

Universidad de las Ciencias Informáticas

Facultad 3



Deconvolución a ciegas para la restauración de imágenes de opacidad de la cápsula posterior en pacientes operados de catarata

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

Autor:

Yanet Pérez Rodríguez

Tutores:

Ing. Michel Alvarez Cancio

Ing. Michel Sariol Fernández

La Habana, junio de 2016

Datos de contacto

Ing. Michel Álvarez Cancio: mcancio@uci.cu

Ing. Michel Sariol Fernández: sariol@uci.cu

Agradecimientos

Declaración jurada de autoría

Declaro que soy la única autora de este trabajo de diploma y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, queda firmado el presente a los _____ días del mes de _____ del año _____.

Yanet Pérez Rodríguez

Ing. Michel Álvarez Cancio

Ing. Michel Sariol Fernández

Resumen

En la actualidad la catarata es la principal causa de pérdida de visión en todo el mundo, para el tratamiento de la misma se realiza una cirugía la cual puede traer complicaciones posoperatorias como lo es la opacidad de la cápsula posterior. Para el diagnóstico de esta enfermedad, son tomadas imágenes que en ocasiones pueden verse afectadas por ruido y desenfoque. Por lo que para dar solución a este problema se desarrolló una aplicación utilizando deconvolución a ciegas basada en Lucy Richardson que permita disminuir el ruido y desenfoque en las imágenes de opacidad de la cápsula posterior, insertándole ruido a las imágenes originales para posteriormente comparar que tan efectiva fue la aplicación. Para esto se utilizaron las métricas de calidad ssim, snr y psnr teniendo como resultado un alto valor de similitud de la imagen original con la imagen borrosa, y una buena calidad en las imágenes restauradas por el algoritmo implementado.

PALABRAS CLAVE:

deconvolución, lámpara de hendidura, desenfoque, catarata.

Índice general

0. INTRODUCCIÓN	1
1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Conceptualización	5
1.1.1. Opacidad de la cápsula posterior (OPC)	5
1.1.2. Lámpara de hendidura	6
1.1.3. Tipos de desenfoque (Blur)	7
1.1.4. Convolución	9
1.1.5. Función de dispersión de puntos (PSF)	10
1.2. Deconvolución	12
1.2.1. Métodos de deconvolución clásica	12
1.2.2. Métodos de deconvolución a ciegas	14
1.2.3. Selección del método de deconvolución a ciegas	20
1.3. Metodología, lenguajes y herramientas de desarrollo	20
1.3.1. Metodología de desarrollo	21
1.3.2. Lenguaje de modelado y herramienta CASE	22
1.3.3. Lenguaje de programación y entorno de desarrollo integrado	22
1.4. Métricas de Validación	23

1.4.1. Índice de similitud estructural (SSIM)	23
1.4.2. Relación señal ruido (SNR) y relación señal ruido de pico (PSNR)	24
1.5. Conclusiones del capítulo	25
2. ANALISIS Y DISEÑO DE LA SOLUCIÓN	26
2.1. Descripción de la solución	26
2.1.1. Requisitos funcionales	29
2.1.2. Requisitos no funcionales	29
2.1.3. Análisis de complejidad temporal	30
2.2. Fase de Planificación	34
2.2.1. Historia de usuario	34
2.2.2. Estimación de esfuerzos por Historias de Usuario	35
2.2.3. Plan de iteraciones	36
2.2.4. Plan de entrega	37
2.3. Diseño	37
2.3.1. Tarjetas CRC	38
2.3.2. Estándares de codificación	38
2.4. Conclusiones del capítulo	39
3. RESULTADOS Y VALIDACIÓN	41
3.1. Fase de implementación	41
3.2. Interfaz del sistema	43
3.3. Resultados de aplicar el algoritmo	45
3.3.1. Validación de los resultados	46
3.4. Pruebas	48
3.4.1. Pruebas de caja blanca	48

3.4.2. Pruebas de caja negra	52
3.5. Conclusiones del capítulo	53
4. CONCLUSIONES	54
REFERENCIAS BIBLIOGRÁFICAS	54
5. Anexos	60

Índice de figuras

1.1. Imagen proveniente de la lámpara de hendidura con la técnica de retroiluminación	7
1.2. Tipos de desenfoque: A la Izquierda se presenta la imagen con desenfoque por movimiento y a la derecha con desenfoque por énfasis	8
1.3. Imagen borrosa e imagen original: A la Izquierda se presenta la imagen borrosa y a la derecha su imagen recuperada	9
2.1. Secuencia lógica del algoritmo	27
2.2. Tarjeta CRC de la clase FBDHSGP.	39
3.1. Interfaz del sistema	44
3.2. 1.a)Imagen original, 1.b)Imagen borrosa, 1.c)Imagen restaurada, 1.d)PSF.	45
3.3. Resultados de aplicar el algoritmo basado en RLA	46
3.4. 1.a)Imagen original, 1.b)Imagen borrosa, 1.c)Imagen restaurada, 1.d)Mapa SSIM.	46
3.5. Código de la función hupdate	49
3.6. Representación del grafo de flujo de camino básico de hupdate	50

Índice de tablas

2.1. Historia de usuario 2 (HU2)	35
2.2. Puntos de estimación por historias de usuario	36
2.3. Iteraciones por historias de usuario	37
2.4. Plan de entrega	37
3.1. Tiempo de implementación de las HU de la primera iteración	42
3.2. Tarea de ingeniería de la transformación a espacio de color	42
3.3. Tiempo de implementación de las HU de la segunda iteración	43
3.4. Tarea de ingeniería de la convolución de la imagen con el PSF	43
3.5. Porción de 6 imágenes a las que se aplicó las métricas SSIM.	47
3.6. Porción de 6 imágenes a las que se le aplicaron las métricas SNR y PSNR.	48
3.7. Caso de prueba para hupdate	52
3.8. Caso de prueba de aceptación de la HU1	53
5.1. Historia de usuario 1 (HU1)	60
5.2. Historia de usuario 3 (HU3)	61
5.3. Historia de usuario 4 (HU4)	62
5.4. Historia de usuario 5 (HU5)	62
5.5. Historia de usuario 6 (HU6)	63



INTRODUCCIÓN

La catarata corresponde a una opacificación del cristalino que interfiere en forma progresiva con la agudeza visual, lo cual trae como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, dando lugar a una serie de imágenes difusas (DE DESARROLLO 2007) . La catarata es la principal causa de pérdida de visión en todo el mundo (se estima que es la responsable del 47 % de los casos de ceguera a nivel mundial) y es la principal causa de ceguera curable. Más de la mitad de la población mayor de 65 años tiene altas probabilidades de desarrollar cataratas, lo que origina una gran discapacidad funcional y una gran demanda asistencial (Santander 2011). El tratamiento de la misma es quirúrgico y puede traer complicaciones posoperatorias como la opacidad de la cápsula posterior OCP. La opacidad de la cápsula posterior es actualmente uno de los aspectos más importante en la cirugía de catarata de los tiempos modernos (Hernández López, Hernández Silva, Castro González, Garcés Fernández, Veitía Rovirosa & Pérez Candelaria 2010) (CRESPO, Yamaris Rosabal; SORIA 2015) (Pérez Candelaria, González Blanco, Fernández Argones, Rodríguez Suárez, Pedroso Llanes & Méndez Duque de Estrada 2010). Sigue siendo la complicación posoperatoria 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 (Hernández Silva, Ramos López, Curbelo Cunill, Fernández Vásquez, Rio Torres & Ruiz Rodríguez 2012). Para garanti-

zar una buena recuperación visual del paciente es imprescindible la transparencia de esta estructura. Su incidencia en la actualidad se encuentra entre 0,7 – 47,6 % en los primeros cinco años de la cirugía, según un metanálisis de 90 estudios publicados (**Hernández López et al.** 2010). En la Universidad de las Ciencias Informáticas se encuentra el grupo de investigación AIRI el cual en conjunto con el instituto cubano de oftalmología, Ramón Pando Ferrer", desarrollaron un software para la cuantificación de opacidad de la cápsula posterior (PANDOC). El software PANDOC provee al oftalmólogo de una herramienta por medio de la cual este 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, Rodríguez & Hernández** 2014). Una de las imágenes con las que trabaja el software son las provenientes de la lámpara de hendidura. La precisión del resultado de la cuantificación está sujeta a los parámetros de calidad de la imagen adquirida en la lámpara de hendidura, ya que en varias ocasiones se ven afectadas por elementos como desenfoque y blur propios del ambiente y las tecnologías de donde son tomadas las mismas. Como lo pueden ser:

- Movimiento ocular del paciente.
- Exceso o escasez de luz en la sala donde se toma la imagen.
- Defectos en el lente de la cámara.

Por ello es importante la obtención de imágenes con poca afectación de blur y desenfoque para el análisis por parte del software. Por tal motivo la presente investigación trata acerca de la creación de un algoritmo de deconvolución a ciegas que atenué los efectos de desenfoque y blur que pueden aparecer en dichas imágenes. 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 restaurar imágenes en retroiluminación de opacidad de la cápsula posterior de los efectos de desenfoque y blur en pacientes operados de catarata?

A raíz de esto se tomó como **objeto de estudio**: Restauración de imágenes digitales, enmarcándonos en el **campo de acción**: Deconvolución a Ciegas en imágenes médicas.

En aras de dar solución al problema planteado se tiene como **objetivo general**: Desarrollar un algoritmo para la corrección de los efectos de desenfoque y blur en imágenes de opacidad de la cápsula posterior de pacientes operados de cataratas.

Teniendo en cuenta el problema a resolver se formuló la siguiente **idea a defender**: Con el desarrollo de un algoritmo de deconvolución a ciegas, entonces se atenuarán los efectos de desenfoque y blur en las imágenes de opacidad de la cápsula posterior de pacientes operados de catarata.

Basándonos en el objetivo generar se formulan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Diseñar la propuesta de solución.
- Implementar los elementos del diseño obtenidos.
- Validar la solución propuesta.

Para dar cumplimiento al objetivo de la investigación se trazan las siguientes **tareas de la investigación**:

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de reconstrucción mediante deconvolución a ciegas en imágenes médicas.
3. Estudio del uso del método de Lucy-Richardson en la atenuación de efectos de desenfoque y blur.
4. Descripción de los pasos a seguir para el uso del algoritmo de deconvolución propuesto.

5. Diseño de la solución propuesta en función de los requisitos especificados.
6. Implementación del algoritmo diseñado para la restauración de imágenes de opacidad de la cápsula posterior con desenfoque y/o blur.
7. Validación de la implementación del sistema a partir de la aplicación de pruebas de caja blanca y caja negra.

La lógica de la investigación se desarrolla a partir de indagaciones teóricas y empíricas en las cuales rige el enfoque general dialéctico materialista del conocimiento. En el nivel teórico del conocimiento se utilizaron los métodos siguientes:

Análisis documental: Permitió realizar el estudio bibliográfico como aspecto esencial en la conformación del marco teórico referencial de la investigación, como sustento de las valoraciones realizadas, así como el estudio de los contenidos relacionados con la identificación de la OCP en pacientes operados de catarata y principales técnicas de preprocesamiento y segmentación de imágenes médicas digitales.

Análisis histórico - lógico: Posibilitó la aproximación a la evolución de los referentes teóricos del tema, analizar diferentes criterios relacionados con 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 logrará un mayor entendimiento y comprensión de este tipo de sistemas, basándose en las experiencias existentes.

Dentro de los métodos del nivel empírico del conocimiento se destacan los siguientes métodos:

Experimento: A partir de este método se pudo probar las distintas técnicas y métodos de deconvolución de imágenes médicas.

Entrevista: Proporciona datos importantes acerca de las necesidades de los especialistas en la identificación de las regiones con opacidad.



FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan algunos conceptos, que por su relación con la temática tratada es importante definir. Se describen los conceptos y métodos, que permitirán el entendimiento del tema para la posterior selección de un algoritmo que de resultado a la problemática tratada. Además se describen las herramientas, lenguajes y tecnologías de desarrollo seleccionadas para la implementación de la solución que se propone, así como los patrones y estrategias de pruebas posibles a emplear.

1.1 Conceptualización

1.1.1. Opacidad de la cápsula posterior (OPC)

La catarata constituye una patología oftalmológica caracterizada por la pérdida de la transparencia de uno de los medios refringentes del ojo, el cristalino. La cápsula posterior es una estructura que anatómicamente queda por detrás del lente intraocular implantado en el paciente operado de cataratas, para garantizar una buena recuperación visual del mismo es imprescindible la transparencia de esta estructura (Alvarez et al. 2014). La cápsula posterior en muchas ocasiones después de la cirugía de cataratas se ve afectada por la opacidad impidiendo la transparencia de la misma. En el campo de la oftalmología

esto es conocido como OCP. La opacidad de la cápsula posterior es actualmente el aspecto más importante en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación posoperatoria 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. Su incidencia en la actualidad se encuentra entre 0,74 y 7,6 % en los primeros cinco años de la cirugía, según un metanálisis de 90 estudios publicados. Que el rango de variación reportado sea tan amplio probablemente responda a diferentes criterios de selección de la muestra, edad, técnica quirúrgica, tipo de lente intraocular (LIO) y sistemas de evaluación de la opacidad (**Hernández López et al.** 2010).

1.1.2. Lámpara de hendidura

El diagnóstico de la OCP se realiza con imágenes provenientes de la lámpara de hendidura. Esta es un dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras del polo anterior del ojo (**Salorio**). Consiste en un microscopio binocular dotado de una potente fuente de luz que permite ver en tres dimensiones y con una amplificación que oscila entre 6 y 40 aumentos, las estructuras del polo anterior del ojo: párpado, conjuntiva, córnea, iris, cristalino y cámara anterior. Se trata de una tecnología esencial en oftalmología y optometría con la que es posible detectar la posición, forma y profundidad de cualquier anomalía en la estructura del ojo (**Kanski**). Para el análisis de la OCP la lámpara de hendidura obtiene las imágenes en retroiluminación que no es más que una técnica que utiliza la luz proveniente de la reflexión interna, en los medios opacos (iris, retina) o semitransparentes (cristalino), para iluminar desde atrás (es decir desde el interior del ojo) la córnea o el cristalino (**Faccia** 2014). En este tipo de imágenes se ve reflejada la opacidad como una textura que sobresale en la cápsula posterior, como se muestra en la figura 1.1.

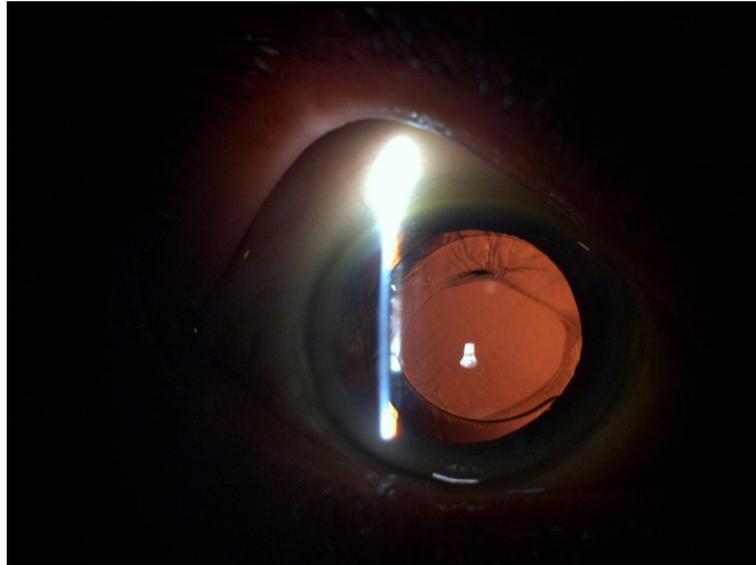


Figura 1.1: Imagen proveniente de la lámpara de hendidura con la técnica de retroiluminación

Las imágenes que son tomadas por la lámpara de hendidura muchas veces no se obtienen con la calidad que es requerida para detectar las diferencias de opacidad dado a que presentan ruido y desenfoque (blur). El (blur) puede ser producido por varios factores debido a variaciones en las condiciones propias del evento bajo análisis, la iluminación, el instrumento de captura y las fuentes de ruido propias del ambiente, siendo causa de pérdidas de información (Archila et al. 2012) importante que brindan estas imágenes.

1.1.3. Tipos de desenfoque (Blur)

El fenómeno de desenfoque es clasificado en dos categorías (Archila et al. 2012), ver figura 1.2

Desenfoque por movimiento: Ocasionado al intentar registrar un evento manipulando el tiempo de abertura del obturador del mecanismo de adquisición, se presenta en eventos ocurridos a alta o baja velocidad (Archila et al. 2012).

Desenfoque global: Se obtiene por énfasis en una parte de la imagen, se encuentra espe-

cialmente en tomas de primeros planos (Archila et al. 2012).



Figura 1.2: Tipos de desenfoco: A la Izquierda se presenta la imagen con desenfoco por movimiento y a la derecha con desenfoco por énfasis

Las imágenes en retroiluminación provenientes de la lámpara de hendidura pueden presentar ambos tipos de desenfoco, y debido a esto cuando se pretenda evaluar la opacidad de un paciente pues los resultados pudieran dar alterados y dañar el diagnóstico del mismo, es por eso que es necesario restaurarlas. La deconvolución de imágenes es un tema que pretende recuperar la imagen original de su versión borrosa y ruidosa (Galatsanos, Nikolas P. 2003) (Kundur & Hatzinakos 1996) figura 1.2, por lo que se estará profundizando en este tema para la restauración de las imágenes en retroiluminación provenientes de la lámpara de hendidura. Considerando la deconvolución estándar, podemos modelar el problema mediante la siguiente ecuación (Galatsanos, Nikolas P. 2003) (Kundur & Hatzinakos 1996):

$$g(x) = f(x) * h(x) + n(x) \quad (1)$$

Donde $g(x)$ es la imagen borrosa observada, dada por la convolución de la imagen real con una función de dispersión de puntos (PSF) desconocido $h(x)$ y se asume la adición de un ruido aditivo $n(x)$.

La clave para ello está en resolver este problema matemático inverso (Hernández López 2010) (Campisi & Egiazarian 2016) (Andrews & Hunt 1977) (Hansen, Nagy & O'leary 2006) donde es apropiado incorporar información a priori de la imagen a recuperar en el proceso de restauración. Para esto se necesita obtener la degradación de la imagen que es conocida como PSF, ver figura 1.3.



Figura 1.3: Imagen borrosa e imagen original: A la Izquierda se presenta la imagen borrosa y a la derecha su imagen recuperada

1.1.4. Convolución

La convolución es una operación sobre un vecindario en el cual cada píxel de salida es una suma ponderada de los píxeles vecinos. La matriz de pesos es llamada núcleo de convolución, también conocido como filtro. A nivel espacial en la convolución, el kernel se transforma y las filas se convierten columnas, pero a nivel de frecuencia es una multiplicación punto a punto. Se evalúa el píxel central de la ventana de acuerdo con los píxeles de su vecindad reducida y sus valores de ponderación. Cuando un nuevo valor es así calculado, se desplaza la ventana sobre el siguiente píxel, realizando la misma operación. Este proceso de evaluar la vecindad ponderada del píxel, se denomina "convolución bidimensional", y a la matriz de filtro se le conoce como "kernel de convolución". (37)

1.1.5. Función de dispersión de puntos (PSF)

Función de dispersión de puntos también conocido como función de apertura de un punto o PSF (*Point Spread Function*), el cual se refiere a cuánto una fracción del objetivo observado incidirá en la zona aledaña a los píxeles que representan dicha fracción en la imagen una vez digitalizada (Cristina, Bria & Russo 2004), es una función que describe la distribución de la luz en el plano focal (Lombardi 2010). En otros palabras, es una función que almacena el desplazamiento de la imagen y la intensidad de los píxeles para que, cuando esta deconvolucione con la imagen dañada, quede eliminado este desplazamiento de la imagen. Los artículos de restauración han estudiado los modelos de PSF que mejor representan las degradaciones en los diferentes campos de aplicación ((Hernández López 2010), (Campisi & Egiazarian 2016),(Hansen et al. 2006)). A continuación se abordará acerca de los modelos para la estimación del PSF, entre ellos están:

1. Emborronamiento por movimiento lineal uniforme (Andrews & Hunt 1977)

Son varios los efectos de degradación que puede tomar la función PSF, entre estos se encuentra (movimiento rectilíneo uniforme). Así pues, considerando una escena que se desplaza respecto a la cámara a una velocidad constante, con un ángulo respecto al eje horizontal y durante un periodo de tiempo, se define la longitud del movimiento como $d = vt$ y la PSF se representa por (Hansen et al. 2006):

$$h(i, j) = \begin{cases} \frac{1}{d} \text{ si } \sqrt{i^2 + j^2} \leq \frac{d}{2} \text{ y } \frac{j}{i} = -\tan(\phi) \\ 0 \text{ En otro caso} \end{cases} \quad (2)$$

2. Emborronamiento por desenfoque uniforme

El desenfoque de una imagen captada por una cámara se modela en cada punto por el denominado círculo de confusión, cuyo radio depende de la longitud focal, la apertura del lente, la longitud de onda de la luz y la distancia entre la cámara y el objeto. Si el grado del desenfoque (radio del círculo) es grande respecto a las

longitudes de onda de la luz que atraviesan la cámara, se utiliza una aproximación geométrica que da lugar a una distribución uniforme dentro del círculo de confusión. En estas condiciones, el PSF se define como (Hansen et al. 2006):

$$h(i, j) = \begin{cases} \frac{1}{\pi R^2} & \text{si } \sqrt{i^2 + j^2} \leq R \\ 0 & \text{En otro caso} \end{cases} \quad (3)$$

donde R corresponde al radio del PSF. Este modelo asume que la distancia del objeto desde la cámara es constante en toda la imagen, lo que le permite ser un PSF invariante espacial (Hansen et al. 2006).

3. Emborronamiento uniforme bidimensional

Esta forma de emborronamiento supone una aproximación a la degradación por desenfoque anterior y se emplea en un gran número de simulaciones sobre restauración de imágenes. Su expresión viene dada por (Hansen et al. 2006):

$$h(i, j) = \begin{cases} \frac{1}{d^2} & \text{si } -\frac{d}{2} \leq i, j \leq \frac{d}{2} \\ 0 & \text{En otro caso} \end{cases} \quad (4)$$

4. Emborronamiento por turbulencia atmosférica

La degradación debida a los cambios en las condiciones de refracción de la atmósfera tiene una importante influencia en la calidad de las imágenes para teledetección. Este tipo de emborronamiento no admite un modelo determinista por su dependencia respecto a factores tales como la temperatura, el tiempo de exposición o las condiciones meteorológicas de la adquisición. Si se considera una exposición prolongada, lo cual suele ser frecuente en las observaciones astronómicas, una buena aproximación de la turbulencia atmosférica es un PSF Gaussiano que responde a la expresión (Hansen et al. 2006)

$$h(i, j) = \frac{1}{2\pi\sigma_R^2} e^{-\frac{i^2 + j^2}{2\sigma_R^2}} \quad (5)$$

Donde σ controla la severidad de la degradación.

1.2 Deconvolución

En la restauración de imágenes clásica se busca una estimación de la imagen verdadera, donde se supone que la degradación es conocida. En cambio, la restauración a ciegas en las imágenes, aborda el problema mucho más difícil, pero más realista, donde es desconocida la degradación. Para llevar a cabo la deconvolución existen distintos métodos y clasificaciones, entre estos está “no blind deconvolution” o deconvolución clásica, el cual se aplica cuando conocemos el PSF. Por otro lado esta “blind deconvolution” o deconvolución a ciegas que es todo lo contrario y se utiliza cuando desconocemos la función PSF (Campisi & Egiazarian 2016).

1.2.1. Métodos de deconvolución clásica

Los métodos clásicos son aquellos que se basan en funciones originales que no dependen del modelado utilizado para el ruido de las observaciones, entre estos se encuentra (Andrews & Hunt 1977):

Filtrado de Wiener (FW)

El **filtro de Wiener** es uno de los filtros lineales óptimos más importantes. El objetivo del filtrado de Wiener es determinar la respuesta al impulso $h(n)$, de forma que el error $e(n)$ sea, en términos estadísticos, mínimo. El criterio elegido es la minimización del valor cuadrático medio del error:

$$\phi = E \{ |e(n)|^2 \} \quad (6)$$

Se considera un filtro causal de longitud p , orden $p - 1$, de respuesta al impulso $w(n)$ y función de red:

$$W(z) = \sum_{n=0}^{p-1} w(n)z^{-n} \quad (7)$$

Si $x(n)$ es la entrada al filtro, la salida es la estimación de $d(n)$ obtenida como la convolución de $w(n)$ con $x(n)$:

$$d(n) = \sum_{l=0}^{p-1} w(l)x(n - l) \quad (8)$$

Se deben encontrar los coeficientes que minimizan la función de costo, por tanto, derivando $e^*(n)$ igualando las derivadas a cero para los valores posibles de k ($k = 0, 1, \dots, p - 1$), al derivar y sustituir, se obtiene el principio de ortogonalidad o teorema de proyección. Este teorema indica que es mínimo y los coeficientes del filtro asumen sus valores óptimos cuando $e(n)$ es ortogonal a cada muestra de entrada $x(n)$, que es utilizada para el cálculo de la estimación. Por tanto, el error también es ortogonal a la salida del filtro. Este principio establece una condición suficiente y necesaria para la optimización. Se tiene un conjunto de p ecuaciones lineales con p incógnitas $w(k)$, para l . En forma matricial, utilizando el hecho de que la secuencia de autocorrelación posee simetría conjugada, $r_x(k) = r_x^*(-k)$, se pueden expresar las ecuaciones de Wiener-Hopf en notación vectorial como:

$$R_x w = r_{dx} \quad (9)$$

Donde R_x es la matriz de autocorrelación Toeplitz hermética $p \times p$, w es el vector de coeficientes del filtro y r_{dx} es el vector de correlación cruzada entre la señal deseada y la

señal recibida . En cada iteración, la actualización de un peso requiere dos multiplicaciones complejas y una suma. Por tanto, para un filtro de p coeficientes, se realizan $2p + 1$ multiplicaciones complejas y $2p$ sumas complejas por iteración; es decir, la carga computacional es del orden de $o(p)$.

Función de correlación La correlación cruzada es una medida de la similitud entre dos señales. Es frecuentemente usada para encontrar características relevantes en una señal desconocida por medio de la comparación con otra que sí se conoce. Dadas dos secuencias de señal reales $x(n)$ y $y(n)$, cada una con energía finita, la correlación cruzada de $x(n)$ y $y(n)$ es una secuencia r_{dx} definida como:

$$r_{dx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n); l \in \mathbb{Z} \quad (10)$$

Matriz Toeplitz

En el álgebra lineal, una matriz de Toeplitz es una matriz cuadrada con todas sus diagonales, de izquierda a derecha, paralelas numéricamente. Una matriz de Toeplitz presenta la siguiente estructura:

$$\forall a_{i,j} \in T \rightarrow a_{i,j} = a_{i+l,j+l} \quad (11)$$

1.2.2. Métodos de deconvolución a ciegas

Los algoritmos de deconvolución ciega, requieren una estimación del tamaño de PSF. Esto se hace en base de prueba y error hasta que se obtiene una imagen restaurada de buena calidad atendiendo a disímiles medidas de calidad según la aplicación (**Fahmy, Raheem, Mohamed, Fahmy et al.** 2012). Dependen específicamente del modelado para el ruido de las observaciones, los estudiados fueron:

1. Maximun Entropy (ME)

El principio de entropía máxima se basa en la premisa de que al estimar la distribu-

ción de probabilidad, debe seleccionar esa distribución que le deja la mayor incertidumbre restante, la entropía máxima, consistente con sus restricciones. De esta manera no ha introducido ninguna suposición o sesgo adicional en sus cálculos (Penfield Jr 2010). La Entropía Máxima en general, es una técnica que nos ayuda a estimar la distribución de probabilidad a partir de datos. El principio de ME es que la distribución debe ser tan uniforme como pueda, cuando no se conoce nada (Nigam, Lafferty & McCallum 1999). Se consideran los datos caracterizados por la distribución de intensidad $I(x, y)$, correspondiente a la observación de un objeto $O(x, y)$ a través de un sistema óptico. Asumimos que el proceso de degradación es lineal con ruido aditivo:

$$I(x, y) = O * P(x, y) + N(x, y) \quad (12)$$

Donde $P(x, y)$ es la función de apertura de puntos (PSF) del sistema de formación de imágenes, y $N(x, y)$ es ruido aditivo. Queremos determinar $O(x, y)$ conociendo $I(x, y)$ y $P(x, y)$. En el caso de Poisson, o Poisson más gaussiano, podemos reducir el problema al caso gaussiano usando la transformación y su generalización. También consideramos que el número de puntos en los datos es el mismo que el número de píxeles en la imagen original O . Este problema inverso está mal planteado debido al ruido, haciendo que el sistema de ecuaciones anterior carezca de una solución única. El método de (ME), que es un enfoque estocástico del problema, fue inicialmente sugerido por (Ratnaparkhi 1997). Entre un número infinito de soluciones a la ecuación anterior, minimiza la cantidad de información contenida utilizando el teorema de Bayes para evaluar la probabilidad de la realización de la imagen original, conociendo los datos tenemos:

$$Prob(O/I) = - \frac{Prob(I/O) + Prob(O)}{Prob(I)} \quad (13)$$

$Prob(I/O)$ es la probabilidad condicional obteniendo los datos de I dada a la imagen original O . Representa la distribución del ruido. Se da, en el caso de ruido gaus-

siano no correlacionado con varianza σ_1^2 por:

$$Prob(I/O) = \hat{P} \frac{(I - P * O)^2}{2\sigma_1^2} \quad (14)$$

En la ecuación anterior el denominador es independiente de O y se considera como una constante. $Prob(O)$ es la distribución a priori de la solución O y $\hat{P} = e^{\sum \text{pixeles}}$. En ausencia de cualquier información sobre la solución O , excepto su positividad, el principio de la entropía máxima sugiere tomar:

$$Prob(O) = e^{\alpha S(O)} \quad (15)$$

Donde $S(O)$ denota la entropía de la imagen O . Dados los datos, el principio de máxima verosimilitud se aplica maximizando $Prob(O/I)$, o equivalentemente maximizando el producto de las dos ecuaciones anteriores. Tomando el logaritmo, necesitamos maximizar:

$$\ln(Prob(O/I)) = \alpha S(O) - \sum \text{pixeles} \frac{(I - P * O)^2}{2\sigma_1^2} \quad (16)$$

Que es una combinación lineal de dos términos: la entropía de la imagen y una cantidad correspondiente a x^2 en la estadística que mide la discrepancia entre los datos y las predicciones del modelo. La solución se encuentra minimizando:

$$J(O) = \sum \text{pixeles} \frac{(I - P * O)^2}{2\sigma_1^2} - \alpha S(O) = x^2 - \alpha S(O) \quad (17)$$

Donde α es un parámetro que puede verse alternativamente como un parámetro lagrangiano o un valor que fija el peso relativo entre la bondad de ajuste y la entropía S . Existen varias definiciones de entropía.

a)

$$S_b = - \sum : (\text{pixeles}) O \ln(O) \quad (18)$$

b)

$$S_j = - \sum : (\text{pixeles}) \ln(O) \quad (19)$$

c)

$$S_b = - \sum : (\text{pixeles}) O - M - O \ln(O/m) \quad (20)$$

La última definición de la entropía tiene la ventaja de tener un máximo cero cuando O es igual a m , pero requiere el concepto de un modelo, m , que es en la práctica el valor del fondo. La determinación del parámetro α no es una tarea fácil y de hecho es un problema muy serio que enfrenta el método ME.

2. Sigma Clean (SC)

La técnica Sigma Clean es utilizada con frecuencia en el procesamiento de datos de síntesis de apertura de radio. Procede localizando el punto máximo en un mapa y restando alguna fracción de su intensidad de fuente puntual estimada a partir del mapa, añadiendo una fuente puntual del mismo flujo a la imagen deconvolucionada. Este proceso continúa hasta que la convergencia se satisface o hasta que nuevas iteraciones diverjan realmente en la estimación de la producción. Todas las operaciones matemáticas son lineales, el único aspecto no lineal de algoritmo está en el orden de selección de los componentes a restar. Un mapa de síntesis de apertura reside en el dominio de Fourier, pero el ruido en las imágenes ópticas existe en el dominio de la imagen y, por tanto, contiene componentes de Fourier en todas las frecuencias hasta píxeles individuales. Además, el ruido varía fuertemente a través de una imagen directa típica, debido a las fluctuaciones estadísticas en las regiones de gran superficie, el brillo puede ser mucho mayor que el real, y significativo en áreas de bajo brillo superficial. Por lo tanto, para situaciones en las que es deseable la deconvolución, el algoritmo SC finalmente encontrará puntos máximos de ruido más cortos que el PSF y producen parejas de componentes positivos y negativos cu-

ya amplitud aumenta con un mayor número de iteraciones, a expensas de las áreas de la imagen que contienen el flujo físicamente significativo ("Bloqueo de ruido"). Este problema es más grave en las imágenes que poseen un amplio rango dinámico, que son sólo las que la deconvolución puede producir ganancias útiles (Keel 1991). Una forma efectiva de eliminar este comportamiento es utilizar un modelo de ruido para la imagen de entrada para seleccionar el punto máximo (pico) que es más significativo, no el mayor en amplitud.

Función de dispersión de puntos: El PSF se normaliza dentro del programa. Los únicos pasos de procesamiento importantes necesarios son la construcción de una versión de el PSF con el punto máximo en el centro de un píxel y asegurar que va a cero en el borde de la matriz PSF. (Keel 1991).

3. Richardson-Lucy (RLA)

El algoritmo de Richardson-Lucy (RLA) es un método de deconvolución que consiste en calcular la imagen más cercana a la imagen real de una imagen observada. La difuminación de las imágenes de movimiento puede ser modelada como la convolución de la imagen sin distorsión y el PSF. El RLA es un método iterativo de restauración de la imagen basada en el principio de la probabilidad máxima utilizando el modelo de Poisson. Para restaurar una imagen el RLA se deriva del teorema de Bayes, basado en la idea de que la imagen de entrada, PSF y la imagen de salida puede ser considerada como probabilidades. El Teorema de Bayes establece una relación entre las distribuciones de probabilidad (de acuerdo con la ecuación) (Anaconda Mosquera 2016).

$$p(O|I) = \frac{p(I|O)p(O)}{p(I)} \quad (21)$$

donde $p(O|I)$ es la distribución de probabilidad de la imagen restaurada, $p(I|O)$ es la distribución de probabilidad de el PSF, $p(O)$ es la distribución de probabilidad

de la imagen real, $p(I)$ es la distribución de probabilidad de imagen borrosa. La probabilidad de esta imagen observada está dada por (Anaconda Mosquera 2016):

$$p(I|O) = \prod_{x,y} \frac{(h(x,y) * f(x,y))^{g(x,y)} e^{-(h(x,y)*f(x,y))}}{g(x,y)!} \quad (22)$$

en el que el máximo se puede calcular tomando la derivada del logaritmo:

$$\frac{\partial \ln(p(I|O)(x,y))}{\partial O(x,y)} = 0 \quad (23)$$

en el que suponiendo que el PSF se normaliza, esto conduce a una ecuación que puede ser resuelta de forma iterativa, como se muestra en la siguiente ecuación:

$$O^{n+1}(x,y) = \left[\frac{I(x,y)}{P * O^n(x,y)} * P^T(x,y) O(x,y) \right] \quad (24)$$

donde $P(x,y) = P(x,y)$, P es el PSF y P^T es la transpuesta del PSF, I es la imagen observada, O^n es la estimación de la imagen real y el operador $*$ es convolución. Teniendo en cuenta que el ruido aditivo es amplificado a medida que pasa cada iteración, por lo que cuando el número de iteraciones tiende a infinito la imagen resultante después de la restauración es ruido. En la práctica se puede establecer criterios para detener el número de iteraciones con el fin de reducir la cantidad de amplificación del ruido. Uno de estos criterios puede medir la calidad de la imagen restaurada en comparación con la imagen original (Anaconda Mosquera 2016).

El algoritmo de Richardson-Lucy es un algoritmo sencillo de implementar, pero su principal inconveniente es la amplificación del ruido y su criterio de parada. Una solución a este problema es el uso de filtros en el comienzo de la restauración o durante la restauración de la imagen. (Anaconda Mosquera 2016). Las principales desventajas de los métodos no lineales son que su comportamiento no es siempre predecible

y generalmente requieren recursos computacionales considerables, la primera desventaja a menudo pierde importancia basado en que los métodos no lineales son superiores a los métodos lineales en una amplia gama de aplicaciones. La segunda desventaja pierde importancia debido al gran desarrollo computacional en la última década (Masache Narvaez & Roldan Monge 2010).

1.2.3. Selección del método de deconvolución a ciegas

- En el método Maximum Entropy la determinación del parámetro α no es una tarea fácil y de hecho es un problema muy serio que enfrenta esta técnica.
- Una ventaja potencial de Sigma Clean es que el PSF sólo necesita ser definido localmente; una familia entera de formas de PSF, en principio hasta una por pixel, podría ser definida y utilizada para la deconvolución, pero no existe basta documentación de cómo trabaja este algoritmo y esto podría ser un inconveniente a la hora de realizar una investigación y posteriormente una implementación.
- El algoritmo de Richardson-Lucy es un algoritmo sencillo de implementar y tiene una gran cantidad de bibliografía donde se puede investigar acerca de este método.
- Para la restauración de las imágenes en retroiluminación provenientes de la lámpara de hendidura se decidió implementar un algoritmo de deconvolución a ciega basado en RLA.

1.3 Metodología, lenguajes y herramientas de desarrollo

En todos los proyectos de desarrollo de software un paso fundamental es la elección de la metodología, lenguajes y herramientas de desarrollo, a continuación se describen los principales argumentos considerados para la ejecución del presente trabajo de diploma.

1.3.1. Metodología de desarrollo

Entre las fases existentes para el desarrollo de un software se encuentra la selección de una metodología, lo cual se hace un poco complejo dada la cantidad existente de la misma que inciden en distintas dimensiones del proceso de desarrollo. Primeramente tenemos propuestas tradicionales o robustas que se enfocan principalmente 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 plasmar cada detalle y en el uso exhaustivo de documentación durante todo el proyecto. Por otra parte están aquellas 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. La calidad de un software depende, a gran escala de la correcta elección de una de estas metodologías(Canós, Letelier & Penadés 2003). Para el desarrollo del software se ha decidido utilizar una metodología ágil ya que es primordial la satisfacción del cliente que forma parte del equipo de desarrollo. Este equipo es conformado por una sola persona lo cual sería complicado adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo.

Metodología XP

Durante la elaboración de la aplicación se utiliza la metodología de software XP programación extrema, dado que 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 en el código.

Características de XP (Sanchez 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.

XP consta de 4 fases:

- Planificación
- Diseño
- Desarrollo
- Pruebas

1.3.2. Lenguaje de modelado y herramienta CASE

Visual Paradigm 8.0 para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (**Paradigm** 2016).

1.3.3. Lenguaje de programación y entorno de desarrollo integrado

Para el desarrollo de esta aplicación se escogió **MatLab** que es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux. Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la

creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets). Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL (**MathWorks**). La aplicación que se estará desarrollando tiene en cuenta matrices, formulas basadas en probabilidades, donde la matemática juega un papel fundamental por lo que esta herramienta es de gran importancia para un correcto desarrollo del software.

1.4 Métricas de Validación

Para validar la eficiencia del algoritmo propuesto es necesario aplicar métricas de calidad que permitan conocer que tan similar es la imagen resultante con respecto a la imagen de referencia (Imagen Original) y así estimar su nivel de ruido. Para esto se deciden utilizar las siguientes métricas:

- Índice de similitud estructural (SSIM)
- Relación señal ruido (SNR) y relación señal ruido de pico (PSNR)

1.4.1. Índice de similitud estructural (SSIM)

Índice de Similitud Estructural (SSIM) se basa en el cálculo de tres términos, a saber, el término de luminancia, el término de contraste y el término estructural. El índice general es una combinación multiplicativa de los tres términos(Wang, Bovik, Sheikh & Simoncelli 2004).

$$SSIM(x, y) = [l(x, y)^\alpha] \cdot [c(x, y)^\beta] \cdot [s(x, y)^y] \quad (25)$$

donde

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (26)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (27)$$

$$c(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (28)$$

Donde $\mu_x, \mu_y, \sigma_x, \sigma_y$ y σ_{xy} son los medios locales, las desviaciones estándar y la covarianza cruzada para las imágenes x, y . Si $\alpha = \beta = y = 1$ (el valor por defecto para los exponentes) y $C_3 = C_2/2$ (selección por defecto de C_3) el índice se simplifica a:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (29)$$

Esta métrica se expresa en el rango de 0 a 1, a medida que el valor *ssim* se acerque más a cero, mas semejanza tiene la imagen restaurada con respecto a la imagen original.

1.4.2. Relación señal ruido (SNR) y relación señal ruido de pico (PSNR)

El PSNR calcula la relación pico de señal a ruido, en decibelios, entre dos imágenes. Esta relación se utiliza a menudo como una medida de calidad entre la imagen original y una imagen comprimida. Cuanto mayor sea el PSNR, mejor será la calidad de la imagen comprimida o reconstruida (**MathWorks** 2017).

$$SNR = \frac{PotenciaImagen}{PotenciaRuido} \quad (30)$$

$$PSNR = 10 \log_{10}\left(\frac{R^2}{SNR}\right) \quad (31)$$

En la ecuación anterior, R es la fluctuación máxima en el tipo de datos de imagen de entrada. Por ejemplo, si la imagen de entrada tiene un tipo de datos de coma flotante de doble precisión, entonces R es 1. Si tiene un tipo de datos enteros sin signo de 8 bits, R es 255, etc (MathWorks 2017).

1.5 Conclusiones del capítulo

- En el capítulo se realizó un estudio sobre los principales conceptos que se deben tomar en cuenta para dar respuesta al problema planteado como lo son: la opacidad de la cápsula posterior, lámpara de hendidura, función de apertura de punto (PSF) de punto, deconvolución, entre otros.
- Se realizó un estudio de diferentes métodos de deconvolución a ciegas el cual tuvo como resultado la selección del algoritmo Richardson-Lucy para la restauración de las imágenes en retroiluminación provenientes de la lámpara de hendidura.
- Se identificaron y describieron las tecnologías y herramientas que se emplearan para desarrollar la propuesta planteada, teniendo como resultado **XP** como metodología de desarrollo, debido a sus características, **MatLab** como lenguaje de programación y el **Visual Paradigm** como herramienta CASE.



ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

En este capítulo se presentan los artefactos generados propios de la metodología de desarrollo utilizada para la implementación de la solución que se propone tales como: historias de usuario y tarjetas CRC. Además se realiza una descripción detallada de la solución propuesta con las herramientas y tecnologías a utilizar y se realiza el modelado completo de la solución.

2.1 Descripción de la solución

Para dar solución a la problemática planteada se decidió la implementación del método de deconvolución a ciegas Richardson Lucy, al cual se le pasa por parámetro la imagen que quiere ser restaurada y el número de iteraciones y el mismo te devuelve la imagen con poca afectación de ruido y desenfoque, para así minimizar el sesgo entre un especialista y otro en cuanto al diagnóstico de la opacidad de la cápsula posterior en pacientes operados de cataratas. A continuación se muestra un diagrama de actividades (figura 2.1) que modela la secuencia de pasos del algoritmo.

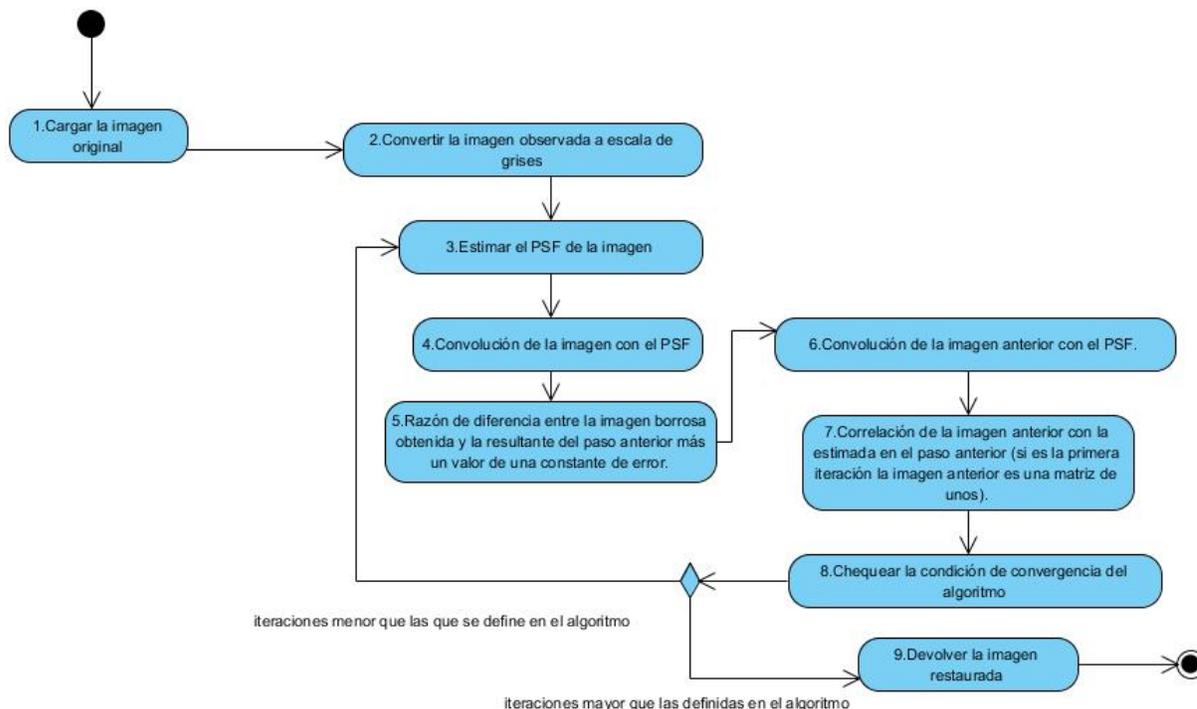


Figura 2.1: Secuencia lógica del algoritmo

Paso 1: Cargar la imagen

En este paso el usuario selecciona la imagen borrosa que desea restaurar mediante la propuesta de RLA presentada.

Paso 2: Convertir la imagen observada a escala de grises

La escala de grises es una escala empleada en la imagen digital en la que el valor de cada píxel posee un valor equivalente a una graduación de gris. Las imágenes representadas de este tipo están compuestas de sombras de grises, por lo que se lleva la imagen original a escala de grises para un mejor trabajo con la misma.

Paso 3: Estimar el PSF de la imagen

Se realiza una estimación de PSF empleando restricciones a partir de **Super Gaussian Prior**, donde a través de parámetros definidos a priori según el posible comportamiento del desenfoque de la imagen, se obtiene el PSF mejor adaptado a la imagen que se desea restaurar.

Paso 4: Convolución de la imagen con el PSF:

Se realiza la convolución de la imagen convertida en escala de grises con la función PSF. La convolución es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g .

Paso 5: Razón de diferencia entre la imagen borrosa obtenida y la resultante del paso anterior más un valor de una constante de error.

Se realiza una división punto a punto entre la matriz de los valores de intensidad de la imagen borrosa obtenida y la estimada en esa iteración y se le suma una constante de error. Esta imagen resultante nos brinda los valores de estimación y la constante nos permite ajustar los resultados del cálculo.

Paso 6: Convolución de la imagen anterior con el PSF.

Se realiza el mismo procedimiento que en el **paso 4** pero con la imagen obtenida en el **paso 5**.

Paso 7: Correlación de la imagen anterior con la estimada en el paso anterior (si es la primera iteración la imagen anterior es una matriz de unos)

Se modifica la representación matricial de los píxeles de la imagen anterior, proceso que se repite iterativamente hasta que se alcanza el máximo en la función optimalidad de estimación de PSF.

Paso 8: Chequear la condición de convergencia del algoritmo

Se valida la condición, si las iteraciones son menor que las definidas se ejecuta nuevamente desde el **paso 2**, sino , se pasa al **paso 9**.

Paso 9: Devolver la imagen restaurada

En este paso se muestra la imagen borrosa, el PSF estimado, y la imagen restaurada.

2.1.1. Requisitos funcionales

Los requisitos funcionales (RF) describen lo que el sistema debe hacer, así como las condiciones y capacidades que debe cumplir la herramienta.

RF1. Cargar la imagen.

RF2. Deconvolucionar imagen.

RF2.1 Transformar a espacio de color.

RF2.2 Hallar la magnitud en la superposición de la imagen.

RF2.3 Encontrar el PSF mas optimo para la imagen.

RF2.4 Estimar imagen original.

RF3. Mostrar la imagen.

2.1.2. 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:

RF1 Usabilidad.

- El sistema podrá ser usado por personas sin avanzados conocimientos informáticos.

RF2 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.

- Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error.

2.1.3. Análisis de complejidad temporal

Paso 1: $O(1)$

Paso 2: $O(n.m)$ % donde n es la cantidad de filas y m la cantidad de columnas de la matriz(imagen).

Paso 3: $O(n.m)(k)$ %donde k es la dimensión del kernel.

+ $O((k)$

+ $O(k)$

+ $O(k)$

+ $O(n.m)$ % complejidad temporal de la transformada de furier y la inversa.(Frigo & Johnson 1998)

+ $O(n^2.m^2)k$

Paso 4: $O(n.m.max(k).w)$ % w es la cantidad de escalas definidas en la aplicación.

Paso 5: $O(n.m) + O(n.m)$

Paso 6: $O(n.m) + O(n.m)$

Paso 7: $O(n.m.k)w$

Paso 8: $O(1)$

Después de haber analizado la complejidad temporal de cada uno de los pasos descritos en la propuesta de solución, se aplicó la regla de la suma teniendo como resultado una complejidad temporal de $O(w(n^2.m^2.max(k))T)$, donde T es la cantidad de iteraciones realizadas por el algoritmo.

Teniendo en cuenta que $w = 7$, $k = 35$, en su peor caso, $T = 100$ en su peor caso según (Anacona Mosquera 2016),(Masache Narvaez & Roldan Monge 2010),(Keel 1991), y todas son contantes por lo que son despreciables se tiene finalmente que la compleji-

dad temporal del algoritmo propuesto basado en RLA tiene una complejidad temporal de $O(n^2.m^2)$.

A continuación se muestra el pseudocódigo de las funcionalidades que dieron solución a algunos de los pasos descritos anteriormente.

Algoritmo 1 FBDHSGP()

Entrada: imagen, parámetros de entrada

Salida: kernel, imagen deconvolucionada final.

```
1: si(imagen = null)
2: error ("No hay imagen")
   fin si
3: sino
4: y=imagen
   fin sino
5: Llevar la imagen a formato de doble precisión
6: Aplicar corrección gamma
7: si(Está el kernel en los parámetros)
8: Inicializar el kernel en un vector vacío
   fin si
9: si(Está el kernel )
10: Asignar el kernel a una variable
11: si(Tiene tres canales la imagen )
12: Llevar la imagen a escala de grises rgb2ycbr();
   fin si
   fin si
13: sino si(tiene tres canales)
14: Llevar la imagen a escala de grises rgb2ycbr();
   fin sino
```

-
- 1: Establecer el tamaño del kernel
 - 2: Inicializar las escalas
 - 3: **mientras**(iteraciones<max_num_iteraciones)
 - 4: Obtener el número de escala
fin mientras
 - 5: **para**(Cada una de las escalas)
 - 6: **si**(s=1) % primera escala
 - 7: Inicializar el valor del kernel usando un kernel gaussiano
 - 8: Ajustar el tamaño del kernel a la escala pasada
 - 9: Normalizar
fin si
 - 10: Redimensionar la imagen según la escala del kernel actual
 - 11: Redimensionar la imagen estimada en cada paso
 - 12: Obtener el centro del kernel shift_kernel_img_space())
 - 13: Usar el kernel estimado para la deconvolución con el método ss_deb
 - 14: Actualizar canal de luminancia de la imagen estimada
 - 15: Convertir de ycbr a formato rgb ycbr2rgb()
fin mientras
-

Algoritmo 2 Ss_deb()

Entrada: imagen, parámetros de la imagen, parámetros del kernel

Salida: Devolver la imagen y el kernel estimado

- 1: Importar valores de la imagen y del kernel
 - 2: Obtener tamaño del kernel y de la imagen
 - 3: Obtener un kernel con respecto a x en formato OTF
 - 4: Obtener un kernel con respecto a y en formato OTF
 - 5: Multiplicación punto a punto del kernel con respecto a x con su conjugada
-

-
-
- 1: Multiplicación punto a punto del kernel con respecto a y con su conjugada
 - 2: Suma de las multiplicaciones anteriores
 - 3: El resultado de la suma lo llevo a dominio de frecuencia
 - 4: **para**($i=1$ hasta iteraciones)
 - 5: Actualizo el valor de sigma
 - 6: Estimo parámetros de deconvolución con prior a la imagen $x_{\text{admn_ubc_bi}}$
 - 7: Estimo parámetros de deconvolución con prior a el kernel $h_{\text{admn_ubc_bi}}$
 - 8: Estima la imagen y el kernel con respecto a los parámetros estimados
- fin**
- fin**
-

Algoritmo 3 Frils_deb_abc()

Entrada: imagen en escala de grises, kernel, parámetros de entrada

Salida: Devolver la imagen recuperada

- 1: Obtener tamaño del kernel
 - 2: Validar que el kernel sea impar
 - 3: Crear filtros de primer y segundo orden
 - 4: Convertir kernel a OTF
 - 5: Convertir el resultado de los filtros a OTF
 - 6: Convolucionar el kernel con su conjugada
 - 7: Convolucionar el kernel con cada uno de los filtros
 - 8: Sumar los resultados de convolucionar el kernel con cada uno de los filtros
 - 9: Importar parámetros de beta, alpha, epsilon, lambda
 - 10: Definir la cantidad de iteraciones
 - 11: Llevar la imagen a dominio de frecuencia
-

-
- 1: Tomar la parte real de la imagen llevada a dominio de frecuencia
 - 2: Iter = 0
 - 3: **mientras** (iter < iteraciones definidas)
 - 4: Estimar la imagen por cada filtro
 - 5: **mientras**(iter= iteraciones definidas)
 - 6: Actualizar el gradiente por cada estimación de la imagen **fin mientras fin mientras**
 - 7: Encontrar coincidencias entre el gradiente de las imágenes estimadas
 - 8: Recuperar la imagen
-

2.2 Fase de Planificación

La metodología de software XP en el desarrollo de aplicaciones, define como fase inicial, la planificación. En 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 es el que define el nivel de prioridad con que se deben implementar las HU, así como el esfuerzo que costará implementar las mismas. Esta fase arroja como resultado un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto de desarrollo. (Beck, Molina & Aguilar 2002).

2.2.1. Historia de usuario

La metodología XP define que entre los artefactos se encuentran las historias de usuario (HU) para especificar las funcionalidades que brindará el sistema. Cada HU representa un requerimiento de software, utilizando el lenguaje común del usuario. Esta fase inicial representa 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. Como parte de la presente investigación, el empleo de este artefacto fue clave, a continuación (2.1) se muestra uno de los ejemplos generados.

Tabla 2.1: Historia de usuario 2 (HU2)

Historia de usuario	
Número: 2.1	Nombre: Transformar a espacio de color
Iteración asignada: 1	
Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 1 semanas
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales: 1 semanas
Descripción: La imagen original esta en formato RGB, lo que presenta tres canales de color, por lo que el algoritmo tendría tres veces el tiempo de ejecución ya que se ejecutaría por cada canal de color de la imagen, por lo que es necesario la transformación a otro espacio de color en el que se pueda obtener la matriz de luminancia de la imagen.	
Observaciones:	

2.2.2. Estimación de esfuerzos por Historias de Usuario

En esta fase se definen las prioridades de cada historia de usuario, y consecuentemente se realiza una estimación del esfuerzo necesario de cada una de ellas. Se definen las entregas con el cliente donde se tiene por norma que una entrega debe adecuarse a tres meses como máximo. Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación del esfuerzo para cada una de las HU, arrojando los siguientes resultados 2.2:

Tabla 2.2: Puntos de estimación por historias de usuario

No.	Historias de usuario (HU)	Puntos de estimación (semanas)
1	Cargar la imagen	1
2	Transformar a espacio de color	1
3	Hallar la magnitud en la superposición de la imagen	2
4	Encontrar el PSF mas optimo par a la imagen.	3
5	Estimar imagen original.	2
6	Mostrar la imagen	1

2.2.3. Plan de iteraciones

Luego de definir las historias de usuarios e identificar el tiempo y el esfuerzo para el desarrollo de cada una de ellas, se propone un plan de iteraciones donde las historias de usuario están contenidas. El plan se propone, regulando el rango establecido para cada una de ellas; teniendo en cuenta todos los aspectos antes tratados, se pretende realizar el desarrollo en 2 iteraciones, descritas a continuación, y recogidas en la tabla 2.3:

Iteración 1 Esta iteración se ubican los requisitos funcionales:

RF1. Cargar la Imagen.

RF3. Mostrar la imagen.

Iteración 2 En esta iteración se encuentran los requisitos funcionales:

RF2. Deconvolucionar imagen.

RF2.1 Transformar a espacio de color.

RF2.2 Hallar la magnitud en la superposición de la imagen.

RF2.3 Encontrar el PSF mas optimo par ala imagen.

RF2.4 Estimar imagen original.

Tabla 2.3: Iteraciones por historias de usuario

Iteraciones	Historias de usuario	Duración
Iteración 1	HU1,HU2,HU3	4
Iteración 2	HU4, HU5, HU6	6

2.2.4. 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.

Tabla 2.4: Plan de entrega

	Iteracion1	Iteracion2
Cantidad de HU	3	3
Fecha de entrega	9/03/2017	1/05/2017

2.3 Diseño

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar. Define que se debe utilizar una correcta especificación de los nombres de métodos y clases que ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código (Beck et al. 2002).

2.3.1. 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 a definir actividades durante el diseño del sistema. Estas tarjetas representan una entidad del sistema, a la cual asignar responsabilidades y colaboraciones. El formato físico de las tarjetas CRC facilita la interacción entre los participantes del proyecto, en sesiones en las que se aplican técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir la de especificación de requisitos, historias de usuarios o casos de uso. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones (Casas & Reinaga 2009). figura 2.2.

2.3.2. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, establezca un estándar de codificación para asegurarse de que se trabaje de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Garcia 2011).

- Utiliza los comentarios para explicar qué hace el código.
- Usa nombres significativos para las clases, métodos, propiedades y variables.
- En los métodos de las clases se define que la primera letra debe ser minúscula, si son varias palabras se debe separar por un guion bajo.
- Se define una declaración por línea.

FBDHSGP	
Descripción: Deconvoluciona la imagen y estima su PSF	
Atributes:	
Nombre	Descripción
Sigma	estimación de ruido (varianza)
Epsilon	valor de ajuste de estimación de PSF
xh_iter	número de iteraciones para estimar imagen
delta	criterio de parada para el PSF
alpha	corrección de valores de estimación
beta_v	peso de penalidad para el valor de gamma
lambda_h	peso de regularización para estimador L1
h_iter	número de iteraciones para estimar PSF en cada iteración de estimación de la imagen
gamma	ajuste de valor gamma de la imagen
kemel	tamaño del la matriz de PSF
Responsabilidades:	
Nombre	Collaborator
FBDHSGP	im2double,rgb2gray,rgb2ycbcr,ycbcr2rgb,mod,sqrt,ceil,
init_kemel	fspecial
init_vars	

Figura 2.2: Tarjeta CRC de la clase FBDHSGP.

- Se evita asignar el mismo valor a varias variables en la misma sentencia, pues dificulta su lectura.

2.4 Conclusiones del capítulo

En este capítulo se propuso el diseño de la solución al problema, describiendo los pasos que la conforman. Se utilizó la metodología XP para realizar el análisis y diseño de la

aplicación, generando como artefactos un total de 6 HU que describen los aspectos principales a tener en cuenta para el desarrollo de la solución. El cliente estableció la prioridad de cada una de las HU que describen las funcionales del sistema, puntualizando el orden de su implementación y las iteraciones en que serán implementadas. También se logra traducir los requerimientos de las funcionalidades a implementar, usando las tarjetas CRC que propone la fase de diseño de la metodología.



RESULTADOS Y VALIDACIÓN

En el siguiente capítulo se estará transitando por las fases de implementación y prueba que propone la metodología XP, generando como artefacto en el caso de la fase de implementación las tareas de desarrollo que dan solución a cada una de las historias de usuarios identificadas en la fase de planificación. También se realizaran un conjunto de pruebas al sistema como las de caja blanca y caja negra. Finalmente se realizara la validación de los resultados del algoritmo basado en RLA que se propuso.

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. Para ello se debe realizar un chequeo de cada HU, en conjunto con el plan de iteraciones y se cambia en caso de ser necesario, por lo cual se crean tareas de desarrollo, para así 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 implementan los métodos donde se convierte la imagen original a escala de grises y se estima el PSF que se utiliza para restaurar la imagen.

Tabla 3.1: Tiempo de implementación de las HU de la primera iteración

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Cargar la imagen	1	1
Transformar a espacio de color	1	1
Hallar la magnitud en la superposición de la imagen	2	2

A continuación se muestran la tarea de ingeniería efectuada para la segunda funcionalidad implementada en esta iteración 3.2:

Tabla 3.2: Tarea de ingeniería de la transformación a espacio de color

Tarea	
Número Tarea: 2	Número HU:2
Nombre: Transformar a espacio de color	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1
Programador responsable: Yanet Pérez Rodríguez	
Descripción: Se obtiene la matriz de luminancia de la imagen	

Iteración 2

En esta iteración se implementan el resto de los requisitos funcionales de la aplicación 3.3.

Tabla 3.3: Tiempo de implementación de las HU de la segunda iteración

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Hallar la magnitud en la superposición de la imagen	2	2
Encontrar el PSF más óptimo para la imagen	3	3
Mostrar la imagen	1	1

A continuación se muestran la tarea de ingeniería efectuada para la primera funcionalidad implementada en la segunda iteración 3.4:

Tabla 3.4: Tarea de ingeniería de la convolución de la imagen con el PSF

Tarea	
Número Tarea: 4	Número HU: 4
Nombre: Encontrar el PSF más óptimo para la imagen	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 3
Programador responsable: Yanet Pérez Rodríguez	
Descripción: Convolucionar la imagen convertida a escala de grises y el PSF.	

3.2 Interfaz del sistema

A continuación se muestra y se describe la interfaz de usuario que permite la interacción con el sistema, figura 3.1.

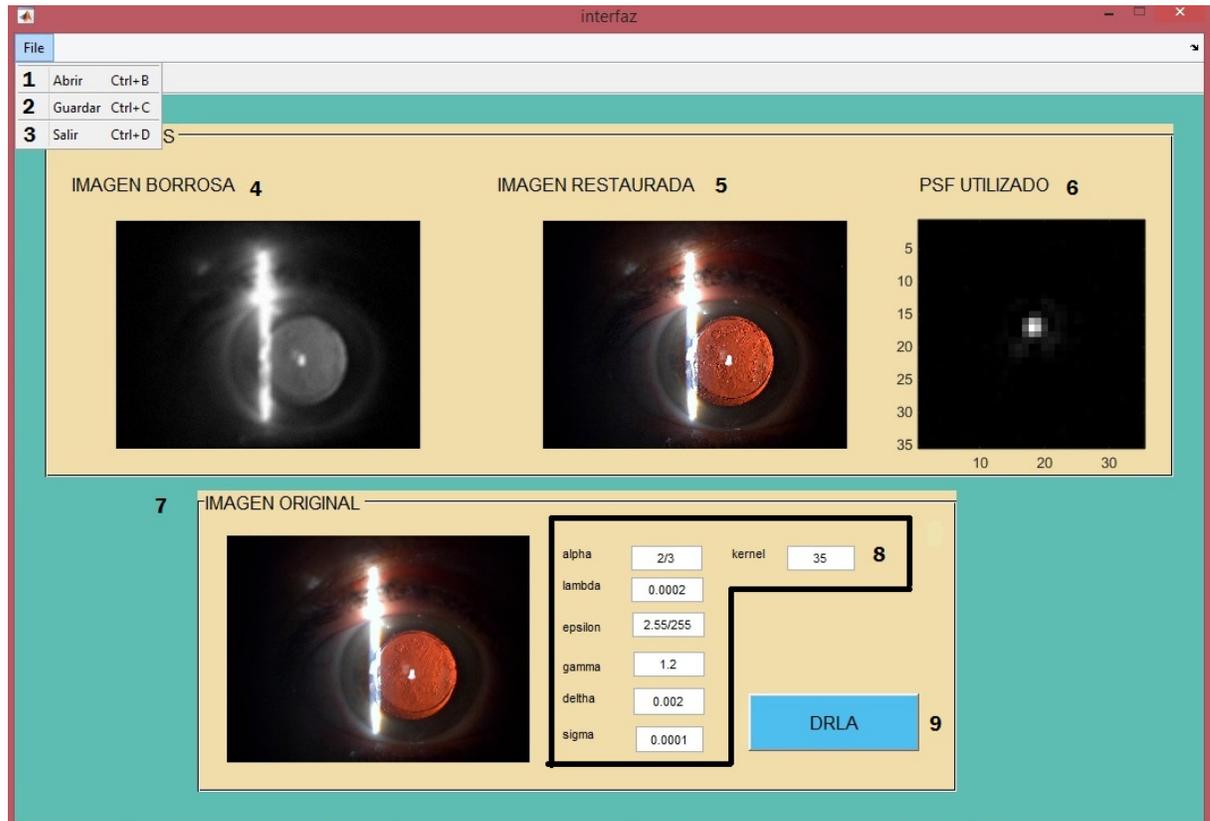


Figura 3.1: Interfaz del sistema

1. Permite cargar la imagen que se quiere restaurar.
2. Guarda la imagen restaurada.
3. Permite salir de la aplicación.
4. Se muestra la imagen borrosa.
5. Se muestra la imagen restaurada.
6. Se muestra el PSF utilizado para la restauración.
7. Se muestra la imagen original

8. Se insertan una serie de parámetros que tienen asignados valores por defecto, pero pueden ser modificados en dependencia de la imagen que se quiere restaurar.
9. Se muestra el botón donde se empieza a ejecutar el algoritmo.

3.3 Resultados de aplicar el algoritmo

El algoritmo fue aplicado a 16 imágenes en retroiluminación provenientes de la lámpara de hendidura. A continuación, en la figura 3.2 se muestran los resultados de aplicar el algoritmo propuesto basado en RLA.

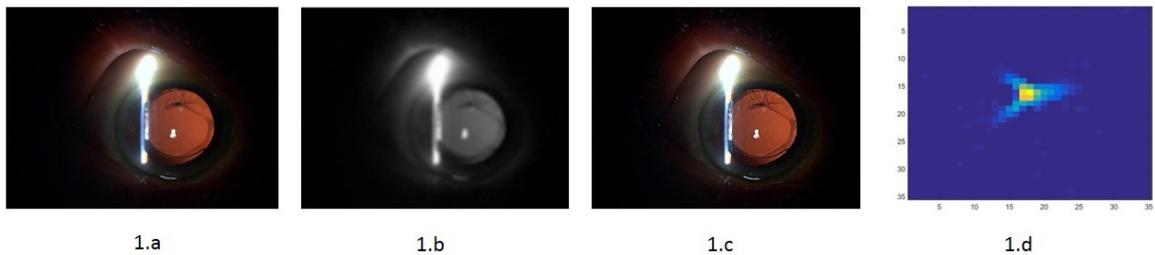


Figura 3.2: 1.a)Imagen original, 1.b)Imagen borrosa, 1.c)Imagen restaurada, 1.d)PSF.

Para ello se tomaron un conjunto de imágenes de prueba con desenfoque y ruido añadidos de manera sintética. Para el desenfoque se usa un kernel gaussiano de tamaño 20×20 y sigma 20, el ruido añadido es un ruido blanco gaussiano con media 0 y varianza 0.001, obteniendo una imagen con alto nivel de desenfoque y que presenta ruido figura 3.2: 1.b. Estas imágenes degradadas son las entradas al algoritmo, las cuales se les estima el PSF 3.2: 1.d y se deconvolucionan como describe el método propuesto en el capítulo 2 3.2: 1.c. A continuación, en la figura 3.3, se presentan los resultados de haber aplicado en 3 imágenes el algoritmo.

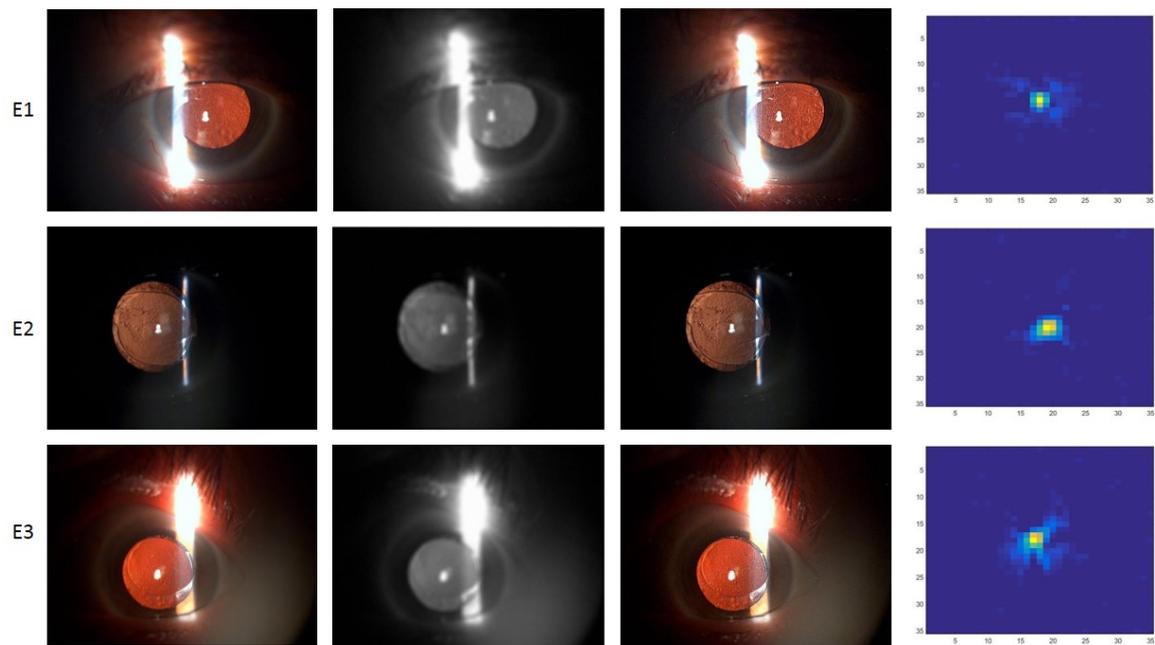


Figura 3.3: Resultados de aplicar el algoritmo basado en RLA

3.3.1. Validación de los resultados

Se seleccionó una población de 16 imágenes en retroiluminación provenientes de la lámpara de hendidura donde se aplicaron las métricas SSIM, SNR y PSNR.

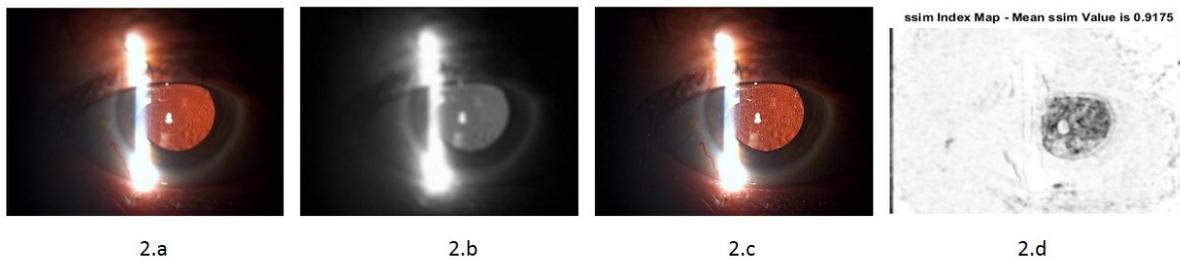


Figura 3.4: 1.a)Imagen original, 1.b)Imagen borrosa, 1.c)Imagen restaurada, 1.d)Mapa SSIM.

Tabla 3.5: Porción de 6 imágenes a las que se aplicó las métricas SSIM.

Imágenes probadas	SSIM imagen restaurada	SSIM imagen borrosa
I1	0.9175	0.5564
I2	0.9340	0.3926
I3	0.9354	0.7086
I4	0.9677	0.7964
I5	0.9396	0.7845
I6	0.8824	0.7725
Media	0.9294	0.6685

El algoritmo propuesto basado en RLA obtuvo resultados favorables sobre una muestra de 16 imágenes, obteniendo una media de $ssim=0,9294$ de la imagen restaurada con respecto a la original y $ssim=0.6685$ de la imagen borrosa con respecto a la original, mientras más cerca este de uno el valor de $ssim$ mas similitud tienen las imágenes, por lo que se puede apreciar que la media de $ssim$ en la imágenes restauradas se acerca bastante a uno, no así con los valores en la imágenes borrosas, lo que significa que las imágenes restauradas con el algoritmo propuesto tienen un alto valor de similitud a las imágenes originales.

Además se obtuvo un promedio de $snr=17,543$ y $psnr=31,1992$ de las imágenes originales con respecto a las restauradas, y de $snr=9,91595$ y $psnr=23,56675$ entre las imágenes originales y las borrosas, por lo que se puede apreciar que el valor de snr en las imágenes restauradas es mayor que en las borrosas, lo que significa que las imágenes restaurada presentan menor valor de ruido que las borrosas. También al comparar la media de $psnr$ de las imágenes restauradas y las borrosas se obtuvo mayor valor en las restauradas lo que significa que estas imágenes presentan mayor calidad que las borrosas, 3.6.

Tabla 3.6: Porción de 6 imágenes a las que se le aplicaron las métricas SNR y PSNR.

Imágenes probadas	SNR y PSNR imagen res- taurada	SNR y PSNR imagen bo- rrosa
I1	20.4900 y 30.369	12.3880 y 22.2971
I2	22.7497 y 33.3010	9.7145 y 20.2658
I3	17.9573 y 32.1901	8.8107 y 23.0435
I4	22.2154 y 36.1528	12.4427 y 26.3801
I5	13.8628 y 31.8027	6.2216 y 24.1615
I6	7.9848 y 23.3191	9.9182 y 25.2525
Media	17.5430 y 31.1992	9,91595 y 23,56675

3.4 Pruebas

XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones (Sommerville 2005). Entre estas están las pruebas de caja blanca y caja negra.

3.4.1. Pruebas de caja blanca

El método prueba de caja blanca se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones, bucles o ambos (Pressman 2005).

```
function [h,alpha,beta] = h_update(d,b,tol,max_iter,alpha)
mb = -b;
if exist('alpha','var')
    alpha = max(alpha,0);
else
    alpha = max(mb-0,0);
end
sd = sum(sum(1./d));
cost_display=0;
for iter = 1:max_iter
    temp = (b+alpha)./d;
    beta = (1-sum(temp(:)))/sd;
    alpha = max(mb-beta,0);
    if cost_display
        h = temp + beta./d;
        h = max(h,0);
        h = h/sum(sum(h));
        f(iter) = 0.5*sum(sum(d.*h.*h))-sum(sum(b.*h));
        fprintf('iter=%d,cost=%f\n',iter,f(iter));
    end
    if iter > 1
        rv_beta = abs(beta_old-beta);
        if rv_beta<tol
            break;
        end
    end
    beta_old = beta;

end
h = temp + beta./d;
end
```

Figura 3.5: Código de la función hupdate

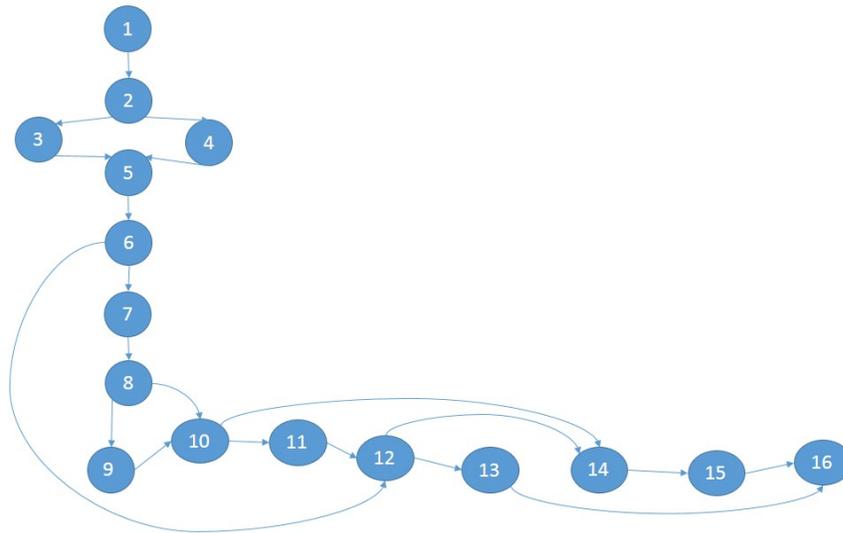


Figura 3.6: Representación del grafo de flujo de camino básico de hupdate

A partir del código fuente se diseñó el grafo de flujo de camino básico como se muestra en la figura 3.6 , luego se procede a calcular la complejidad ciclomática $V(G)$, la cual define el número de caminos independientes del conjunto básico de un programa, dando un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

$$V(G) = Aristas(A) - Nodos(N) + 2 \quad (32)$$

$$V(G) = 20 - 16 + 2$$

$$V(G) = 6$$

$$V(G) = 1 + nodospredicados \quad (33)$$

$$V(G) = 1 + 5$$

$$V(G) = 6$$

$$V(G) = r \quad (34)$$

$$V(G) = 6$$

Donde r es el número de regiones y $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, los caminos que se definen son los siguientes:

C1:1-2-3-5-6-7-8-10-14-15-16

C2:1-2-3-5-6-7-8-9-10-11-12-13-16

C3:1-2-4-5-6-12-14-15-16

C4:1-2-4-5-6-7-8-10-14-15-16

C5:1-2-3-5-6-7-8-10-11-12-14-15-16

C6:1-2-4-5-6-7-8-10-11-12-14-15-16

Los casos de pruebas se definen a partir de los caminos básicos detectados, de forma que los datos señalados causen que se visiten las sentencias vinculadas a cada nodo del camino. Para este método se calcularon seis caminos básicos, por lo que se debe hacer igual número de casos de prueba. A continuación se muestra uno de los casos de pruebas realizados.

Tabla 3.7: Caso de prueba para hupdate

Entradas	kernel estimado en función de la variable de penalización beta, kernel con PSF centrado en el origen, constante de regularización, número de iteraciones parámetro de estimación de super gaussian prior alpha.
Resultados Esperados	Modificación del kernel estimado y de alpha.
Condiciones	Deba haberse cargado la imagen y debe existir un kernel para la deconvolución

Después de aplicar el mismo procedimiento a todas las funcionalidades de la aplicación no se detectó ninguna inconformidad en las mismas.

3.4.2. Pruebas de caja negra

Estas pruebas intentan encontrar casos de prueba en que el módulo no se atiene a su especificación. Se enfocan especialmente en los módulos que se relacionan con la interfaz de usuario ya que no requieren el conocimiento de la estructura interna del programa para su puesta en marcha (Pressman 2005). A continuación se presenta el diseño de caso de prueba de la HU1 a la cual se le realizaron pruebas.

Tabla 3.8: Caso de prueba de aceptación de la HU1

Historia de usuario: 1
Nombre: Cargar Imagen
Descripción: Se carga la imagen que el usuario desea restaurar
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución:
<ul style="list-style-type: none"> ▪ El usuario selecciona la imagen en formato jpg, bmp o png. ▪ El usuario carga la imagen en el formato seleccionado. ▪ El sistema muestra la imagen en pantalla y su imagen borrosa correspondiente.
Resultado Esperado: Se muestra la imagen en pantalla y su imagen borrosa correspondiente.
Evaluación de la Prueba: Prueba satisfactoria

Por cada HU se realizó una prueba de aceptación. Fueron detectadas un total de 9 no conformidades en las 2 iteraciones realizadas, las cuales ya han sido solucionadas.

3.5 Conclusiones del capítulo

- Los resultados obtenidos aplicando las métricas ssim, snr y psnr demuestran que la imagen restaurada tiene alta similitud respecto a la imagen original.
- Las aplicación de pruebas de caja blanca y pruebas de caja negra demuestran la calidad de la implementación realizada para dar solución al problema.



CONCLUSIONES

- Se realizó un estudio de diferentes métodos de deconvolución a ciegas el cual tuvo como resultado el uso del método Richardson-Lucy en el algoritmo propuesto para la restauración de las imágenes en retroiluminación provenientes de la lámpara de hendidura.
- El algoritmo propuesto basado en RLA obtuvo resultados favorables sobre una muestra de 16 imágenes, obteniendo una media de $ssim = 0,9294$, lo que significa que las imágenes restauradas con el algoritmo propuesto tienen un alto valor de similitud con las imágenes originales.
- Se obtuvo un promedio de $snr = 17,543$ y $psnr = 31,1992$ de la imagen original con respecto a la restaurada, además de $snr = 9,91595$ y $psnr = 23,56675$ entre la imagen original y la borrosa, por lo que se puede afirmar que el valor de ruido en la imagen restaurada es menor que en la imagen borrosa.

REFERENCIAS BIBLIOGRÁFICAS

().

Alvarez, M, R Rodríguez & I Hernández. *Diseño de un sistema basado en casos para la identificación de opacidad mediante el pentacam.* UCIENCIA, La Habana (2014).

Anacona Mosquera, Oscar Eduardo. *Implementação do algoritmo de richardson-lucy em arquiteturas reconfiguráveis aplicado ao problema de borrado de imagens* (2016).

Andrews, Harry C & Bobby Ray Hunt. *Digital image restoration* (1977).

Archila, John F et al. *Evaluación de técnicas usadas en la restauración de imágenes afectadas por desenfoque (blur).* Scientia et technica, tomo 2(51):145–154 (2012).

Beck, Kent, Jesús García Molina & Luis Joyanes Aguilar. Una explicación de la programación extrema: aceptar el cambio. Addison Wesley (2002).

Campisi, Patrizio & Karen Egiazarian. *Blind image deconvolution: theory and applications.* CRC press (2016).

Canós, José H, Patricio Letelier & M^a Carmen Penadés. *Metodologías ágiles en el desarrollo de software.* Metodologías Ágiles en el Desarrollo de Software, tomo 1(10):1–8 (2003).

Casas, Sandra I & Héctor H Reinaga. *Aspectos tempranos: un enfoque basado en tarjetas crc.* Avances en Sistemas e Informática, tomo 6(1):85–92 (2009).

CRESPO, Maelis Hernández; GALIANO Vivian Solange Céspedes., Yamaris Rosabal; SORIA. *Capsulotomía posterior en pacientes operados de catarata.* Revista Electrónica Dr. Zoilo E. Marinello Vidaurreta, tomo 40 (2015).

Cristina, Federico, Oscar N Bria & Claudia Cecilia Russo. *Generación de imágenes de alta resolución utilizando secuencias de video de baja resolución.* En X Congreso Argentino de Ciencias de la Computación (2004).

DE DESARROLLO, GERENCIA DE PROYECTOS. *Proyecto de implementación del tamizaje ocular pediátrico para costa rica* (2007).

Faccia, Paula A. *Técnicas de iluminación.* Exploración con biomicroscopio ocular Técnicas y protocolo de intervención, pág. 38 (2014).

Fahmy, Mamdouh F, Gamal M Abdel Raheem, Usama S Mohamed, Omar F Fahmy et al. *A new fast iterative blind deconvolution algorithm.* Journal of Signal and Information Processing, tomo 3(01):98 (2012).

Frijo, Matteo & Steven G Johnson. FFTW: An adaptive software architecture for the FFT, tomo 3 (1998).

Galatsanos, Segall C. Andrew y Katsaggelos Aggelos K., Nikolas P. *Digital image enhancement.* Marcel Dekker, Inc, 2003, Encyclopedia of Optical Engineering. DOI: 10.1081/E-EOE 120009510. (2003).

Garcia, O. *Nomenclatura estándar del código java* (2011).
URL <http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/>.

Hansen, Per Christian, James G Nagy & Dianne P O'leary. Deblurring images: matrices, spectra, and filtering. SIAM (2006).

Hernández López, Iván, Juan Raúl Hernández Silva, Yadira Castro González, Ailén Garcés Fernández, Zucell Veitía Rovirosa & Eneida Pérez Candelaria. *Estrategias de prevención de la opacidad de la cápsula posterior.* Revista Cubana de Oftalmología, tomo 23:608–623 (2010).

Hernández Silva, Juan Raúl, Meisy Ramos López, Luis Curbelo Cunill, Gilberto Fernández Vásquez, Marcelino Rio Torres & Yanele Ruiz Rodríguez. *Astigmatismo posquirúrgico en la facoemulsificación según el lugar de la incisión.* Revista Cubana de Oftalmología, tomo 25(1):2–11 (2012).

Hernández López, Eymar. *Un acercamiento a la deconvolución a ciegas usando el algoritmo de lucy-richardson.* (2010).

Kanski, Jack J. *Oftalmología clínica, 5ª edición.* Informe técnico, ISBN 978-84-8174-758-4 ().

Keel, William C. *A simple, photometrically accurate algorithm for deconvolution of optical images.* Publications of the Astronomical Society of the Pacific, tomo 103(665):723 (1991).

Kundur, Deepa & Dimitrios Hatzinakos. *Blind image deconvolution revisited.* IEEE signal processing magazine, tomo 13(6):61–63 (1996).

Lombardi, G. *Astronomical site testing in the era of extremely large telescopes: evaluating the image quality degradation induced by the earth atmosphere.* Boletín de la Asociación Argentina de Astronomía La Plata Argentina, tomo 53:325–336 (2010).

Masache Narvaez, Jose Luis & Ernesto Javier Roldan Monge. Implementación de un algoritmo para el cálculo estimado del movimiento lineal en imágenes y su aplicación en la restauración de imágenes movidas. B.S. thesis (2010).

MathWorks ().

———. *Mathworks* (2017). URL <https://de.mathworks.com/help/>.

- Nigam, Kamal, John Lafferty & Andrew McCallum.** *Using maximum entropy for text classification.* En IJCAI-99 workshop on machine learning for information filtering, tomo 1, págs. 61–67 (1999).
- Paradigm, Visual.** *Paradigm, visual* (2016). URL <https://www.visual-paradigm.com>.
- Penfield Jr, Paul.** *Principle of maximum entropy.* Information, Entropy and Computation (2010).
- Pérez Candelaria, Eneida de la C, Yanileidy González Blanco, Liamet Fernández Argones, Belkys Rodríguez Suárez, Alina Pedroso Llanes & Ana M Méndez Duque de Estrada.** *Correlación entre la localización del lente intraocular, según biomicroscopia ultrasónica, y diferentes parámetros quirúrgicos en pacientes operados de catarata.* Revista Cubana de Oftalmología, tomo 23:780–493 (2010).
- Pressman, Roger S.** *Software engineering: a practitioner's approach.* Palgrave Macmillan (2005).
- Ratnaparkhi, Adwait.** *A simple introduction to maximum entropy models for natural language processing.* IRCS Technical Reports Series, pág. 81 (1997).
- Salorio, Demetrio Pita.** *Diccionario terminológico de oftalmología.* Informe técnico ().
- Sanchez, María A Mendoza.** *Metodologías de desarrollo de software.* María A. Mendoza Sánchez (2004).
- Santander, Nuria Alonso.** *Sistema de ayuda a la decisión para cirugía de cataratas.* Universidad Nacional de Educación a Distancia. ETSI Informática. Departamento de Inteligencia Artificial.[en línea]:<<http://www.cisiad.uned.es/tesis/tesis-Nuria-Alonso.pdf>> (2011).
- Sommerville, Ian.** *Ingeniería del software.* Pearson Educación (2005).

Wang, Zhou, Alan C Bovik, Hamid R Sheikh & Eero P Simoncelli. *Image quality assessment: from error visibility to structural similarity.* IEEE transactions on image processing, tomo 13(4):600–612 (2004).



Anexos

Tabla 5.1: Historia de usuario 1 (HU1)

Historia de usuario

Número: 1	Nombre: Cargar la imagen
-----------	--------------------------

Iteración asignada: 1

Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 1 semanas
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales: 1 semanas

Descripción: Se carga la imagen que el usuario desea restaurar

Observaciones:

Tabla 5.2: Historia de usuario 3 (HU3)

Historia de usuario

Número: 3	Nombre: Hallar la magnitud en la superposición de la imagen.
-----------	--

Iteración asignada: 1

Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales: 1 semanas

Descripción: Se realiza la convolución de la imagen convertida en escala de grises con la función PSF. La convolución es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g .

Observaciones:

Tabla 5.3: Historia de usuario 4 (HU4)

Historia de usuario	
Número: 4	Nombre: Encontrar el PSF mas optimo para la imagen.
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alta (Alto / Medio / Bajo)	Puntos reales:3 semanas
Descripción: Se modifica la representación matricial de los pixeles de la imagen anterior, proceso que se repite iterativamente hasta que se alcanza el máximo en la función optimalidad de estimación de PSF.	
Observaciones:	

Tabla 5.4: Historia de usuario 5 (HU5)

Historia de usuario	
Número: 5	Nombre: Estimar imagen original.
Iteración asignada: 2	
Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales:2 semanas
Descripción: Se realiza una estimacion de la imagen con el PSF estimado previamente	
Observaciones:	

Tabla 5.5: Historia de usuario 6 (HU6)

Historia de usuario	
Número: 6	Nombre: Devolver la imagen restaurada
Iteración asignada: 2	
Prioridad en negocio: Media (Alta / Media / Baja)	Puntos estimados: 1 semanas
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales: 1 semanas
Descripción: Se muestra la imagen borrosa, el PSF estimado, y la imagen restaurada.	
Observaciones:	