

Universidad de las Ciencias Informáticas

Facultad 3



Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Trabajo Final presentado en opción al título de Ingeniero en Ciencias Informáticas

Autor:

Carlos Andrés Alonso Peña

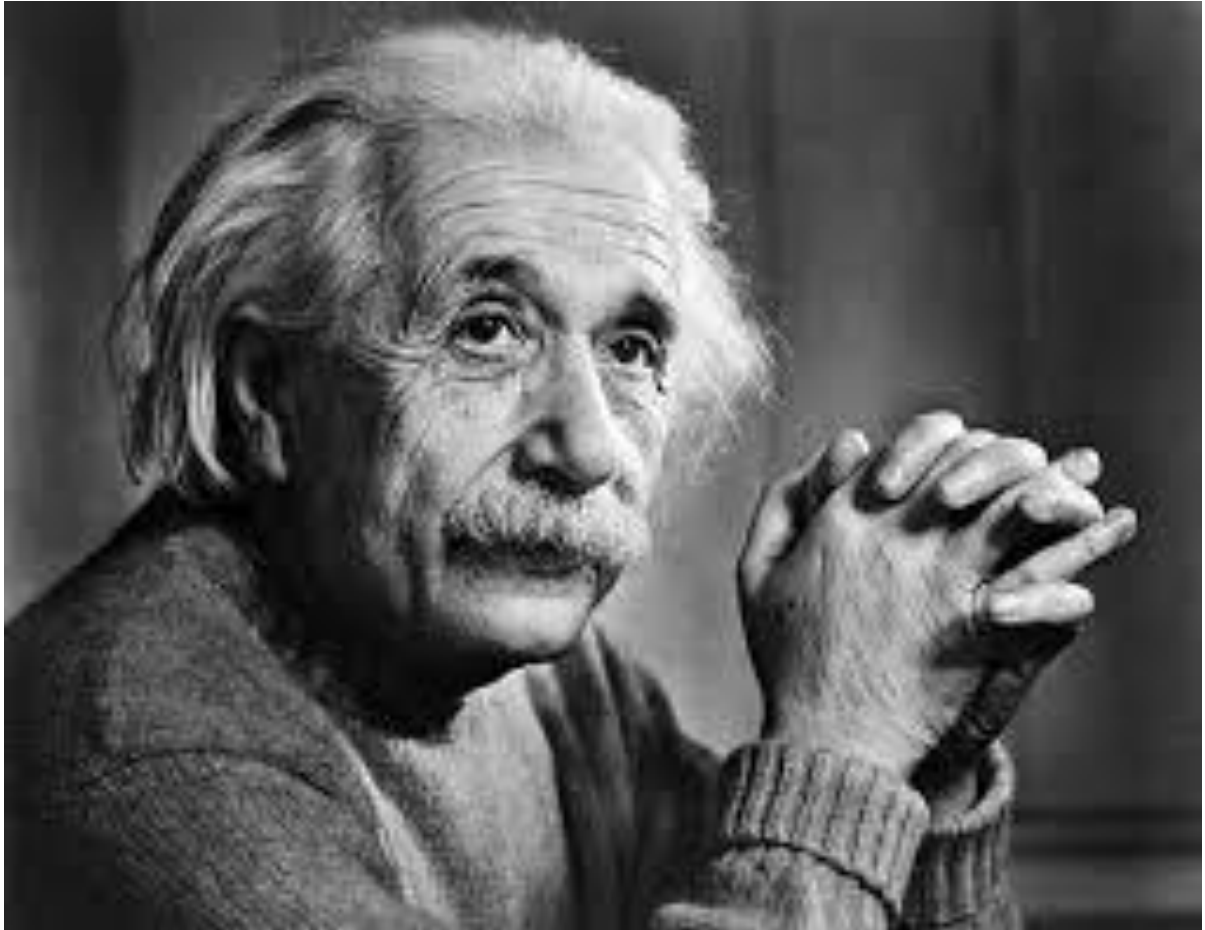
Pedro Luis Machado Sánchez

Tutores:

Ing. Michel Alvarez Cancio

Ing. Héctor Pérez Baranda

Ciudad de la Habana, junio de 2016



*Hay una fuerza motriz más poderosa que el vapor, la electricidad y
la energía atómica: la voluntad.*

Albert Einstein

Declaración de Autoría

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Carlos Andrés Alonso Pedro Luis Machado Ing. Michel Alvarez Cancio.

Ing. Héctor Pérez Baranda.

Dedicatoria

Carlos: quiero dedicar mi tesis a dos personas muy importantes en mi vida: mi madre Mayda y mi abuela Olga.

Pedro: quiero dedicar mi tesis a las tres personas muy importantes de mi vida: mi hermana Jennifer, mi madre y mi padre Pedro.

Agradecimientos

Carlos: a mis padres por el apoyo infinito que me han dado en la vida y ser los principales autores de mi formación como profesional. A mi hermana, mi padrino y toda mi familia en general brindarme todo el apoyo necesario. A mis tutores Michel y Héctor de los que estoy eternamente agradecido por el apoyo que me han dado en la tesis, a pesar de las críticas y sugerencias en mi beneficio. A la Universidad de las Ciencias Informáticas por darme la oportunidad de formarme como profesional. A todos los profesores que me han ayudado en mi vida estudiantil. A mis compañeros de aula por compartir todo este tiempo juntos y aprender algo de cada uno. A Yoel, Andres, Pedro, Manolo, Joenis, Keylier, Leo, La china por ser ejemplo de amigo durante estos cinco años por mostrarme siempre su confianza y apoyo. A mi madrina que la quiero con la vida. Y a toda la gran familia los quiero.

Pedro: a mis padres por el apoyo infinito que me han dado en la vida y ser los principales autores de mi formación como profesional. A mi hermana, mis abuelos y toda mi familia en general apoyarme siempre en mis decisiones. A mis tutores Michel y Héctor de los que estoy infinitamente agradecido por el apoyo que me han dado en la tesis, a pesar de las críticas y sugerencias en mi beneficio. A la Universidad de las Ciencias Informáticas por darme la oportunidad de formarme como profesional. A todos los profesores que me han ayudado en mi vida estudiantil. A mis compañeros de aula por compartir todo este tiempo juntos y aprender algo de cada uno. A Yoel, Andres, Carlos, Manolo, Joenis, Keylier, Leo, La china por ser ejemplo de amigo durante estos cinco años por mostrarme siempre su confianza y apoyo. A mi madrina que la quiero con la vida. Y a toda la gran familia los quiero.

Resumen

Resumen

La opacidad de la cápsula posterior es actualmente uno de los aspectos más importante en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación postoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas. Por tal motivo la presente investigación tiene como objetivo la creación de una aplicación denominada PANDOC 2.0, que permita mediante un sistema basado en casos identificar y cuantificar objetivamente esta opacidad mediante el uso de las imágenes resultantes de la cámara Scheimpflug del Pentacam y evaluar su correlación con la graduación subjetiva mediante lámpara de hendidura. Con el desarrollo de la investigación se provee a los oftalmólogos una herramienta capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano) mejorando versiones anteriores, logrando así una mejor evaluación objetiva del grado de afectación. Esto brinda facilidades a los especialistas para la toma de decisiones, rapidez para llevar a cabo el proceso de evaluación y emisión del diagnóstico, puesto que reduce la curva de aprendizaje por su facilidad de uso.

Palabras claves: Cirugía de catarata, opacidad de la cápsula posterior, sistema basado en casos, tomogramas Scheimpflug del Pentacam.

Índice

Índice

Introducción	13
Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales.	
Bases conceptuales	18
1.1 Opacidad de Cápsula Posterior: Conceptualización	18
1.2 Tratamiento de la OCP: Estudio médico	18
1.3 Prevención de la OCP	19
1.4 Sistemas desarrollados para la identificación de OCP	20
1.5 Segmentación basada en el uso de un umbral	23
1.5.1 Métodos de segmentación mediante umbralización	23
1.6 Componentes Conexas.....	24
1.7 Sistemas Basados en el conocimiento.....	25
1.7.1 Composición de los Sistemas Basados en Casos.....	26
1.7.2 Sistemas Basados en Reglas.....	27
1.7.3 Sistemas Basados en Probabilidades.....	29
1.7.4 Razonamiento Basado en Casos.....	30
1.7.5 Valoración del estudio de los Sistemas Basados en Casos.....	32
1.7.6 Tipos de Búsqueda en los Sistemas Basados en Casos.....	32
1.8 Justificación de la selección de Búsqueda por Semejanza	33
1.9 Base de Conocimientos	33
1.10 Herramientas usadas en el desarrollo.....	34
1.10.1 Metodología de Software.	34
1.10.2 Lenguaje de Programación.....	35
1.10.3 Entorno integrado de Desarrollo.....	36
1.10.4 Modelado de Lenguaje.....	36
1.10.5 Sistema Gestor de Base de Datos.....	37
Conclusiones del capítulo	38
Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas	39
2.1 Fase de Planificación	39
2.1.1 Historias de Usuario.....	39
2.1.2 Estimación de esfuerzos por Historias de Usuario.....	40
2.1.3 Requisitos no funcionales	41
2.1.4 Plan de iteraciones.....	42
2.1.5 Plan de entrega.....	43
2.2 Implementación de la propuesta de solución	44
2.2.1 Tarjetas CRC.....	44

Índice

2.2.2 Componentes conexas	44
2.2.3 Análisis de complejidad	56
2.3 Identificación de la OCP	58
2.3.1 Definición de los rasgos predictores y los rasgos objetivos	58
2.3.2 Dominio de definición de cada rasgo	59
2.3.3 Peso informacional de cada rasgo.....	59
2.3.4 Selección de las funciones de comparación de rasgos.....	60
2.3.5 Selección de la función de comparación de casos	60
2.3.6 Representación de los casos.....	61
2.3.7 Obtener el nuevo caso a resolver.....	61
2.3.8 Comparación del nuevo caso con la Base de Casos	62
2.3.9 Obtención de los 'k' casos más semejantes	62
2.3.10 Selección del rasgo objetivo.....	62
Conclusiones parciales del capítulo	62
Capítulo 3. Resultados y validación del sistema.	64
3.1 Fase de implementación.....	64
3.2 Pruebas de aceptación	68
3.2.1 Resultado de las pruebas de aceptación.....	72
3.3 Pruebas Unitarias	72
3.4 Métrica de evaluación de la clasificación por regiones.....	76
Conclusiones parciales del capítulo	78
Conclusiones Generales	79
Recomendaciones.....	80
Bibliografía.....	81
Anexos.....	85
Anexo 1: Historias de Usuarios	85
Anexo 2: Tarjetas CRC.....	89
Anexo 3: Tareas de Ingeniería generadas en la iteración 3.....	91
Anexo 4: Tareas de Ingeniería generadas en la iteración 4.....	92

Índice de Figuras

Índice de Figuras

Ilustración 1: Software POCO.....	20
Ilustración 2: Software EPCO.....	21
Ilustración 3: Software AQUA.....	21
Ilustración 4: Software AA.....	22
Ilustración 5: Software Image J.....	22
Ilustración 6: Ejemplo conectividad-4 y conectividad-8.....	25
Ilustración 7: Composición de los Sistemas Basados en Casos.....	26
Ilustración 8: Componentes del Razonamiento Basado en Casos.....	31
Ilustración 9: Imagen del Pentacam binarizada.....	45
Ilustración 10: Caso 1_Conectividad-8.....	47
Ilustración 11: Caso 2_Conectividad-8.....	47
Ilustración 12: Caso 3_Conectividad-8.....	47
Ilustración 13: Regiones identificadas rellenas.....	51
Ilustración 14: Generación de ventanas.....	54
Ilustración 15: Análisis realizado por PANDOC al tomograma del Paciente 1.....	69
Ilustración 16: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 1.....	70
Ilustración 17: Análisis realizado por PANDOC al tomograma del Paciente 2.....	70
Ilustración 18: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 2.....	70
Ilustración 19: Análisis realizado por PANDOC al tomograma del Paciente 3.....	71
Ilustración 20: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 3.....	71
Ilustración 21: Análisis realizado por PANDOC al tomograma del Paciente 4.....	71
Ilustración 22: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 4.....	72
Ilustración 23: Grafo de flujo del método Binarización.....	73
Ilustración 24: Resultados de las pruebas unitarias.....	75
Ilustración 25: Resultados obtenidos al aplicar la métrica.....	77

Índice de Tablas

Índice de Tablas

Tabla 1: HU 1. Construir BC	40
Tabla 2: HU 2. Calcular certeza de solución	40
Tabla 3: Puntos de estimación por historias de usuario.....	41
Tabla 4: Plan de duración de las iteraciones	43
Tabla 5: Plan de entrega	43
Tabla 6: Tarjeta CRC para la clase Geometria_Computacional.....	44
Tabla 7: Dominio de definición de cada rasgo.....	59
Tabla 8: Peso informacional de cada rasgo	60
Tabla 9: Tiempo de implementación de las HU de la iteración #1	64
Tabla 10: Tarea de ingeniería de la funcionalidad Crear BC	65
Tabla 11: Tarea de ingeniería de la funcionalidad Conexión	65
Tabla 12: Tarea de ingeniería de la funcionalidad Reconocer imagen de tomograma.....	65
Tabla 13: Tarea de ingeniería de la funcionalidad Reconocer centro de la estructura ocular	66
Tabla 14: Tiempo de implementación de las HU de la iteración # 2	66
Tabla 15: Tarea de ingeniería de la funcionalidad Llenar BC	66
Tabla 16: Tarea de ingeniería de la funcionalidad Obtener nuevo caso	67
Tabla 17: Tarea de ingeniería de la funcionalidad Buscar región.....	67
Tabla 18: Tiempo de implementación de las HU de la iteración # 3	67
Tabla 19: Tiempo de implementación de las HU de la iteración # 4	68
Tabla 20: Caso de prueba de aceptación para la HU Definir región del caso.....	68
Tabla 21: Caso de prueba de aceptación para la HU Calcular certeza de solución	69
Tabla 22: Resultado de las pruebas de aceptación	72
Tabla 23: Camino Básico del flujo	75
Tabla 24: Caso de Prueba Camino Básico # 1.....	75
Tabla 25: Matriz de confusión.....	76
Tabla 26: Matriz de confusión para las 80 regiones	77
Tabla 27: Historia de usuario #2	85
Tabla 28: Historia de usuario #3	85
Tabla 29: Historia de usuario #4	86
Tabla 30: Historia de usuario #5	86
Tabla 31: Historia de usuario #6	86
Tabla 32: Historia de usuario # 7	87
Tabla 33: Historia de usuario #8	87
Tabla 34: Historia de usuario # 9	88

Índice de Tablas

Tabla 35: Historia de usuario # 10	88
Tabla 36: Historia de usuario # 12	88
Tabla 37: Historia de usuario # 13	89
Tabla 38: Tarjeta CRC de la clase Conexión	89
Tabla 39: Tarjeta CRC de la clase Lienzo	89
Tabla 40: Tarjeta CRC de la clase Recortador	90
Tabla 41: Tarjeta CRC de la clase Configuración	90
Tabla 42: Tarjeta CRC de la clase Editor	90
Tabla 43: Tarea de ingeniería de la funcionalidad Cargar casos similares de la BC	91
Tabla 44: Tarea de ingeniería de la funcionalidad Calcular similitud de rasgos	91
Tabla 45: Tarea de ingeniería de la funcionalidad Calcular similitud de casos	91
Tabla 46: Tarea de ingeniería de la funcionalidad Calcular certeza de solución	92
Tabla 47: Tarea de ingeniería de la funcionalidad Brindar datos al usuario	92
Tabla 48: Tarea de ingeniería de la funcionalidad Recuperar el caso a la BC.....	92

Introducción

Introducción

En la actualidad personas de todo el mundo padecen de la enfermedad de cataratas, sufriendo como consecuencia la pérdida total o parcial de la visión. La catarata es la opacidad en el cristalino del ojo, trayendo como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, de esta forma crea una serie de imágenes difusas. Las cataratas generan problemas con la apreciación de cambios de colores y cambios de contrastes (Acosta R y otros, 2006).

Es la causa más común de ceguera tratable con cirugía, tiene diversas causas, pero se le atribuye mayormente a la edad, acelerando este proceso si el paciente padece de enfermedades como la diabetes o hipertensión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad (Acosta R y otros, 2006).

Tras una intervención quirúrgica el paciente puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito a largo plazo, pues en muchos de ellos el paciente puede presentar complicaciones postoperatorias. La complicación postoperatoria más frecuente a largo o mediano plazo es la Opacidad de la Cápsula posterior (OCP). La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata. Para garantizar una buena recuperación visual del paciente es imprescindible la transparencia de esta estructura. Su incidencia en la actualidad se encuentra entre 0,7% y 47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados. Siendo en Cuba una cifra considerable que asciende hasta el 50% de los casos. (Hernández López, 2010)

Se han identificados los principales factores que influyen en la aparición de esta anomalía postoperatoria, entre estos se encuentran:

- Los relacionados con el diseño del lente intraocular (LIO). En el citado estudio se puede observar cómo influye la biocompatibilidad de la lente intraocular en relación con el saco capsular. (Findl O y otros, 2008)
- Otro factor influyente son los implantes adicionales como los anillos de tensión capsular (Capsular Tension Ring: CTR), que a pesar de su eficacia para la reducción significativa de la OCP no logran eliminarla totalmente; este efecto limitado y el costo adicional de la cirugía han dificultado que se generalice su uso (Hernández López I, 2009).
- Las intervenciones farmacológicas han contribuido a interrumpir o inhibir la regeneración de las células epiteliales cristalinas, antimetabolitos (Hernández López y otros, 2009), inmunotoxinas (Tenten Y y otros, 2001), antiinflamatorios (Cortina P y otros, 1997) y trypanblue (Portes ALF y otros, 2010).
- La aplicación de las nuevas tecnologías también ha permitido obtener cierto grado de avance en este campo, el uso de técnicas como Cleanbagjet (Dewey S, 2006), la fotólisis láser

Introducción

(Pollhammer M y otros, 2007) y la pieza de mano láser (Henderson BA. 2009), se encuentran actualmente en estudio para definir su utilidad en la reducción de la OCP.

- También se menciona la terapia genética, campo en el cual se realizaron estudios importantes, evitando la diferenciación de las células epiteliales cristalinas (LEC: Lens Epitelial Cells) en células fibroblásticas que resultan en OCP (Gwon AE y otros, 1992).

No obstante, hasta el presente, estas estrategias no han logrado eliminar la aparición de la OCP de manera definitiva (Hernández López, 2011).

En la comunidad médica internacional no existe consenso alguno acerca de la correcta cuantificación de la OCP, se deja a cada especialista la elección de alguna de las técnicas propuestas (Alvarez Cancio y otros, 2014). La mayoría de los esfuerzos para el diseño de sistemas que cumplan con los objetivos de la investigación, tales como POCO del inglés Posterior Capsule Opacification (Barman SA y otros, 2000), AQUA (Cuantificación automática de catarata secundaria) (Findl O y otros, 2003), EPCO (Evaluación de la opacidad de la cápsula posterior) (Tetz MR y otros, 1997), y el sistema AA del inglés Aslam Analysis (Aslam TM y otros, 2005, Aslam TM y otros, 2005), están basados en las imágenes en retroiluminación obtenidas por las lámparas de hendiduras (Friedman DS y otros, 1999). La mayoría de estos sistemas tienen en común una desventaja muy peculiar producto de las imágenes resultantes de la lámpara de hendidura, como resultado de la reflexión del artefacto, efecto producido por la reflexión de la luz en la córnea, que es conocido como imágenes de Purkinje. Han sido desarrollados varios softwares propietarios para eliminar el efecto indeseable de dichas imágenes, con la función de múltiples imágenes de un mismo ojo (Wolf Buehl, 2002), fotografiadas en direcciones ligeramente diferentes de la mirada (Manfred R Tetz, 1997), pero ello involucra una curva de aprendizaje muy larga, ya que es un procesamiento tedioso y engorroso para el paciente.

El Pentacam es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo. Estas imágenes son reconstruidas a partir de múltiples fotografías tomadas mediante una cámara rotacional del sistema Scheimpflug. Este moderno sistema Scheimpflug con que cuenta el Pentacam es capaz de capturar imágenes en múltiples meridianos (hasta 50 imágenes en 2 segundos) en un único escaneo automatizado (Hernández López I, 2011). El análisis de estos tomogramas en 3D es de gran utilidad para la cuantificación objetiva de la OCP. Los tomogramas del Pentacam tienen además la ventaja de no presentar destellos de luz que interfieran con el análisis como sucede con las fotografías en retroiluminación, tomadas por los sistemas de adquisición de imágenes en lámpara de hendidura (Findl O, y otros, 2010) (Hernández López I, 2011).

Introducción

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Grupo de Investigación AIRI (Artificial Intelligence: Research and Innovation), el cual tiene como resultado el software PANDOC (Programa Analizador de Opacidad Capsular), realizado en colaboración con el Instituto Cubano de Oftalmología (ICO) "Ramón Pando Ferrer". El software PANDOC es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, minimizando el sesgo de observación entre un médico y otro. (Alvarez Cancio y otros, 2014)

"El software PANDOC es un sistema basado en casos para identificar y cuantificar objetivamente esta opacidad mediante el uso de las imágenes resultantes de la cámara Scheimpflug del PENTACAM y evaluar su correlación con la gradación subjetiva mediante lámpara de hendidura.

En una entrevista realizada al oftalmólogo Iván Hernández López se identificaron las siguientes deficiencias relacionadas con el uso del software PANDOC:

- Demora en identificar la OCP.
- Identifica pixel con OCP que no los son.
- Plantea la necesidad de identificar opacidad por regiones atendiendo al proceso natural de acumulación de la misma.

Es por ello que surge como necesidad la identificación automática de opacidad de la cápsula posterior (OCP) en imágenes del PENTACAM por regiones".

Por este motivo la presente investigación tiene como propósito fundamental el desarrollo de una herramienta que realice la segmentación de la OCP en componentes conexas, con el objetivo de identificar automáticamente las regiones pertenecientes a esta complicación postoperatoria para su posterior cuantificación.

Una vez culminada la etapa de Análisis se dio paso a la próxima etapa de desarrollo donde se identifica el siguiente **problema a resolver**: ¿cómo detectar automáticamente las regiones de opacidad capsular en tomogramas de Scheimpflug del Pentacam?

Objeto de estudio: reconocimiento de patrones en imágenes del Pentacam.

Campo de acción: identificación de OCP en imágenes resultante de los tomogramas de Scheimpflug del Pentacam en pacientes operados de catarata.

Objetivo general: desarrollar una herramienta para la detección automática de las regiones con OCP mediante tomogramas Scheimpflug del PENTACAM.

Introducción

Para dar cumplimiento al objetivo de la investigación se han descrito los siguientes **objetivos específicos**:

- Elaborar el marco teórico referencial relacionado con la identificación de regiones en imágenes médicas.
- Desarrollar un algoritmo que permita identificar regiones con opacidad en imágenes provenientes del PENTACAM.
- Validar la solución propuesta aplicando diferentes métodos de validación de software.

Durante el desarrollo de la investigación es necesario utilizar varios métodos científicos, los cuales se mencionan a continuación.

Observación: método que se aplicará de manera continua durante todo el proceso investigativo, observando los resultados de los algoritmos que se van a utilizar y así decidir los que mejor se adecuan al proceso.

Experimento: método que se utilizará para el desarrollo de experimentos de los algoritmos más importantes, basados en la identificación automática de la opacidad, así como el cálculo de la OCP. Luego con los resultados se seleccionarán los mejores en cuanto a algunos medidores.

Analítico-Sintético: se utiliza con el objetivo de analizar las teorías, documentos e información referente al tema en cuestión, para luego extraer los elementos más importantes que se relacionan con la identificación de la OCP en pacientes operados de catarata.

Histórico-Lógico: permite estudiar y analizar la trayectoria y evolución de software de análisis de OCP en pacientes operados de catarata, para poder comprender el nivel de desarrollo alcanzado por este tema en el mundo, además con estas experiencias se podrá tener un mayor entendimiento y comprensión de este tipo de sistemas, basándose en las experiencias existentes.

Inductivo-Deductivo: se utiliza con el objetivo de inducir una serie de conocimientos referentes a la cuantificación objetiva de la OCP, para luego arribar a razonamientos que conlleven a la deducción de conocimientos que pueden ser aplicables al problema en cuestión.

La investigación está estructurada de la siguiente forma:

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Se realiza un estudio de los aspectos fundamentales relacionados con la OCP, así como un estudio del arte de los Sistemas Basados en el Conocimiento y el Razonamiento Basado en Casos, así como el análisis de sistemas que tratan la cuantificación de la Opacidad de la Capsula Posterior.

Introducción

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Se profundiza en el razonamiento basado en casos y en cada uno de los pasos a llevar a cabo en el proceso evolutivo del sistema.

Capítulo 3. Resultados y validación del sistema

Se muestra el conjunto de pruebas realizadas a la herramienta y los resultados obtenidos.

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales. Bases conceptuales

En el presente capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación. Como puntos fundamentales se abordan los principales conceptos asociados al dominio del problema, como el de incidencia y prevención de la OCP. También se realiza un estudio del arte de los principales sistemas para la identificación de OCP que existen en el mundo, se estudian los sistemas basados en casos y las bases de conocimiento como técnicas de la inteligencia artificial a utilizar, las mismas ayudarán en la identificación de la opacidad capsular en pacientes sometidos a cirugía de cataratas.

1.1 Opacidad de Cápsula Posterior: Conceptualización

La extracción extracapsular de la catarata (EECC) con implante de lente intraocular es la técnica quirúrgica preferida para el tratamiento de catarata en la actualidad. En ella, a diferencia de la extracción intracapsular de la catarata (EICC), se extrae del ojo únicamente el cristalino opacificado dejando la cápsula posterior intacta. Esta estructura constituye una barrera anatómica entre los segmentos anterior y posterior del ojo, que se cree reduce el riesgo de complicaciones en el segmento posterior después de la cirugía (como el edema macular cistoideo y el desprendimiento de retina). Sin embargo, dejar la cápsula intacta puede dar lugar a complicaciones a largo plazo, principalmente la opacificación de la cápsula posterior. (Findl O y otros, 2008).

La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de cataratas, para garantizar una buena recuperación visual del mismo es imprescindible la transparencia de esta estructura. Desafortunadamente no son pocos los casos en los que se desarrolla esta anomalía, la opacificación de la cápsula posterior repercute negativamente en el resultado visual del paciente (Cancio, 2014). En el campo de la oftalmología esto es conocido como OCP y constituye la complicación tardía más importante en la cirugía de cataratas en los tiempos modernos (Michel Alvarez Cancio, 2013).

1.2 Tratamiento de la OCP: Estudio médico

Anteriormente para la realización de la capsulotomía como tratamiento de la opacidad de cápsula posterior, se efectuaba la discisión quirúrgica, ahora el Nd-YAG láser es el procedimiento ideal dado que el paciente no requiere hospitalización ni preparación preoperatoria, no necesita anestesia, es indolora, más facilidad, seguridad y precisión; riesgo y complicaciones mínimas, sin incisión y menor trauma quirúrgico. Requiere la transparencia de la córnea y del humor acuoso para su mayor

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

efectividad, así como la menor reacción inflamatoria ocular y el control previo de la presión intraocular, poca o ninguna medicación tras su aplicación y sin convalecencia o muy breve. (Ge J y otros 2000)

El Nd-YAG láser utilizado en este procedimiento actúa por la estimulación de un cristal de Yttrium-Aluminium Granate (YAG) enriquecido por iones de Neodymium (Nd, número atómico: 60 y peso atómico: 144,24). Su mecanismo de acción consiste en la emisión de unos pulsos muy cortos de luz de un alto poder (poder: energía/s), que provocan una explosión debido a la materia. (Ge J y otros 2000)

1.3 Prevención de la OCP

Actualmente el tratamiento de la opacidad de la cápsula posterior se basa en la realización de una abertura en la cápsula posterior con un láser granate de neodimio: itrio aluminio (Nd:YAG), o rara vez por capsulotomía quirúrgica. (American Academy of Ophthalmology,2008) Aunque es un proceder efectivo, la capsulotomía láser dista mucho de ser el método ideal puesto que se puede asociar a múltiples complicaciones como el daño de la óptica del LIO, elevación transitoria de la presión intraocular, edema macular cistoide y desprendimiento de retina entre otras, además del incremento de los costes para el sistema asistencial de salud y la falta de disponibilidad del tratamiento en muchos países subdesarrollados. (Hernández López I, 2009)

Debido a esto las investigaciones recientemente han dirigido su atención a estrategias de prevención de la OCP (7,38). El conocimiento de los mecanismos etiopatogénicos involucrados en el desarrollo de la opacidad de la cápsula posterior tras la cirugía de catarata han permitido concluir que para prevenir la aparición de esta complicación se necesita: bloquear la migración de las células epiteliales cristalinas residuales, impedir la proliferación y/o transformación fibroblástica de las LECs o eliminarlas completamente del saco capsular. (Amon M, 2001)

Las intervenciones para lograrlo se dividen a su vez en seis categorías principales:

1. Diseño de los lentes intraoculares
2. Modificaciones de las técnicas quirúrgicas
3. Implantes adicionales
4. Intervenciones farmacológicas
5. Aplicación de nuevas tecnologías

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

6. Terapia génica

1.4 Sistemas desarrollados para la identificación de OCP.

Los sistemas más usados en la actualidad por el oftalmólogo que le permite visualizar la OCP son los sistemas basados en la lámpara de hendidura, también conocida como biomicroscopio. Es un instrumento de visualización directa de la estructura ocular, pero su limitación radica en que la evaluación del grado de opacidad de la cápsula posterior la hace el médico de manera subjetiva según su apreciación visual, lo que puede variar según la experiencia del cirujano (López, 2011). Es por esta razón que algunos autores han desarrollado soluciones informáticas para lograr una evaluación objetiva y reproducible de esta complicación. A continuación, se describen algunos de estos sistemas, así como la influencia de la calidad de las imágenes en la obtención de resultados investigativos de elevada calidad.

POCO (Posterior Capsule Opacity): En este sistema las imágenes obtenidas son por retroiluminación, las mismas son evaluadas usando un análisis de los píxeles basado en la diferencia de texturas. Esta evaluación es semiobjetiva, ya que el examinador debe elegir el área afectada y clasificarla según su severidad: el programa divide la zona dentro de la capsulorrexis en 56 pequeños segmentos de igual área, el examinador debe marcar los segmentos que estén opacificados en más del 50% de su área y luego asignar a éstos un nivel de severidad (leve, moderado o severo). El resultado viene dado en forma de porcentaje de opacificación de 0 a 100%. Se calcula la severidad de la opacificación mediante la fórmula (SA, 2000):

$$[(\text{área de grado 1} \times 1) + (\text{área de grado 2} \times 2) + (\text{área de grado 3} \times 3)] / \text{área total.}$$

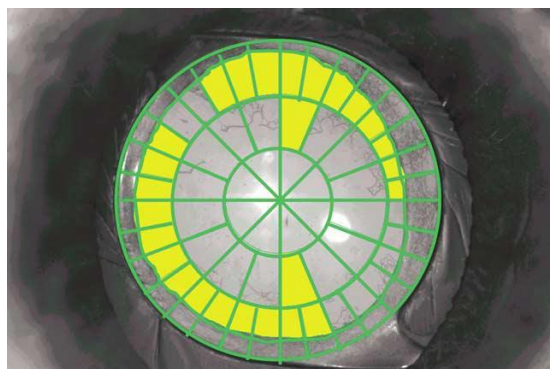


Ilustración 1: Software POCO

EPCO (Evaluación de la opacidad de la cápsula posterior) desarrollado por Tetz y Col, en 1997. Se obtienen imágenes de la lente por retroiluminación, se transfieren al programa, se marca el área a estudiar (algunos autores analizan el área tras la óptica, otros el área dentro de la capsulorrexis anterior y también es posible examinar la zona central comprendida por 3-4 mm), luego se remarcan

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

las zonas opacificadas con el ratón y se clasifican subjetivamente en 4 grupos: mínimo, leve, moderado y severo. El índice de OCP viene dado al multiplicar el grado de opacificación por el área seleccionada. El resultado obtenido es un índice de 0 a 4. Al igual que el software POCO calcula la severidad de la opacificación mediante la fórmula (SA, 2000):

$$[(\text{área de grado 1} \times 1) + (\text{área de grado 2} \times 2) + (\text{área de grado 3} \times 3)] / \text{área total.}$$

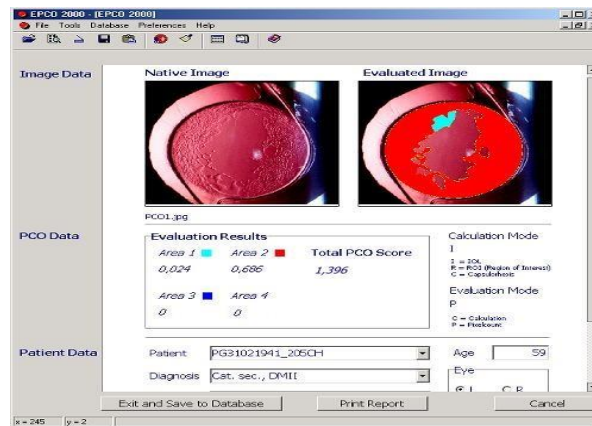


Ilustración 2: Software EPCO

AQUA (Automated Quantification of After-Cataract): Se obtienen imágenes con retroiluminación y se importan al programa. Se selecciona y analiza el área dentro de la capsulorrexis. Este programa está basado en texturas. Para calcular la no homogeneidad de la imagen se calcula la entropía (grado de desorden) de un mapa de bits. Es totalmente automático y no tiene pasos subjetivos. El resultado que se obtiene es un índice de 0 a 10 (Oliver Findl, 2003).

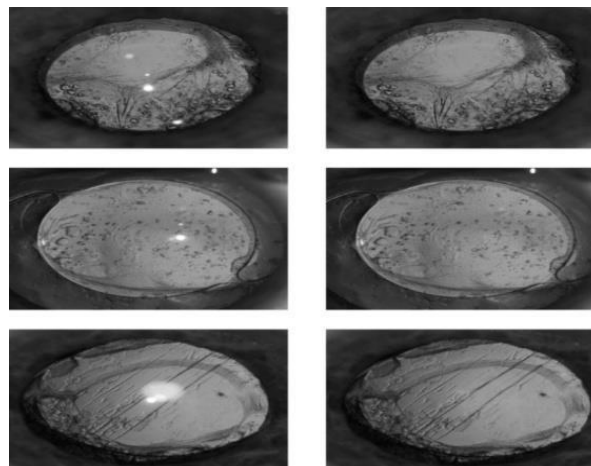


Ilustración 3: Software AQUA

AA: Este sistema fue diseñado y programado utilizando la plataforma de programación MatLab. Todo el diseño de la programación y la escritura de código fueron por TM Aslam y para mayor claridad el sistema se conoce por lo tanto como el análisis Aslam (AA) (Aslam TM, 2005). El primer

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

problema encontrado fue uno de iluminación desigual en las imágenes. Incluso con las grandes áreas de reflejos de luz aberrantes retirados, utilizando la fusión con imágenes similares, pero vírgenes, una iluminación de fondo generalmente variables de toda la imagen puede causar errores en el análisis de imágenes. Aunque un sistema para la eliminación de registro basado de imágenes de luz está siendo probado y desarrollado por TM Aslam, todavía se necesita una curva de aprendizaje larga y sigue influyendo el efecto negativo de la iluminación de fondo.

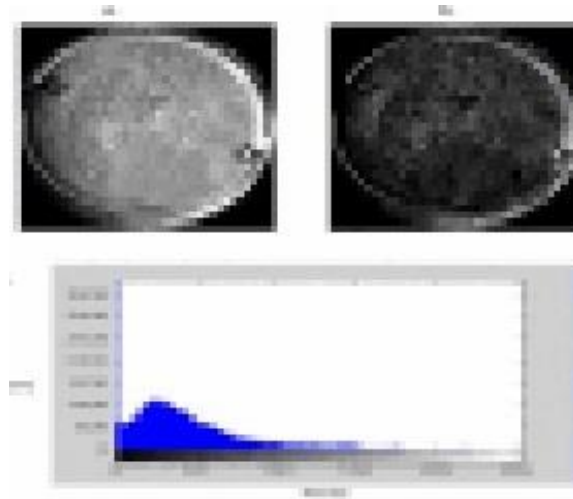


Ilustración 4: Software AA

Image J: Desarrollado por el National Institute of Health de Estados Unidos. Está basado en los cálculos realizados por Grewal y Jain, quienes determinaron que para los tomogramas del Pentacam un área de 4 mm corresponde a 16 752 píxeles en la imagen (D Grewal, 2008). Se utilizan las herramientas Find Edges (hallar bordes) y Measure (medición) para calcular el valor promedio de la intensidad de píxeles del área seleccionada en los tomogramas Scheimpflug, como indicador de la opacidad capsular posterior, expresado en unidades (U). Esta herramienta es creada con otros fines, lo que hace que la curva de aprendizaje sea larga, además de que no diferencia los píxeles de fondo a la hora de dar el resultado.

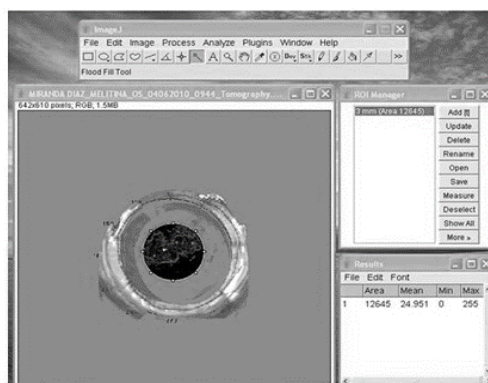


Ilustración 5: Software Image J

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Todos los sistemas anteriormente descritos tienen como deficiencia que realizan el tratamiento sobre imágenes con retroiluminación obtenidas por lámparas de hendidura, que provocan un destello de luz o ruido sobre el tomograma y de esta manera dificulta el resultado o diagnóstico del especialista.

1.5 Segmentación basada en el uso de un umbral

Este tipo de segmentación, permite separar un objeto dentro de la imagen del fondo que lo circunda, la técnica se basa en comparar alguna propiedad de una imagen con un umbral fijo o variable, realizando tal comparación para cada uno de los píxeles que conforman la imagen, si el valor de la propiedad de un píxel supera el valor del umbral, entonces el píxel pertenece al objeto, en caso contrario, el píxel pertenece al fondo. Cuando la segmentación se realiza basada en el nivel de gris de la imagen, el valor del nivel de gris de cada píxel debe ser comparado con el umbral, para decidir si tal píxel pertenece al objeto o al fondo. La imagen de salida, es una imagen binaria en la cual aquellos píxeles cuyo valor es uno, pertenecen al objeto y los píxeles cuyo valor es cero, pertenecen al fondo (Jaime Esqueda J, 2002)).

La selección del valor del umbral, se realiza generalmente a partir del histograma de la imagen. Así si una imagen está compuesta de un objeto que aparece en la escena sobre un fondo, entonces es de esperar que el histograma sea bimodal, es decir, si el objeto es más claro que el fondo, pues en el histograma aparecerán dos picos, el ubicado en los valores de gris más elevados correspondiente al objeto y otro pico para niveles de gris más bajos, correspondientes al fondo (Jaime Esqueda J, 2002)).

1.5.1 Métodos de segmentación mediante umbralización

Los métodos umbralización o del valor umbral son algoritmos cuya finalidad es segmentar gráficos rasterizados, separando los objetos de una imagen que nos interesen del resto. Con la ayuda de los métodos de valor umbral en las situaciones más sencillas se puede decidir qué píxeles conforman los objetos que buscamos y qué píxeles son sólo el entorno de estos objetos. Este método es especialmente útil para separar el texto de un documento del fondo de la imagen (papel amarillento, con manchas y arruguitas por ejemplo) y así poder llevar a cabo el reconocimiento óptico de texto (OCR) con más garantías de obtener el texto correcto. Esto es especialmente útil si queremos digitalizar libros antiguos, en los que el contraste entre el texto (que ya ha perdido parte de sus pigmentos) y el papel (oscurecido y manoseado) no es demasiado elevado. La umbralización es un método que busca segmentar imágenes escalares creando una partición binaria de las intensidades de las imágenes. De acuerdo a (Gonzales RC, 2002) la umbralización es cuando una imagen en gris es binarizada consiguiendo un umbral óptimo T y con ese valor se separan los píxeles en dos regiones, una de zonas claras y otra de zonas oscuras (Jaime Esqueda J, 2002). En la umbralización hay dos posibles situaciones:

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

1. **Umbral único** (Global thresholding). Se da cuando solamente hay dos regiones de píxeles, para separarlos se establece un único umbral T . Este tipo de umbral se obtiene fácilmente a partir de histogramas bimodales.
2. **Umbral multinivel** (Local thresholding). Dada una imagen con varios objetos, para separarlos hace falta más de un umbral, de forma que los píxeles que se encuentren entre cada par de umbrales T_i y T_j representarán a un objeto. Los umbrales elegidos pueden ser de varios tipos, dependiendo de las características tenidas en cuenta para su elección

Se escoge el método umbral único debido a que los valores intensidad de fondo son constante pues los introduce el propio PENTACAM lo que permite establecer un nivel de umbralización único idéntico para todas las imágenes.

1.6 Componentes Conexas

El uso de las componentes conexas para el procesamiento de imágenes digitales no es nuevo, pero dada la información que estas componentes recogen a partir de una determinada imagen se encuentran constantemente nuevas aplicaciones en varios mecanismos relacionados con el procesamiento de imágenes (De Giusti M y otros, 2005).

Básicamente, las componentes conexas permiten asociar elementos o partes de imágenes entre sí para luego realizar operaciones a partir de la composición de cada imagen en regiones. Una componente conexa no es más que un conjunto de puntos o píxeles de las imágenes que se han agrupado a partir de cierta característica que los identifica. Esta agrupación permite, por ejemplo, discriminar partes de una imagen, encontrar relaciones entre los elementos de una misma componente o incluso relaciones entre distintas componentes, así como aplicar ciertas operaciones y filtros a porciones de una imagen (De Giusti M y otros, 2005).

La imagen color está formada por tres canales RGB correspondiendo a Rojo, Verde y Azul y por tanto cada píxel viene dado por tres valores numéricos que se supone están en el rango de 0 a 255, 0 indicando, por ejemplo, la ausencia de un color y 255 la máxima representación de ese color en ese punto (De Giusti M y otros, 2005).

Un conjunto de componentes conexas puede verse como un conjunto de puntos agrupados en Clases de Equivalencia, donde un determinado punto "p" pertenece a exactamente una clase de equivalencia, y donde no existen clases de equivalencia vacías (De Giusti M y otros, 2005).

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

La binarización es el primer aspecto a tener en cuenta dentro del concepto de componentes conexas puesto que permite básicamente convertir imágenes con niveles de gris, en una imagen binaria (blanco y negro). De acuerdo a esta técnica, los valores de pixel en la imagen de entrada que son menores a un cierto umbral pre-especificado, son convertidos a negro, mientras que los pixeles con valores mayores al umbral, son convertidos a blanco. En algunas ocasiones se desea realizar una binarización tal que, a una banda especificada por dos umbrales, se les asigne el color blanco, mientras que los pixeles de la imagen de entrada cuyos valores están fuera de la banda especificada, se les asigne el color negro (Elizondo, 2002).

Como parte de las componentes conexas se debe tener en cuenta el concepto conectividad. En las imágenes digitales que se utilizan se podrán tener en cuenta 2 tipos de conectividad. Según el tipo de conectividad que se considere, el algoritmo de etiquetaje deberá dar resultados diferentes. Para un ejemplo concreto en este caso se supondrá que la opacidad está en negro y el fondo en blanco. Considerando la conectividad a cuatro al resultado de etiquetar dará cuatro regiones con diferentes etiquetas. Mientras que considerando la conectividad a ocho el resultado dará solamente dos regiones, puesto que en este caso la región verde y la roja formarán parte de la misma región por una vecindad diagonal y lo mismo pasa con la azul de la amarilla (Elizondo, 2002).

4-Conectividad: Considera la conectividad horizontal y vertical.



8-Conectividad: Considera la conectividad horizontal, vertical y diagonal.

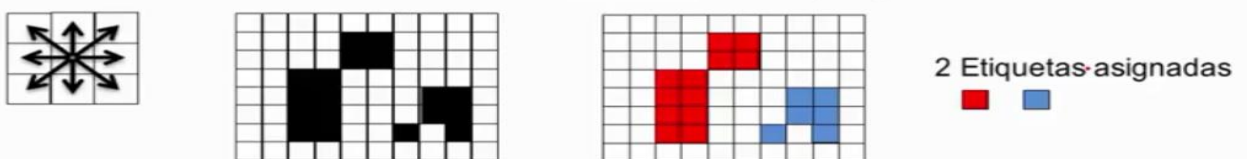


Ilustración 6: Ejemplo conectividad-4 y conectividad-8

Tomando lo antes descrito se puede concluir que será más factible la utilización de conectividad a ocho, puesto que ofrece como resultado menos regiones a analizar y las consultas a la base de datos se realizarán con menos frecuencia lo que agiliza el tiempo de respuesta del sistema.

1.7 Sistemas Basados en el conocimiento

Los Sistemas basados en Conocimiento representan un paso delante de los sistemas de información convencionales al pretender representar funciones cognitivas del ser humano como el aprendizaje y el razonamiento (Peña Ayala A, 2006). Se pueden definir como aquel programa de

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

ordenador que contiene la erudición de un especialista humano versado en un determinado campo de aplicación (Pino Diez R y otros, 2001).

Estos sistemas tienen notables ventajas sobre los expertos humanos. Por una parte, el conocimiento contenido en ellos es más fácil de documentar y de transferir que el de los expertos humanos. Por otra parte dicho conocimiento permanece tras la desaparición de los expertos, por lo que constituye lo que se ha dado en llamar la memoria institucional del organismo o empresa que lo ha desarrollado. Ahora bien, presentan claras limitaciones al compararlos con los expertos humanos. Ante todo, carecen por completo de creatividad y sentido común. Además, solo sirven para parcelas bien acotadas de conocimiento frente a la mayor universalidad del conocimiento humano (Pino Diez R y otros, 2001).

1.7.1 Composición de los Sistemas Basados en Casos

Estos sistemas emplean una amplia variedad de arquitecturas. Estas son específicas según su aplicación, sin embargo, se puede generalizar un módulo de componentes que normalmente se deben integrar en cualquier ámbito, cuyos elementos se muestran en la Ilustración 7:

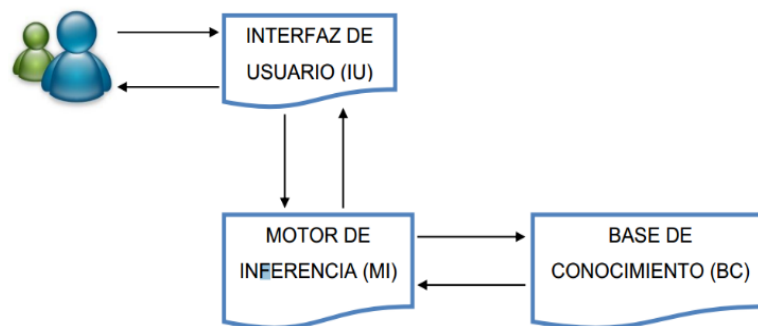


Ilustración 7: Composición de los Sistemas Basados en Casos

Una característica distintiva de los sistemas basados en casos (SBC) es la separación del conocimiento (base de conocimiento) del método de solución del problema (máquina de inferencia). La construcción de la base de conocimiento lleva implícito un arduo proceso de adquisición del conocimiento y es particular para cada sistema, por lo que será necesario construirla para cada aplicación. Sin embargo, la máquina de inferencia puede reutilizarse en la construcción de varios SBC siempre que el tipo de conocimiento y el tipo de razonamiento sea similar. La máquina de inferencia es el método implementado que utiliza el conocimiento de la base para resolver los problemas de su dominio (Soria Francis S, 2010).

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

El conocimiento que se almacena en la Base de Conocimientos es de diferentes tipos (Hernández Orallo J y otros, 2004):

- Simbólico, sobre cómo resolver los problemas del dominio.
- Probabilidades o frecuencias que modelan como se relacionan los valores de los diferentes rasgos que caracterizan el dominio.
- Pesos de una red neuronal.
- Casos o ejemplos de problemas del dominio.

Estos diferentes tipos de conocimiento dan lugar a diferentes tipos de SBC (Hernández Orallo J y otros, 2004):

- Sistemas Basados en Reglas.
- Sistemas Basados en Probabilidades.
- Sistemas Basados en Casos.

1.7.2 Sistemas Basados en Reglas

En general los primeros sistemas expertos pertenecían a este tipo de sistemas (Gámez Marín J. A. y otros, 1998). El proceso de solución de problemas en un sistema basado en reglas (SBR) es crear una cadena de inferencias que constituye un camino entre la definición del problema y su solución. Estos se definen a partir de un conjunto de objetos, que representen las variables del modelo considerado, ligadas mediante un conjunto de reglas, que representarán las relaciones entre las variables (Sánchez Ortiz S, 2011).

Las reglas nos permiten representar el conocimiento del siguiente tipo (Gámez Marín J. A. y otros, 1998):

SI la temperatura es superior a 37°C ENTONCES el paciente tiene fiebre.

SI un libro es anterior al siglo XVII Y es raro ENTONCES es un libro caro.

En general las reglas son del tipo:

SI condición ENTONCES acción

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Siendo la parte del SI de la regla (también llamada premisa o antecedente) evaluada y en caso de ser cierta, la parte ENTONCES (también llamada acción o consecuente) se activa, dando como resultado un nuevo conjunto de hechos.

A pesar de las múltiples ventajas de este tipo de sistemas y que la expresión del conocimiento basado en reglas se aproxima a la forma natural de razonar de los expertos humanos, también presentan inconvenientes, como, por ejemplo:

1. Mantenimiento de la coherencia entre las reglas de la base de conocimiento (Gámez Marín J. A. y otros, 1998):

- Un encadenamiento infinito, que aparece cuando en la base de conocimiento encontramos reglas del tipo:

SI A ENTONCES B

...

SI B ENTONCES A

- Problemas de ampliación de la base de conocimiento:

En algunas situaciones es necesario realizar una actualización del conocimiento, por ejemplo, añadiendo excepciones para un determinado conjunto de reglas o bien incorporando un nuevo conocimiento. En estos casos, y con la finalidad de mantener la coherencia entre las reglas, puede ser necesario incluir un elevado número de reglas, provocando que la base de conocimiento se haga innecesariamente grande. En estas situaciones, puede ser preferible reconstruir la base de conocimiento con el coste que esto implica.

2. Dificultades para retractarse de conclusiones anteriores (Gámez Marín J. A. y otros, 1998):

- Este problema viene provocado por el carácter modular y monótono de este tipo de sistemas. Así cuando se cumple la premisa de una regla, nos da licencia para actuar sin tener en cuenta el resto del conocimiento.

3. Opacidad (Gámez Marín J. A. y otros, 1998):

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

- La división de la base de conocimiento en pequeñas reglas tiene como ventaja que cada una de ellas es fácil de usar individualmente, ganando el sistema en modularidad. Sin embargo, se tiene que pagar un precio por ello (que en muchos casos es muy elevado) consiste en una pérdida de una perspectiva global sobre el problema que estamos considerando.

4. Ineficiencia (Gámez Marín J. A. y otros, 1998):

- Durante el proceso de inferencia, en cada iteración es necesario chequear cada regla para ver si es aplicable. Este proceso, aun cuando se han hecho avances para tratar de solucionarlo, es altamente costoso.

1.7.3 Sistemas Basados en Probabilidades

Desde un punto de vista general, en un sistema experto probabilístico la base de conocimiento está formada por un conjunto de variables X_1, \dots, X_n y una distribución de probabilidad conjunta sobre ellas $P(X_1, \dots, X_n)$. Por otro lado, un motor de inferencia será aquel que nos permita actualizar nuestra información sobre determinada variable (o conjunto de ellas) X , ante la presencia de un conjunto de hechos, evidencias o síntomas determinados E . En teoría de la probabilidad este motor de inferencia no es más que el cálculo de la probabilidad condicional $P(X|E)$ (Gámez Marín J. A. y otros, 1998).

En este tipo de sistemas la adquisición del conocimiento consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimiento. Dentro de sus principales dificultades encontramos:

- No son viables para explicar el razonamiento, ya que los métodos y modelos que utiliza están aún lejos de ofrecer explicaciones comprensibles (Soria Francis S y otros, 2010).
- No son factibles para todo tipo de dominio, pues se dificulta construir las redes con ayuda de expertos humanos cuando existen carencias de conocimiento (Soria Francis S y otros, 2010).
- El alto número de parámetros que manejan, lo que hace que sea difícil su especificación y definición (Pérez Lara A y otros, 2011).
- La implementación del motor de inferencia es más complicada y se vuelve costosa computacionalmente en la mayoría de los casos (Pérez Lara A y otros, 2011).

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

1.7.4 Razonamiento Basado en Casos

El enfoque de razonamiento basado en casos (RBC) se basa en dos principios sobre la naturaleza del mundo. El primer principio es que el mundo es regular: problemas similares tienen soluciones similares.

En consecuencia, las soluciones para problemas similares anteriores son un buen punto de partida para la nueva resolución de problemas. El segundo principio es que los tipos de problemas de un agente de encuentros tienden a repetirse. En consecuencia, problemas en el futuro es probable que sean similares a los problemas actuales. El RBC es una estrategia de razonamiento efectivo (Leake D, 1996).

Este sistema de razonamiento se basa en una unidad mínima llamada caso, el cual tiene dos componentes: rasgos predictores (descripción del problema) y rasgos objetivos (solución del problema). Un caso es una pieza contextualizada de conocimiento, la cual representa una experiencia que enseña una lección fundamental para el logro de los objetivos del razonador (Kolodner J, 1993).

El RBC es un paradigma para la resolución de problemas nuevos partiendo del análisis y adaptación de soluciones que fueron dadas a problemas previos, que se encuentran almacenados y organizados en una biblioteca o base de casos. Cuando un nuevo problema es encontrado, el RBC recuerda casos similares y adapta las soluciones que funcionaron en el pasado al problema actual (Soria Francis S y otros, 2010). Si encuentra un caso similar, retorna la solución acompañando a ese caso. Sino encuentra ningún caso idéntico, el razonador buscará casos de entrenamiento que tienen componentes similares a aquellas del nuevo caso. El sistema es capaz de aprender a través de la agregación de nuevos casos (Bello Pérez R. E, 2002).

Problemas cuya solución posea forma similar presentan soluciones similares. En consecuencia, con lo anterior, soluciones de problemas previos similares al actual son un punto de partida útil para las soluciones de un nuevo problema (Soria Francis S y otros, 2010). Además, incrementa el conocimiento almacenando el nuevo caso para ser aplicado en situaciones futuras. Esto permite que el conocimiento se mantenga actualizado en todo momento (Febles Rodríguez J. P. y otros, 2002).

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Componentes del Razonamiento Basado en Casos.

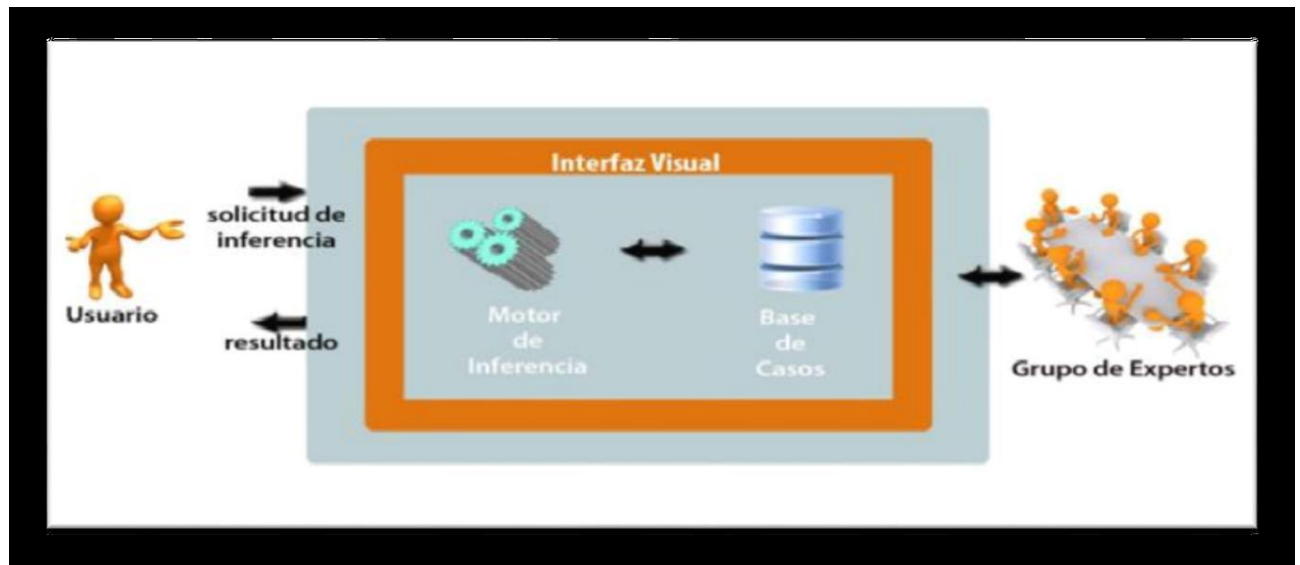


Ilustración 8: Componentes del Razonamiento Basado en Casos

En la Base de Casos (BC), se almacena el conocimiento necesario para resolver los problemas del dominio de aplicación, puede ser representada a través de una tabla en la cual las columnas son etiquetadas por variables que representan los rasgos predictores y los rasgos objetivos (decisiones) y las filas representan casos. El motor de Inferencia (MI), es un procedimiento basado en un esquema de razonamiento o método de solución de problemas, que utiliza el conocimiento para resolver los problemas de ese dominio y la Interfaz de Usuario (IU) permite la comunicación entre el usuario y el ordenador (Bello Pérez R. E, 2002). La interfaz visual posibilita a los expertos, de una forma interactiva y amigable, introducir en la BC todo el conocimiento sobre el tema en cuestión. También es el componente a través del cual el usuario solicita al razonador que haga la búsqueda que requiere (Rodríguez K, 2010).

Por las utilidades del RBC ha sido utilizado en diversos dominios de aplicación, como el campo jurídico, la medicina, el diseño, tareas de planificación, etc. Su uso ha brindado muchas ventajas, como las siguientes:

1. El esfuerzo en la solución de problemas puede ser capturado para ahorrar trabajo en el futuro.
2. Experiencias previas que hayan sido exitosas pueden ser utilizadas para justificar nuevas soluciones.
3. Experiencias previas que no hayan sido exitosas se pueden utilizar para anticipar problemas.
4. La comunicación entre el sistema y los expertos se realiza en base a ejemplos concretos, es decir, el sistema explica sus decisiones citando precedentes.

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

5. Trabaja a partir de casos existentes por lo que no se requieren demasiadas entrevistas con los expertos, simplificándose la adquisición del conocimiento.
6. Es un algoritmo de aprendizaje acumulativo, el aprendizaje tiene lugar tan pronto como un nuevo ejemplo está disponible, sin excesivo costo computacional.
7. Permite proponer soluciones para los problemas rápidamente, evitando el tiempo necesario para derivar respuestas desde el estado inicial de un proceso de búsqueda de soluciones.
8. Permite proponer soluciones en dominios que no se comprenden completamente.
9. Los casos ayudan a focalizar el razonamiento sobre las partes importantes de un problema señalando qué rasgos del problema son importantes.

A pesar de las numerosas ventajas del RBC existen elementos que lo hacen inconveniente para algunos dominios de aplicación como los siguientes:

1. El sistema no explora todo el espacio de soluciones, por lo que no siempre encuentra la solución óptima.
2. Requiere de una base de datos considerablemente grande y bien seleccionada.
3. La consistencia entre varios casos es difícil de mantener. Depende de una adecuada función de semejanza la cual no es fácil de encontrar para cada aplicación.

1.7.5 Valoración del estudio de los Sistemas Basados en Casos

Se concluye que no todos los paradigmas para crear sistemas basados en el conocimiento facilitan la identificación de opacidad en imágenes del Pentacam. Se escoge el uso del Razonamiento Basado en Casos, dado que el conocimiento puede expresarse en función de casos o vivencias, además representan experiencias anteriores, cada caso constituye una solución, sin necesitar una interrelación entre ellos para obtener una solución al problema. Los sistemas anteriormente expuestos presentan deficiencias como la falta de viabilidad para explicar el razonamiento, la posibilidad de encadenamiento infinito al insertar una nueva regla y el uso de la probabilidad no es una teoría aceptada en un diagnóstico médico, sobre el dominio de aplicación de la presente investigación.

1.7.6 Tipos de Búsqueda en los Sistemas Basados en Casos

Durante la recuperación de los casos, se necesita realizar un proceso de búsqueda, el cual brindará como resultados el o los casos que podrán ser tomados como solución al problema. A continuación, se expone una breve explicación de los tipos de búsqueda que se pueden utilizar durante el proceso antes mencionado:

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

- Búsqueda por analogía

La búsqueda por analogía, significa determinar si la descripción del problema es equivalente a la descripción de alguno de los casos almacenados, es decir, probar la equivalencia entre los valores de los rasgos predictores, aunque sean diferentes, comprobando si semánticamente tienen igual significado (Corona Prendes F. D y otros, 2013).

Para este análisis se utiliza una red semántica, en la cual los nodos representan valores y rasgos. Además, los enlaces indican el grado de relación entre ellos. Dos valores son equivalentes si están enlazados con el mismo concepto por el mismo tipo de relación, a partir del mismo punto de vista (Agudo B, 2001).

- Búsqueda por semejanzas

La búsqueda por semejanzas, tiene como objetivo obtener el caso almacenado que más se parece al nuevo problema. Existen diversas formas de medir el grado de semejanza, estas pueden ser, desde expresiones analíticas de distancia hasta la descripción de algoritmos para obtener el grado de parecido entre dos objetos. Esto depende del tipo de información que se presente (Rodríguez Y, 2003).

Considerando que un caso está descrito por un número de características, una aproximación sencilla sería contabilizar los atributos comunes entre el caso almacenado y el nuevo problema. Se suele distinguir entre medidas de semejanzas locales y globales, las primeras calculan la semejanza entre valores de un mismo atributo en los dos casos y la segunda combina los resultados de aplicar la similitud local a cada uno de los atributos de los casos que se están comparando. Una estrategia para definir semejanza entre atributos simbólicos, consiste en utilizar una tabla de semejanzas, donde se obtiene explícitamente la similitud entre cada par de valores (Rodríguez Y, 2003).

1.8 Justificación de la selección de Búsqueda por Semejanza

Para la identificación de regiones con opacidad en imágenes médicas tomadas con el PENTACAM, se propone usar una búsqueda por semejanza. Al existir un caso nuevo se buscarán los más semejantes en la BC para realizar así la recuperación de los casos, usando las funciones de semejanza de Rasgos y Casos propuestas por (Ruíz Shulcloper J y otros, 1995).

1.9 Base de Conocimientos

En las últimas décadas la forma de enfrentarnos a la obtención y gestión de conocimientos ha conocido un desarrollo notable gracias a los avances registrados en el campo de la Inteligencia

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Artificial. Pero quizás uno de los factores que más ha contribuido a este desarrollo es la introducción de la distinción entre bases de datos y bases de conocimiento (Borja A, 2005).

El autor (Moreno Ortiz, 2000) define el concepto de "base de conocimientos" como el sistema que contiene una representación simbólica de ciertas entidades, objetos, relaciones y hechos que constituyen el acervo de conocimientos involucrados en el desempeño de una tarea.

La base de conocimientos contiene representaciones simbólicas del conocimiento de los expertos, definiciones de los términos, interconexiones y relaciones causa-efecto entre los componentes (Borja A, 2005).

1.10 Herramientas usadas en el desarrollo

1.10.1 Metodología de Software.

El desarrollo de software no es una tarea fácil; prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales o robustas que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán, o sea se basa en el uso exhaustivo de documentación durante todo el ciclo del proyecto. Por otra parte tenemos las metodologías ágiles o ligeras las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (Canós José H, 2003). Seleccionar una metodología apropiada, resulta en ocasiones un factor esencial para obtener un software con la calidad esperada.

Luego del análisis de las metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software; el cliente es parte del equipo de desarrollo, el equipo de desarrollo es de solo una persona, además de la dificultad para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo.

Metodología de software XP

Durante la elaboración de la aplicación se utiliza para establecer el control y utilizar un marco de trabajo definido y de probada eficiencia, la metodología de software XP programación extrema, dado que una de las prácticas más significativas que posee, es que con XP es posible simplificar el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que no se necesiten. El ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

en el código. XP tiene como objetivo fundamental la satisfacción del cliente. Es recomendable utilizar esta metodología debido a que los requisitos tienen altas probabilidades de cambiar con el tiempo y el proyecto cuenta con un grupo pequeño de programadores. Además, esta metodología ha sido la utilizada durante el desarrollo del software PANDOC.

Algunas de las características de XP son (Sánchez Mendoza, 2004):

- Pruebas unitarias: se basa en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.
- La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.

¿Qué nos brinda XP? (Sánchez Mendoza, 2004):

- Comienza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- El cliente o el usuario se convierte en parte del equipo.

XP consta de 4 fases:

- Planificación
- Diseño
- Desarrollo
- Pruebas

1.10.2 Lenguaje de Programación.

Para el desarrollo de esta aplicación se escogió como lenguaje de programación a Java, este lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel. La selección de dicho lenguaje se debe a que es puro orientado a objetos, lo que propina una gran reusabilidad, independencia de la plataforma, esto significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware (Oracle, 2015). Existe además variada documentación y una amplia comunidad sobre este lenguaje y el mismo se caracteriza por ser muy utilizado en la implementación de aplicaciones que incorporan técnicas de inteligencia artificial.

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

Características fundamentales de Java:

- **Simple:** elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.
- **Robusto:** maneja la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución.
- **Portable:** un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java, ya que su código compilado es interpretado.
- **Multiproceso:** puede ejecutar diferentes líneas de código al mismo tiempo.
- **Dinámico:** no es necesario que compile todas las clases de un programa para que este funcione. Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases.
- **Interpretado:** corre en máquina virtual.

1.10.3 Entorno integrado de Desarrollo.

Para el desarrollo de la solución se seleccionó como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) el NetBeans 8.0. Es libre y de código abierto; posee todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente. Java es Además es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris (Oracle, 2015).

1.10.4 Modelado de Lenguaje.

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo necesario para

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos.

Ofrece interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo. Hace posible la generación de código Java desde los diagramas.

1.10.5 Sistema Gestor de Base de Datos.

Como sistema gestor de base de datos (SGBD) se utilizó SQLite, es un SGBD relacional que está contenida en una relativamente pequeña (~500kb) biblioteca en C. Es un proyecto de dominio público creado por Dr. Richard Hipp. A diferencia de los SGBD cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones (SQLite, 2015).

Esto reduce la latencia en el acceso a la base de datos (BD), debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la BD (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de BD al principio de cada transacción. Este SGBD dispone de una completa interfaz orientada a objetos, con distintas funciones que facilitan la manipulación de datos y posee funciones muy similares a las que se pueden manejar con MySQL.

Combina el motor y la interfaz de la BD en una única biblioteca, y almacena los datos en un único archivo de texto plano. Esto hace que cada usuario pueda crear tantas BD como desee sin la necesidad de la intervención de un administrador de BD que gestione los espacios de trabajo, usuarios y permisos de acceso

La potencia de SQLite se basa fundamentalmente en la simplicidad, lo que hace que no sea una buena solución en entornos de tráfico muy elevado y/o alto acceso concurrente a datos, además encapsula toda la BD en un único fichero. El hecho de almacenar toda la BD en un único archivo, facilita la portabilidad de los datos, y solamente tiene la restricción del espacio de disco asignado.

Se puede utilizar SQLite de dos formas:

- Como gestor de base de datos local en un computador. De esta forma, se pueden gestionar BD con SQLite igual que si se estuviese trabajando con un sistema gestor de base de datos como MySQL, sin necesidad de instalar nada, ya que SQLite se compone de un único archivo ejecutable.

Capítulo 1. Opacidad capsular en pacientes sometidos a cirugía de catarata: Bases conceptuales

- Como una extensión más de PHP, utilizando las funcionalidades de SQLite configuradas, o bien como módulo de PHP, o como biblioteca; sin necesidad de tener instalado o conectar con un servidor de base de datos. Ofrece una rápida interfaz de la BD almacenada en el archivo de texto plano.

Varios procesos o hilos pueden acceder a la misma BD sin problemas. Varios accesos de lectura pueden ser servidos en paralelo. Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente. En caso contrario, el acceso de escritura falla devolviendo un código de error (o puede automáticamente reintentarse hasta que expira un timeout configurable). Esta situación de acceso concurrente podría cambiar cuando se está trabajando con tablas temporales. Sin embargo, podría producirse un deadlock debido al proceso multihilo.

SQLite Admin es un administrador de bases de datos SQLite que básicamente permite (SQLite Admin, 2015):

- Abrir, crear, compactar bases de datos.
- Crear, renombrar y eliminar tablas vacías.
- Crear, renombrar y eliminar columnas.
- Crear, modificar y eliminar registros.
- Encriptar o reencriptar bases de datos
- Correr sentencias SQL personalizadas.

Conclusiones del capítulo

Para poder evaluar objetivamente, el efecto de las diferentes intervenciones que actualmente se realizan para reducir la aparición de la OCP se realiza un estudio y análisis de dicha complicación posoperatoria, siendo esta la enfermedad más frecuente tras una cirugía exitosa de catarata. Luego de un estudio de las técnicas posibles a utilizar para la identificación automática de las regiones con opacidad en tomogramas del Pentacam, se determinó utilizar el algoritmo de componentes conexas. Se llevó a cabo el estudio de los SBC, para tener un mayor entendimiento y conocer más a fondo el RBC como técnica de la IA empleada. Se describen los conceptos principales de la ingeniería del conocimiento, los diferentes tipos de conocimiento y los elementos que componen el almacenamiento de este en las BC. Además, se seleccionaron y describieron las herramientas y tecnologías a ser usadas en el desarrollo de la propuesta de solución. El estudio realizado sobre metodologías de desarrollo permitió arribar a la conclusión de que XP, debido a sus características, es la metodología más apropiada para guiar el desarrollo del sistema. El estudio realizado sobre herramientas y tecnologías para el desarrollo de software, permitió seleccionar Java como lenguaje de programación, NetBeans como IDE, SQLite como SGBD y Visual Paradigm como herramienta para el modelado.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Este capítulo tiene como objetivo plantear el problema el cual dará origen al sistema que se va a desarrollar. Se nombran los procesos que se necesitan automatizar. Se definen los requisitos ya sean funcionales o no funcionales que el sistema debe respetar y cumplir. Se describen las historias de usuario que permitirán planificar las iteraciones en que se desarrollará el sistema además del plan de entrega de versiones especificándose la fecha en las que se entregan las versiones del sistema al cliente.

2.1 Fase de Planificación

La metodología XP define como fase inicial del desarrollo de software la planificación. Durante el transcurso de esta etapa se realiza el proceso de identificación y elaboración de las historias de usuario (HU), además el equipo de trabajo se familiariza con las tecnologías y herramientas seleccionadas para el desarrollo. El cliente define el nivel de prioridad con que se deben implementar las HU, así como la estimación del esfuerzo que costará implementar las mismas. El resultado de la presente fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración, de forma tal que sea una guía durante el desarrollo (Beck, 2002). Además, se enunciarán algunos de los requisitos no funcionales que de manera íntegra satisfacen las necesidades del cliente.

2.1.1 Historias de Usuario

Entre los artefactos que define la metodología XP se encuentran las Historias de Usuarios (HU) que son utilizadas para especificar las funcionalidades que brindará el sistema. Cada HU es una representación de un requerimiento de software escrito en una o dos frases, utilizando el lenguaje común del usuario. Representan una forma rápida de administrar los requerimientos de los usuarios, sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos (Beck, 2002). De acuerdo a la prioridad asignada por el cliente a cada HU y teniendo en cuenta la complejidad y riesgo determinado por el programador, se realiza la estimación de cada una de las identificadas. La unidad de estimación es el punto, un punto equivale a una semana ideal de programación.

Como resultado del trabajo realizado durante la fase se identificaron un total de 13 HU, a continuación, se muestran algunas de ellas.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Historia de usuario	
Número: 1	Nombre: Construir BC
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se realiza la declaración y se construye la base de conocimientos	
Observaciones:	

Tabla 1: HU 1. Construir BC

Historia de usuario	
Número: 11	Nombre: Calcular certeza de solución
Iteración asignada: 4	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se calcula la certeza de la solución.	
Observaciones:	

Tabla 2: HU 2. Calcular certeza de solución

2.1.2 Estimación de esfuerzos por Historias de Usuario

Como parte del desarrollo de la solución en cuestión, se realiza una estimación de esfuerzo por cada una de las Historias de Usuario, plasmando los resultados siguientes:

No	Historias de usuario	Puntos de estimación
1	Construir la BC	2
2	Conectar la BC con la aplicación	1

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

3	Reconocer imagen de tomograma	2
4	Reconocer el centro de la estructura ocular	2
5	Llenar la BC	3
6	Obtener el nuevo caso	2
7	Definir región del caso	2
8	Cargar Casos similares de la BC	2
9	Calcular similitud de rasgos	2
10	Calcular similitud de casos	2
11	Calcular certeza de solución	2
12	Brindar datos al usuario	2
13	Recuperar el caso a la BC	2

Tabla 3: Puntos de estimación por historias de usuario

2.1.3 Requisitos no funcionales

Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación. Se relacionan con atributos de calidad sistémica que conllevan al buen funcionamiento del sistema como un todo. A continuación se relacionan algunos de ellos:

RNF1: Usabilidad

- El sistema podrá ser usado por personas sin avanzados conocimientos informáticos.
- El sistema tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

RNF3: Soporte

- La herramienta recibirá mantenimiento en el período de tiempo determinado por el equipo de trabajo y los clientes involucrados en el desarrollo de un sistema que haga uso de la herramienta.
- El software contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento y configuración.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

RNF4: Interfaz

- El sistema debe contar con una interfaz fácil de usar, sencilla, amigable, permitiendo que los usuarios sean capaces de interactuar con la aplicación.
- Los mensajes, títulos y demás textos que aparezcan en la interfaz del sistema deben aparecer en idioma español.
- Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español y tener una apariencia uniforme.
- Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error.

2.1.4 Plan de iteraciones

Luego de identificar y definir las HU y estimar el esfuerzo propuesto para la realización de cada una de ellas, se precisa establecer el contenido de trabajo de las mismas y es aquí donde se establece el plan de iteraciones, regulando la cantidad de HU a implementar dentro del rango establecido por la estimación efectuada. Tomando como referencia los aspectos antes tratados la aplicación que se pretende construir se desarrollará en 4 iteraciones, explicadas más detalladamente a continuación:

Iteración 1

La iteración tiene como finalidad implementar las HU que se consideraron más necesarias atendiendo a su relevancia e impacto. En esta iteración se construye la BC y se conecta con la aplicación; además se implementan las funcionalidades que permiten reconocer imagen de tomograma y reconocer el centro de la estructura ocular.

Iteración 2

En esta iteración se realizan todas las HU relacionadas con la ingeniería y gestión del conocimiento, se llena la BC, se realiza la obtención del nuevo caso, además de definir la región del caso.

Iteración 3

En esta iteración se realizan las HU que definen la mayor parte del motor de inferencias, ya que donde se implementan los elementos y funciones de similitud, definiendo también como cargar los casos similares de la BC, calcular similitud de rasgos y calcular similitud de casos.

Iteración 4

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

En esta iteración se realizan todas las HU relacionadas con la interacción con el especialista y el método de recuperación definido en el motor de inferencias, aquí se calcula la certeza de solución, se le brindan los datos al usuario y se realiza la recuperación del caso a la BC.

A modo de resumen se presenta a continuación la tabla que muestra las 4 iteraciones analizadas previamente con las HU que incluyen y su duración:

Iteraciones	Historias de usuario	Duración
Iteración 1	Construir la BC	7
	Conectar la BC con la aplicación	
	Reconocer imagen de tomograma	
	Reconocer el centro de la estructura ocular	
Iteración 2	Llenar la BC	7
	Obtener el nuevo caso	
	Definir región del caso	
Iteración 3	Cargar Casos similares de la BC	6
	Calcular similitud de rasgos	
	Calcular similitud de casos	
Iteración 4	Calcular certeza de solución	6
	Brindar datos al usuario	
	Recuperar el caso a la BC	

Tabla 4: Plan de duración de las iteraciones

2.1.5 Plan de entrega

El plan de entregas tiene como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una.

	Iteración 1	Iteración 2	Iteración 3	Iteración 4
Cantidad de HU	4	3	3	3
Fecha de entrega	14/01/2016	4/03/2016	15/04/2016	27/05/2016

Tabla 5: Plan de entrega

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

2.2 Implementación de la propuesta de solución

A continuación, se hace referencia a los principales elementos que componen la base de conocimientos y el motor de inferencias. Siguiendo los planteamientos de la metodología XP debe ser un diseño simple y de fácil interacción, basándose fundamentalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC). También se detallan las cuatro iteraciones llevadas a cabo, así como las tareas generadas por cada HU.

2.2.1 Tarjetas CRC

La metodología XP en lugar de utilizar diagramas para desarrollar modelos representa las clases mediante tarjetas. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo a definir actividades durante el diseño del sistema. Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Clase: Geometria_Computacional	
Responsabilidades	Colaboradores
1. Permite crear la elipse simulada con las rectas que delimitan cada una de las regiones. 2. Permite identificar la región donde se localiza el caso que se está analizando	1. Lienzo 2. Pixel

Tabla 6: Tarjeta CRC para la clase Geometria_Computacional

2.2.2 Componentes conexas

La imagen digital resultante del PENTACAM será la entrada a nuestro sistema que al aplicarle el concepto componentes conexas dará como salida la ventana o conjunto de ventanas que representan las regiones que se han detectado. Para esto se aplica un algoritmo de detección de regiones conexas atendiendo a lo siguiente:

Conjunto de pasos lógicos de componentes conexas

1. Binarizar la imagen.
2. Etiquetar las regiones.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

- Etiquetar los pixeles y guardar posibles equivalencias.
- Resolver equivalencias.
- 3. Marcar las regiones.

1 - Binarizar la imagen

Partiendo del descriptor más simple, el cuál utiliza características propias del píxel, en este caso las características de color. Dada una imagen de entrada el objetivo es detectar la opacidad. Para hacerlo simplemente usaremos el color distinto del color de fondo. Se entenderá como opacidad una región conectada o no de pixeles de color distinto al fondo que será una simplificación, pero será de gran utilidad inicialmente.

Se obtendrá el valor RGB de cada píxel como descriptor, seguidamente se establecerán los límites de cada característica de color para determinar los márgenes de tipo opacidad como variante de clasificación.

Los resultados de aplicar lo antes descrito sobre una imagen se resume de la siguiente manera:

La clasificación de la opacidad usando la variante definida anteriormente conlleva a que se ha calculado el centro y el área que se va a analizar. Dicha área se encuentra comprendida entre dos elipses con radio 3 y 5 mm respectivamente con respecto al centro de la estructura ocular. En negro aparecerán los pixeles que identifican la opacidad y en blanco los pertenecientes al color de fondo.

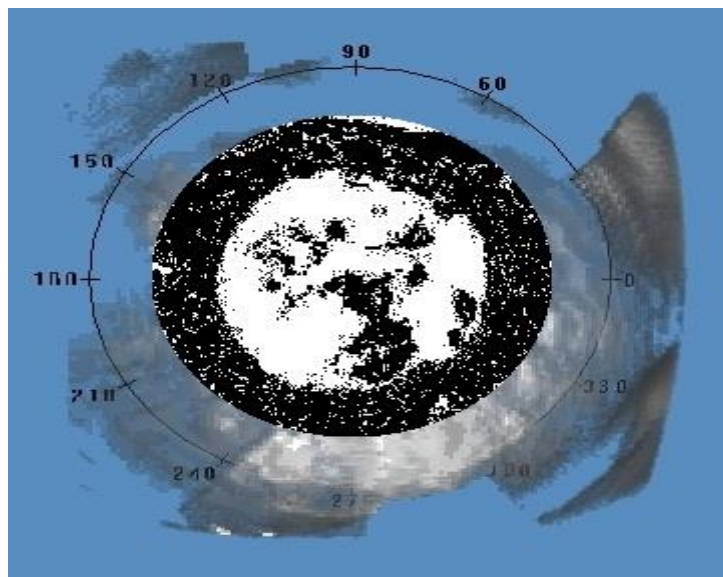


Ilustración 9: Imagen del Pentacam binarizada

Para el proceso de binarización se diseñó el siguiente algoritmo:

Algoritmo 1: Binarización

Entradas: Ellipse2Dmenor(EMEN), Ellipse2Dmayor(EMAY)

Salida: imagenBinarizada(IB)

1: mediaPixel \leftarrow 0

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

```
2: copialmg ← Imagen(ancho, largo, tipo)
3: copialmg ← Datos de ImageActual
4: para todo pixel ∈ EMAY hacer
5: para todo pixel ∈ EMEN hacer
6: si pixel ∈ EMEN y pixel ∈ EMAY entonces
7: colorAux ← Nuevo color creado con las características del pixel
8: mediaPixel ← RGB de colorAux
9: si mediaPixel ≤ 141 y mediaPixel ≥ 139 entonces
10: copialmg ← color blanco
11: Fin si
12: sino entonces
13: copialmg ← color negro
14: Fin sino
15: Fin si
16: Fin para
17: Fin para
18: retornar copialmg Binarizada
```

2 - Etiquetar las regiones

A partir de este resultado anterior ahora se planteará la generación de las ventanas que han de localizar la opacidad encontrada. Esta generación de ventanas se realizará en dos pasos. Primero se etiquetarán las componentes conexas de la imagen, es el procedimiento conocido como labeling en inglés. Y seguidamente para cada región etiquetada se calculará la ventana que la contiene.

Se aplicará inicialmente un algoritmo de etiquetaje que plantea: Dada una imagen binaria retorna una imagen con etiquetas numéricas, de manera que todos los pixeles que pertenecen a una misma región conectada compartan la misma etiqueta.

El algoritmo está basado en realizar dos recorridos sobre la imagen, Los recorridos pasan por cada pixel de la imagen de izquierda a derecha y de arriba abajo. Se supondrá que la imagen binaria que se desea etiquetar, los pixeles que forman parte de una región están en **1** y los que no están en **0**.

Paso 1: Etiquetar cada pixel de la imagen atendiendo a sus vecinos superior e izquierdo según la conectividad y guarda posibles equivalencias.

En el primer recorrido se asignará una etiqueta a cada pixel que esté a uno. El valor de la etiqueta dependerá de sus vecinos superior e izquierdo y guardará las posibles equivalencias. Existirán tres situaciones diferentes:

Caso 1: Si (Vecinos=0) **entonces** Asignar etiqueta NUEVA.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante

Componentes Conexas

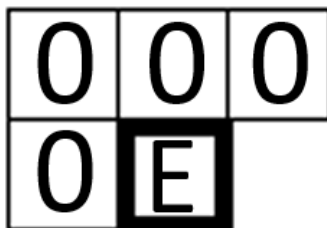


Ilustración 10: Caso 1_Conectividad-8

Caso 2: Si (Un Vecino!=0) **entonces** Asignar la etiqueta del VECINO.

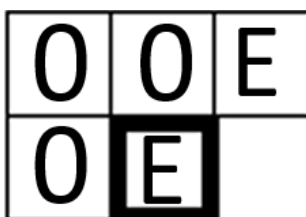


Ilustración 11: Caso 2_Conectividad-8

Caso 3: Si (Más de un Vecino!=0) **entonces** Asignar una etiqueta del VECINO e indicar equivalencia.



Ilustración 12: Caso 3_Conectividad-8

Caso uno y más simple es cuando el pixel actual que se analiza que por tanto está a uno y sus vecinos superior e izquierdo están todos a cero. Este es el caso en que se podría decir que se ha encontrado una nueva región. En conectividad a ocho se evalúan los píxeles izquierdo y superior, además del que va en diagonal y derecho superior. Así pues, para este caso de toda la vecindad a cero se asigna el pixel al pixel una etiqueta nueva. Se deberá mantener un contador de etiquetas que vaya creciendo a medida que vamos encontrando nuevas regiones.

Caso dos, es cuando el pixel actual tiene un vecino distinto a cero, pero solo uno. En este caso se asignará al pixel actual la misma etiqueta que la del vecino que está a uno. Simplemente se encontrará un nuevo pixel de una región que ya conocíamos.

Caso tres, es cuando el pixel actual tiene más de un vecino que está a uno. En este caso se podría asignarle dos etiquetas. La solución será asignar una de estas diversas etiquetas con un criterio preestablecido, (por ejemplo, la más inferior a la izquierda). Y se guardará la equivalencia que se

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

ha encontrado. En este caso se ha identificado un pixel que por los vecinos que tiene pudiera pertenecer a dos regiones. Por tanto, a partir de ahora la región etiquetada como E1 y la etiquetada como E2 serían la misma región. De esta manera se guardará esta equivalencia.

Para llevar a cabo el paso de etiquetado y guardar equivalencias se diseña el algoritmo que se muestra a continuación:

Algoritmo 2: Etiquetar8

Entradas: imagenBinarizada (imageB), Ellipse2Dmenor(EMEN), Ellipse2Dmayor(EMAY).

Salida: no devuelve ningún resultado

1: Raster \leftarrow *raster de imageB*

2: contAux \leftarrow 0

3: **para** $y \in EMAY$ **hacer**

4: **para** $x \in EMAY$ **hacer**

5: **si** $x, y \in EMAY$ **y** $x, y \in EMEN$ **entonces**

6: *cont* \leftarrow 0

7: **si** *arrayEtiq*[Izquierdo] \neq *vacio* **y** *imageBcolor*(Izquierdo) **es igual** *colorOpacidad* **entonces**

8: **si** *cont* $>$ 0 **entonces**

9: **si** *etiquetaDpixel* [Izquierdo] \neq *etiquetaDpixel* [actual] **entonces**

10: **si** **!EstaRelaciona**(*etiquetaDpixel* [Izquierdo], *etiquetaDpixel* [actual]) **entonces**

// La lista de etiquetas en la posición de etiqueta actual va a añadir a su lista de relaciones la etiqueta del pixel izquierdo

11: *listaEtiq* (*etiquetaDpixel* [actual]) \leftarrow *listaRelac* **adicionar** *etiquetaDpixel* [Izquierdo]

12: **Fin si**

13: **Fin si**

14: **Fin si**

15: **Sino entonces**

16: *etiquetaDpixel* [actual] \leftarrow *etiquetaDpixel* [Izquierdo]

17: **Fin sino**

18: *cont* \leftarrow +1

19: **Fin si**

20: **si** *arrayEtiq*[IzqArriba] \neq *vacio* **y** *imageBcolor*(IzqArriba) **es igual** *colorOpacidad* **entonces**

21: **si** *cont* $>$ 0 **entonces**

22: **si** *etiquetaDpixel* [IzqArriba] \neq *etiquetaDpixel* [actual] **entonces**

23: **si** **!EstaRelaciona**(*etiquetaDpixel* [IzqArriba], *etiquetaDpixel* [actual]) **entonces**

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

// La lista de etiquetas en la posición de etiqueta actual va a añadir a su lista de relaciones la etiqueta del pixel izquierdo superior

24: *listaEtiq (etiquetaDpixel [actual])* ← *listaRelac* **adicionar** *etiquetaDpixel [IzqArriba]*

25: **Fin si**

26: **Fin si**

27: **Fin si**

28: **Sino entonces**

29: *etiquetaDpixel [actual]* ← *etiquetaDpixel [IzqArriba]*

30: **Fin sino**

31: *cont* ← +1

32: **Fin si**

33: **si** *arrayEtiq[superior] != vacio* **y** *imageBcolor(superior)* **es igual** *colorOpacidad* **entonces**

34: **si** *cont > 0* **entonces**

35: **si** *etiquetaDpixel [superior] ≠ etiquetaDpixel [actual]* **entonces**

36: **si !EstaRelaciona**(*etiquetaDpixel [superior]*, *etiquetaDpixel [actual]*) **entonces**

// La lista de etiquetas en la posición de etiqueta actual va a añadir a su lista de relaciones la etiqueta del pixel superior inmediato

37: *listaEtiq (etiquetaDpixel [actual])* ← *listaRelac* **adicionar** *etiquetaDpixel [superior]*

38: **Fin si**

39: **Fin si**

40: **Fin si**

41: **Sino entonces**

42: *etiquetaDpixel [actual]* ← *etiquetaDpixel [superior]*

43: **Fin sino**

44: *cont* ← +1

45: **Fin si**

46: **si** *arrayEtiq[DerArriba] != vacio* **y** *imageBcolor(DerArriba)* **es igual** *colorOpacidad* **entonces**

47: **si** *cont > 0* **entonces**

48: **si** *etiquetaDpixel [DerArriba] ≠ etiquetaDpixel [actual]* **entonces**

49: **si !EstaRelaciona**(*etiquetaDpixel [DerArriba]*, *etiquetaDpixel [actual]*) **entonces**

// La lista de etiquetas en la posición de etiqueta actual va a añadir a su lista de relaciones la etiqueta del pixel derecho superior

50: *listaEtiq (etiquetaDpixel [actual])* ← *listaRelac* **adicionar** *etiquetaDpixel [DerArriba]*

51: **Fin si**

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

```
52: Fin si
53: Fin si
54: Sino entonces
55: etiquetaDpixel [actual] ← etiquetaDpixel [DerArriba]
56: Fin sino
57: cont ← +1
58: Fin si
59: si cont = 0 entonces
60: contAux ← +1
61: Random ← nuevo Random
62: etiAux ← nueva etiqueta con con los parámetros que componen su constructor
63: listaEtiq adicionar etiAux
64: array Etiq[actual] ← etiAux
65: fin si
66: Fin si
67: Fin para
68: Fin para
```

Algoritmo 3: EstaRelacionada

Entradas: valor entero de una etiqueta(etiq), posición(pos) entera

Salida: verdadero o falso

```
1: para  $i \in Long$  hacer //Longitud de las relaciones de la lista de etiquetas en la posición de entrada
2: si etiRelacion = etiq entonces //Comparar relación en pos con etiq de entrada
3: retornar verdadero
4: Fin si
5: Fin para
6: retornar falso
```

Paso 2: Resuelve todas las equivalencias detectadas de etiquetas y selecciona una etiqueta para cada equivalencia, el segundo recorrido las resuelve y asigna la etiqueta seleccionada.

En el segundo recorrido lo que haremos será primero resolver todas las equivalencias que se han ido encontrando y entonces se volverán a etiquetar los pixeles que así lo requieran según las equivalencias. Considerando que los pixeles negros, los de objeto están a uno y los pixeles blanco de fondo están a cero.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

En el caso de **conectividad a ocho** existirá un primer pixel cuya vecindad a considerar podría ser el lateral izquierdo, el superior izquierdo, el inmediato superior o el superior derecho, entonces el pixel que se encuentra analizando podría tener uno o varios vecinos etiquetados de los anteriormente mencionados. El pixel en análisis tomará la equivalencia del primero que identifique como etiquetado obteniendo su etiqueta. De esta manera, así sucesivamente se van aplicando los mismos casos.

Luego en el paso dos se resolverán todas las equivalencias. Cada pixel se analizará conjuntamente con sus equivalencias de manera que los pixeles equivalentes tomarán la etiqueta del pixel en análisis. Y a partir de aquí ya se tiene el resultado final.

Si se cambia etiquetas por colores se podrá ver mucho mejor los resultados. Con conectividad a ocho se tienen menos regiones más grandes y con ramificaciones debidas a las conectividades de las diagonales.

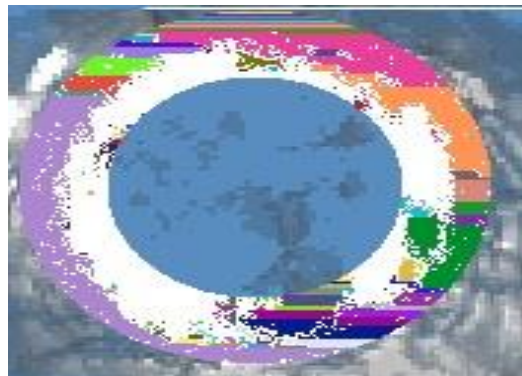


Ilustración 13: Regiones identificadas rellenas

Para la resolución el paso resolver asociaciones se codificó el siguiente algoritmo:

Algoritmo 4: ResolverAsociaciones

Entradas: no tiene entradas

Salidas: no tiene salidas, el algoritmo modifica listas y variables ya existentes declaradas de manera global

1: asociaciones \leftarrow *declaración de una lista de listas de enteros creada globalmente*

2: l \leftarrow *nueva lista de enteros declarada dentro del algoritmo*

3: **Para** $i \in etqList$ **hacer**

4: **Si** $etqList[actual] = i$ **entonces**

5: l \leftarrow *nueva lista de enteros*

6: hold $\leftarrow i$ //hold: variable que contendrá el primer valor de la lista

7: $etqList[actual] \leftarrow longAsociaciones * -1$

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

- 8: *l* **adicionar** *i*
- 9: **Para** $j \in listRelaciones$ **hacer**
- 10: **AsociarHijos2**(*l*, *listRelaciones*[*actual*])
- 11: **Fin para**
- 12: **Si** $longitudl > 0$ y $L[0] = Hold$ **entonces**
- 13: *Asociaciones* **adicionar** *l*
- 14: **Fin si**
- 15: **Fin si**
- 16: **Fin para**

Algoritmo 5: AsociarHijos2

Entradas: lista de enteros, posición(*pos*)

Salida: no tiene salidas, el algoritmo modifica listas y variables ya existentes declaradas de manera global

- 1: **Si** $etqlist[pos] = pos$ **entonces**
- 2 **Si** $longRelaciones = 0$ **entonces**
- 3: **Si** $l[0] = Hold$ **entonces**
- 4: *l* **adicionar** $etqlist[pos]$
- 5: $etqlist[pos] \leftarrow hold$
- 6: **Fin si**
- 7: **Sino** **entonces**
- 8: $posList \leftarrow buscarPos(l[0])$ // Algoritmo de búsqueda de posición de un elemento en la lista
- 9: $asociaciones[poslist]$ **adicionar** $etqlist[pos]$
- 10: $etqlist[pos] \leftarrow hold$
- 11: **Fin sino**

- 12: **Fin si**
- 13: **Sino si** $longl > 0$ **entonces**
- 14: **Si** $l[0] = hold$ **entonces**
- 15: *l* **adicionar** $etqlist[pos]$
- 16: $etqlist[pos] \leftarrow hold$
- 17: **Para** $i \in listRelaciones$ **hacer**
- 18: **asociarHijos2**(*l*, *listRelaciones*[*i*])
- 19: **Fin para**
- 20: **Fin si**
- 21: **Sino** **entonces**

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

```
22: posList ← buscarPos(l[0])
23: asociaciones[poslist] adicionar etqlist[pos]
24: etqlist[pos] ← hold
25: Para  $i \in listRelaciones$  hacer
26:   asociarHijos2( $l, listRelaciones[i]$ )
27: Fin para
28: Fin sino
29: Fin si
30: Fin si
31: Sino si  $etqlist[pos] < 1$  entonces
32:    $value \leftarrow Math.abs(etqlist[pos])$ 
33:    $hold \leftarrow asociaciones[value]$ 
34:   Si  $longl > 0$  entonces
35:     agregarAlFinal( $asociaciones[value], l, value$ )
36:   Fin si
37: Fin si
38: Sino si  $etqlist[pos] \neq hold$  entonces
39:    $value \leftarrow buscarPos(pos)$ 
40:    $hold \leftarrow asociaciones[value]$ 
41:   Si  $longl > 0$  entonces
42:     agregarAlFinal( $asociaciones[value], l, etqlist[pos]$ )
43:   Fin si
44: Fin si
```

3 – Marcar las regiones

El paso de la localización es el más simple, para cada etiqueta de la imagen se buscarán los extremos superior izquierdo e inferior derecho como localizadores de la ventana del objeto. Estos se encontrarán a partir de minimizar, maximizar los valores de las coordenadas de los pixeles dentro de la imagen.

Para cada etiqueta : ${}_k E$

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

$$\begin{aligned}
 k_1^X &= \min_{i^X} \{ (i^X, j^Y) : etiqueta(i^X, j^Y) = k^E \} \\
 k_1^Y &= \min_{i^Y} \{ (i^X, j^Y) : etiqueta(i^X, j^Y) = k^E \} \\
 k_2^X &= \max_{i^X} \{ (i^X, j^Y) : etiqueta(i^X, j^Y) = k^E \} \\
 k_2^Y &= \max_{i^Y} \{ (i^X, j^Y) : etiqueta(i^X, j^Y) = k^E \}
 \end{aligned}$$

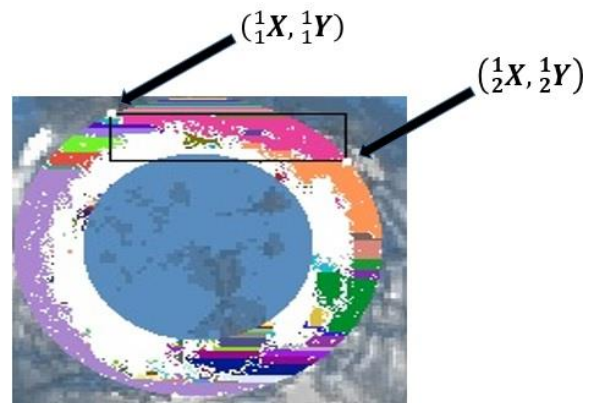


Ilustración 14: Generación de ventanas

Para la resolución el paso Marcar las regiones se diseñó el siguiente algoritmo:

Algoritmo 6: MarcarRegiones

Entradas: no tiene entradas

Salida: no tiene salidas, el algoritmo modifica listas y variables ya existentes declaradas de manera global

- 1: $g2 \leftarrow$ creador de gráficos
- 2: **Para** $i \in longAsociaciones$ **hacer**
- 3: **Si** $longAsociaciones[i] > 1$ **entonces**
- 4: $ve \leftarrow$ **BuscarVentanaXEtiqueta**(asociaciones[i])
- 5: ventanas **adicionar** ve
- 6: $alto \leftarrow$ $Math.abs(MinXPixel - MaxXPixel)$
- 7: $ancho \leftarrow$ $Math.abs(MinYPixel - MaxYPixel)$
- 8: $x \leftarrow$ $MinXPixel$
- 9: $y \leftarrow$ $MinYPixel$
- 10: $g2 \leftarrow$ Color Rojo
- 11: $g2 \text{ drawRect}(x, y, alto, ancho)$ // Pintar la ventana en la imagen analizada
- 12: **Fin si**
- 13: **Fin para**

Algoritmo 7: BuscarVentanaXEtiqueta

Entradas: etiqueta (etiq)

Salida: nueva VentanaXEtiqueta

- $menorX \leftarrow -1$
- $menorY \leftarrow -1$
- $mayorX \leftarrow -1$

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

mayorY ← -1

Para $y \in \text{longEtiqueta} - 100$ **hacer**

Para $x \in \text{longEtiqueta}[0] - 100$ **hacer**

si $\text{Etiqueta}[x][y] \neq \text{vacío}$ **entonces**

si $\text{Etiqueta}[x][y] = \text{etiq y menorX} = -1$ **entonces**

menorX ← x

menorY ← y

mayorX ← x

mayorY ← y

Fin si

Sino si $\text{Etiqueta}[x][y] = \text{etiq}$ **entonces**

Si $\text{menorX} > x$ **entonces**

menorX ← x

Fin si

Si $\text{menorY} > y$ **entonces**

menorY ← y

Fin si

Si $\text{mayorX} < x$ **entonces**

mayorX ← x

Fin si

Si $\text{mayorY} < y$ **entonces**

mayorY ← y

Fin si

Fin si

Fin si

Fin para

Fin para

Retornar *nueva VentanaXEtiqueta(menorX, mayorX, menorY, mayorY)*

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

2.2.3 Análisis de complejidad

En este epígrafe se calcula la complejidad temporal, según el enfoque teórico, de los algoritmos propuestos en la selección anterior.

Análisis del algoritmo 1: Binarización

Haciendo uso de la estructura de datos lineales (listas) y estructuras de datos homogéneos (arreglos) se facilita el trabajo con matrices. Teniendo en cuenta lo antes mencionado se muestra la complejidad de cada paso del algoritmo.

- Pasos 1 - 3 $O(1)$
- Pasos 4 – 17 $O(n * m)$ Note que en estos 13 pasos se puede ver la aparición de un ciclo anidado con complejidad $O(n)$, donde n es la cantidad de valores que puede tomar la coordenada X dentro del área a analizar, el otro ciclo tiene complejidad $O(m)$, siendo m la cantidad de valores que puede tomar la coordenada Y .
- Pasos 5 – 16 $O(m)$ Note que dentro de este ciclo solo se encuentran operaciones con complejidad $O(1)$ por lo que se mantiene en el peor de los casos $O(m)$, y trae consigo que se multiplique.
- Pasos 18 $O(1)$

Se concluye que para el algoritmo anterior la complejidad en el peor de los casos es $O(n * m)$.

Análisis del algoritmo 2 y 3: Etiquetar

- Pasos 1 y 2 $O(1)$
- Pasos 3– 68 $O(n * m * k)$ Note que en estos 65 pasos aparece un ciclo anidado con complejidad $O(n)$, donde n es la cantidad de valores que puede tomar la coordenada X dentro del área a analizar, el otro ciclo tiene complejidad $O(m)$, siendo m la cantidad de valores que puede tomar la coordenada Y . También se puede observar que existe un método dentro de los dos ciclos anidados denominado *EstaRelacionada* que tiene complejidad $O(k)$.
- Pasos 4 – 67 $O(m * k)$ Note que dentro de este ciclo con complejidad $O(m)$ se encuentra la llamada al método *EstaRelacionada* con complejidad $O(k)$. Además se encuentran operaciones con complejidad $O(1)$ lo que trae consigo que se multiplique y se mantiene en el peor de los casos $O(m * k)$.
- Pasos 5 – 66 $O(k)$ Note que dentro de esta condicional se encuentra la llamada al método *EstaRelacionada* con complejidad $O(k)$. Además se encuentran operaciones con complejidad $O(1)$ por lo que se mantiene en el peor de los casos $O(k)$.
- Pasos 7 – 19, 20 – 32, 33 – 45, 46 – 58 $O(k)$ Note que dentro de estas condicionales se encuentra la llamada al método *EstaRelacionada* con complejidad $O(k)$. Además se

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

encuentran operaciones con complejidad $O(1)$ por lo que se mantiene en el peor de los casos $O(k)$.

- Pasos 59 – 65 $O(1)$ Note que dentro de esta condicional se encuentran operaciones con complejidad $O(1)$ por lo que se mantiene en el peor de los casos $O(1)$.

Se concluye que para el algoritmo anterior la complejidad en el peor de los casos es $O(n * m * k)$.

Análisis del algoritmo 4 y 5: ResolverAsociaciones

- Pasos 1 y 2 $O(1)$
- Pasos 3 – 16 $O(n * m * k)$ Note que en estos 13 pasos se puede ver la aparición de un ciclo anidado con complejidad $O(n)$, donde n es la cantidad de etiquetas dentro de una lista, el otro ciclo tiene complejidad $O(m)$, siendo m la cantidad de relaciones que contiene cada una de las etiquetas. Además se puede observar que existe un método dentro de los dos ciclos anidados denominado *AsociarHijos2* que tiene complejidad $O(k)$.
- Pasos 4 – 15 $O(m * k)$ Note que dentro de esta condicional se encuentra un ciclo con complejidad $O(m)$ así como la llamada al método *AsociarHijos2* con complejidad $O(k)$. Conjuntamente se aprecian operaciones con complejidad $O(1)$ lo que trae consigo que se multiplique y se mantiene en el peor de los casos $O(m * k)$.
- Pasos 9 – 11 $O(k)$ Note que dentro de estas condicionales se encuentra la llamada al método *AsociarHijos2* con complejidad $O(k)$. De esta manera quedaría en el peor de los casos $O(k)$.

Análisis del algoritmo 6 y 7: MarcarRegiones

- Pasos 1 $O(1)$
- Pasos 2 – 13 $O(n * m * k)$ Note que en estos 11 pasos se puede ver la aparición de un ciclo con complejidad $O(n)$, donde n es la cantidad total de una lista de asociaciones. Además se puede observar que existe un método dentro del ciclo mencionado anteriormente denominado *BuscarVentanaXEtiqueta* que tiene complejidad $O(m * k)$.
- Pasos 4 – 15 $O(m * k)$ Note que dentro de esta condicional se encuentra la llamada al método *BuscarVentanaXEtiqueta* que tiene complejidad $O(m * k)$ ya que contiene 2 ciclos anidados, el primero con complejidad $O(m)$ donde m es la cantidad de etiquetas dentro de un arreglo y el segundo con complejidad $O(k)$ donde k es la longitud del valor que se encuentra en la primera posición del arreglo de etiquetas. También se aprecian operaciones con complejidad $O(1)$ tanto en el método *BuscarVentanaXEtiqueta* como en el más global *MarcarRegiones* lo que trae consigo que se multiplique y se mantiene en el peor de los casos $O(m * k)$.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

2.3 Identificación de la OCP

Para la identificación automática de la opacidad de la cápsula posterior se plantea la utilización de un sistema basado en casos, el cual está compuesto por tres componentes principales: una base de casos, un analizador de problemas y un recuperador de casos (Kolodner, 1992). A partir de estos componentes se enuncian un conjunto de pasos para el desarrollo del SBC con el objetivo de la identificación automática de opacidad en pacientes operados de cataratas.

Secuencia de pasos para el desarrollo del SBC (Alvarez Cancio, et al., 2014):

1. Definir los rasgos predictores y los rasgos objetivos.
2. Determinar el dominio de definición de cada rasgo.
3. Determinar el peso informacional de cada rasgo.
4. Definir las funciones de comparación de rasgos.
5. Definir las funciones de comparación de casos.
6. Representar los casos.
7. Obtener el nuevo caso a resolver.
8. Comparar el nuevo caso con la Base de Casos.
9. Obtener los 'k' casos más semejantes.
10. Seleccionar el rasgo objetivo, siendo este el paso fundamental para nuestra solución.

2.3.1 Definición de los rasgos predictores y los rasgos objetivos

Para definir los rasgos predictores, nos basamos en los conocimientos brindados por los especialistas en cuanto a su criterio de evaluación de la opacidad.

Se comienza por expresar la importancia de la región a evaluar dentro de la estructura ocular. Obteniéndose el rasgo: **Región**, la misma estará enfocada entre los 3 y 5 mm en correspondencia con el centro de la estructura ocular. La imagen se dividirá en 32 regiones equivalentes, lo que permitirá comparar la región que se encuentre en análisis y sus dos regiones vecinas con el objetivo de encontrar las semejanzas entre casos.

Además, la identificación y selección de la **Ventanas o regiones conexas** a analizar, dichas áreas serán tomadas siempre de izquierda a derecha obteniendo como punto de partida el pixel superior izquierdo con opacidad, luego el superior derecho seguido del inferior izquierdo y finalmente el inferior derecho tomando como resultado una Ventana con largo y ancho como características propias. También la cercanía del objeto al centro de la estructura ocular. Obteniéndose el rasgo: **Cercanía al centro**. El valor de este rasgo está dado por el resultado del cálculo de la distancia euclidiana entre el punto medio de la Ventana y el centro de la estructura ocular a analizar.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

Cada región o ventana será tratada como un objeto o caso. Siendo importante la composición de colores Rojo, Verde y Azul de cada objeto, dado que es otro dato comparable con significatividad para el problema en cuestión. Surgiendo el rasgo: **Intensidad**. Este resultado está dado por el promedio de la intensidad de los colores primarios que componen la región analizada.

Cantidad de Píxeles, rasgo que proporciona el dato de cuántos píxeles se encuentran incluidos dentro de la ventana que se encuentra en cuestión.

Cantidad de Asociaciones, este rasgo permitirá obtener información de cuantas subregiones estaban interconectadas dentro de la ventana.

Después de definidos los rasgos predictores se define el rasgo objetivo: **Opacidad**. Siendo este el resultado del análisis final de si el objeto (pixel) representa opacidad dentro de la estructura ocular o no.

2.3.2 Dominio de definición de cada rasgo

En este paso se procede a determinar el dominio de definición de cada rasgo, expresados en la siguiente tabla:

Rasgo	Tipo de valor	Dominio
Región	Entero	[1,.....,32]
Ventana	Vectorial	(L,A) donde L[1,.....,n], A[1,.....,n]
Cantidad de Asociaciones	Entero	[50,.....,100]
Intensidad	Entero	[100,.....,230]
Cantidad de Píxeles	Entero	[1,.....,n]
Cantidad de Cambios	Entero	[0,.....,n]
Opacidad	Binario	[0,1]

Tabla 7: Dominio de definición de cada rasgo

2.3.3 Peso informacional de cada rasgo

Después de tener los rasgos y su dominio de definición se hace importante definir el peso informacional que posee cada uno de estos rasgos para nuestro problema, siendo algunos más significativos que otros. Quedado cada uno confeccionado según el estudio de comportamiento y el criterio de los especialistas en oftalmología de la siguiente forma:

Rasgo	Peso Informacional{1,.....,10}
-------	--------------------------------

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

1. Región	8
2. Ventana	6
3. Cercanía al centro.	6
4. Intensidad	5
5. Cantidad de píxeles	3
6. Cantidad de Asociaciones	2

Tabla 8: Peso informacional de cada rasgo

2.3.4 Selección de las funciones de comparación de rasgos

A continuación se muestran las funciones para evaluar la semejanza de los rasgos que han sido extraídos del nuevo caso con los casos existentes en la BC, las cuales fueron tomadas del libro “Enfoque Lógico Combinatorio al Reconocimiento de Patrones” (Shulcloper, 1999).

Siendo la función: $C_s(X_s(O_i), X_s(O_j)) = \begin{cases} 1 & \text{Si } X_s(O_i) = X_s(O_j) \\ 0 & \text{en otro caso} \end{cases}$ un concepto de igualdad, donde (O_i) es el valor del rasgo X_s en el objeto O_i . Aplicable a los rasgos: *Región* y *Cantidad de Asociaciones*

La función: $C_s(X_s(O_i), X_s(O_j)) = \begin{cases} 1 & \text{Si } X_s(O_i), X_s(O_j) \in [A_p, A_k] \\ 0 & \text{en otro caso} \end{cases}$ es un concepto de intervalo,

Fórmula 2

Donde $X_s(O_i)$ es el valor del rasgo X_s en el objeto O_i y A_p, A_k son los valores iniciales y finales del intervalo. Siendo esta aplicable al rasgo: *Intensidad* y *Cantidad de píxeles*.

Y la función: $C_s(X_s(O_i), X_s(O_j)) = \begin{cases} 1 & \text{Si } (X_s(O_i) - X_s(O_j)) < \varepsilon_s \\ 0 & \text{en otro caso} \end{cases}$ es un concepto de semejanza

Fórmula 3

Asociado a un umbral, donde $X_s(O_i)$ es el valor del rasgo X_s en el objeto O_i y ε_s es el umbral asociado al rasgo X_s . Es aplicable a los rasgos: *Cercanía al centro*.

2.3.5 Selección de la función de comparación de casos

Después de definidas las funciones de comparación de rasgos se procede a comparar el objeto (caso) nuevo con los existentes en la BC. La función de semejanza que se utiliza para la

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

comparación de dichos casos, tomada del libro “Enfoque Lógico Combinatorio al Reconocimiento de Patrones” (Shulcloper, 1999) es:

$$\beta(\Omega I(O_i), \Omega I(O_j)) = \left(\frac{\sum_{X_i \in S} P(X_i)}{\sum_{X_i \in \Omega} P(X_i)} \right)$$

Fórmula 4

Donde $P(X_i)$ es una magnitud asociada a cada rasgo X_i que refleja su relevancia (Peso Informacional), S es el conjunto de rasgos coincidentes entre los objetos (O_i, O_j), Ω es el conjunto total de rasgos que se consideran y $\Omega I(O_i)$ es el conjunto de rasgos del objeto O_i .

2.3.6 Representación de los casos

Esta etapa de almacenamiento consiste en registrar en la base de conocimiento la información derivada del nuevo caso.

El almacenamiento de los casos se realiza en una base de casos, la cual se nutre de los casos nuevos proporcionados por el sistema y los casos previamente almacenados en ella. Coexistiendo entonces los conocimientos, las experiencias brindadas por los especialistas y las recreadas por el sistema. En esta base de conocimiento se almacena el conocimiento necesario para resolver los problemas del dominio de aplicación.

El conocimiento se representa a través de una tabla en la cual las columnas son etiquetadas por variables que representan los rasgos predictores y el rasgo objetivo (decisión) y las filas representan los casos. Para la representación de los casos, además de las columnas descritas existirá una columna con una variable (Valor Global) que mediante un valor numérico hace énfasis en la descripción general de cada uno de los rasgos predictores del caso en cuestión. Esta variable nos beneficia en la organización de la base de casos en función de su valor.

2.3.7 Obtener el nuevo caso a resolver

La obtención del nuevo caso a resolver se desencadena cuando el usuario del sistema carga la imagen médica resultante de un tomograma SCHEIMPFLUG del PENTACAM. Los casos están vinculados a cada pixel cuya variación de colores sea distinta a la del fondo y se encuentre entre los 3 y 6 mm de radio (área de interés donde se encuentra la opacidad) con respecto al centro de la estructura ocular, siendo cada uno de estos un nuevo caso a dar solución.

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

2.3.8 Comparación del nuevo caso con la Base de Casos

En la base de casos todos los datos estarán organizados de menor a mayor por el valor de la variable "Valor Global", de forma que, para comparar el nuevo caso con los existentes en la base de casos, se seleccionan de la base de conocimiento aquellos valores cuya descripción se ajusta más a la información presentada en el nuevo caso. A estos casos más cercanos al valor del nuevo se le aplican las funciones de comparación de rasgos para comparar cada uno de los rasgos de los mismos, y después se utiliza la función de comparación de casos para comparar finalmente los casos, definiendo posteriormente el valor del rasgo objetivo de este nuevo caso.

2.3.9 Obtención de los 'k' casos más semejantes

Después de la selección de los casos más semejantes de la base de casos y la comparación cada uno de estos con el nuevo caso, se procede a escoger los "k" casos más semejantes, para lo que se propone usar (k=5) siendo la comparación en un espectro de casos más amplio demasiado costosa computacionalmente. (Peña, 2015)

2.3.10 Selección del rasgo objetivo

Para la selección del resultado del rasgo objetivo, revisaremos los k más semejantes ordenándolos de menor a mayor según su valor de semejanza. Después aplicamos la siguiente fórmula para obtener el resultado:

$$\theta(Ck) = \begin{cases} 1 & \text{Si } \left(\sum_{x=1}^k (Vn(Cx) * x) * V \right) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Fórmula 5

Donde $Vn(Cx)$ es una magnitud asociada a cada caso C_x derivada por el resultado de su función de comparación $\beta(\Omega I(O_i), \Omega I(O_j))$ con el nuevo caso, K es el conjunto de los casos con mayor valor de coincidencia, V es un valor que denota la presencia de opacidad en el caso dado por el rasgo objetivo Opacidad del caso almacenado, el cual toma valor 1 si es opacidad y -1 en caso de ausencia de la misma, y el valor de $\theta(Ck)$ nos representa la presencia o la ausencia de opacidad en el caso en cuestión, representando este el resultado del rasgo objetivo.

Conclusiones parciales del capítulo

En este capítulo se ha presentado el ciclo de vida de la solución propuesta según la metodología XP, generando las HU que caracterizan al sistema, así como la estimación del esfuerzo necesario

Capítulo 2. Sistema de cuantificación de opacidad capsular mediante Componentes Conexas

para la implementación de las mismas. Además, se construyó el plan de iteraciones dando lugar a la conformación de la estimación del tiempo que demoraría cada una. En adición se llevó a cabo el plan de entregas en el que se define qué historias de usuario formarían parte de cada entrega. También fueron confeccionadas las tarjetas CRC correspondientes a cada clase. Siendo parte del concepto de componentes conexas fueron descritos los pasos lógicos pertenecientes al algoritmo de detección de regiones conexas implementado en la solución informática. Finalmente quedan descritos cada uno de los pasos realizados durante el desarrollo del SBC teniendo en cuenta las características propias que exige el sistema.

Capítulo 3. Resultados y validación del sistema

Capítulo 3. Resultados y validación del sistema

Las pruebas de software constituyen un instrumento para determinar el nivel de calidad de un producto. Dentro de las fases propuestas por la metodología XP se lleva a cabo la implementación y las pruebas del sistema. Dicha metodología divide las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código, la cual es diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al terminar una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (J .J . Gutiérrez, 2010). Durante las iteraciones las historias de usuarios seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Las pruebas unitarias se realizan con el objetivo de verificar el correcto funcionamiento de los procedimientos, agrupados en los distintos componentes.

3.1 Fase de implementación

Se especifica en esta fase la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Lo primero es hacer un chequeo de cada HU, en conjunto con el plan de iteraciones y se modifica en caso de ser necesario, para esto se crean tareas de desarrollo, para de esta forma poder organizar la implementación. Estas tareas, al contrario de las HU escritas en el lenguaje del cliente, son escritas en un lenguaje técnico. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones de desarrollo sobre el sistema.

Iteración 1

En esta iteración se implementaron las funcionalidades de mayor prioridad para el sistema:

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Construir la BC	2	2
Conectar la BC con la aplicación	1	0.5
Reconocer imagen de tomograma	2	1.5
Reconocer el centro de la estructura ocular	2	2

Tabla 9: Tiempo de implementación de las HU de la iteración #1

A continuación, se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración:

Capítulo 3. Resultados y validación del sistema

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre: Crear la BC	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 2
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se declaran todos los elementos necesarios para la creación de la base de conocimientos.	

Tabla 10: Tarea de ingeniería de la funcionalidad Crear BC

Tarea	
Número de tarea: 2	Número de HU: 2
Nombre: Conectar la BC con la aplicación	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se conecta la BC con el software PANDOC.	

Tabla 11: Tarea de ingeniería de la funcionalidad Conexión

Tarea	
Número de tarea: 3	Número de HU: 3
Nombre: Reconocer imagen de tomograma	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se declaran los elementos necesarios para reconocer un tomograma.	

Tabla 12: Tarea de ingeniería de la funcionalidad Reconocer imagen de tomograma

Tarea

Capítulo 3. Resultados y validación del sistema

Número de tarea: 4	Número de HU: 4
Nombre: Reconocer el centro de la estructura ocular	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se declaran los elementos necesarios para reconocer el centro de la estructura ocular con la utilización de trilateración celular.	

Tabla 13: Tarea de ingeniería de la funcionalidad Reconocer centro de la estructura ocular

Iteración 2

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Llenar la BC	3	3
Obtener el nuevo caso	2	2
Definir región del caso	2	1.5

Tabla 14: Tiempo de implementación de las HU de la iteración # 2

Tarea	
Número de tarea: 5	Número de HU: 5
Nombre: Llenar la BC	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 2.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se llena la BC con la mayor cantidad de casos de imágenes existentes que presentan opacidad.	

Tabla 15: Tarea de ingeniería de la funcionalidad Llenar BC

Tarea	
Número de tarea: 6	Número de HU: 6

Capítulo 3. Resultados y validación del sistema

Nombre: Obtener el nuevo caso	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Una vez que se carga en el sistema un tomograma cada pixel que se encuentra entre los 3 y los 5 mm del centro de la estructura ocular va a significar un nuevo caso.	

Tabla 16: Tarea de ingeniería de la funcionalidad Obtener nuevo caso

Tarea	
Número de tarea: 7	Número de HU: 7
Nombre: Buscar región.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se implementa el método Buscar_Region, que divide la imagen en 32 partes equivalentes lo que permite identificar la región a la que pertenece el nuevo caso a resolver y de esta forma minimizar el rango de búsqueda.	

Tabla 17: Tarea de ingeniería de la funcionalidad Buscar región

Iteración 3

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Cargar Casos similares de la BC	2	1.5
Calcular similitud de rasgos	2	2
Calcular similitud de casos	2	2

Tabla 18: Tiempo de implementación de las HU de la iteración # 3

Las tareas de ingeniería generadas por la iteración 3 se encuentran en el apartado "Anexo # 3".

Iteración 4

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Calcular certeza de solución	2	2

Capítulo 3. Resultados y validación del sistema

Brindar datos al usuario	2	1.5
Recuperar el caso a la BC	2	2

Tabla 19: Tiempo de implementación de las HU de la iteración # 4

Las tareas de ingeniería generadas por la iteración 4 se encuentran en el apartado “Anexo # 4”.

3.2 Pruebas de aceptación

Cada historia de usuario está asociada a una prueba de aceptación, conocidas también como pruebas funcionales, las mismas se realizan en esta etapa del proyecto y en ellas se describen las posibles formas de utilización del software. Las pruebas funcionales no solo validan la transformación de una entrada en una salida, sino que validan una característica completa.

En estos documentos de prueba se indican las posibles respuestas que tiene el software en la utilización de cada funcionalidad, así como los posibles mensajes de error, información o de aceptación que emite el software cuando se utiliza dicha funcionalidad (Canós José H, 2003).

A continuación, se ejemplifican algunas de las pruebas de aceptación:

Caso de prueba de aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Definir región del caso	
Descripción: Prueba para la funcionalidad definir región del caso.	
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de un tomograma Scheimpflug del PENTACAM. Se debe haber reconocido automáticamente el centro de la estructura ocular.	
Resultado Esperado: El sistema calcula y devuelve la región donde se localiza el caso que se analiza.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 20: Caso de prueba de aceptación para la HU Definir región del caso

Caso de prueba de aceptación

Capítulo 3. Resultados y validación del sistema

Código: HU11_P2	Historia de Usuario: 11
Nombre: Calcular OCP en áreas definidas	
Descripción: Prueba para la funcionalidad que calcula la opacidad en el área seleccionada por el sistema.	
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de un tomograma Scheimpflug del PENTACAM. Se debe haber reconocido automáticamente el centro de la estructura ocular. Se debe haber seleccionado automáticamente el área que presenta opacidad.	
Resultado Esperado: El sistema guarda la configuración seleccionada y muestra un mensaje informando los resultados del cálculo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 21: Caso de prueba de aceptación para la HU Calcular certeza de solución

Como parte de estas pruebas se procedió a la creación de 8 casos de estudio para verificar los resultados del software, en los mismos se tomaron imágenes que corresponden a diferentes pacientes, a los que llamaremos (Paciente 1, Paciente 2, Paciente 3, ..., Paciente 8). A continuación, se presenta una muestra de 4 casos de estudio de los 8 realizados. Las pruebas arrojaron los siguientes resultados:

1. Caso de estudio correspondiente a la imagen resultante del tomograma Scheimpflug del ojo derecho del paciente "Paciente 1" tomado el 28/09/2010.

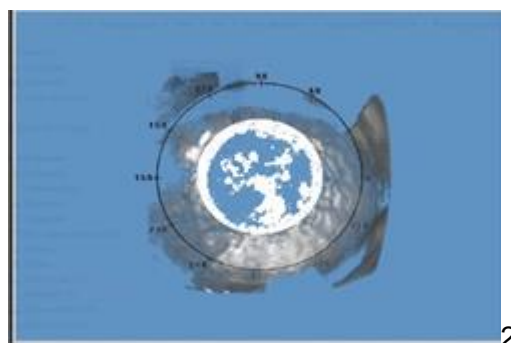


Ilustración 15: Análisis realizado por PANDOC al tomograma del Paciente 1

Capítulo 3. Resultados y validación del sistema

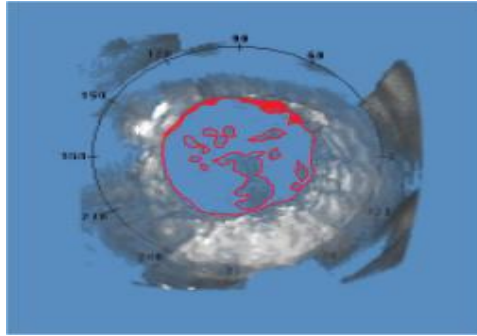


Ilustración 16: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 1

2. Caso de estudio correspondiente a la imagen resultante del tomograma Scheimpflug del ojo derecho del paciente "Paciente 2" tomado el 21/09/2010

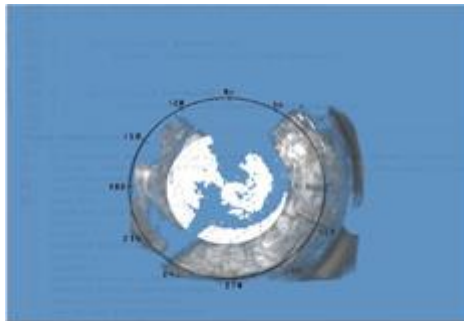


Ilustración 17: Análisis realizado por PANDOC al tomograma del Paciente 2

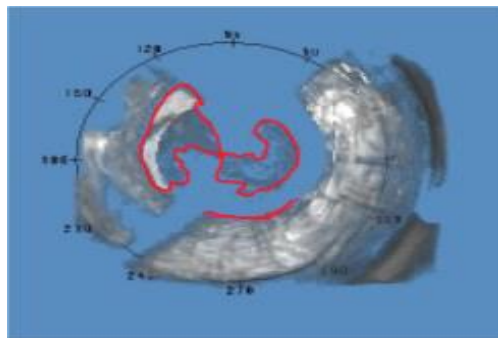


Ilustración 18: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 2

3. Caso de estudio correspondiente a la imagen resultante del tomograma Scheimpflug del ojo izquierdo del paciente "Paciente 3" tomado el 18/05/2010

Capítulo 3. Resultados y validación del sistema

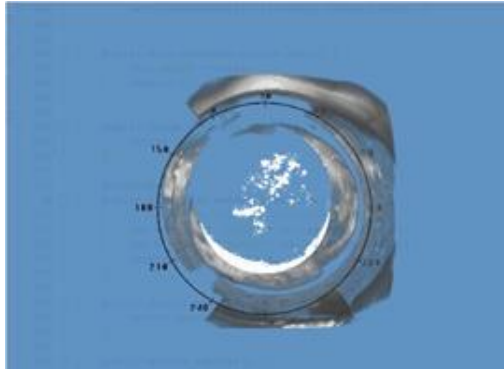


Ilustración 19: Análisis realizado por PANDOC al tomograma del Paciente 3

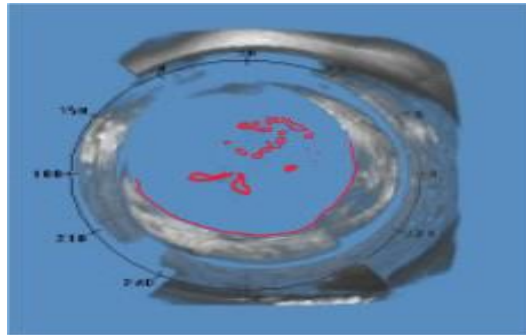


Ilustración 20: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 3

4. Caso de estudio correspondiente a la imagen resultante del tomograma Scheimpflug del ojo izquierdo del paciente "Paciente 4" tomado el 29/06/2010

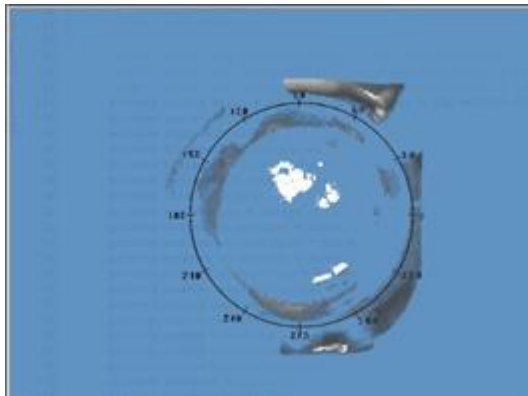


Ilustración 21: Análisis realizado por PANDOC al tomograma del Paciente 4

Capítulo 3. Resultados y validación del sistema

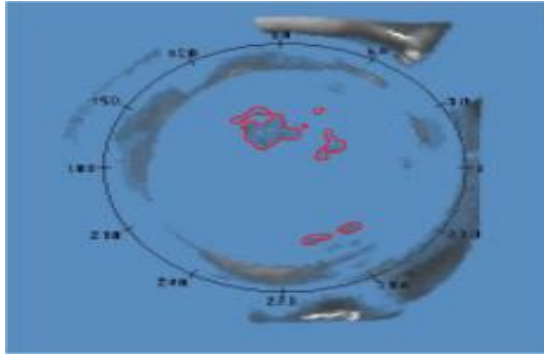


Ilustración 22: Región considerada por el especialista como opacidad según su percepción al tomograma del Paciente 4

3.2.1 Resultado de las pruebas de aceptación

Número de iteración	No conformidades		Recomendaciones	No conformidades resueltas
	Significativas	No significativas		
1	1	1	0	2
2	0	2	2	4
3	1	0	1	2
4	0	1	1	2

Tabla 22: Resultado de las pruebas de aceptación

Las pruebas se realizaron de forma iterativa e incremental, y se comprobó en cada iteración la corrección de los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del sistema, por lo que el sistema quedó aprobado por parte del cliente, quien evaluó el 100 % de los casos de prueba de forma satisfactoria.

3.3 Pruebas Unitarias

Para la aplicación de las pruebas unitarias se hizo uso de la técnica camino básico. El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Para la realización de técnica se toma como ejemplo el método Binarización, el cual consiste en binarizar la imagen que se está analizando. A continuación, se muestra el código fuente.

```
public BufferedImage Binarizacion(Ellipse2D elipseMenor, Ellipse2D elipseMayor)
{
    int mediaPixel;//1
```


Capítulo 3. Resultados y validación del sistema

```
Color colorAux;//1
BufferedImage copialmg = new BufferedImage(imageActual.getWidth(), imageActual.getHeight(),
imageActual.getType());//1
copialmg.setData(imageActual.getData());//1
//Recorremos la imagen píxel a píxel
for (int i = (int) pos[1] - 200; i < (int) pos[1] + 210; i++)
{//2
  for (int j = (int) pos[0] - 200; j < (int) pos[0] + 200; j++)
  {//3
    if (!elipseMenor.contains(i, j) && elipseMayor.contains(i, j))
    {//4
      colorAux = new Color(imageActual.getRGB(i, j));//5
      mediaPixel = (int) ((colorAux.getRed() + colorAux.getGreen() + colorAux.getBlue()) / 3);//5
      if (mediaPixel <= 141 && mediaPixel >= 139) {//6
        copialmg.setRGB(i, j, (255 << 16) | (255 << 8) | 255);//7
      } else {//
        copialmg.setRGB(i, j, (0 << 16) | (0 << 8) | 0);//8
      }
    }
  }
} //9
} //10
//Retornamos la imagen
return copialmg;//11
}
```

Partiendo del fragmento de código tomado se obtiene el siguiente grafo de flujo:

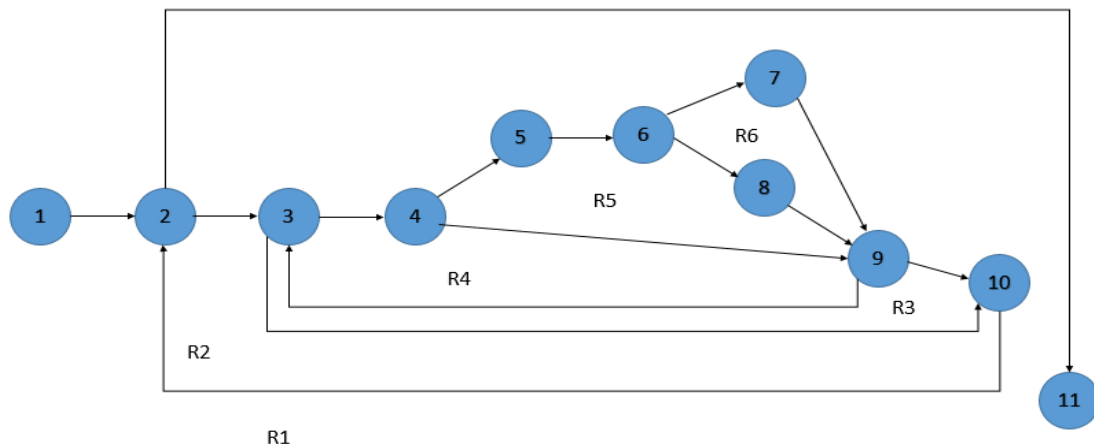


Ilustración 23: Grafo de flujo del método Binarización

Capítulo 3. Resultados y validación del sistema

Una vez definido el grafo de flujo se procede al cálculo de la complejidad ciclomática siendo esta una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa (Pressman, 2005). Para esta operación existen tres vías de solución, las cuales se enuncian a continuación:

- ✓ $V(G)=(A-N) +2$
- ✓ $V(G)=P +1$
- ✓ $V(G)=R$

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , se define como:

$V(G)= A-N+2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , también se define como:

$V(G)= P + 1$, donde P es el número de nodos predicados contenidos en el grafo de flujo G .

$V(G)= R$, donde R es la cantidad total de regiones.

Del grafo de flujo G se obtiene que:

$$A = 15 \quad V(G) = A-N+2 \quad P= 5 \quad R=6$$

$$N = 11 \quad V(G) = 15-11+2 \quad V(G) = P+1 \quad V(G) = R$$

$$V(G) =6 \quad V(G) = 5+1 \quad V(G) = 6$$

$$V(G) =6$$

Se aplican las tres formas para afirmar un resultado seguro y confiable. Luego se obtiene una complejidad ciclomática $V(G)=6$ esta cifra representa la cantidad de caminos independientes para el grafo de flujo construido para el método. Luego de tener elaborado el grafo de flujo e identificados los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos (ver Tabla 23). Se escogen los datos de manera que las condiciones de los nodos predicados estén adecuadamente establecidas, con el fin de comprobar cada camino. A continuación, se especifican estos casos de prueba.

Números	Caminos básicos
1	1-2-11
2	1-2-3-10-2-11
3	1-2-3-4-9-3-10-2-11
4	1-2-3-4-5-6-7-9-3-10-2-11
5	1-2-3-4-5-6-8-9-3-10-2-11
Fq 6SG	1-2-3-4-9-3-10-2-11

Capítulo 3. Resultados y validación del sistema

Tabla 23: Camino Básico del flujo

Caso de Prueba Camino Básico # 1
Descripción: Este método dada una imagen, devuelve la imagen binarizada.
Condiciones de ejecución: Se debe haber cargado la imagen antes.
Entradas\Pasos de ejecución: Ellipse2D elipseMenor, Ellipse2D elipseMayor F75Red 72AW
Resultado esperado: La imagen binarizada.
Evaluación de la prueba: Satisfactoria.

Tabla 24: Caso de Prueba Camino Básico # 1

En la siguiente gráfica se muestran los resultados de las pruebas unitarias, a través del método de caja blanca aplicando la técnica de camino básico al método Binarizacion.

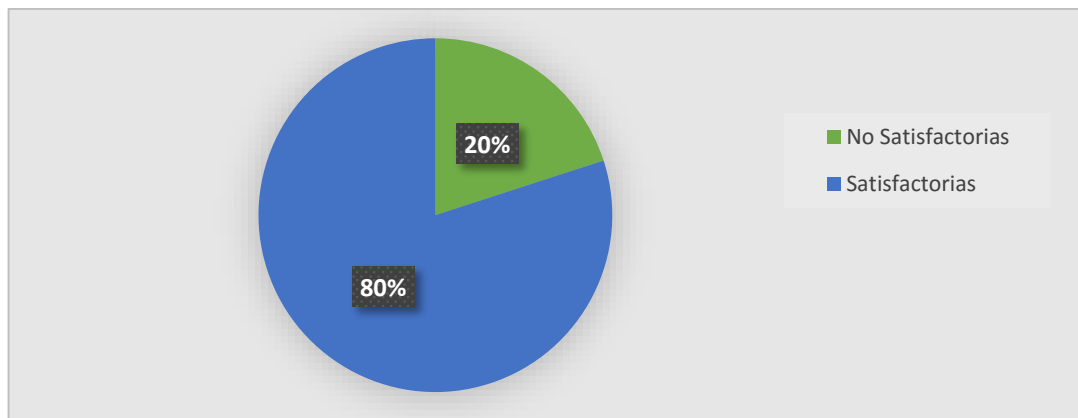


Ilustración 24: Resultados de las pruebas unitarias.

Se realizaron 35 casos de prueba de caja blanca (pruebas unitarias), de los cuales 30 resultaron satisfactorios representando el 80% del total de los casos de prueba. Se detectaron un total de cuatro errores, ligados a funcionalidades del sistema, a los que se les dio el tratamiento requerido para el logro de un 100% de pruebas satisfactorias.

Capítulo 3. Resultados y validación del sistema

3.4 Métrica de evaluación de la clasificación por regiones

Como parte de la fase de pruebas se aplicó también la métrica Evaluación de la Clasificación por Regiones (Vanrell, M. 2016), para ello se define la matriz de confusión y sobre esta se obtienen cuatro medidas la exactitud, precisión, sensibilidad y especificidad.

La matriz de confusión es definida como la herramienta básica que permite visualizar el nivel de error de un clasificador. Las filas de la matriz representan las instancias reales (regiones que marca el especialista) y las columnas representan las regiones con opacidad (marcadas por el sistema).

La matriz confusión contendrá los siguientes valores:

- Si la región analizada ha sido correctamente identificada por el sistema como opacidad se denotan como reales positivos.
- Si las regiones que son opacidad y fueron identificados incorrectamente por el sistema se denotan como falsos positivos.
- Si las regiones que son opacidad y el sistema no los detecta serán denotados como falsos negativos.
- Si las regiones que no son opacidad y el sistema los identifica correctamente se denotaran reales negativos quedando la matriz confusión estructurada de la siguiente manera.

Instancias Reales/Resultados de Calcificación	Opacidad	No Opacidad
Opacidad	Reales positivos	Falsos negativos
No Opacidad	Falsos positivos	Reales negativos

Tabla 25: Matriz de confusión

Las medidas usadas se calculan a partir de la matriz de confusión de la siguiente forma:

- La exactitud describe la proximidad entre el resultado global del pacificador y la clasificación exacta y se calcula cuantitativamente de la siguiente forma: $Exactitud = \frac{Reales\ positivos + Reales\ Negativos}{Predicciones\ Totales}$, siendo las Predicciones Totales la cantidad de regiones analizada.
- La precisión mide la calidad de las respuestas positivas del clasificador y se calcula cuantitativamente de la siguiente forma: $Presicion = \frac{Reales\ positivos}{Reales\ positivos + Falsos\ positivos}$.
- La sensibilidad mide la eficiencia en la clasificación de todos los elementos que son de la clase y se calcula cuantitativamente de la siguiente forma: $Sensibilidad = \frac{Reales\ positivos}{Reales\ positivos + Falsos\ negativos}$.

Capítulo 3. Resultados y validación del sistema

- La especificidad mide la eficiencia en la clasificación de todos los elementos que no son de la clase y se calcula cuantitativamente de la siguiente forma: $Especificidad = \frac{Reales\ negativos}{Reales\ negativos + Falsos\ positivos}$.

Las medidas toman valores desde 0 hasta 1, para el valor 1 se obtiene la coincidencia perfecta entre las regiones analizadas.

Se analiza la clasificación en 15 imágenes resultantes del PENTACAM marcadas por el especialista y luego analizadas por el sistema, de las que se derivaron 80 regiones quedando la matriz de confusión de la siguiente manera.

Instancias Reales/Resultados de Calcificación	Opacidad	No Opacidad
Opacidad	63	13
No Opacidad	0	4

Tabla 26: Matriz de confusión para las 80 regiones

$$Exactitud = \frac{63 + 4}{80} = 0.84$$

$$Presicion = \frac{63}{63 + 0} = 1$$

$$Sensibilidad = \frac{63}{63 + 4} = 0.94$$

$$Especificidad = \frac{4}{4 + 0} = 1$$

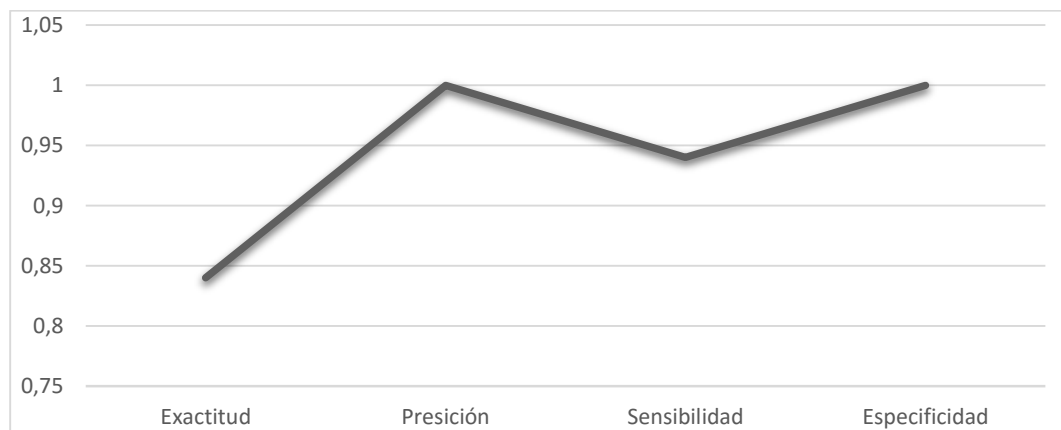


Ilustración 25: Resultados obtenidos al aplicar la métrica

Capítulo 3. Resultados y validación del sistema

Dados los resultados anteriores se puede concluir que la proximidad entre el resultado y la imagen marcada por el especialista es corta ya que presenta un índice de exactitud de 84%, la calidad de la respuesta es elevada pues se obtuvo el máximo de precisión, la eficiencia en la clasificación de cada región es buena ya que presenta alto nivel de sensibilidad superior al 90% y el máximo de especificidad.

Conclusiones parciales del capítulo

Una vez terminado el sistema fueron realizadas las pruebas, donde se demostró que una historia de usuario es un resumen de una funcionalidad de un software, mientras que los desarrolladores van ampliando sus conocimientos a medida que se va implementando. Las mismas son necesarias a la hora de corregir errores no deseados; donde el cliente comprueba el cumplimiento y la satisfacción de lo que esperaba con el mismo, por lo que se realizaron las pruebas de aceptación y las pruebas unitarias con el empleo de los métodos de caja blanca y caja negra. Se pudo constatar después de 4 iteraciones, que el sistema quedó libre de no conformidades, por lo que el cliente puede utilizarlo en un entorno real. Luego se aplicó la métrica Evaluación de la Clasificación por Regiones para validar el SBC, demostrando que las imágenes resultantes del sistema se ajustan a lo especificado por el cliente.

Conclusiones Generales

Conclusiones Generales

Sobre la base del análisis, interpretación y sistematización de las investigaciones teóricas y empíricas, a continuación, se presentan las siguientes conclusiones de la investigación:

- Se realizó el marco teórico de la investigación donde se establecieron las tendencias marcadas en cuanto al desarrollo de software, relacionados con la OCP como complicación postoperatoria más frecuente y la identificación de regiones en imágenes médicas lo que propició la necesidad de desarrollar una herramienta informática para apoyar el criterio de los especialistas.
- Como resultado del diseño fue implementado un algoritmo de detección de regiones conexas para la identificación de las zonas de opacidad en imágenes provenientes del PENTACAM.
- Para clasificar las regiones q son opacidad y dado que el conocimiento se puede expresar como casos o vivencias se diseñó un sistema basado en casos.
- Para validar la herramienta se realizaron pruebas unitarias y de aceptación, comprobándose el correcto funcionamiento de la misma, evidenciándose mediante un acta de liberación redactada por el especialista Iván Hernández López.
- El clasificador se validó apoyado en la métrica Evaluación de la Clasificación por Regiones lo que proporcionó resultados satisfactorios atendiendo a los indicadores de la exactitud, precisión, sensibilidad y especificidad.
- Este sistema podrá ser utilizado además como software de apoyo en investigaciones médicas del campo de la OCP ya q muestran las regiones afectadas por esta complicación postoperatoria.

Recomendaciones

Recomendaciones

Las recomendaciones de la presente investigación están encaminadas a incrementar las funcionalidades que ya presenta la misma, por lo que se recomienda:

- Diseñar una base de datos para almacenar los resultados de los cálculos realizados por los especialistas, de manera que le puedan servir de ayuda para estudios y diagnósticos por parte de aquellos menos experimentados.
- Realizar un estudio de autómatas celulares para el etiquetado de regiones, lo que permitirá detectar mejor las regiones a analizar.

Bibliografía

Bibliografía

- Acosta R, Hoffmeister L, Roman R, Comas M, Castilla M, Castells X. (2006). *Revisión sistemática de estudios poblacionales de prevalencia de cataratas*. Barcelona, España.
- Iliana Gutiérrez Martínez, Rafael E. Bello Pérez y Andrés Tellería Rodríguez. (2002). UN SISTEMA BASADO EN CASOS PARA LA TOMA DE DECISIONES EN CONDICIONES DE INCERTIDUMBRE. 23, 2.
- Iván Hernández López, Juan Raúl Hernández Silva, Yadira Castro González, Ailén Garcés Fernández, Zucell Veitía Roviroso, Eneida Pérez Candelaria. (2010). Estrategias de prevención de la opacidad de la cápsula posterior. *Revista Cubana de Oftalmología*, 23.
- Laura Lozano, Javier Fernández. (2008). *Razonamiento Basado en Casos: Una Visión General*. España .
- Natalia Martínez Sánchez, María M García Lorenzo, Zoila Zenaida García Valdivia, Gheisa Ferreira Lorenzo. (2009). *EL PARADIGMA DEL RAZONAMIENTO BASADO EN CASOS EN EL ÁMBITO DE LOS SISTEMAS DE ENSEÑANZA/APRENDIZAJE INTELIGENTES*. Santa Clara.
- Abrego, E. (26 de Octubre de 2014). *Aportes de los celulares en las diferentes especialidades*. Obtenido de <http://es.slideshare.net/elsaivoneabrego/aportes-de-los-celulares-en-las-diferentes-especialidades-40727633>
- Alvarez Cancio, Michel, Rodriguez Puentes, Rafael y Hernandez Lopez, Ivan. (06 de 03 de 2015). *DISEÑO DE UN SISTEMA BASADO EN CASOS PARA LA IDENTIFICACIÓN DE OPACIDAD MEDIANTE EL PENTACAM*. Obtenido de Uciencia: uciencia.uci.cu/?q=tax
- Anibal Bregón, Arancha Simón, Carlos Alonso, Belarmino Pulido, I-saac Moro ,Juan José Rodríguez. (2005). *Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos*.
- Aslam TM, P. N. (2005). A freely accessible, evidence based, objective system of analysis of posterior capsular opacification; evidence for its validity and reliability. 5(9).
- Augusto Cortez Vásquez, Carlos Navarro Depaz, Jaime Pariona Quispe. (2010). Sistemas de razonamiento basado en casos aplicado a sistemas de líneas de productos de software.. 2010. 2., Vol. 7. 7, 2.
- Beck, K. (2002). Una explicación de la programación extrema: aceptar el cambio .
- Blanco, Paco. (2009). *Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone*. Madrid: Universidad Politécnica de Madrid.

Bibliografía

- BOOCH, Grady. (1999). *El lenguaje unificado de modelado*.
- Cancio, M. A. (2014). PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata.
- Canós José H, L. P. (2003). Metodologías Ágiles en el Desarrollo de Software.
- D Grewall, R. J. (2008). Pentacam tomograms: A Novel Method for Quantification of Poste-. 49(5).
- D. S Friedman, D D Duncan. (1999). Digital image capture and automated analysis of posterior capsular opacification. *Investigative Ophthalmology & Visual Science*, 40, 8.
- Echarte, P. (30 de Noviembre de 2010). *eslomas.com*. Obtenido de <http://www.eslomas.com/2010/11/como-usar-una-base-de-datos-sqlite-en-android/>
- Elizondo, M. C. José Jaime Esqueda. (2002). *Fundamentos de Procesamiento de Imágenes*. California: CONATEC.
- J .J . Gutiérrez, M. J. (2010). PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA.
- Kolodner., Janet L. (09 de 03 de 2015). *Artificial Intelligence Review*.
- Lara, A. L. (4 de Agosto de 2013). *SaduceBlog* . Obtenido de <http://blog.saducelabs.com/android/android-y-sqlite/>
- López, I. H. (2011). Cuantificación objetiva de la opacidad de la cápsula posteriormediante tomogramas Scheimpflug del Pentacam. *Revista Cubana de Oftalmología*, 24(2).
- Manfred R Tetz, G. U. (1997). Photographic image analysis system of posterior capsule opacification. *Journal of Cataract & Refractive Surgery*, 23(10).
- Marisa R. De Giusti, Maria Marta Vila, Gonzalo Lujan Villarreal. (2005). Manuscript Document Digitalization and recognition: a first approach. *Journal of Computer Science and Technology*, 158-163.
- Michel Alvarez Cancio, A. H. (2013). PANDOC: SOFTWARE PARA LA CUANTIFICACIÓN OBJETIVA DE LA OPACIDAD DE LA CÁPSULA POSTERIOR MEDIANTE TOMOGRAMAS SCHEIMPFLUG DEL PENTACAM.
- Michel Alvarez Cancio, Adrián Hernández Barrios,Rafael Rodríguez Puentes, Iván Hernández López. (2013). PANDOC: SOFTWARE PARA LA CUANTIFICACIÓN OBJETIVA DE LA OPACIDAD DE LA CÁPSULA POSTERIOR MEDIANTE TOMOGRAMAS SCHEIMPFLUG DEL PENTACAM.

Bibliografía

- Natalia Martínez Sánchez, María Matilde García Lorenzo, Zoila Zenaida García Valdivia. (2009). *Modelo para diseñar sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos*. Colombia.
- Oliver Findl, W. B. (2003). Comparison of 4 methods for quantifying posterior capsule opacification. *Journal of Cataract & Refractive Surgery*, 29.
- Oliver Findl, Wolf Buehl, Peter Bauer, Thomas Sycha. (2010). Intervenciones para la prevención de la opacificación de la cápsula posterior. *Biblioteca Cochrane Plus*, 2.
- Oracle. (2015). *Java*. Recuperado el 14 de 1 de 2015, de <http://www.java.com/es/about/>:
<http://www.java.com/es/about/>.
- Pacheco, A. (10 de Marzo de 1999). *Ventajas y Desventajas de los Sistemas Expertos*. Obtenido de <http://www.depi.itch.edu.mx/apacheco/ai/ventdesv.htm>
- Peña, C. A. (2015). *HERRAMIENTA PARA EL ANALISIS DE OPACIDAD CAPSULAR PARA ANDROID*. Obtenido de https://compumat.uci.cu/sites/default/files/public/p2622-ponencia-4638_0.pdf
- Pressman, R. (2005). *Ingeniería del Software un enfoque práctico*.
- Ríos, I. G. (2013). Modelo Basado en Casos para el diseño de transmisiones por engranajes cilíndricos con contacto exterior. *Trabajo de Diploma*, 64.
- Rodríguez, A. (2006). ¿QUÉ ES JAVA? CONCEPTO DE PROGRAMACIÓN ORIENTADA A OBJETOS FRENTE A PROGRAMACIÓN ESTRUCTURADA . *APRENDER A PROGRAMAR*, 4.
- SA, B. (2000). Quantification of posterior capsular opacification in digital images after cataract surgery. 3979.
- Sánchez Mendoza, M. A. (2004). Metodologías del desarrollo de software.
- Sánchez, T. R. (2014-2015). *Metodología de desarrollo para la Actividad productiva de la UCI*. Ciudad de La Habana, La Habana, Cuba.
- Shulcloper, J. R. (1999). Selección de variables y clasificación supervisada. En J. R. Shulcloper, *Enfoque Lógico Combinatorio al Reconocimiento de Patrones* (Vol. 1, pág. 150). Mexico: Instituto Politécnico Nacional.
- SQLite Admin*. (2015). Recuperado el 10 de febrero de 2015, de *SQLite Administrador*:
sqliteadmin.orbmu2k.de/

Bibliografía

- SQTLite. (2015). Recuperado el 5 de febrero de 2015, de SQTLiteManagenet.org:
<http://www.sqlitemanager.org/>
- Sucar, L. E. (2014). Sistemas Basados en Conocimientos. 38.
- Tariq M Aslan, Niall Patton, Baljean Dhillon. (2005). Assessment of systems of analyzing PCO. *Journal of Cataract and Refractive Surgery, Vol. 31.*
- Valentín Tinguaro Díaz Alemán, D. Perera Sanz, V. Lozano López, Javier Rodríguez Martín. (2005). Anillos de tensión capsular. Nuestra experiencia. *Sociedad Canaria de Oftalmología.*, 16.
- Vanrell, M. (2016). *Lecture 22 - L3.5.a. Evaluación de la clasificación por ventana (I)*. Obtenido de <https://www.coursera.org/learn/deteccion-objetos/lecture/A34LS/l3-5-a-evaluacion-de-la-clasificacion-por-ventana-i>
- Vique, Robert Ramírez. (2014). *Métodos para el desarrollo de aplicaciones móviles.* . Barcelona.
- Visual-Paradigm.* (02 de 03 de 2015). Obtenido de Sitio Web oficial del producto Visual-Paradigm:
<http://www.visual-paradigm.com/product/vpum/>.
- Wolf Buehl, O. F. (2002). Reproducibility of standardized retroillumination photography for quantification of posterior capsule opacification. *Journal of Cataract & Refractive Surgery, 28(2).*

Anexos

Anexos

Anexo 1: Historias de Usuarios

Historia de usuario	
Número: 2	Nombre: Conectar la BC con la aplicación
Iteración asignada: 1	
Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: Se conecta la BC con el software PANDOC.	
Observaciones:	

Tabla 27: Historia de usuario #2

Historia de usuario	
Número: 3	Nombre: Reconocer imagen de tomograma
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: El sistema reconoce la imagen del tomograma que ha sido cargada.	
Observaciones:	

Tabla 28: Historia de usuario #3

Historia de usuario	
Número: 4	Nombre: Reconocer el centro de la estructura ocular
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas

Anexos

Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se reconoce el centro de la estructura ocular para delimitar el espacio de búsqueda.	
Observaciones:	

Tabla 29: Historia de usuario #4

Historia de usuario	
Número: 5	Nombre: Llenar la BC
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 3 semanas
Descripción: Se llena la base de conocimientos.	
Observaciones:	

Tabla 30: Historia de usuario #5

Historia de usuario	
Número: 6	Nombre: Obtener el nuevo caso
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se obtiene el nuevo caso a resolver a partir de una imagen de tomograma SCHEIMPFLUG del PENTACAM.	
Observaciones: Tiene que haber sido cargada una imagen de tomograma SCHEIMPFLUG del PENTACAM.	

Tabla 31: Historia de usuario #6

Anexos

Historia de usuario	
Número: 7	Nombre: Definir región del caso
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se definen las características necesarias para la identificación de la región en la cual se encuentra el caso que está siendo analizado.	
Observaciones:	

Tabla 32: Historia de usuario # 7

Historia de usuario	
Número: 8	Nombre: Cargar Casos similares de la BC
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se cargan de la BC los casos que resultaron más semejantes al nuevo caso.	
Observaciones:	

Tabla 33: Historia de usuario #8

Historia de usuario	
Número: 9	Nombre: Calcular similitud de rasgos
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas

Anexos

Descripción: Se calcula la similitud de cada uno de los rasgos del nuevo caso con los casos cargados de la BC.

Observaciones:

Tabla 34: Historia de usuario # 9

Historia de usuario	
Número: 10	Nombre: Calcular similitud de casos
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se calcula la similitud entre el nuevo caso y los casos cargados de la BC teniendo en cuenta cada uno de los rasgos que resultaron semejantes.	
Observaciones:	

Tabla 35: Historia de usuario # 10

Historia de usuario	
Número: 12	Nombre: Brindar datos al usuario
Iteración asignada: 4	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se le brinda al usuario los datos obtenidos del tomograma SCHEIMPFLUG del PENTACAM	
Observaciones:	

Tabla 36: Historia de usuario # 12

Historia de usuario	
Número: 13	Nombre: Recuperar el caso a la BC

Anexos

Iteración asignada: 4	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Se almacena el nuevo caso en la BC.	
Observaciones:	

Tabla 37: Historia de usuario # 13

Anexo 2: Tarjetas CRC

Clase: Conexión	
Responsabilidades	Colaboradores
1. Permite realizar consultas y operaciones con la base de conocimientos.	1. Pixel

Tabla 38: Tarjeta CRC de la clase Conexión

Clase: Lienzo	
Responsabilidades	Colaboradores
1. Permite las operaciones de extracción de los nuevos casos 2. Realiza los cálculos sobre la opacidad señalada por el sistema en la imagen 3. Establece los nuevos casos a comparar.	1. Editor 2. Pixel

Tabla 39: Tarjeta CRC de la clase Lienzo

Clase: Recortador

Anexos

Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Realiza las operaciones sobre la imagen cargada.2. Realiza los cálculos de la imagen cargada.3. Manea la configuración de la aplicación.	<ol style="list-style-type: none">1. Lienzo2. Configuración

Tabla 40: Tarjeta CRC de la clase Recortador

Clase: Configuración	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Realiza las operaciones de configuración sobre las herramientas.2. Realiza las operaciones de configuración sobre los resultados.	<ol style="list-style-type: none">1. Recortador2. Pixel

Tabla 41: Tarjeta CRC de la clase Configuración

Clase: Editor	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite realizar las operaciones con la base de conocimientos.2. Realizar el escaneo automático.3. Realizar el cálculo de la OCP.	<ol style="list-style-type: none">1. Conexión2. Configuración3. Pixel4. Lienzo

Tabla 42: Tarjeta CRC de la clase Editor

Anexos

Anexo 3: Tareas de Ingeniería generadas en la iteración 3

Tarea	
Número de tarea: 8	Número de HU: 8
Nombre: Cargar Casos similares de la BC	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 2
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se cargan de la BC los casos que resultaron más similares teniendo en cuenta el valor de la variable "Valor Global".	

Tabla 43: Tarea de ingeniería de la funcionalidad Cargar casos similares de la BC

Tarea	
Número de tarea: 9	Número de HU: 9
Nombre: Calcular similitud de rasgos	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se aplican las funciones de comparación de rasgos a aquellos casos de la BC que resultaron semejantes teniendo en cuenta el valor de la variable "Valor Global".	

Tabla 44: Tarea de ingeniería de la funcionalidad Calcular similitud de rasgos

Tarea	
Número de tarea: 10	Número de HU: 10
Nombre: Calcular similitud de casos	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 2
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se aplica la función de comparación de casos a los casos de la BC que resultaron más semejantes al nuevo caso teniendo en cuenta el resultado de la función de comparación de rasgos.	

Tabla 45: Tarea de ingeniería de la funcionalidad Calcular similitud de casos

Anexos

Anexo 4: Tareas de Ingeniería generadas en la iteración 4

Tarea	
Número de tarea: 11	Número de HU: 11
Nombre: Calcular certeza de solución	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se calcula la certeza de la solución brindada por el sistema con la herramienta calcular.	

Tabla 46: Tarea de ingeniería de la funcionalidad Calcular certeza de solución

Tarea	
Número de tarea: 12	Número de HU: 12
Nombre: Brindar datos al usuario	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Se brinda al usuario del sistema los datos referentes al grado y porcentaje de opacidad que presenta la imagen analizada.	

Tabla 47: Tarea de ingeniería de la funcionalidad Brindar datos al usuario

Tarea	
Número de tarea: 13	Número de HU: 13
Nombre: Recuperar el caso a la BC	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Programador responsable: Pedro Luis Machado, Carlos Andres Alonso	
Descripción: Si el especialista está de acuerdo con el análisis realizado por el sistema según su percepción, el nuevo caso será recuperado en la BC.	

Tabla 48: Tarea de ingeniería de la funcionalidad Recuperar el caso a la BC