocio desde toda clase de dispositivos, plataformas y orígenes de datos.



## **CAPÍTULO 3**



### 3.1 Captura de Requerimientos

De acuerdo a las necesidades que tiene el médico especialista coordinador del proyecto se realiza un análisis de los requerimientos que van a ser puestos en marcha durante el desarrollo del sistema, para ello se establecerán módulos que contengan los requerimientos especificados:

- Diseño e Implementación del módulo de inferencia para el apoyo en la toma de decisiones del médico especialista.
- Procesamiento de imágenes en la prueba de campimetría. Que será capaz de:
  - En una primera visita del paciente comparar los resultados con un patrón de visión normal con la finalidad de apoyar al diagnóstico.
  - En visitas posteriores, comparará con los resultados obtenidos anteriormente y de esta manera evaluar el progreso de la enfermedad.
- Con la finalidad de hacer un seguimiento de la enfermedad del paciente se implementará un módulo de manejo de información acerca de los mismos.
- Implementación del método S.T.A.R (Scoring Tool for Assessing Risk, Método para la Valoración de Riesgo de Glaucoma a 5 años).
- Implementación de un módulo de ilustración de la anatomía del ojo humano mediante el uso de imágenes t animaciones 2D. El módulo estará en la capacidad de mostrar:
  - Vista de la anatomía de ojo humano con parámetros normales con fines ilustrativos.
  - Imágenes del ojo humano con parámetros normales de visión.
  - Imágenes del ojo humano con diferentes anomalías en el campo visual.

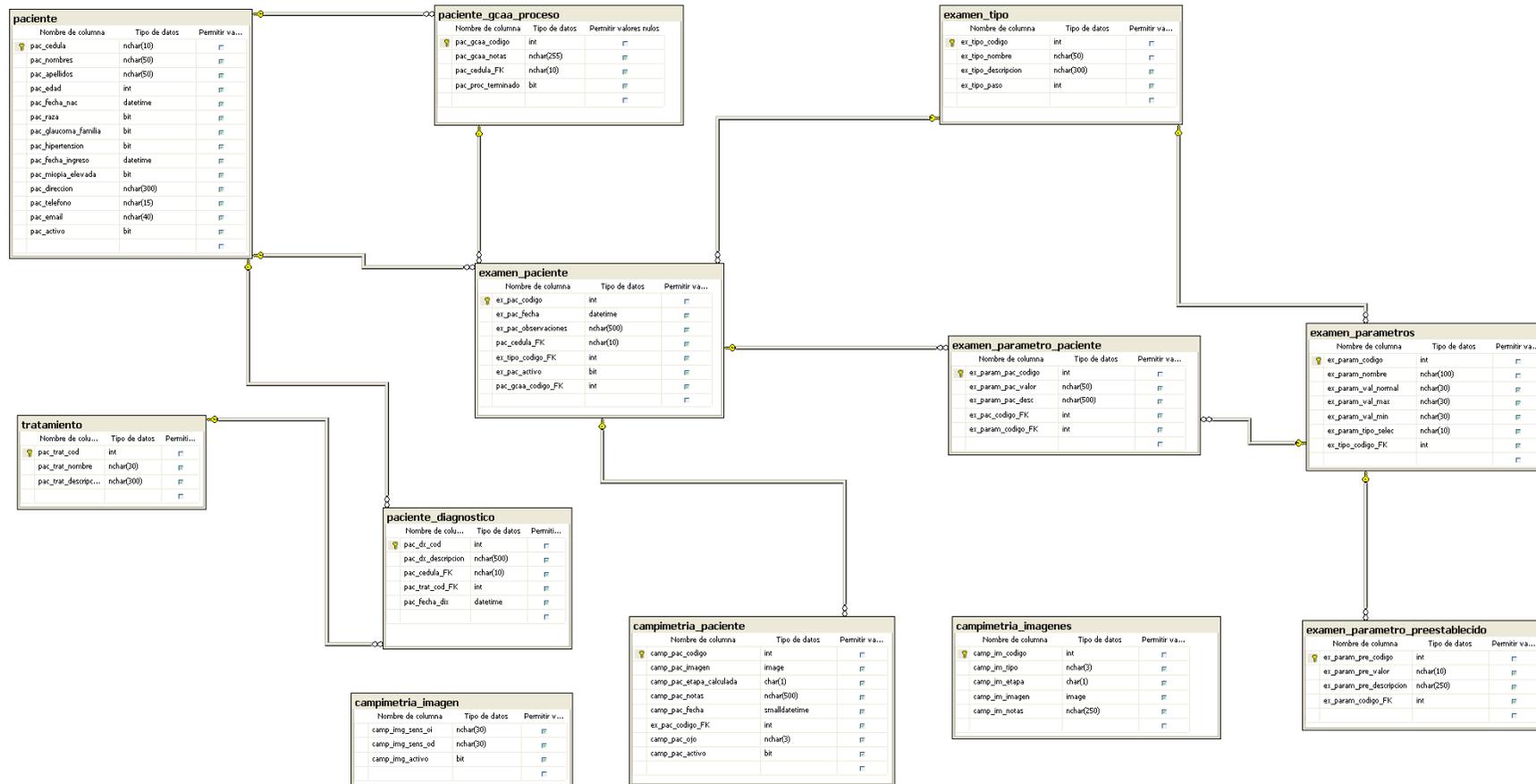


### **3.2 Análisis y Diseño de la Base de Datos**

A continuación se presenta el detalle de las tablas que contiene la base de datos de Sql Server 2005, esta base de datos contiene un total de 12 tablas, las cuales soportan la estructura del sistema del sistema experto, dicho modelo de datos fue diseñado para permitir la escalabilidad del sistema mediante el ingreso de nuevos exámenes y parámetros para dichos exmanes.



### Diagrama E-R Base de Datos DBGCAA





### 3.2.1 Diccionario de datos

Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción
<b>campimetria_imagenes</b>		Entidad			Entidad que almacena información sobre parámetros de referencia de una campimetría
	camp_im_codigo	<b>PK(int)</b>			Código de la Imagen de Referencia de Campimetría
	camp_im_etapa	char(1)			Etapa en la que se encuentra la imagen, puede tomar los valores a,b,c,d,e,f.
	camp_im_imagen	image			Archivo de Imagen de la etapa correspondiente para comparación.
	camp_im_notas	nchar(250)			Notas acerca de la imagen
<b>campimetria_paciente</b>		Entidad			Entidad en la que se almacena las imágenes de una campimetría tomada desde un



Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción
	camp_pac_codigo	<b>PK (int)</b>			paciente. Código del examen de campimetría de paciente.
	ex_pac_codigo_FK	<b>FK (int)</b>	examen_paciente	ex_pac_codigo	Llave externa para referenciar al examen al que pertenece este parámetro.
	camp_pac_imagen	image			Archivo de imagen de la campimetría del paciente.
	camp_pac_etapa_calculada	char(1)			Etapas calculadas, que indican el avance de la enfermedad, puede tomar los valores a,b,c,d,e,f.
	camp_pac_notas	nchar(500)			Notas sobre el examen como referencia para el especialista.
	camp_pac_fecha	smalldatetime			Fecha en la que se realizó el examen.
	camp_pac_ojo	nchar(3)			Indica cuál fue el ojo analizado, posibles valores (IZQ,DER)
	camp_pac_activo	bit			Indica el estado del examen (Eliminación lógica) 0 = FALSE, 1=TRUE



Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción
<b>examen_paciente</b>		Entidad			Entidad que almacena la información de un examen realizado a un paciente.
	ex_pac_codigo	<b>PK (int)</b>			Identificador único del examen del paciente.
	pac_cedula_FK	<b>FK nchar(10)</b>	paciente	pac_cedula	Identificador externo que indica el ID del paciente dueño del examen realizado.
	ex_tipo_codigo_FK	<b>FK (int)</b>	examen_tipo	ex_tipo_codigo	Identificador externo que indica el tipo de examen de esta instancia.
	pac_gcaa_codigo_FK	<b>FK (int)</b>	paciente_gcaa_proceso	pac_gcaa_codigo	Identificador Externo que indica el proceso de diagnóstico al que pertenece el examen, ya que un mismo paciente puede tener más de un proceso de diagnóstico asociado a su Historia Clínica.



Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción
	ex_pac_fecha	datetime			Fecha en la que se ingresó los datos del examen en el Sistema.
	ex_pac_observaciones	nchar(500)			Notas del especialista
	ex_pac_activo	bit			Estado de examen de paciente (Eliminación lógica) 0 = inactivo, 1=activo.
<b>examen_parametro_paciente</b>		Entidad			Entidad en la que se almacena la información de un parámetro asociado a un examen de paciente.
	ex_param_pac_codigo	<b>PK (int)</b>			Identificador único del parámetro de examen.



Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción
	ex_pac_codigo_FK	<b>FK (int)</b>	examen_paciente	ex_pac_codigo	Identificador externo que asocia al parámetro a un examen de paciente existente.
	ex_param_codigo_FK	<b>FK (int)</b>	examen_parametros	ex_param_codigo	Identificador externo que asocia al parámetro a un parámetro existente en la tabla examen_parametros
	ex_param_pac_valor	nchar(50)			Valor del parámetro
	ex_param_pac_desc	nchar(500)			Notas del especialista
<b>examen_parametro_pre establecido</b>		Entidad			Entidad en la que se almacena información de los valores posibles que puede tomar un parámetro (opcional) según el tipo en examen_parametros
	ex_param_pre_codigo	<b>PK (int)</b>			Identificador único de parámetro.
	ex_param_codigo_FK	<b>FK (int)</b>	examen_parametros	ex_param_codigo	Identificador externo del parámetro de examen asociado al valor.
Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción



	ex_param_pre_valor	nchar(10)			Valor del parámetro
	ex_param_pre_descripcion	nchar(250)			Descripción del valor de parámetro.
<b>examen_parametros</b>		Entidad			Entidad que almacena información de cada parámetro establecido en el sistema asociado al tipo de examen de la tabla examen_tipo.
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>
	ex_param_codigo	<b>PK (int)</b>			Identificador único del parámetro de examen creado.
	ex_tipo_codigo_FK	<b>FK (int)</b>	examen_tipo	ex_tipo_codigo	Identificador externo que asocia el parámetro a un tipo de examen existente.
	ex_param_nombre	nchar(100)			Nombre del parámetro
	ex_param_val_normal	nchar(30)			Valor normal del parámetro
	ex_param_val_max	nchar(30)			Valor máximo preestablecido
	ex_param_val_min	nchar(30)			Valor mínimo preestablecido
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>



	ex_param_tipo_selec	nchar(10)			Tipo de variable, valores posibles .SIMPLE = no tiene valores preestablecidos, SELECTED= tiene valores preestablecidos en tabla examen_parametro_preestablecido.
<b>examen_tipo</b>		Entidad			Entidad que almacena los tipos de exámenes existentes para un proceso de valoración (Diagnóstico)
	ex_tipo_codigo	<b>PK (int)</b>			Identificador primario único para el tipo de examen.
	ex_tipo_nombre	nchar(50)			Nombre del Examen
	ex_tipo_descripcion	nchar(300)			Descripción del examen
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>



	ex_tipo_paso	int			Identificador único que indica el número de paso del examen dentro del proceso de diagnóstico, de esta manera se puede reanudar en cualquier punto del diagnóstico de un paciente. Parámetro importante en el módulo de orquestación de ventanas y seguimiento del flujo.
<b>paciente</b>		Entidad			Entidad que almacena la información relacionada a la Historia Clínica de un paciente.
	pac_cedula	<b>PK nchar(10)</b>			Identificador único de un paciente en el sistema.
	pac_nombres	nchar(50)			Nombres del paciente
	pac_apellidos	nchar(50)			Apellidos del paciente
	pac_edad	int			Edad del paciente
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>



	pac_fecha_nac	datetime			Fecha de nacimiento del paciente
	pac_raza	bit			Raza del paciente, se evalúa como factor de riesgo la raza negra (raza negra = 1, otra raza = 0)
	pac_glaucoma_familia	bit			Antecedentes de glaucoma familiar, 0=FALSE, 1=TRUE
	pac_hipertension	bit			Presencia de Hipertensión en paciente, 0=FALSE, 1=TRUE
	pac_fecha_ingreso	datetime			Fecha de ingreso de la Historia clínica del paciente en la aplicación.
	pac_miopia_elevada	bit			Presencia de miopía elevada en paciente. 0=FALSE, 1=TRUE
	pac_direccion	nchar(300)			Dirección del paciente
	pac_telefono	nchar(15)			Número telefónico del paciente
	pac_email	nchar(40)			Correo electrónico del paciente
Tabla	Campo	Tipo	Tabla Relacionada	Campo Relacionado	Descripción



	pac_activo	bit			Estado del paciente (Eliminación lógica) activo = 1,inactivo =0
<b>paciente_diagnostico</b>		Entidad			Entidad que almacena un diagnóstico de un paciente, luego de terminado un proceso de diagnóstico
	pac_dx_cod	<b>PK (int)</b>			Identificador único para un diagnóstico creado.
	pac_cedula_FK	<b>FK nchar(10)</b>	paciente	pac_cedula	Identificador externo que asocia a un diagnóstico con un paciente.
	pac_trat_cod_FK	<b>FK (int)</b>	tratamiento	pac_trat_cod	Identificador externo que asocia a un diagnóstico de un paciente con un tratamiento predeterminado (opcional)
	pac_dx_descripcion	nchar(500)			Notas o descripción del diagnóstico
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>



	pac_fecha_dix	datetime			Fecha de ingreso del examen en el sistema.
<b>paciente_gcaa_proceso</b>		Entidad			Entidad que almacena la información relacionada a un proceso de diagnóstico (conjunto de pasos ordenados de forma esquemática para arrojar un resultado)
	pac_gcaa_codigo	<b>PK (int)</b>			Identificador único de un proceso de diagnóstico
	pac_cedula_FK	<b>FK nchar(10)</b>	paciente	pac_cedula	Identificador externo que asocia un proceso de diagnóstico a un paciente.
	pac_gcaa_notas	nchar(255)			Notas del especialista
	pac_proc_terminado	bit			Valor que indica si el proceso ha llegado hasta su finalización, 0=Proceso Activo, 1=Proceso Finalizado.
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>



<b>tratamiento</b>		Entidad			Entidad en la que se almacena información sobre tratamientos habituales (uso opcional)
	pac_trat_cod	<b>PK (int)</b>			Identificador único de un tratamiento.
	pac_trat_nombre	nchar(30)			Nombre del tratamiento
	pac_trat_descripcion	nchar(300)			Descripción del tratamiento
campimetria_imagen		Entidad			Entidad que sirve como puente de comunicación temporal entre MatLab y la aplicación .NET, es una entidad temporal
	camp_img_sens_oi	nchar(30)			Valor en porcentaje de la sensibilidad restante del ojo izquierdo (valor obtenido del procesamiento de imagen desde matlab)
<b>Tabla</b>	<b>Campo</b>	<b>Tipo</b>	<b>Tabla Relacionada</b>	<b>Campo Relacionado</b>	<b>Descripción</b>
	camp_img_sens_od	nchar(30)			Valor en porcentaje de la



					sensibilidad restante del ojo derecho (valor obtenido del procesamiento de imagen desde matlab)
	camp_img_activo	bit			Indica si el parámetro ya fue usado, en cada iteración, el contenido de la tabla es eliminado, emulando una entidad virtual que solo disponible en la ejecución del sistema.



### 3.3 Análisis y Diseño de la Aplicación

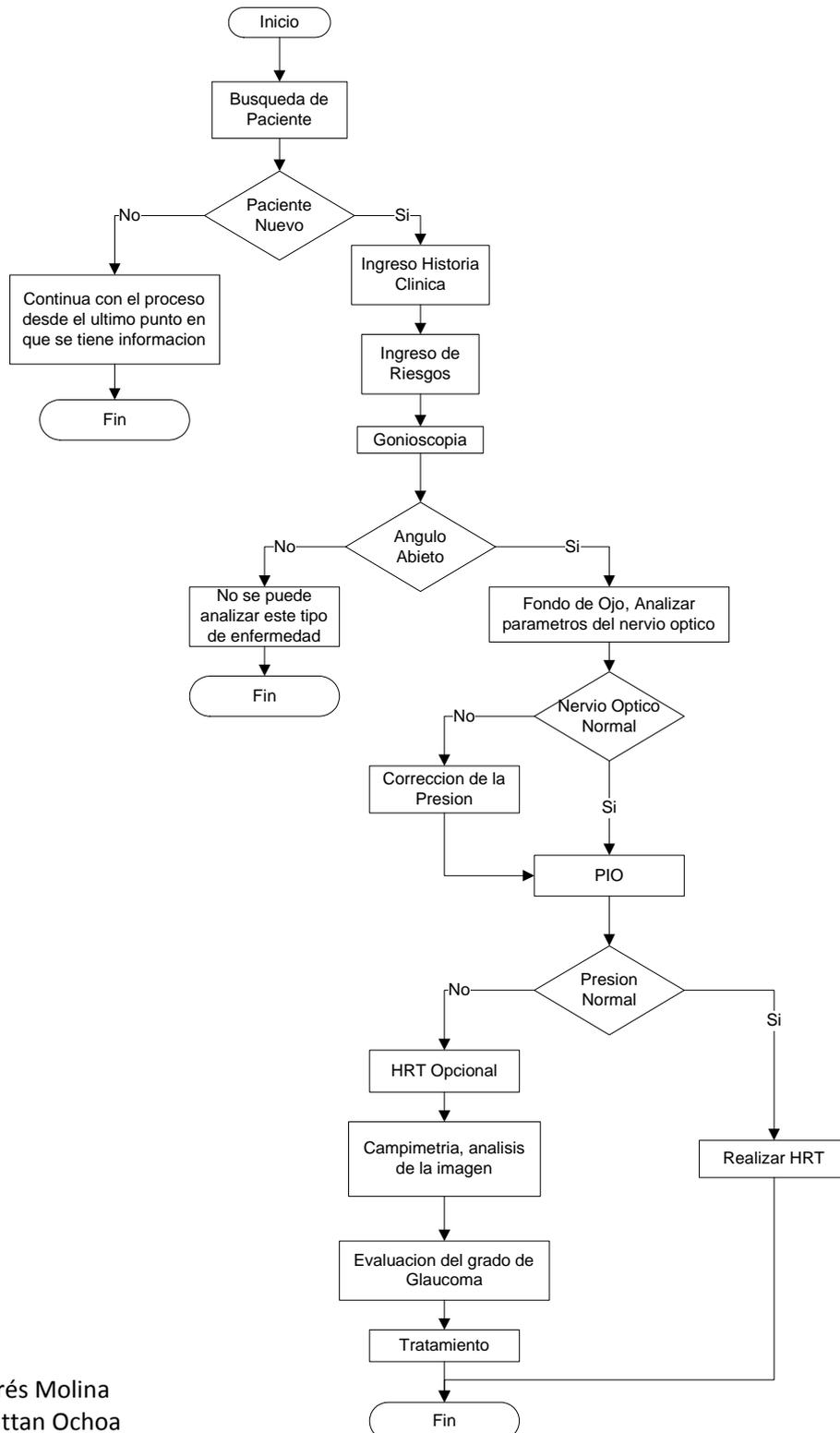
El Sistema Experto Para el Apoyo en el Diagnóstico de Glaucoma Crónico de Ángulo Abierto, está orientado al ambiente Web, está dirigido para el siguiente usuario:

**Especialista:** Posee todos los privilegios para manejo del sistema, es decir, realiza los mantenimientos necesarios de cada examen y paciente; además realiza todo el flujo necesario para llevar a cabo un diagnóstico de un paciente y los análisis de las imágenes.



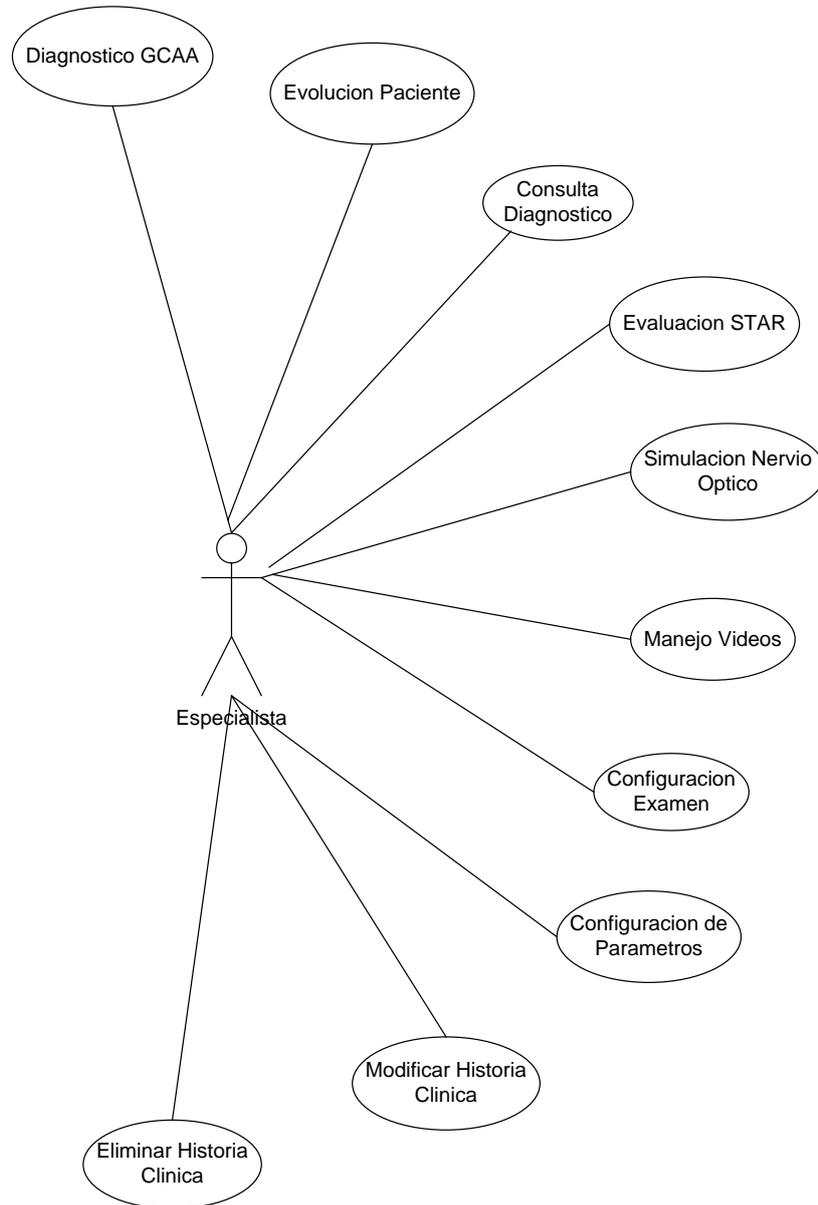
### 3.4 Análisis y Diseño de la Aplicación

#### 3.4.1 Flujo de Proceso de Diagnostico





### 3.4.2 Modelo de Casos de Uso





## Modelación de Casos de Uso

### Caso de Uso: Diagnostico GCAA

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Diagnostico”, dentro del menú Diagnostico GCAA.

<b>Caso de uso:</b> Diagnostico GCAA
<b>Actores:</b> Especialista
<b>Precondiciones:</b>
<b>Flujo de Eventos:</b> <ol style="list-style-type: none"><li>1. El especialista escoge la opción “Diagnostico”.</li><li>2. Se ingresa la historia clínica del paciente.</li><li>3. Se ingresan los factores de riesgo del paciente.</li><li>4. Se determina si el Glaucoma es del tipo Crónico de Angulo Abierto.</li><li>5. El especialista determina el estado del nervio óptico.<ol style="list-style-type: none"><li>5.1 En caso de ser normal no se realiza ningún proceso</li><li>5.2 Si presenta alguna anomalía es necesario realizar la corrección de la presión Intraocular.</li></ol></li><li>6. Realiza el análisis de la PIO<ol style="list-style-type: none"><li>6.1 En el caso de ser normal se debe realizar un HRT.</li><li>6.2 En el caso de presentar anomalías el HRT es opcional.</li></ol></li><li>7. Se procede con el procesamiento de las imágenes de la Campimetría.</li><li>8. Evaluación del grado del Glaucoma.</li><li>9. Se determina el tratamiento.</li></ol>
<b>Pos condiciones:</b>



1. Si en los tratamientos en la revisión que se presenta luego de seis meses no se presenta ningún cambio se debe cambiar de tratamiento.

**Caso de Uso:** Evolución Paciente

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Evolución Paciente”, dentro del menú Diagnostico GCAA.

<b>Caso de uso:</b> Evolución Paciente
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se debió realizar al menos un diagnostico previo.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción “Evolución Paciente”. 2. Se ingresa un dato que sirva de parámetro de búsqueda del paciente. 3. El sistema visualiza la información de los diagnósticos anteriores del paciente. 4. El especialista puede comparar los resultados de los diagnósticos realizados anteriormente.
<b>Pos condiciones:</b>

**Caso de Uso:** Consulta Diagnostico

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Consulta Diagnostico”, dentro del menú Diagnostico GCAA.

**Caso de uso:** Consulta Diagnostico



<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se debió realizar al menos un diagnostico previo.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción "Consulta Diagnostico". 2. Se ingresa un dato que sirva de parámetro de búsqueda del paciente. 3. El sistema visualiza la información de los diagnósticos anteriores del paciente. 4. El especialista puede visualizar el resultado de un diagnóstico específico realizado anteriormente.
<b>Pos condiciones:</b>

**Caso de Uso:** Evaluación STAR

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción "Evaluación STAR", dentro del menú STAR.

<b>Caso de uso:</b> Evaluación STAR
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se necesita información del paciente sobre su PIO, PSD, GCC, C/D, Diabetes.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción "Evaluación STAR". 2. Se ingresa los datos del paciente correspondiente a: edad, PIO, PSD, GCC, C/D, Diabetes. 3. El sistema se basa en el algoritmo STAR para el cálculo de la probabilidad del riesgo de glaucoma a 5 años.



4. Se presentan los datos obtenidos por el sistema.
<b>Pos condiciones:</b>

**Caso de Uso:** Simulación del Nervio Óptico

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Nervio Óptico”, dentro del menú Multimedia.

<b>Caso de uso:</b> Simulación del Nervio Óptico
<b>Actores:</b> Especialista
<b>Precondiciones:</b>
<b>Flujo de Eventos:</b>  <ol style="list-style-type: none"><li>1. El especialista escoge la opción “Nervio Óptico”.</li><li>2. Se ingresa los datos del nervio óptico.</li><li>3. El sistema se presenta una animación correspondiente a los datos ingresados.</li><li>4. En caso de ser necesario se pueden modificar los parámetros del nervio para cuestiones didácticas.</li></ol>
<b>Pos condiciones:</b>

**Caso de Uso:** Manejo Videos

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Videos Ilustrativos”, dentro del menú Multimedia.

<b>Caso de uso:</b> Manejo Videos
<b>Actores:</b> Especialista
<b>Precondiciones:</b>
<b>Flujo de Eventos:</b>



1. El especialista escoge la opción "Videos Ilustrativos".
2. El sistema presenta los videos disponibles.
3. Al Seleccionar un video del listado, este es reproducido dentro del programa.

**Pos condiciones:**

1. Al momento de cerrar el video se puede continuar con el manejo normal de la aplicación.

**Caso de Uso:** Configuración Examen

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción "Configuración Examen", dentro del menú Configuración Sistema Experto.

<b>Caso de uso:</b> Configuración Examen
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  <ol style="list-style-type: none"><li>1. Se necesita que se haya ingresado al menos un examen en la aplicación.</li></ol>
<b>Flujo de Eventos:</b>  <ol style="list-style-type: none"><li>1. El especialista escoge la opción "Configuración de Examen".</li><li>2. El sistema presenta los exámenes disponibles.</li><li>3. De la lista de exámenes el especialista selecciona uno de ellos para su modificación.</li><li>4. El sistema guarda las modificaciones realizadas.</li></ol>
<b>Pos condiciones:</b>  <ol style="list-style-type: none"><li>1. Al realizar un nuevo examen se visualizarán los cambios realizados en los</li></ol>



exámenes.

#### **Caso de Uso:** Configuración Parámetros

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Configuración Parámetros”, dentro del menú Configuración Sistema Experto.

<b>Caso de uso:</b> Configuración Parámetros
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se necesita que se haya ingresado al menos un examen en la aplicación.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción “Configuración de Parámetros”. 2. El sistema presenta los parámetros de los exámenes disponibles. 3. De la lista de parámetros el especialista selecciona uno de ellos para su modificación. 4. El sistema guarda las modificaciones realizadas.
<b>Pos condiciones:</b>  1. Al realizar un nuevo examen se visualizarán los cambios realizados en los parámetros.

#### **Caso de Uso:** Modificar Historia Clínica

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Modificar Historia Clínica”, dentro del menú Pacientes.



<b>Caso de uso:</b> Modificar Historia Clínica
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se necesita que se haya ingresado al menos un paciente en la aplicación.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción “Modificar Historia Clínica”. 2. El sistema solicita la identificación del paciente para la búsqueda. 3. Se presenta la información correspondiente al paciente solicitado. 4. El especialista puede modificar la información actual del paciente. 5. El sistema guarda las modificaciones a la historia clínica del paciente.
<b>Pos condiciones:</b>  1. Al realizar un nuevo examen se visualizarán los cambios realizados en el paciente.

**Caso de Uso:** Eliminar Historia Clínica

**Descripción:** Este Caso de Uso se inicia cuando se elige la opción “Eliminar Historia Clínica”, dentro del menú Pacientes.

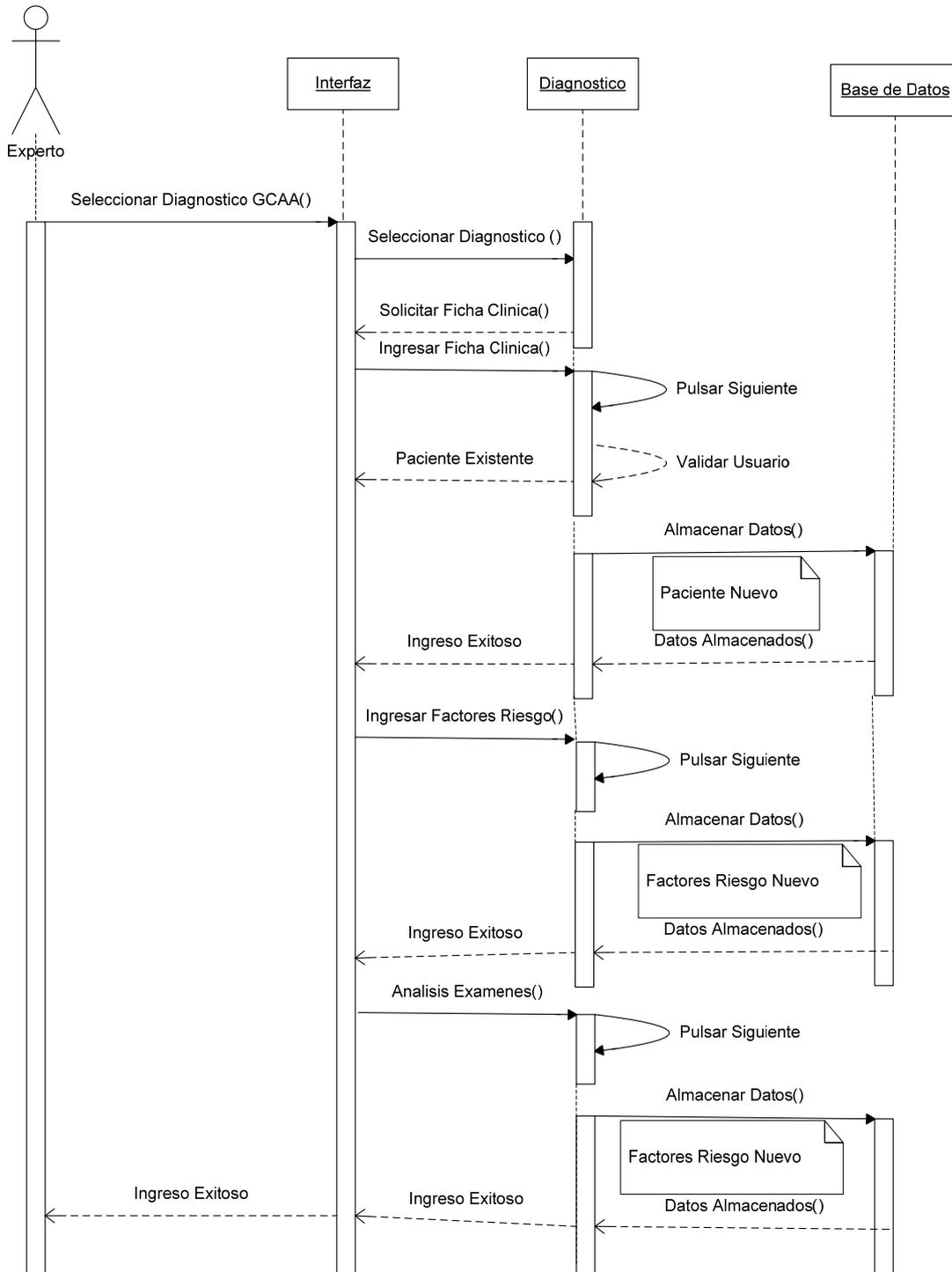
<b>Caso de uso:</b> Eliminar Historia Clínica
<b>Actores:</b> Especialista
<b>Precondiciones:</b>  1. Se necesita que se haya ingresado al menos un paciente en la aplicación.
<b>Flujo de Eventos:</b>  1. El especialista escoge la opción “Eliminar Historia Clínica”. 2. El sistema solicita la identificación del paciente para la búsqueda. 3. Se presenta la información correspondiente al paciente solicitado. 4. El especialista pulsa el botón “Eliminar Historia Clínica”. 5. El sistema elimina la historia clínica del paciente.
<b>Pos condiciones:</b>





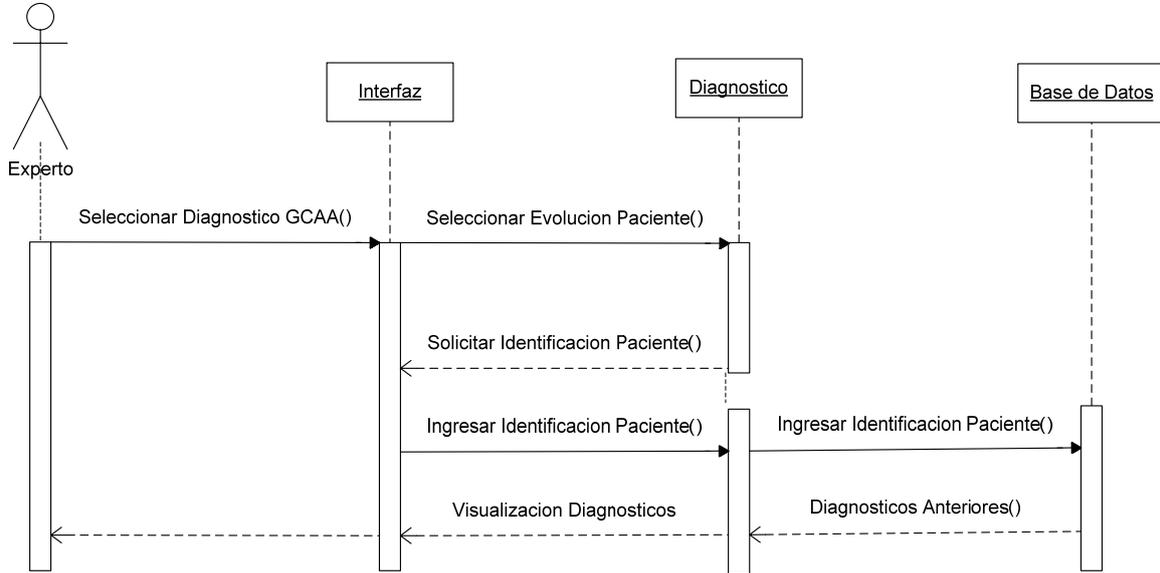
### 3.4.3 Diagramas de Secuencia

#### Diagnostico

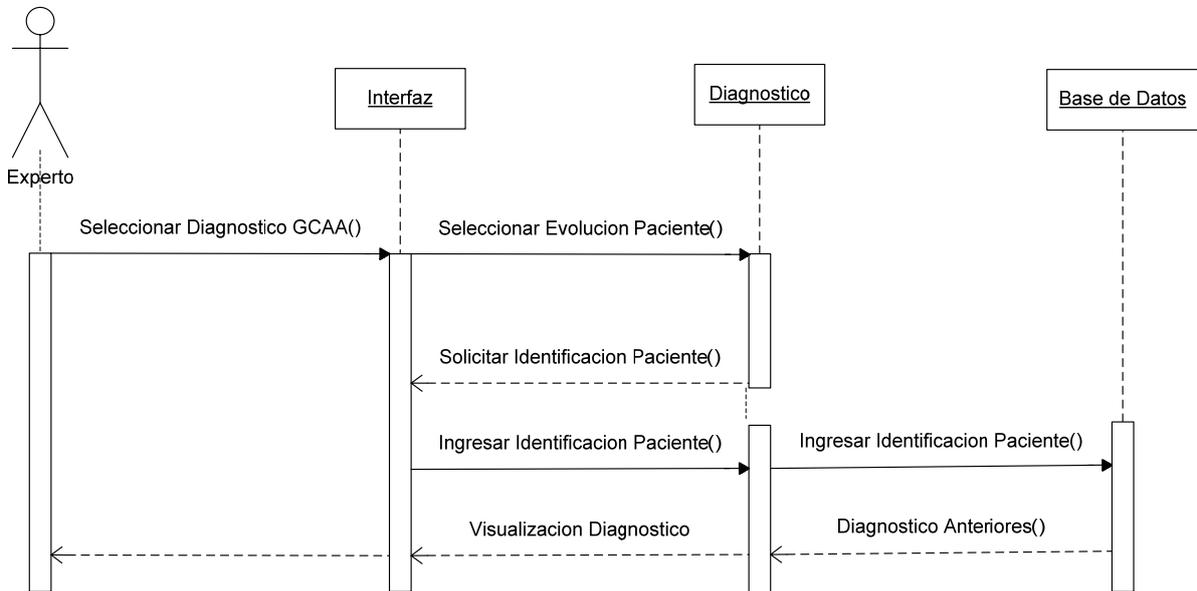




### Evolución Paciente

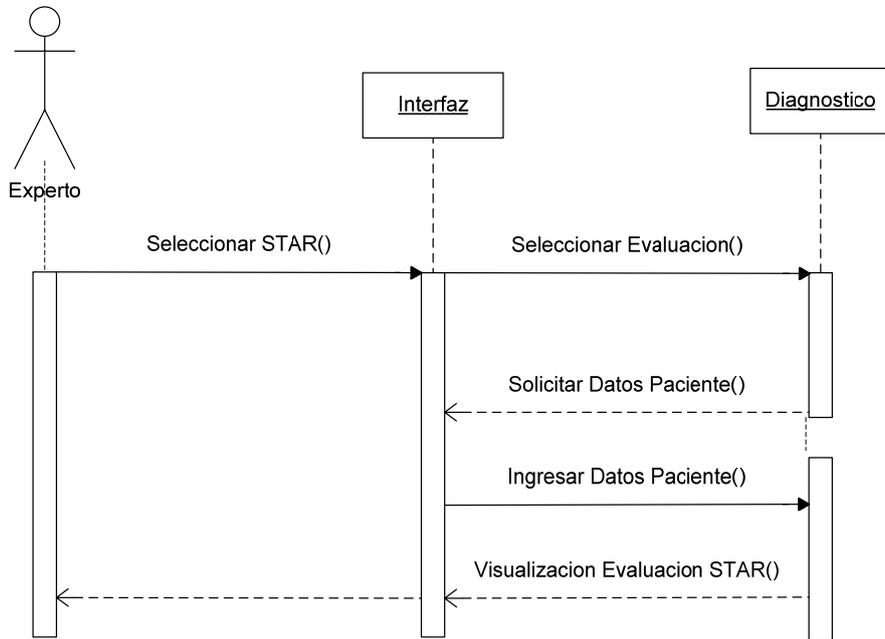


### Consultar Diagnostico

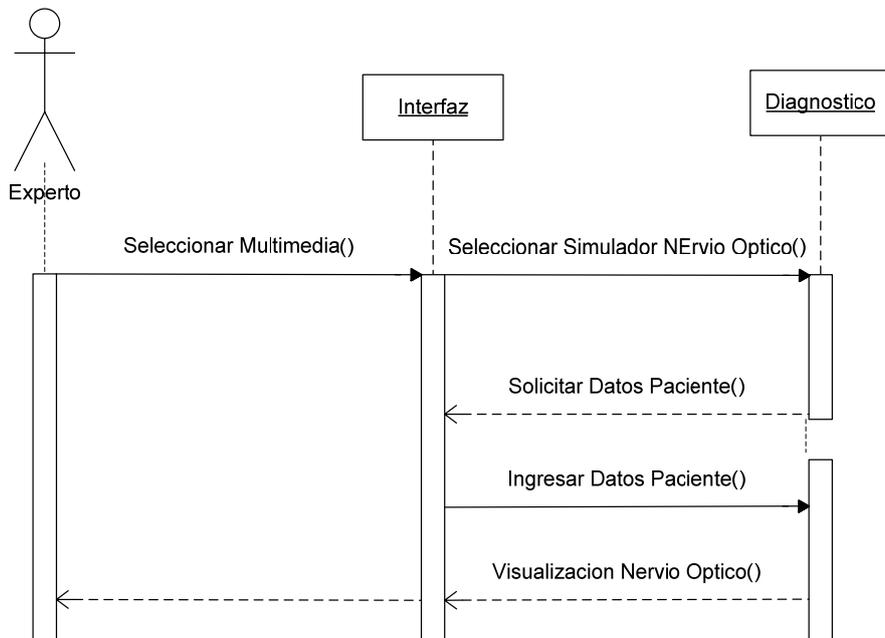




### Evaluación STAR

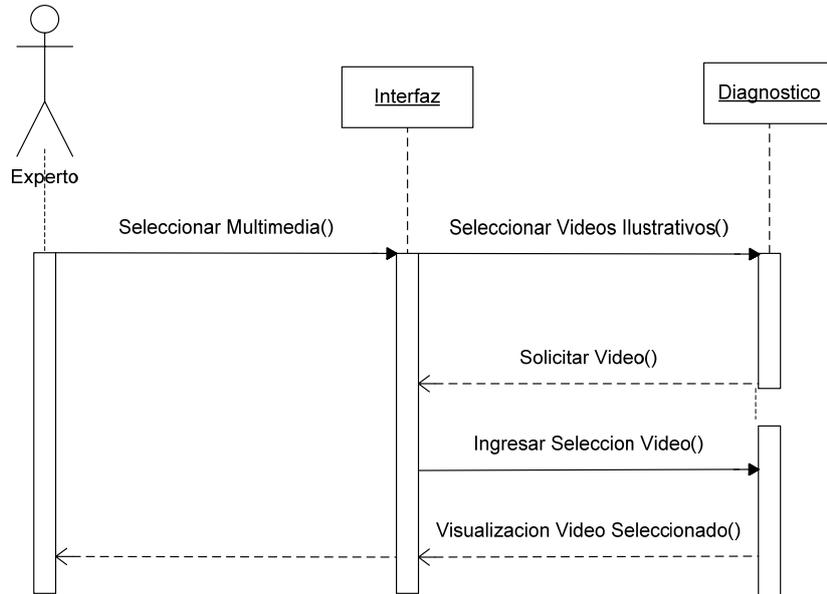


### Simulación del Nervio Óptico

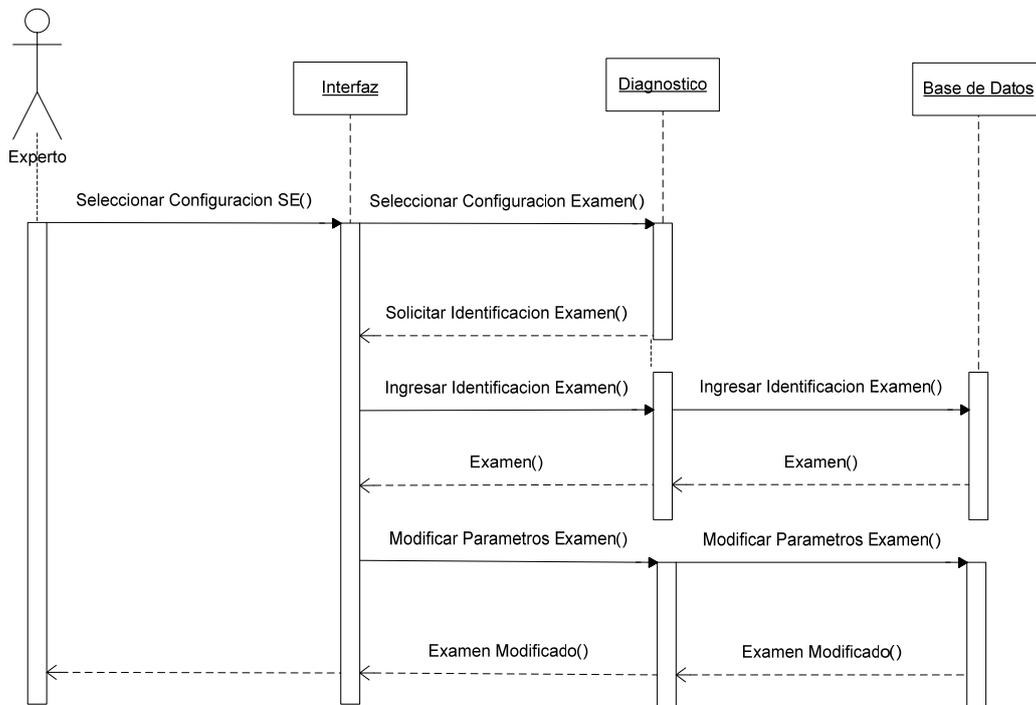




### Manejo Videos

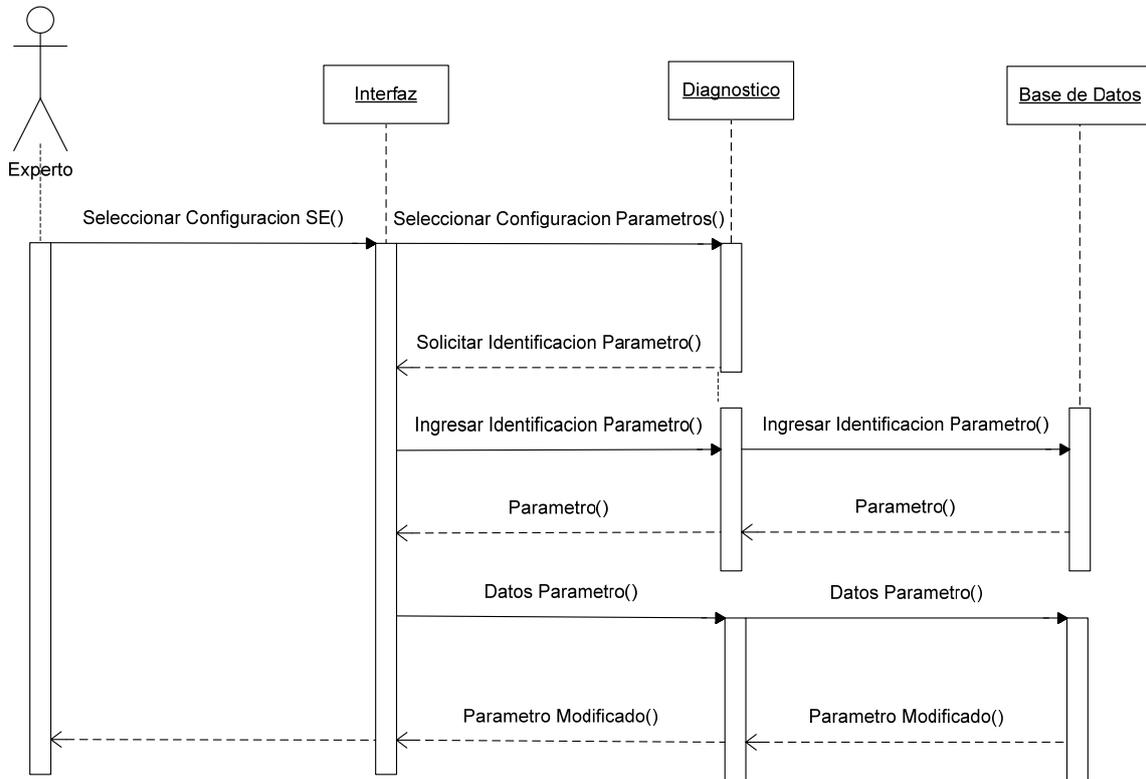


### Configuración Examen



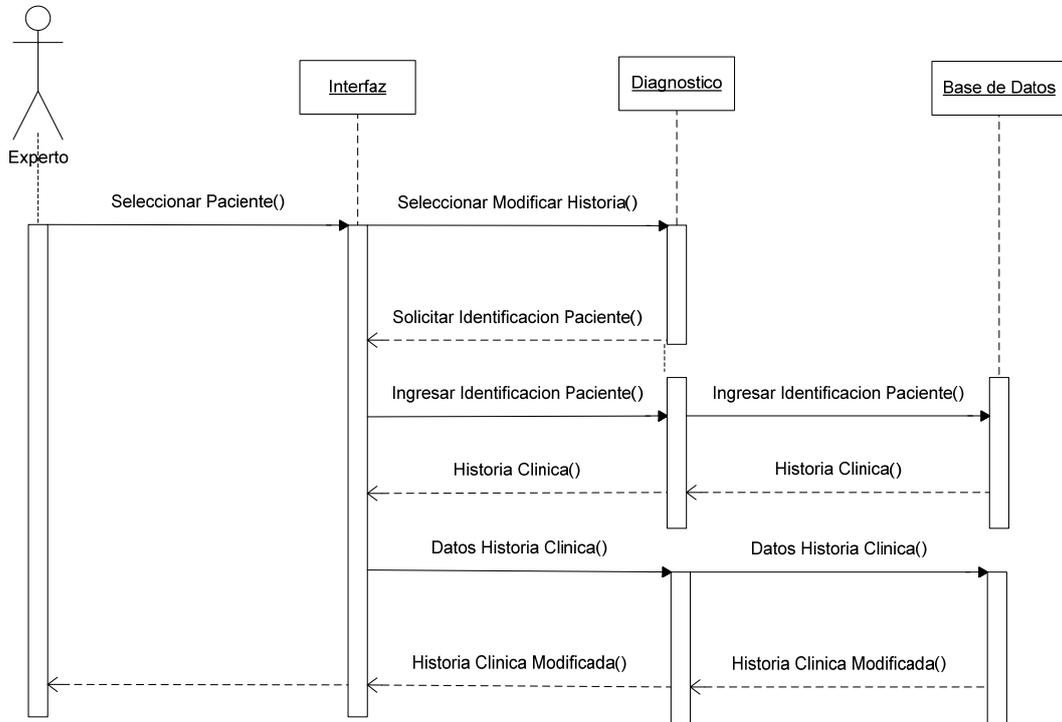


### Configuración Parámetros



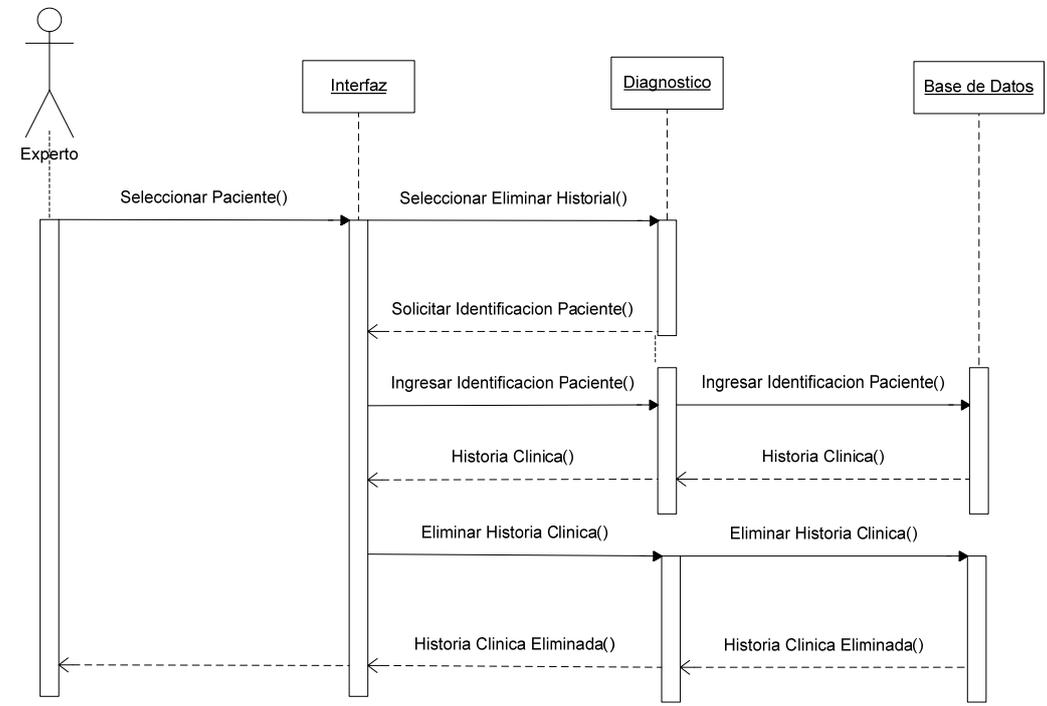


### Modificar Historia Clínica





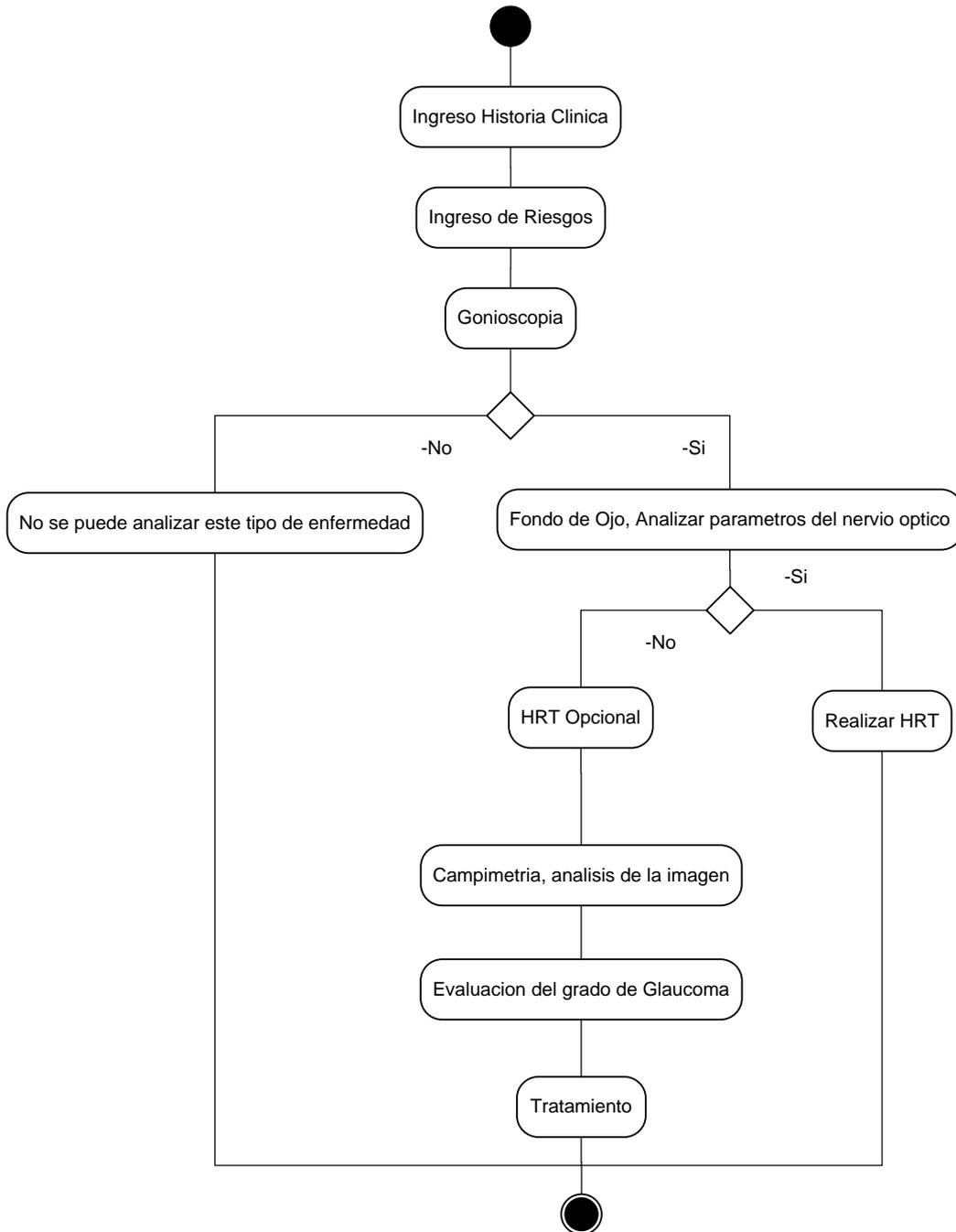
## Eliminar Historia Clínica





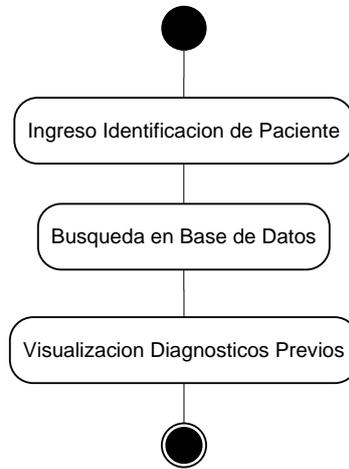
### 3.4.4 Diagramas de Actividad

#### Diagnostico GCAA

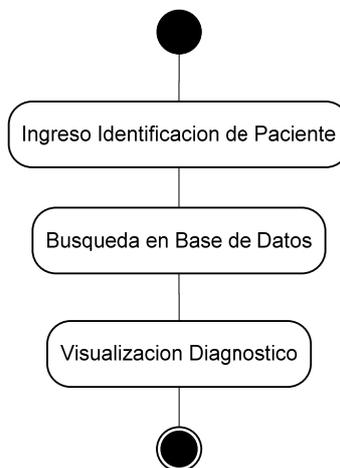




### Evolución Paciente

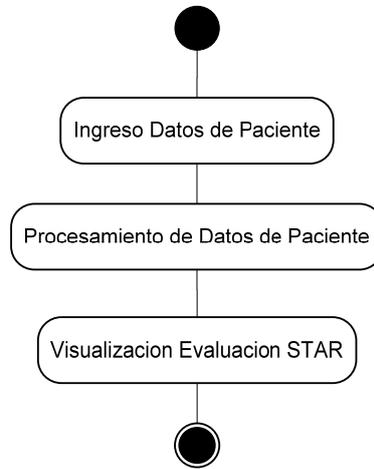


### Consulta Diagnósticos

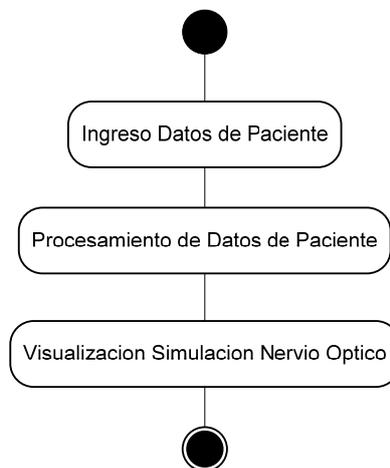




### Evaluación STAR

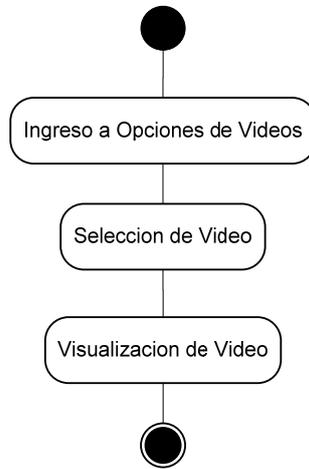


### Simulación Nervio Óptico

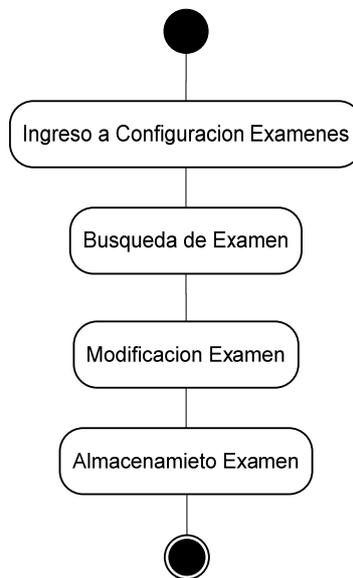




## Manejo de Videos

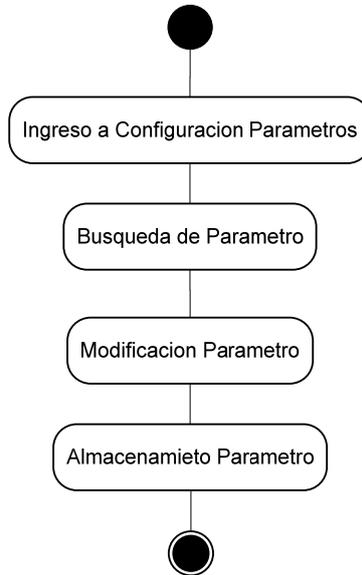


## Configuración Examen

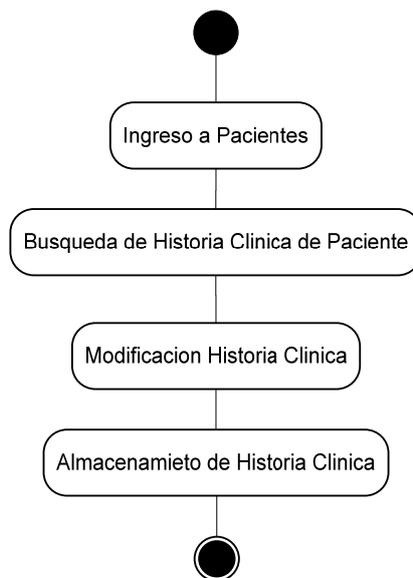




### Configuración Parámetros

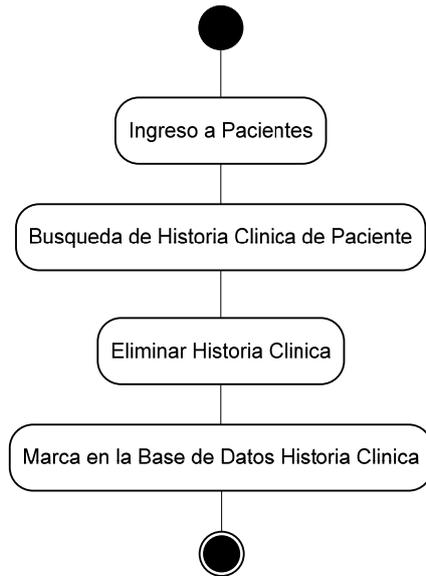


### Modificar Historia Clínica





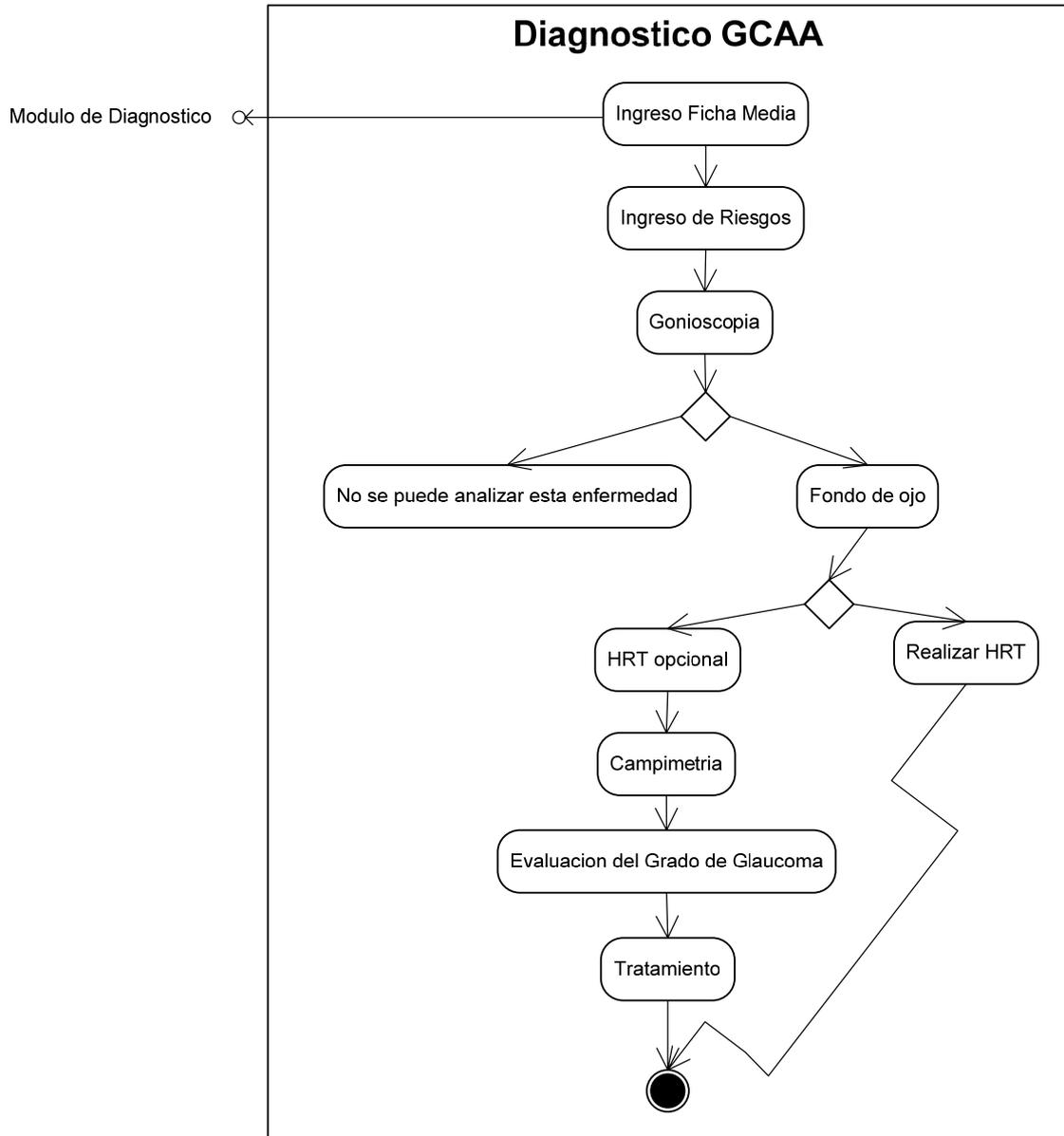
## Eliminar Historia Clínica





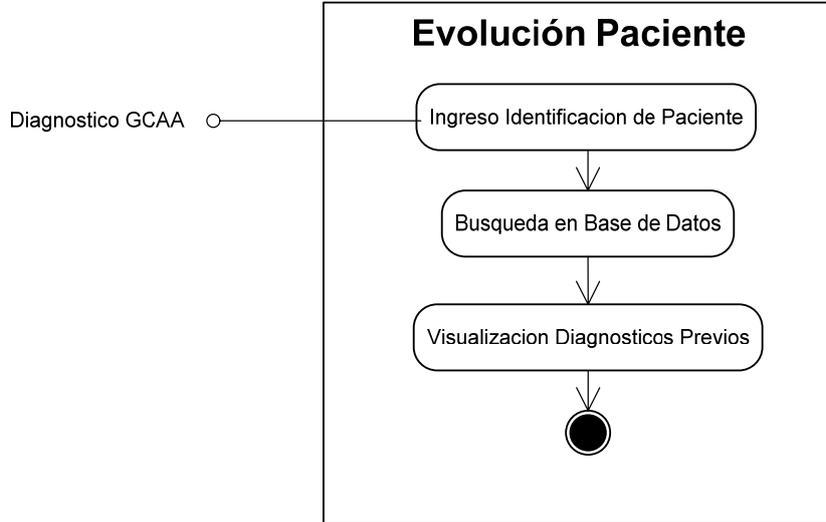
### 3.4.5 Diagramas de Estado

#### Diagnostico GCAA

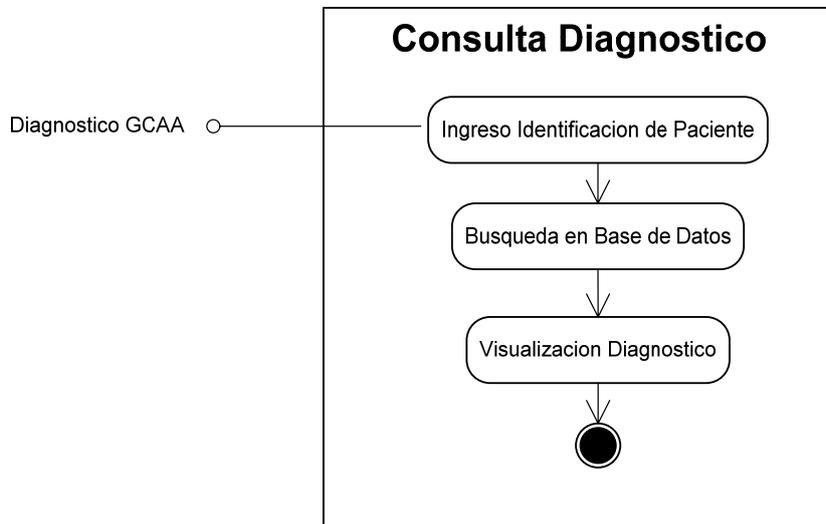




### Evolución Paciente

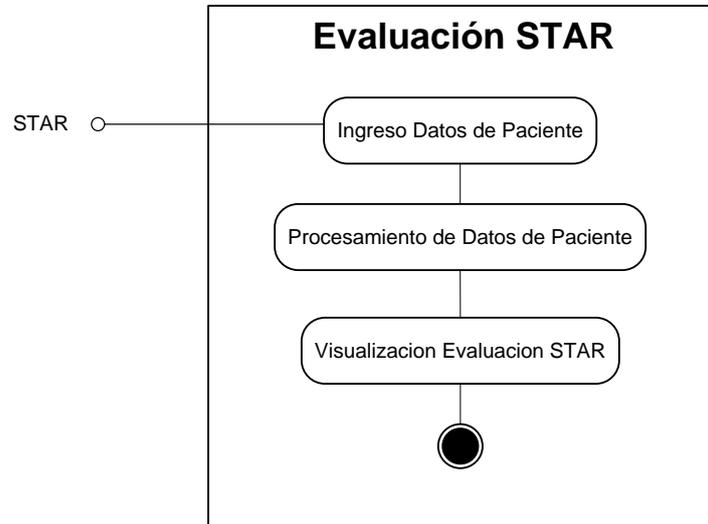


### Consulta Diagnostico

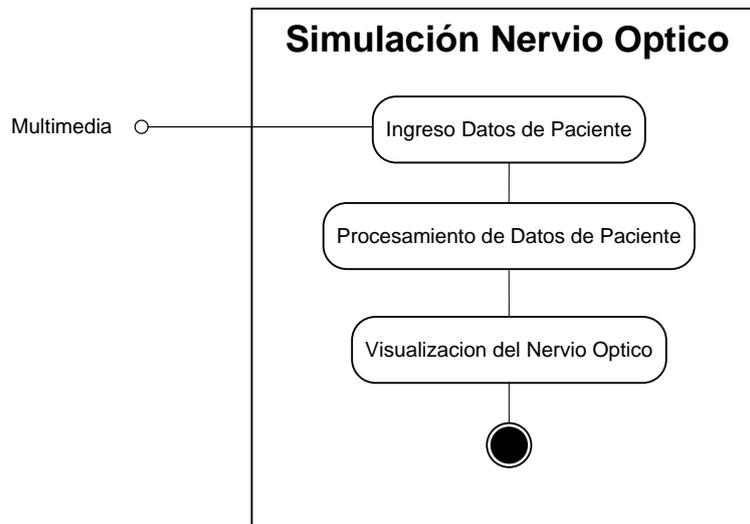




### Evaluación STAR

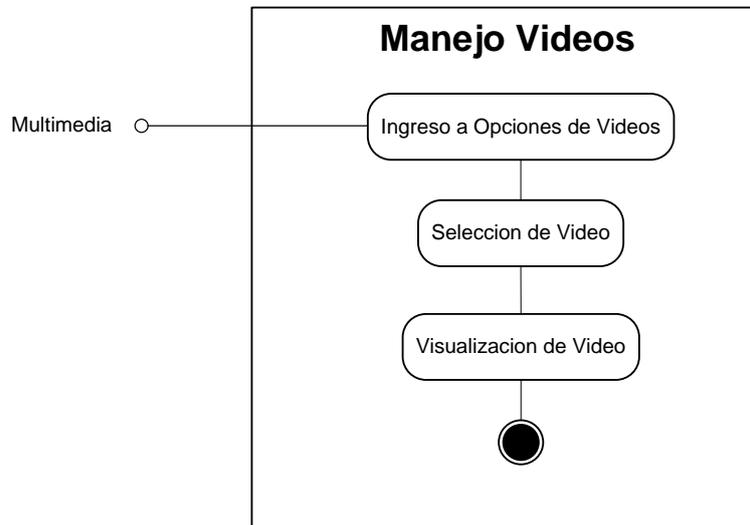


### Simulación Nervio Óptico

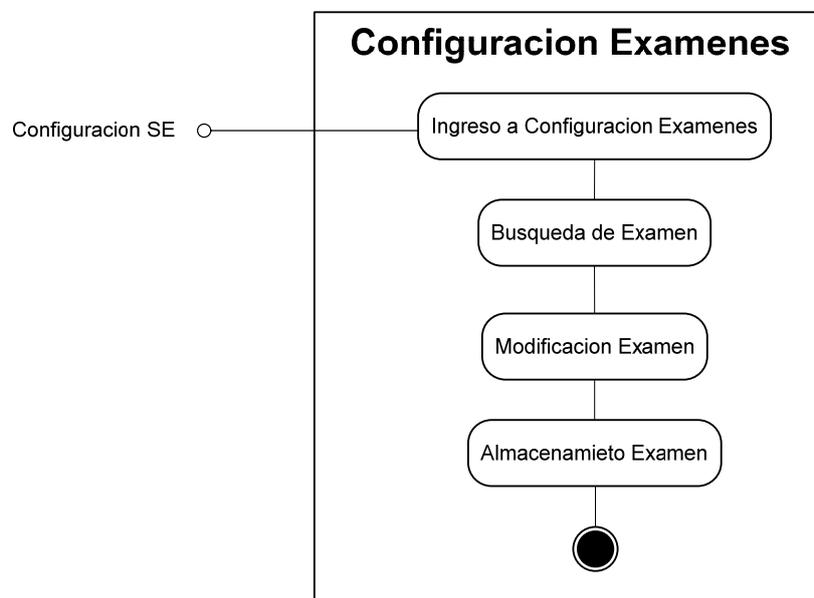




## Manejo de Videos

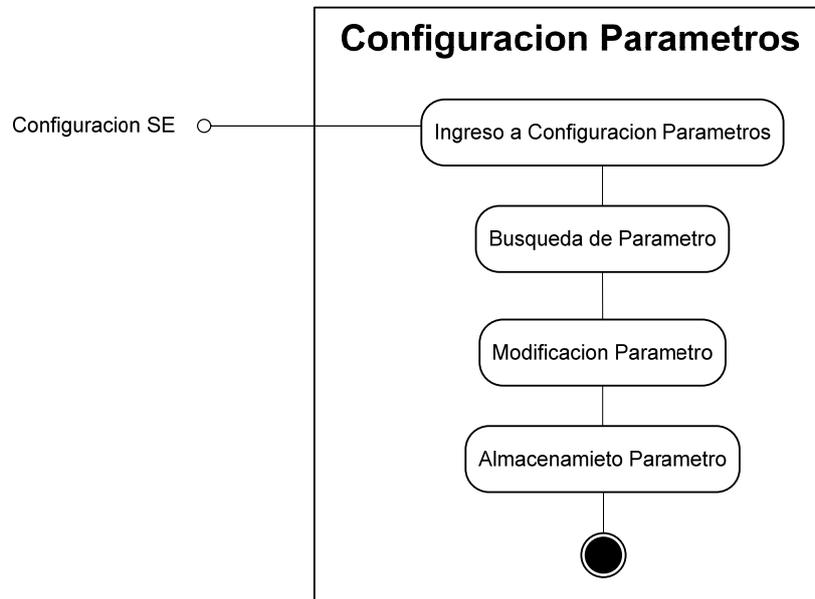


## Configuración Examen

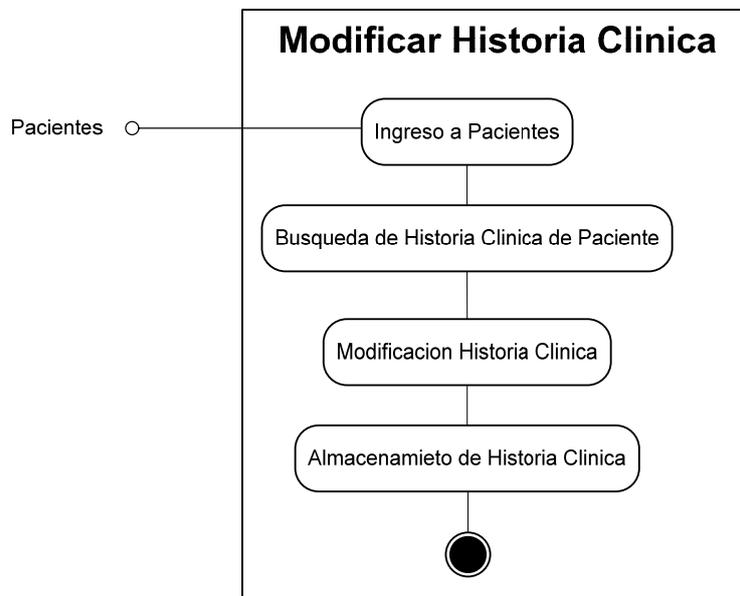




## Configuración Parámetros

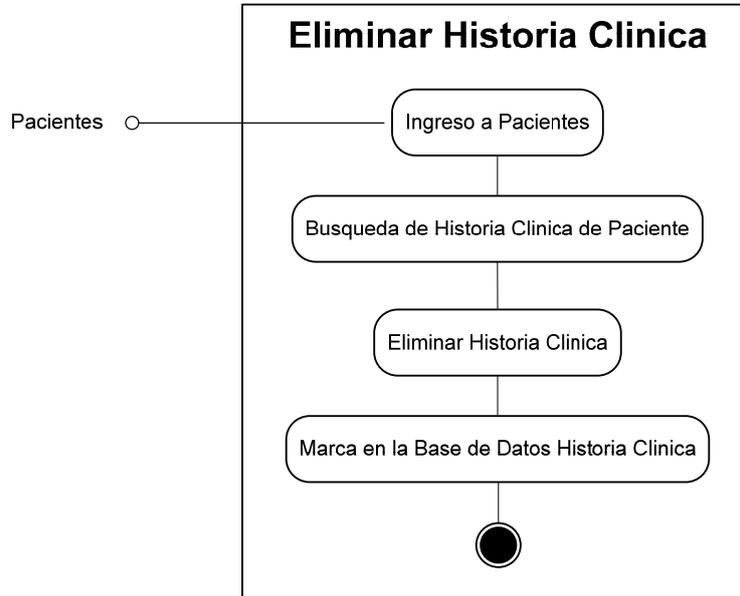


## Modificar Historia Clínica





## Eliminar Historia Clínica





## **CAPITULO 4**



## 4.1 Implementación de la Aplicación

Luego del diseño realizado en la fase anterior, se procedió a la implementación del sistema, como se había establecido en un principio, se optó por utilizar la arquitectura de 3 capas y de esta manera tenemos:

Capa de interfaz: Son todas la interfaces del sistema

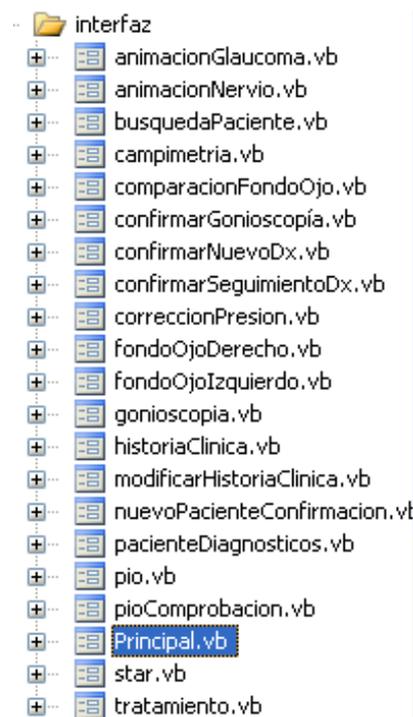


Figura 4.1 Interfaces del sistema



Capa de negocio: Se realizan todas las operaciones del sistema

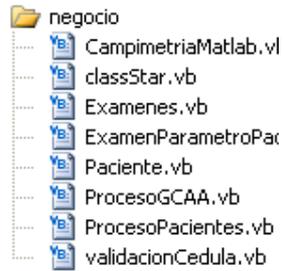


Figura 4.2 Capa de negocio

Capa de Datos: Contiene toda la información para la conexión hacia la base de datos, además del control de excepciones, dentro la el manejador de base de datos también se encuentran procedimientos almacenados con el objetivo de ayudar al mantenimiento de la base de datos



Figura 4.3 Capa de datos Visual Basic .Net

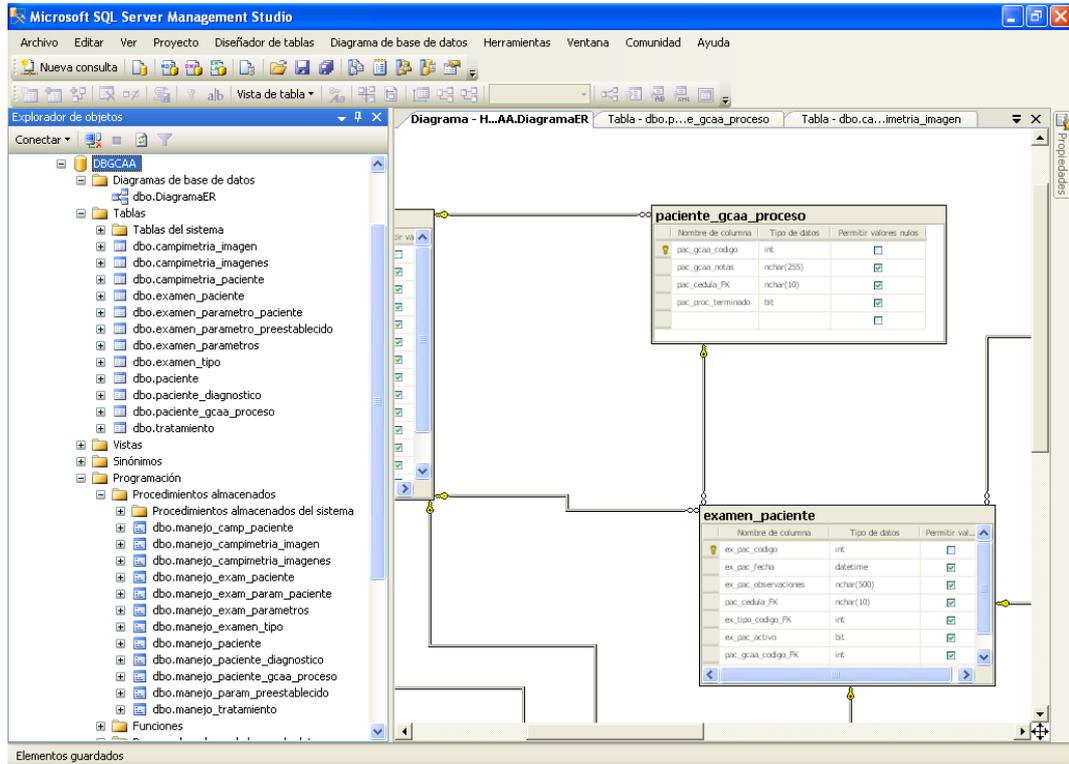


Figura 4.4 Imagen de la base de datos implementada en el sistema



## 4.2 Implementación de Procesamiento de Imágenes

Para el procesamiento de imágenes se deben considerar varios aspectos con respecto al manejo de estas, por ejemplo Matlab almacena la mayoría de las imágenes como arreglos bidimensionales (matrices) en los cuales cada elemento de la matriz corresponde a la intensidad de un píxel de la imagen. Algunas imágenes, como las imágenes a color (RGB), requieren de un arreglo tridimensional, donde en el primer plano en el espacio tridimensional representa la intensidad de rojo de los píxeles, el segundo plano representa la intensidad de verde de los píxeles y el tercer plano representa la intensidad de azul de los píxeles. Para reducir el espacio en memoria requerido para almacenar imágenes, Matlab almacena los datos en arreglos de 8 o 16 bits sin signo, clases uint8 y uint16, respectivamente.

### 4.2.1 Tipos de imágenes en Matlab

El toolbox de Procesamiento de Imágenes maneja cuatro tipos de imágenes básicas: imágenes indexadas, imágenes con intensidad (escala de grises), imágenes binarias e imágenes RGB.

### 4.2.2 Lectura de imágenes en Matlab

El comando *imread* lee una imagen desde un archivo gráfico. Si la imagen es en escala de grises, entonces devuelve una matriz bidimensional. Si la imagen es RGB, entonces devuelve un arreglo tridimensional. Su sintaxis es:

```
I = imread("filename")
```

En Matlab se soportan los siguientes formatos de imagen: JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO y CUR.

### 4.2.3 Conversiones entre tipos de imágenes

Para ciertas operaciones es necesario convertir una imagen de su tipo original a otro tipo de imagen que facilite su procesamiento. En la Tabla 3.1 se presentan algunos comandos usados para la conversión entre tipos de imágenes.



Comando	Descripción
gray2ind	Crea una imagen indexada a partir de una imagen de intensidad en escala de gris.
im2bw	Crea una imagen binaria a partir de una imagen de intensidad, imagen indexada o RGB basado en un umbral de luminancia.
ind2rgb	Crea una imagen RGB a partir de una imagen indexada
rgb2gray	Crea una imagen de intensidad en escala de gris a partir de una imagen RGB
rgb2ind	Crea una imagen indexada a partir de una imagen RGB

Tabla 3.1 Comandos de conversión de imágenes en Matlab

#### 4.2.4 Selección de una sección de una imagen en Matlab

Para tomar una sección de cualquier imagen se utiliza el comando `imcrop`. Su sintaxis es: `I2 = imcrop(I)`. Para seleccionar la región que se va a cortar, simplemente arrastre el ratón y forme un rectángulo sobre la región deseada. Cuando se suelta el botón del ratón, el comando regresa la sección seleccionada al argumento de salida especificado (`I2` en este caso).

#### 4.2.5 Determinar el tamaño de una imagen

Para determinar el tamaño de una imagen podemos usar el comando `size` de matlab, que devuelve el número de filas, columnas y planos de colores.

```
» size(I)
```

```
ans =
```

```
86 96 3
```



Si queremos almacenar el resultado en variables separadas, sería:

```
» [M, N] = size(I)
```

Si queremos obtener información más detallada de la imagen usamos el comando whos:

```
» whos I
```

Name	Size	Bytes	Class
I	86x96x3	24768	uint8 array

Grand total is 24768 elements using 24768 bytes

## 4.2.6 Transformaciones Geométricas

Las transformaciones geométricas modifican las relaciones espaciales entre píxeles. A continuación se presentan algunas de ellas:

### 4.2.6.1 Interpolación

La interpolación es el proceso en el cual se estiman los valores de una imagen en una sección específica, por ejemplo, cuando se amplía una imagen, en la nueva imagen existen más píxeles que en la imagen original.

Dentro de Matlab los comandos `imresize` e `imrotate` utilizan interpolación bidimensional como paso intermedio en sus procesos e implementa los siguientes métodos de interpolación:

- Vecino más próximo (nearest): al píxel interpolado se le asigna el valor del píxel que corresponde. Es el método por defecto si no se especifica alguno.
- Interpolación bilineal (bilinear): el valor del píxel interpolado es el promedio ponderado de los píxeles en la vecindad 2x2 más cercana.
- Interpolación bicúbica (bicubic): el valor del píxel interpolado es el promedio ponderado de los píxeles presentes en la vecindad 4x4 más cercana.



Para imágenes RGB, la interpolación se ejecuta en los planos de color rojo, verde y azul de forma individual.

#### 4.2.6.2 Amplificación y Reducción de imágenes

Para el cambio de tamaño de una imagen (sea amplificación o reducción) se utiliza el comando *imresize*. Este comando permite especificar: el tamaño de la imagen de salida (procesada), el método de interpolación utilizado y el filtro a usar para evitar el *efecto alias*. El efecto alias se presenta al reducir el tamaño de una imagen debido a que se presenta una pérdida de información.

#### 4.2.6.3 Correlación de Matrices

La correlación es una operación en la cual el valor de un píxel de salida se calcula como la suma ponderada de los píxeles vecinos. La correlación se utiliza para encontrar el parecido entre píxeles de una imagen. Si los píxeles son iguales o parecidos, se dice que están altamente correlacionados entre sí. La correlación permite hallar patrones, y se utiliza el comando *corr2*, el cual calcula el coeficiente de correlación entre dos matrices del mismo tamaño.

El coeficiente entre dos matrices es un número real comprendido entre el rango [-1 y 1] y se dice que las matrices están altamente correlacionadas si el coeficiente tiende a estos límites (-1 o 1); y una baja correlación, si tienden a cero.

### 4.3 Detección de Contornos

La detección de contornos es un paso intermedio en el reconocimiento de patrones en imágenes digitales. En una imagen, los contornos corresponden a los límites de los objetos presentes en la imagen. Para hallar los contornos se buscan los lugares en la imagen en los que la intensidad del píxel cambia rápidamente. La detección de contornos se encuentra implementada en Matlab en el comando *edge* y se aplica a imágenes de intensidad. Su sintaxis es: `BW = edge(I, método)` Esta función devuelve una imagen de bordes binaria, y se puede obtener con diversos métodos, como: *sobel*, *prewitt*, *robert*, *canny*.



## 4.4 Elementos Estructurantes

Examinar la estructura geométrica de una imagen usando como sonda un patrón de ajuste que se denomina elemento estructurante (SE.). El SE puede tener cualquier tamaño y forma (horizontal, vertical, cuadrado, circular, etc.).

En Matlab se encuentra implementado en el comando ***strel***, el cual crea un elemento de estructura morfológica. Su sintaxis se muestra a continuación:

SE = *strel*(forma, parámetros)

## 4.5 Operadores Morfológicos

### 4.5.1 Dilatación

La dilatación expande los píxeles de la imagen sobre la que se aplica.

### 4.5.2 Erosión

La erosión adelgaza la imagen sobre la que se aplica siendo, en un sentido no estricto, opuesta a la dilatación.

### 4.5.3 Apertura (Opening)

Erosión seguida de una dilatación. Elimina pequeños píxeles aislados que haya en la imagen.

### 4.5.4 Cierre (Closing)

Dilatación seguida de una erosión. Rellena los pequeños agujeros que existan en la imagen. La Apertura/cierre elimina picos positivos/negativos más estrechos que el elemento estructurante.

En Matlab se encuentran implementados en los siguientes comandos: *imdilate*, *imerode*, *imclose*, *imopen* respectivamente para crear un elemento estructurante, dilatar, erosionar, cierre y apertura.



## 4.6 SEGMENTACIÓN

La segmentación es un proceso que consiste en dividir una imagen digital en regiones homogéneas o similares con respecto a una o más características (como por ejemplo el brillo, el color, tamaño, longitud, forma) con el fin de facilitar su posterior análisis y reconocimiento automático. Es una de las áreas más importantes y complejas de la visión artificial, la cual ha sido estudiada extensamente y continúa siendo tema de discusión.

### 4.6.1 Segmentación basada en Umbralizado

La umbralización es un proceso que permite convertir una imagen de niveles de gris o en color en una imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto de los píxeles del fondo.

La umbralización es una técnica de segmentación rápida, que tiene un coste computacional bajo y que incluso puede ser realizada en tiempo real durante la captura de la imagen usando un computador.

### 4.6.2 Segmentación por Regiones

La segmentación por regiones es utilizada para separar los objetos de interés. En este caso, la imagen es particionada en diferentes regiones, quedándose cada una relacionada en ciertas características y conjuntos de píxeles conectados. Así, a partir de la segmentación de la imagen en regiones, pueden ser realizadas las medidas sobre cada región y las relaciones entre las regiones adyacentes pueden ser establecidas.

La segmentación por regiones está implementada en Matlab básicamente a través de los comandos `bwlabel` y `Regionprops`.

**Bwlabel** etiqueta los componentes conectados en una imagen binaria. Su sintaxis es:

```
L = bwlabel(bw, n)
```



Donde retorna una matriz L del mismo tamaño de la imagen bw, que contiene etiquetas para los objetos conectados en bw. El valor de n puede ser 4 u 8, donde especifica objetos 4-conectados y 8-conectados respectivamente. Si el argumento n se omite, el valor es 8.

**Regionprops** mide las propiedades de las regiones de una imagen. Su sintaxis es:

```
STATS = regionprops (L, propiedades)
```

Devuelve la medida de un grupo de propiedades para cada región etiquetada en la matriz L.

### 3.11 MÉTODOS DE CLASIFICACIÓN DE PATRONES

Existen varios métodos de los clasificadores de patrones, estos se pueden ordenar atendiendo a diferentes criterios como: la forma de construirse, el tipo de muestra, la información disponible. Se muestra una posible división de los algoritmos de clasificación:

Adaptación (Pattern Matching), representan cada clase mediante un patrón prototipo. Algunos métodos de este tipo son:

- Clasificador de mínima distancia
- Adaptación por correlación

Clasificadores estadísticamente óptimos, se fundamentan en la Teoría de la decisión estadística. En este método tenemos:

- Clasificador Bayesiano para clases gaussianas

Redes neuronales engloba a un conjunto de técnicas que proporcionan soluciones flexibles, adaptables a cada problema. Se fundamentan en la teoría del aprendizaje estadístico.

Cada uno de estos métodos proporciona diferentes ventajas e inconvenientes. La elección de uno depende del problema que se vaya a resolver y de los resultados esperados del mismo.



### 3.12 Desarrollo de la aplicación para reconocimiento de caracteres

El primer paso para el desarrollo de la aplicación consiste en determinar cada uno de las regiones en las que se encontraran los números a ser identificados por el sistema para lo cual haremos uso de las herramientas ya anteriormente descritas.



Figura 3.1 Recorte de la Imagen

En la figura 3.1 podemos ver que en la primera figura se encuentra convertida la imagen de color RGB a escala de grises para un posterior uso de esta imagen.

En la figura de la mitad se encuentra delimitada la zona en la que se encuentra el número a ser identificado, debido a que cada campimetría cuenta con resultados tanto para el ojo derecho como para el izquierdo por separado se toman doce regiones a analizar en cada uno de los respectivos resultados, lo que nos da un total de veinte y cuatro regiones a procesar.

Finalmente en la última figura se encuentra ya la imagen recortada de la original, dicha figura servirá para continuar con el proceso de reconocimiento.

Una vez obtenida la región de trabajo debemos determinar las aéreas correspondientes a cada número, para posteriormente realizar la correlación con las imágenes base.

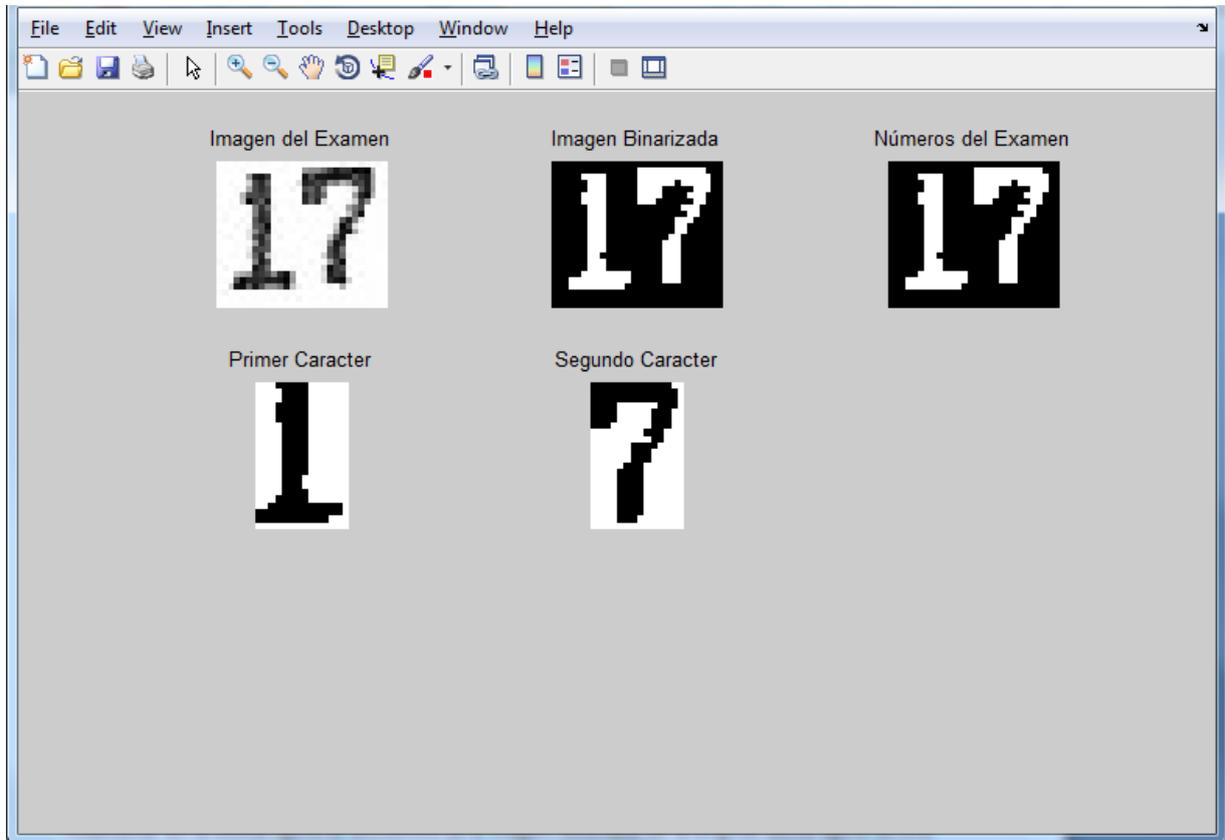


Figura 3.2 Procesamiento de Imagen Recortada

En la figura 3.2 tenemos la región recortada con la cual empezaremos el análisis. Para la segunda figura se presenta la imagen binarizada, esto quiere decir que la imagen que originalmente se encontraba en escala de grises ahora pasa a tener solo dos números en su matriz (0,1) lo cual convierte la imagen a uno en blanco y negro, este proceso nos permite encontrar las aéreas correspondientes a cada número mediante el uso de discriminantes, lo cual nos da de resultado las dos últimas imágenes en donde podemos apreciar que el sistema ya ha determinado el área de cada uno de los números los cuales están listos para realizar la correlación con las imágenes que se encuentran previamente almacenadas.

Una vez realizada la correlación podemos utilizar la matriz resultante de dicha operación para asignar el valor numérico correspondiente a la figura procesada.

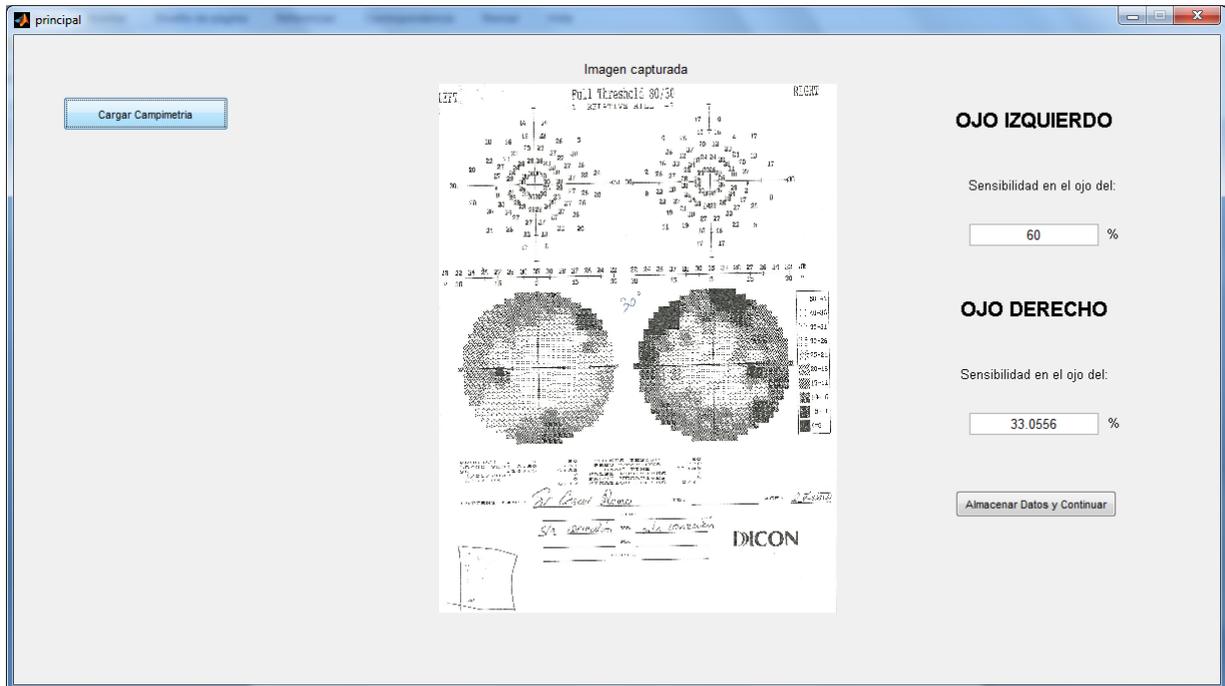


Figura 3.3 Resultados Obtenidos

Finalizado todo el proceso de reconocimiento de imagen se toman todos los valores obtenidos del resultado de cada ojo para calcular la sensibilidad que tiene en la actualidad el paciente Figura 3.3.



## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

Al finalizar el presente proyecto se ha llegado a las siguientes conclusiones:

- El sistema creado será de mucha ayuda para médicos especialistas en el área de la oftalmología debido a que al ser flexible puede acoplarse a nuevos exámenes que se vayan desarrollando para el estudio del glaucoma crónico de ángulo abierto.
- Se puede mantener un registro no solo de los exámenes del paciente sino también de los tratamientos que les son suministrados, lo cual permite evaluarlo al cabo de 6 meses y verificar si el estado de la enfermedad se ha detenido o continúa degenerando la visión del paciente.
- Gracias a que el sistema agrega el módulo de procesamiento de imágenes, además de añadir grillas a los exámenes conocidos como fondo de ojo, se reduce considerablemente el error al realizar mediciones ya que estas eran subjetivas y dependían de la experiencia del especialista para evitar confusiones o malas interpretaciones.
- Para el momento de escoger entre el uso de un sistema basado en redes neuronales o uno basado en reglas se determinó que la mejor opción sería la basada en reglas debido a que los valores resultantes de referencia a tomarse en los exámenes se han mantenido estables a lo largo del estudio del glaucoma.
- La metodología usada junto con el diseño en capas fueron acertadas para el desarrollo de este tipo de proyectos ya que nos permitió obtener un producto de calidad el cual satisface las necesidades del médico especialista.



## Recomendaciones

- Al momento de usar el sistema en el módulo de procesamiento de imágenes se recomienda siempre el uso de campimetrías de 30 grados ya que con estas se realizaron las calibraciones para un óptimo resultado.
- Si se es un usuario nuevo se recomienda el uso del manual ya que costa una guía de cómo manejar el sistema, de esta manera se garantiza un funcionamiento óptimo y eficiente del mismo.
- El usuario del sistema debería llenar los campos de observaciones que se presentan en cada uno de los exámenes que conforman el diagnóstico del glaucoma crónico de ángulo abierto ya esto permite tener un mayor control sobre los tratamientos que se le realizan al paciente.



## **ANEXOS**

**Anexo 1: Análisis de Parámetros**

Objeto (Parámetro)	Descripción	Unidad	Tipo Dato	Rango Permitido	Política de Selección	Valores Preestablecidos
Edad	La Edad del Paciente al momento de la visita	Años	Integer	40 -80	Single	40-44 45-49 50-54 55-59 60-64 65-69 70-74 75-80
Presión Intraocular (PIO)	Promedio de la PIO sin tratamiento para ambos ojos, medida en 2 a 4 visitas durante 6 meses. Como Alternativa, los Investigadores del Estudio sugieren que tal vez sea posible usar la PIO promedio de ambos ojos de la visita 1 para proporcionar una estimación dentro del 3% del riesgo calculado para el 95% de los pacientes.	mmHg (milímetros de mercurio)	Integer	23 -32	Single	32 30,31 29 28 27 26 25 24 23
GCC	Un Promedio de 3 mediciones diferentes de ambos ojos en la misma visita	µm (micrometros)	Double	450 - 689	Single	450-469 470-489 490-509 510-529 530-549 550-569 570-589 590-609

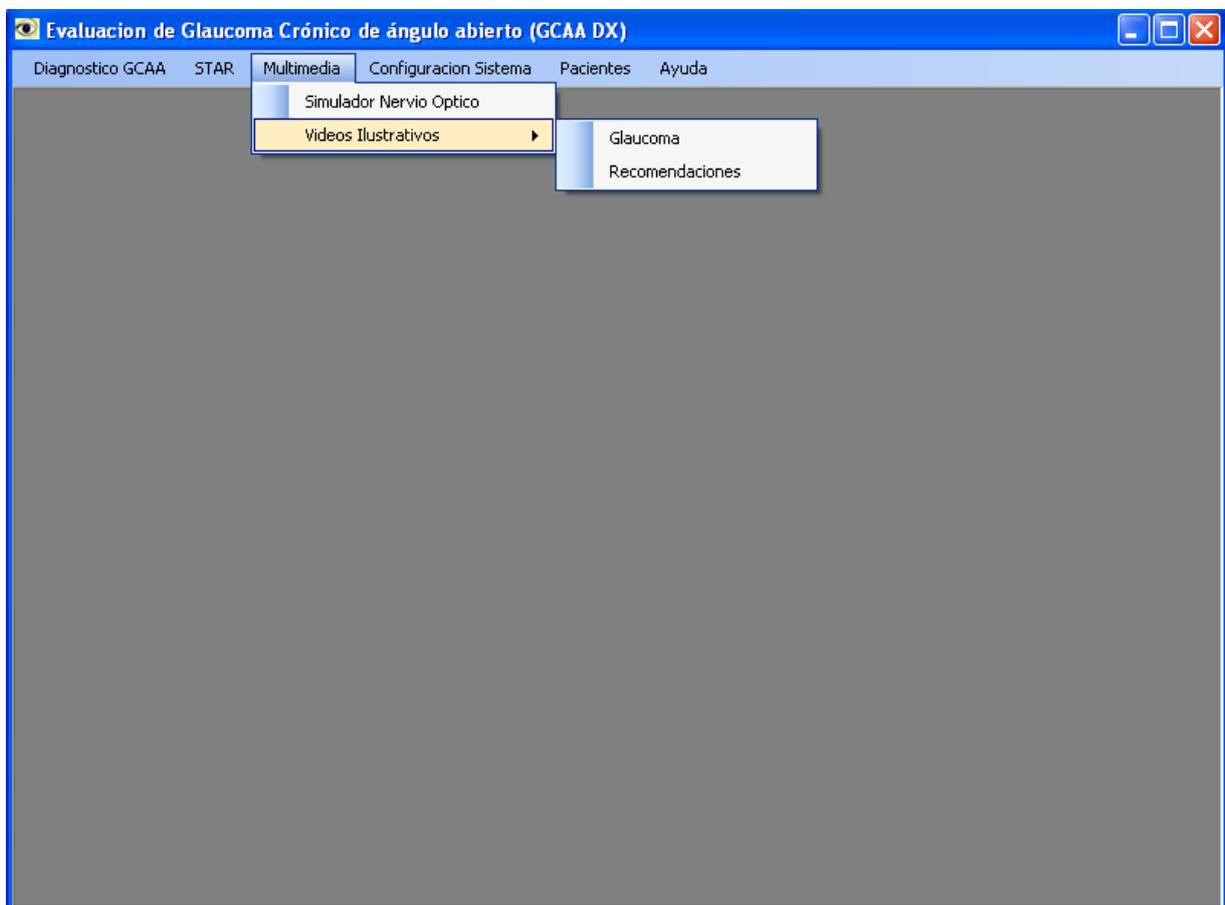


						610-629 630-649 650-669 670-689
PSD	Promedio de ambos ojos establecido a partir del informe más reciente del índice del campo visual.	dB (decibelios)	Double	1 - 2.59	Single	1.00-1.19 1.20-1.39 1.40-1.59 1.60-1.79 1.80-1.99 2.00-2.19 2.20-2.39 2.40-2.59
Índice C/D	Índice entre profundidad de la depresión vertical/papila óptica (C/D) – un promedio de ambos ojos.		Double	0.1 - 0.9	Single	0.9 0.8 0.7 .... 0.1
Diabetes mellitus	Paciente sufre de diabetes		Boolean	si, no	Single	Si No
Umbrales de Riesgo Global	Porcentaje de Riesgo de GCAA	Porcentaje	Double	1% - 100 %	Single	< 5% observación> <5% - 10% Considerar Tratamiento> > 15% Indicar Tratamiento



## Anexo 2: Manual de Usuario

El sistema se encuentra elaborado en base a pestañas que nos sirven para navegar entre las diferentes opciones disponibles para el usuario y que se encuentran organizados por la funcionalidad de los mismos. Aprovechando que la lógica de los procedimientos están basados en un diagrama de flujo, se pudo reflejar el funcionamiento del mismo a manera de un asistente o wizard, a través de pasos ordenados y claramente definidos.





Dentro de la pestaña Diagnostico GCAA tenemos las opciones Nuevo Paciente, Evolución Paciente desde aquí podemos controlar cada uno de los procesos de diagnostico que se siguen para cada paciente, finalmente la opción Salir con la cual podemos abandonar el sistema.



Al seleccionar Nuevo Paciente se nos presentará una pestaña con la información general del paciente, debajo de esta se encuentran casillas de verificación en las que podemos seleccionar factores de riesgo como son el tipo de raza, antecedentes familiares, hipertensión entre otras.

**Ingreso de Historia Clínica**

**Informacion General del Paciente**

Cédula de Identidad: 0104178685

Nombres: JUAN FRANCISCO

Apellidos: VASCONEZ ABRIL

Edad: 56

Fecha de Nacimiento: 15/12/1955

Direccion: CUENCA - ECUADOR

Telefono: 2555666

Correo Electronico: juan@dominio.com

**Seleccione los factores de riesgo que presenta el paciente:**

<input type="checkbox"/> Raza mestiza, negra o hispanoamericana	<input checked="" type="checkbox"/> Hipertension Arterial
<input checked="" type="checkbox"/> Antecedentes familiares de presencia de Glaucoma	<input checked="" type="checkbox"/> Miopia Elevada

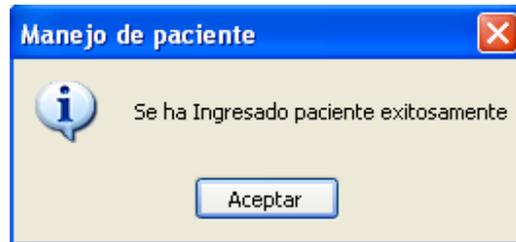
**Observaciones**

EL PACIENTE PRESENTA ANTECEDENTES DE GLAUCOMA FAMILIAR POR PARTE DE MADRE

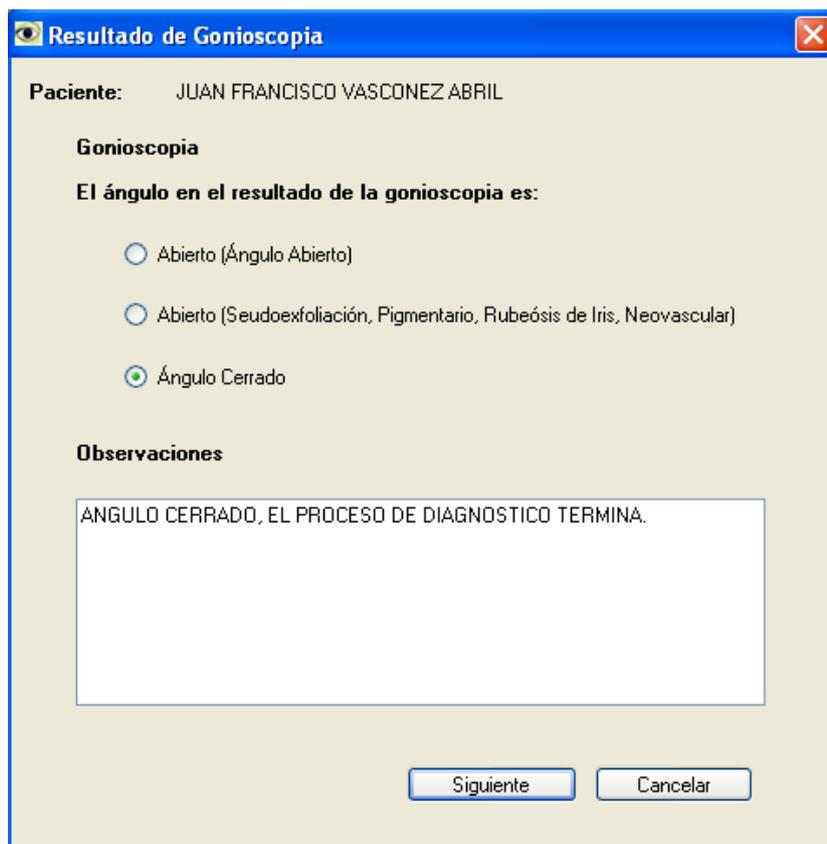
Siguiente Cancelar



Una vez que la ficha se ha completado con los datos correspondientes y damos click en el botón siguiente se presenta un mensaje que nos indica que los datos del paciente se han guardado exitosamente.

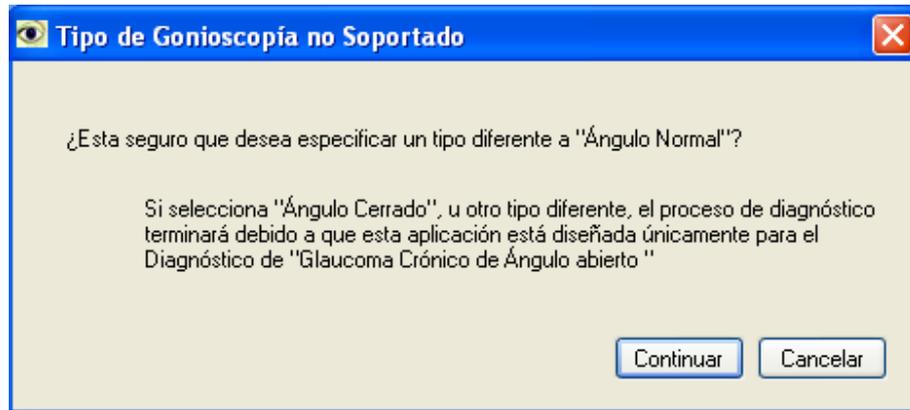


El primer examen que forma parte del diagnóstico es la gonioscopia, de acuerdo con el resultado de este examen tenemos tres tipos de ángulos: abierto, abierto por pseudoexfoliación, Rubeosis de Iris, etc. y ángulo cerrado.





El sistema se encuentra habilitado para funcionar únicamente con un ángulo abierto, en el caso de seleccionar un tipo de resultado no soportado se nos aparecerá un mensaje informándonos acerca del error y el proceso de diagnóstico será cerrado por el programa.



En el caso de ser necesario se encuentra presente en todos los exámenes la opción de anotar observaciones que se pueden presentar a lo largo del diagnóstico.



**Resultado de Gonioscopia**

**Paciente:** JUAN FRANCISCO VASCONEZ ABRIL

**Gonioscopia**

**El ángulo en el resultado de la gonioscopia es:**

Abierto (Ángulo Abierto)

Abierto (Seudoexfoliación, Pigmentario, Rubeosis de Iris, Neovascular)

Ángulo Cerrado

**Observaciones**

ANGULO ABIERTO GCAA

Para el examen de fondo de ojo podemos interactuar con imágenes que pueden ser cargadas desde el computador, cuenta también con la opción de colocar una grilla con el objetivo de realizar mediciones mucho más exactas.



**Examen de Fondo de Ojo Derecho**

Cargar Imagen    Construir Grilla

SUPERIOR(S)

TEMPORAL (T)

NASAL (N)

INFERIOR(I)

Excavación: 0,7

- Excavación nasal
- Atrofia peripapilar
- Excavación profunda con lámina cribosa prominente
- Vasos de bayoneta

Seleccione del elemento más ancho al más delgado (Formato ISNT):

I S N T

Reset

Grosor Corneal (micras): 560

Observaciones:

ADELGAZAMIENTO DEL ANILLO NEURORETINIANO CON ALTERACION DEL PATRON ISNT

Paciente: JUAN FRANCISCO VASCONEZ ABRIL

Cancelar    Siguiente

Se debe tener en cuenta que este examen se debe realizar primero para el ojo derecho y luego para el ojo izquierdo.



**Examen de Fondo de Ojo Izquierdo**

Cargar Imagen    Construir Grilla

SUPERIOR(S)

N  
A  
S  
A  
L  
(N)

INFERIOR(I)

Paciente: JUAN FRANCISCO VASCONEZ ABRIL

Excavación: 0,3

- Excavación nasal
- Atrofia peripapilar
- Excavación profunda con lámina cribosa prominente
- Vasos de bayoneta

Seleccione del elemento más ancho al más delgado (Formato ISNT):

T I N S

Reset

Grosor Corneal (micras): 400

Observaciones: SIN NOVEDADES

Cancelar    Siguiente

Una vez concluido el ingreso de los datos del paciente para este examen se presenta una tabla comparativa de los resultados de ambos ojos, junto con cada resultado se indica si el parámetro presenta alguna anomalía, en la parte inferior de la tabla se presenta el campo asimetría de la excavación que resulta de la diferencia entre las excavaciones de cada ojo, en el caso de ser mayor a 0.2 presenta anomalías.



**Resultados de Comparación de Exámenes de Fondo de Ojo**

**Paciente:** JUAN FRANCISCO VASCONEZ ABRIL

<b>Resultados de Exámen de Fondo de Ojo Izquierdo:</b>		<b>Resultados de Exámen de Fondo de Ojo Derecho:</b>	
Excavación:	0,3 (RANGO NORMAL)	Excavación:	0,7 (ANOMALÍA)
Excavación Nasal	NO (PARÁMETRO NORMAL)	Excavación Nasal	SI (ANOMALÍA)
Atrofia Peripapilar:	NO (PARÁMETRO NORMAL)	Atrofia Peripapilar:	SI (ANOMALÍA)
Adelgazamiento ISNT:	NO (PARÁMETRO NORMAL)	Adelgazamiento ISNT:	NO (PARÁMETRO NORMAL)
Excavación Profunda:	NO (PARÁMETRO NORMAL)	Excavación Profunda:	NO (PARÁMETRO NORMAL)
Vasos de Bayoneta:	NO (PARÁMETRO NORMAL)	Vasos de Bayoneta:	NO (PARÁMETRO NORMAL)
Grosor Corneal:	400	Grosor Corneal:	560

**RESULTADOS** Los Valores de los parámetros son Normales

**RESULTADOS** Existen anomalías en valores de los Parámetros

**Resultados de Comparación de Fondo de Ojo (Ojo Izquierdo - Ojo Derecho)**

Asimetría de Excavación: 0,4 (ANOMALÍA)

Diagnóstico Sugerido: Resultados de Exámen presentan anomalías, se recomienda realizar un Exámen de "Corrección de la presión"

Debido a que la presión ocular se encuentra estrechamente ligado con el grosor corneal se ha realizado una ventana en la que se ejecutan las correcciones de la PIO de acuerdo a valores previamente establecidos.



**Examen de corrección de la Presión Intraocular (PIO)**

<b>Grosor Corneal OI (micras)</b>	<b>Grosor Corneal OD (micras)</b>
400	560
<b>Valor de Corrección PIO - OI (mmHg)</b>	<b>Valor de Corrección PIO - OD (mmHg)</b>
7,25	-0,75

**Observaciones:**

EL PACIENTE PRESENTA ANOMALIAS EN EL OJO IZQUIERDO

Con los valores obtenidos para la corrección se procede al cálculo de la presión intraocular, estos valores serán sumados o restados de la presión del paciente que es ingresada por el usuario. La presión corregida es presentada inmediatamente después de que la presión original es ingresada.



**Calculo de Presión Intraocular (PIO)**

Paciente: JUAN FRANCISCO VASCONEZ ABRIL

OJO IZQUIERDO	OJO DERECHO
PIO del Paciente (mmHg): 20,1	PIO del Paciente (mmHg): 20,5
Corrección de la PIO (mmHg): 7,25	Corrección de la PIO (mmHg): -0,75
<b>PIO Resultante (mmHg) 27,35</b>	<b>PIO Resultante (mmHg) 19,75</b>

**Observaciones:**  
PACIENTE PRESENTA ELEVADA PIO EN OJO IZQUIERDO

Siguiente Cancelar

Concluido el proceso se presenta una tabla de resumen con las presiones de cada ojo y en caso de ser anómalas también se informa al usuario un cual se posee problemas.

**Examen de Comprobación de PIO (Presión Intraocular)**

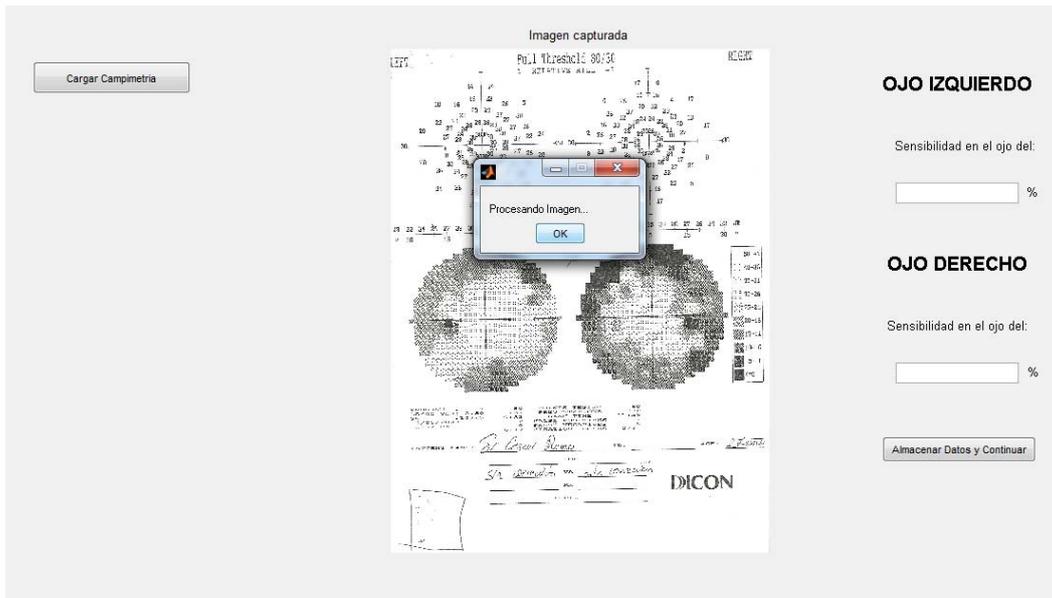
PIO TOTAL OI (mmHg)	PIO TOTAL OD (mmHg)
27,35 (ANOMALÍA)	19,75 (PIO NORMAL)

**Resultado:** Resultados en la PIO con valores elevados (sobre los 21 mmHg)  
Exámen de HRT es opcional. Se recomienda la práctica de una Campimetría

Continuar Cancelar



El siguiente paso en el flujo es el procesamiento de la campimetría, para lo cual el usuario deberá cargar una campimetría previamente almacenada, luego de seleccionada la imagen el sistema automáticamente comenzará con la interpretación del examen y realizará los cálculos de sensibilidad de cada ojo





**Resultado de Campimetría**

**OJO DERECHO**  
Sensibilidad Restante del Ojo (%) 33,0556

**OJO IZQUIERDO**  
Sensibilidad Restante del Ojo (%) 60

Siguiente Cancelar

El último paso en el diagnóstico del glaucoma crónico de ángulo abierto es determinar el tratamiento a seguir por parte del paciente. Es necesario que el usuario lleve un control de los tratamientos realizados al paciente para facilitar el cambio de tratamiento en caso de ser necesario.

**Tratamiento sugerido**

**Paciente:** JUAN FRANCISCO VASCONEZ ABRIL

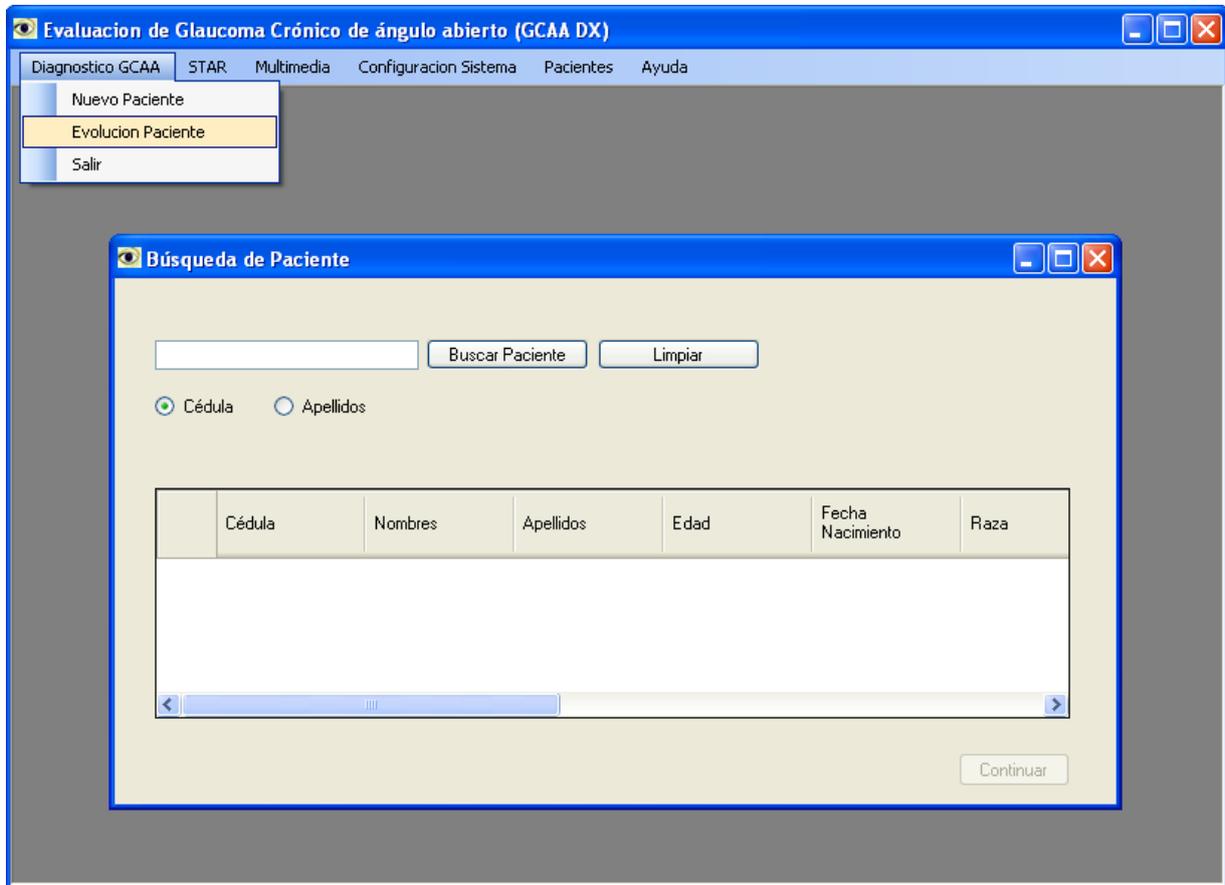
**Descripción del Tratamiento**

REDUCCION DE PIO EN OJO IZQUIERDO HASTA ESTABILIZARLA

Aceptar Cancelar



Para seguir la evolución de un paciente debemos entrar en Diagnostico GCAA y seleccionar la opción paciente, aquí podremos buscar tanto por apellido como por el número de cedula.





**Búsqueda de Paciente**

0104178685       

Cédula     Apellidos

	Cédula	Nombres	Apellidos	Edad	Fecha Nacimiento	Raza
▶	0104178685	JUAN FRANCISCO ...	VASCONEZ ABRIL ...	56	15/12/1955	

**Búsqueda de Paciente**

vasconez       

Cédula     Apellidos

	Cédula	Nombres	Apellidos	Edad	Fecha Nacimiento	Raza
▶	0104178685	JUAN FRANCISCO ...	VASCONEZ ABRIL ...	56	15/12/1955	

Una vez seleccionado el paciente se nos presenta una ventana en la cual en la parte superior se muestran todos los diagnósticos que ya fueron concluidos y q servirán de referencia para el médico



especialista, en la parte inferior se encuentra el último examen que fue realizado a un paciente pero sin finalizar el diagnóstico.

Fecha de Diagnóstico	Resumen DX	Tratamiento	Descripción del Tratamiento

Fecha de Exámen	Exámen	Observaciones	Notas
22/04/2011 15:35	Gonioscopía	ANGULO ABIERTO, CONTINUA CON DIAGNOSTICO ...	

En el caso de que deseásemos continuar con el diagnóstico basta con seleccionar el último examen realizado y dar click en la opción continuar diagnóstico, el sistema preguntará si está seguro de seguir con el diagnóstico.

¿Desea continuar con el proceso de diagnóstico para este paciente?

Continuar Cancelar



En el caso de ser necesario se puede modificar la historia clínica de un paciente ya sea para actualizar sus datos o para cambiar algún parámetro que pudo haber sido ingresado erróneamente.

**Modificación de Historia Clínica**

Información General del Paciente

Cedula: 0104178685

Nombres: JUAN FRANCISCO

Apellidos: VASCONEZ ABRIL

Edad: 56

Fecha de Nacimiento: 15/12/1955

Direccion: CUENCA - ECUADOR

Telefono: 2555666

Correo Electronico: juan@dominio.com

Seleccione los factores de riesgo que presenta el paciente:

Raza mestiza, negra o hispanoamericana

Antecedentes familiares de presencia de Glaucoma

Hipertension Arterial

Miopia Elevada

Posteriormente al cambio de la historia clínica se presenta un mensaje indicándonos que la modificación se dio de una manera exitosa.



The screenshot shows a software window titled "Modificación de Historia Clínica" with a blue title bar and standard window controls. The main area is titled "Información General del Paciente" and contains several input fields: "Cedula:" with the value "0104178685" and a "Buscar" button; "Nombres:" with "NOMBRE CAMBIADO"; "Apellidos:" with "APELLIDOS CAMBIADOS"; "Edad:" with "56"; "Fecha de Nacimiento:" with "15/10/1955"; "Dirección:"; "Teléfono:"; and "Correo Electrónico:". Below these fields is a section titled "Seleccione los factores de riesgo que presenta el paciente:" with four checkboxes: "Raza mestiza, negra o hispanoamericana" (unchecked), "Antecedentes familiares de presencia de Glaucoma" (unchecked), "Hipertension Arterial" (checked), and "Miopia Elevada" (checked). At the bottom right are "Modificar" and "Cancelar" buttons. A smaller dialog box titled "Manejo de paciente" is overlaid on top, containing an information icon, the text "Se ha Modificado paciente exitosamente", and an "Aceptar" button.

Finalmente se puede ver reflejado el cambio que se llevo a cabo en el momento que se realiza la búsqueda de un paciente y los datos son mostrados en la tabla.

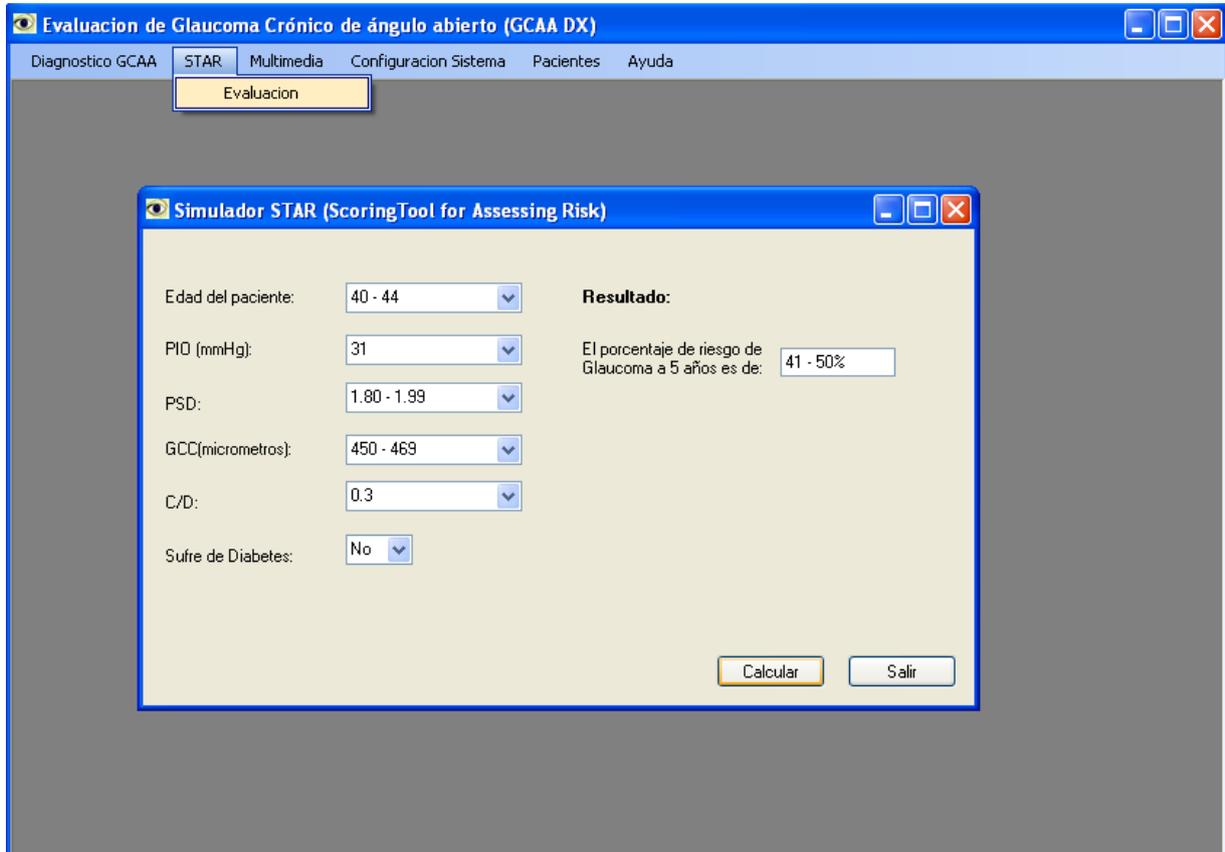


The screenshot shows a software window titled "Búsqueda de Paciente" with a search interface. At the top, there is a text input field containing "0104178685", a "Buscar Paciente" button, and a "Limpiar" button. Below this, there are two radio buttons: "Cédula" (which is selected) and "Apellidos". The main area contains a table with the following data:

	Cédula	Nombres	Apellidos	Edad	Fecha Nacimiento
▶	0104178685	NOMBRE CAMBIADO ...	APELLIDOS CAMBIADOS ...	56	15/12/1955

At the bottom right of the window, there is a "Continuar" button.

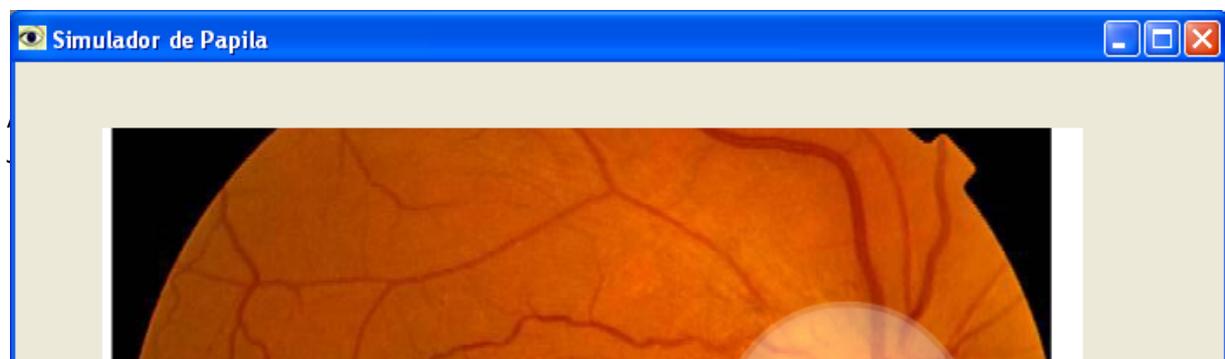
Como una herramienta adicional al sistema se encuentra un módulo que nos permite calcular el riesgo de padecer glaucoma dentro de cinco años, para el uso de esto se deben llenar los campos que se indican y en la parte derecha se presentará el porcentaje de riesgo de padecimiento de glaucoma.



Como elementos ilustrativos se encuentran integradas animaciones en las cuales se presenta como evoluciona la pérdida de la visión al momento de padecer glaucoma, para entrar a estos elementos debemos dirigirnos a la pestaña multimedia y posteriormente a videos.



De la misma manera se presenta una animación en la cual se puede reproducir el estado de la papila, excavaciones y verificar el patrón ISNT que puede presentar un paciente. Al igual que el caso anterior para acceder a este elemento debemos dirigirnos a la pestaña multimedia.





# SIMULADOR DE PAPILA

The simulator interface displays a retinal fundus with a central optic disc (papilla) highlighted in a semi-transparent circle. A control panel is overlaid at the bottom right, featuring a 'CONTROLES' header and three main sections: 'PAPILA', 'EXCAVACION', and 'CONTROLES'. The 'PAPILA' section has four directional arrows and displays height (h) and width (v) measurements. The 'EXCAVACION' section has four directional arrows and displays height (h'), width (v'), and depth (I-S) measurements. The 'CONTROLES' section includes buttons for 'RESET', 'ALEATOREO', and 'MEDIDAS'. The Universidad de Cuenca logo and name are visible in the bottom left corner of the simulator window.

CONTROLES					
PAPILA		EXCAVACION		CONTROLES	
h 1.7 mm.	v 1.7 mm.	h' 0.47 mm.	v' 0.47 mm.	N-T0 mm.	I-S 0 mm.
[Arrows]		[Arrows]		[RESET]	
[Arrows]		[Arrows]		[ALEATOREO]	
[Arrows]		[Arrows]		[MEDIDAS]	

UNIVERSIDAD DE CUENCA



## GLOSARIO

**Criptoaritmética:** Método que consiste en sustituir cada letra por un dígito, de modo que la operaciones sean correctas. A igual letra, igual digito, y a distinta letra distinto dígito. Como es habitual los números no pueden tener ceros a la izquierda.

**Torres de Hanoi:** Es un rompecabezas o juego matemático inventado en 1883 por el matemático francés Edward Lucas.

**Heurístico:** capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos.



## BIBLIOGRAFÍA

- **.NET2008**

[msdn.microsoft.com/en-us/vstudio/default.aspx](http://msdn.microsoft.com/en-us/vstudio/default.aspx)

- **SQL Server 2005**

[technet.microsoft.com/es-es/library/ms166016.aspx](http://technet.microsoft.com/es-es/library/ms166016.aspx)

- **Sistemas Expertos**

[SE1][http://es.wikipedia.org/wiki/Sistema\\_experto](http://es.wikipedia.org/wiki/Sistema_experto)

[SE2][http://es.wikipedia.org/wiki/Ingenier%C3%ADa\\_del\\_conocimiento](http://es.wikipedia.org/wiki/Ingenier%C3%ADa_del_conocimiento)

[SE3] “*Sistemas Expertos - Principios y Programación*”, JOSEPH GIARRATANO-GARY RILEY

“*Visión Artificial y Procesamiento Digital de Imágenes Usando Matlab*”, IVÁN GARCÍA

- **Oftalmología**

[ofta1] “Oftalmología”, *Trabajar con los ojos de la gente*, <http://www.metodobases.it/buena-vista/trabajar.html>.

[ofta2] Doctor Pablo Zeas (Oftalmólogo Especialista)

[ofta3] Juan Carlos Betancourt “*Guía de Manejo GLAUCOMA*”, Editorial “THOMSON PLM Ecuador SA”, 2009

“*Diagnóstico y Tratamiento del Paciente Adulto con Glaucoma de Ángulo Abierto*”, CONSEJO DE SALUBRIDAD GENERAL MEXICO, 2009

