

**Universidad Autónoma Metropolitana**  
**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería  
Ingeniería en Computación

---

---

**Manual de Usuario**

Sistema Computarizado Auxiliar en el Diagnóstico de  
Enfermedades mediante el Método de la Iridología  
(SCADE-Iris)



**UAM – AZCAPOTZALCO | MARIANA BELTRÁN ACOSTA**

**Asesor | DR. ARMANDO JIMÉNEZ FLORES**

## TABLA DE CONTENIDO

<b>TABLA DE CONTENIDO.....</b>	<b>- 2 -</b>
<b>REQUERIMIENTOS PARA SU EJECUCIÓN .....</b>	<b>- 3 -</b>
Hardware .....	- 3 -
Software .....	- 3 -
Proyecto .....	- 3 -
<b>PRIMERA VISTA DEL SIMULADOR .....</b>	<b>- 4 -</b>
Interfaz Gráfica .....	- 4 -
Área de Visualización de imagen.....	- 5 -
Área de Grafica .....	- 5 -
Panel de Control.....	- 6 -
<i>Imagen</i> .....	- 6 -
<i>Análisis</i> .....	- 7 -
<i>Reporte</i> .....	- 8 -
<b>EJECUTAR EL SIMULADOR.....</b>	<b>- 9 -</b>
<b>REALIZAR UNA SIMULACIÓN .....</b>	<b>- 9 -</b>
Seleccionar imagen .....	- 9 -
<i>Búsqueda</i> .....	- 10 -
<i>Captura</i> .....	- 10 -
Delimitar zona.....	- 12 -
<i>Mover</i> .....	- 13 -
<i>Redimensionar</i> .....	- 13 -
Generar Mapa Iridológico .....	- 14 -
Analizar Imagen.....	- 14 -
Guardar datos del paciente.....	- 16 -
Generar reporte .....	- 17 -

## REQUERIMIENTOS PARA SU EJECUCIÓN

Para ejecutar el simulador se necesita una computadora con las siguientes características:

### Hardware

Lector de CD  
256 MB en memoria RAM como mínimo  
Procesador Pentium III o superior

### Software

SO Windows XP o superior  
Microsoft .NET Framework 4

### Proyecto

Se debe tener la carpeta **SCADE** que contiene el proyecto para llevar a cabo el **Análisis Iridológico**.  
La ruta en la que se debe colocar dicha carpeta es: **C:\**

## PRIMERA VISTA DEL SIMULADOR

El **Simulador** es una **Interfaz Gráfica** que tiene como finalidad permitir que el usuario pueda utilizarla de manera intuitiva desde el inicio y pueda llevar a cabo la ejecución de un análisis **Iridológico** y visualizar la gráfica de las diez zonas mas afectadas de la persona en estudio.

### Interfaz Gráfica

La **Interfaz Gráfica** del sistema SCADE-Iris (ver *Figura 1*), cuenta con **tres** secciones principales, estas son:

1. Área de Visualización de imagen
2. Área de Gráfica
3. Panel de control

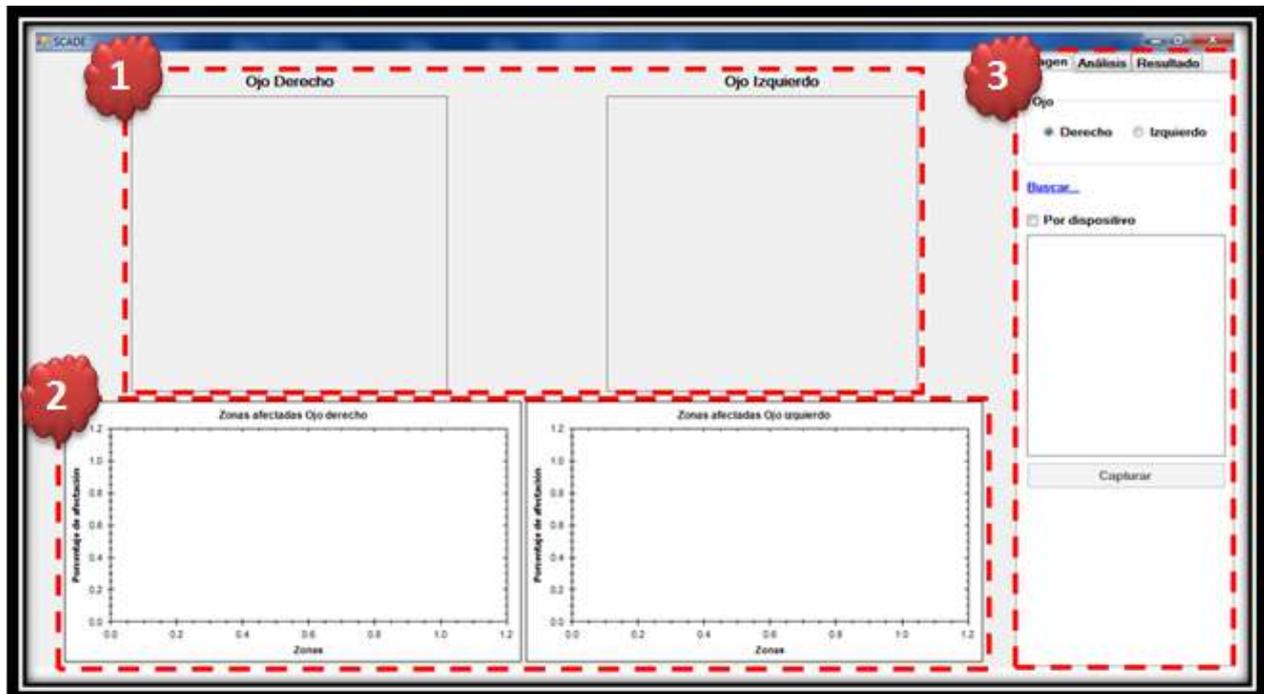


Figura 1. Interfaz Gráfica.

## Área de Visualización de imagen

Esta área es donde se visualiza la imagen de cada uno de los ojos (ver *Figura 2*), tanto **derecho** como **izquierdo**. Se visualizan las imágenes dependiendo el modo de carga, es decir, ya sea por **búsqueda** de un archivo previamente almacenado o por **captura** de imagen mediante el uso de una *webcam*.



Figura 2. Área de Visualización de imagen.

## Área de Grafica

En ella se muestra la representación gráfica de las **diez** zonas más afectadas en cada uno de los ojos, tanto izquierdo como derecho (ver *Figura 3*). Para poder visualizar el resultado en esta área, es necesario ejecutar un análisis de las imágenes previamente cargadas en el **Área de visualización de imagen**.

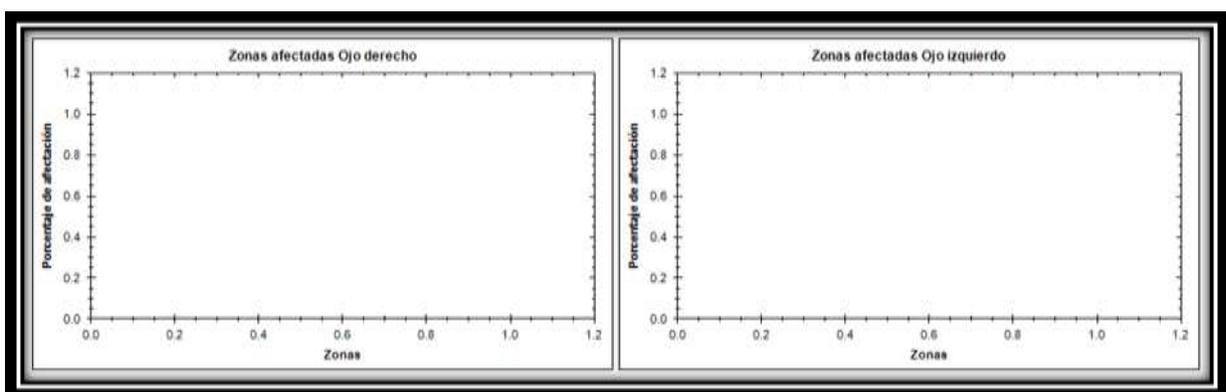


Figura 3. Área de Gráfica.

## Panel de Control

Esta área permite realizar las acciones correspondientes a un análisis **Iridológico**. El **Panel de Control** está dividido en **tres** pestañas:

### *Imagen*

Esta pestaña permite cargar la imagen en cada área destinada para el ojo derecho y el ojo izquierdo (ver *Figura 4*), para ello se deberá seleccionar el tipo de ojo a cargar (**Derecho** o **Izquierdo**). La carga de imagen se puede hacer mediante el vínculo **Buscar...** o seleccionando la casilla de la opción **Por dispositivo**. En caso de seleccionar la opción **Por dispositivo**, el botón **Capturar** servirá para realizar la toma de la imagen desde la *webcam*.



**Figura 4.** Pestaña Imagen.

## Análisis

Esta pestaña está dividida en tres secciones (ver *Figura 5*):

- **Delimitar zona:** Permite cargar la **plantilla Iridológica** haciendo *clic* en el botón **Delimitar** y posteriormente haciendo *clic* en alguna de las imágenes cargadas en el **Área de visualización de imagen**. Se pueden realizar dos acciones sobre la plantilla Iridológica:
  - **Mover:** Al seleccionar esta acción, se podrán mover cada uno de los tres círculos pintados en la **plantilla Iridológica**, esto para ambos ojos. El movimiento es de manera individual en cada círculo.
  - **Redimensionar:** Esta opción permite cambiar el tamaño de cualquiera de los tres círculos que componen la **plantilla Iridológica**.
- **Limpiar zona:** Realiza el borrado de la imagen del **Área de visualización de imagen**, para ello debe seleccionarse que cuadro de esta área será borrado, ya sea el del ojo **izquierdo** o el ojo **derecho**.
- **Análisis:** Contiene dos botones, el primero **Generar Mapa**, permite generar el **Mapa Iridológico** sobre la plantilla previamente ajustada. El pintado de dicho mapa se realiza en ambos ojos en una sola ejecución, el segundo botón, **Analizar Imagen**, lleva a cabo el proceso de **análisis Iridológico** sobre ambos ojos.

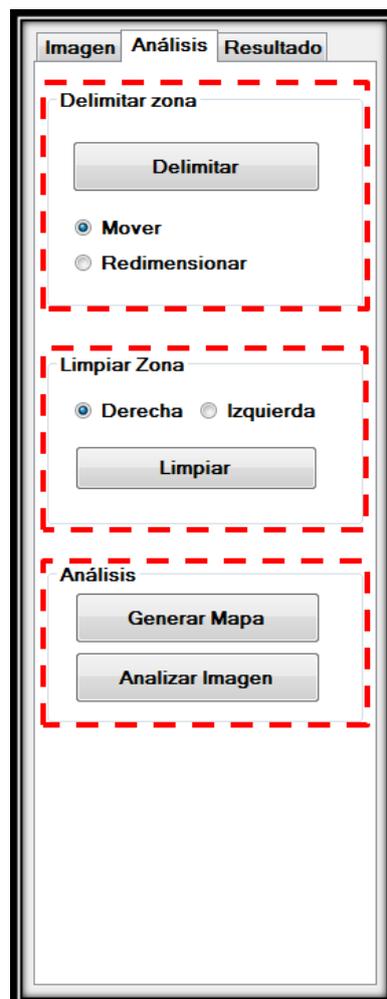


Figura 5. Pestaña: Análisis

## Reporte

En esta pestaña (ver *Figura 6*) se ingresan los datos para registrar el usuario en la **Base de Datos**. Adicionalmente existe el botón **Generar Reporte** que crea el reporte del análisis **Iridológico** en curso.

The screenshot shows a web interface with three tabs: 'Imagen', 'Análisis', and 'Resultado'. The 'Análisis' tab is active. Below the tabs is a form titled 'Paciente' with the following fields:

- Nombre \*
- Apellido Paterno \*
- Apellido Materno
- Edad \*
- Sexo \*
  - Masculino
  - Femenino
- Domicilio
- Email
- Teléfono ( ) \_\_\_\_\_

Below the 'Paciente' section is a 'Reporte' section containing a 'Generar Reporte' button.

Figura 6. Pestaña: Reporte

## EJECUTAR EL SIMULADOR

Para ejecutar el **SCADE-Iris** se debe ingresar al **CD** y copiar la carpeta **SCADE** en la ruta **C:\**.

Ingresar a la carpeta **C:\SCADE\Simulador** y dar doble *clic* en el archivo ejecutable “**Scade.exe**”, aparecerá la interfaz del simulador.

## REALIZAR UNA SIMULACIÓN

Para crear una simulación de deben de seguir los siguientes pasos:

1. Seccionar Imagen
  - a. Búsqueda
  - b. Captura
2. Delimitar zona
  - a. Mover
  - b. Redimensionar
3. Generar Mapa Iridológico
4. Analizar imagen
5. Guardar datos del paciente.
6. Generar reporte.

### Seleccionar imagen

Primeramente se debe elegir el tipo de ojo a cargar, seleccionando la pestaña **Imagen** (ver *Figura 7*).

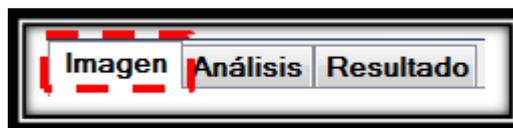


Figura 7. Seleccionar pestaña Imagen

En este ejemplo se comienza por seleccionar el **ojo derecho** (ver *Figura 8*).



Figura 8. Selección del ojo derecho.

## Búsqueda

Seleccionar la opción **Buscar...** (ver *Figura 9*) para elegir un archivo desde una ruta específica:



Figura 9. Botón Buscar

Se abrirá una **ventana de diálogo** (ver *Figura 10*) para poder seleccionar un archivo **.bmp** desde una ruta específica y poder verlo en el **Área de visualización de imagen**.

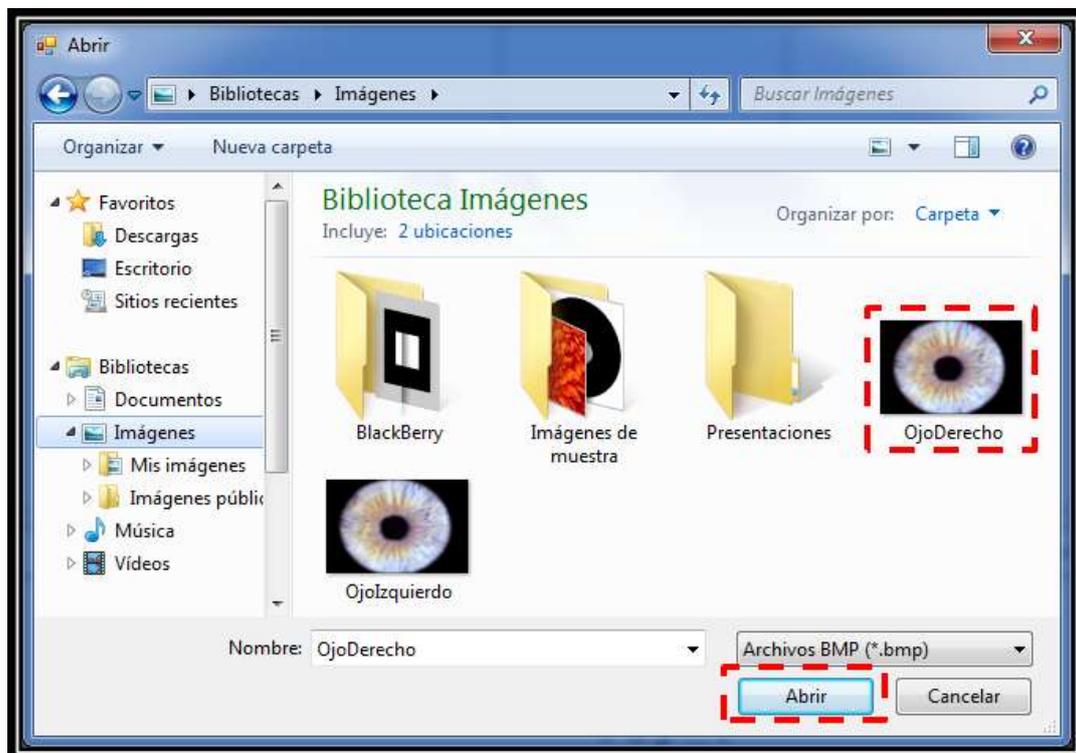


Figura 10. Ventana de Abrir archivo

## Captura

Seleccionar la opción **Por dispositivo** (ver *Figura 11*) para activar la *webcam*:



Figura 11. Botón Por dispositivo

Una vez activada la *webcam*, se debe hacer clic en **Capturar** (ver *Figura 12*) para realizar la “toma” de dicha imagen.



Figura 12. Capturar imagen

Continuando con el ejemplo, debe hacerse el mismo procedimiento de captura de imagen para el **ojo izquierdo** (ver *Figura 13*).



Figura 13. Selección del ojo izquierdo

Finalmente se debe tener el **Área de Visualización imagen** (ver *Figura 14*) con ambas imágenes cargadas:

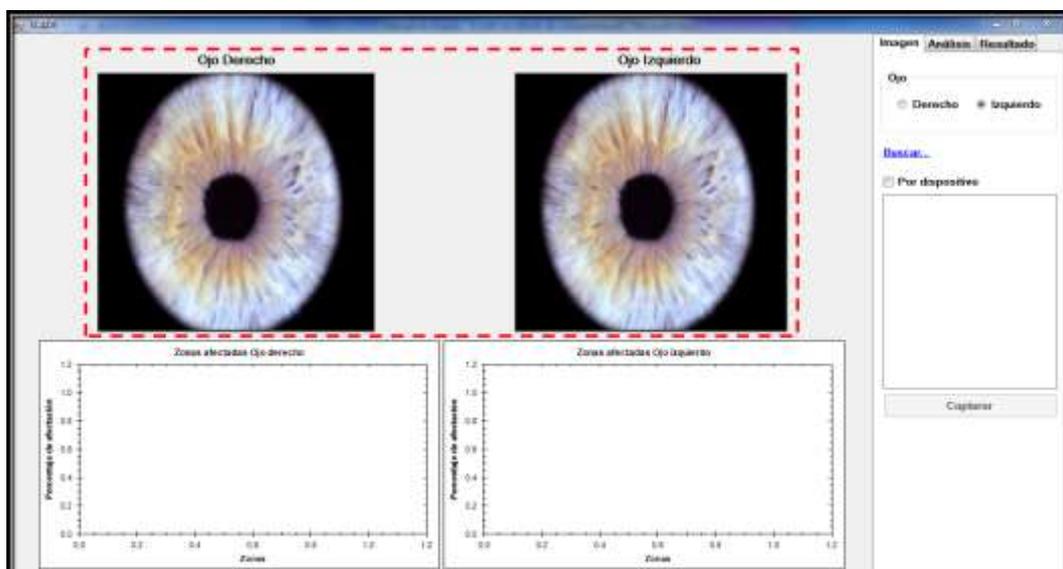


Figura 14. Área de visualización de imagen

## Delimitar zona

Para comenzar a trabajar en la delimitación de la zona, se debe elegir la pestaña **Análisis** (ver *Figura 15*).



Figura 15. Seleccionar pestaña Análisis

Una vez posicionados en dicha pestaña, deberá hacerse *clic* en el botón **Delimitar** (ver *Figura 16*).



Figura 16. Botón Delimitar

Hacer *clic* en el centro del iris de la imagen del **ojo derecho** y posteriormente en la imagen del **ojo izquierdo** (ver *Figura 17*).

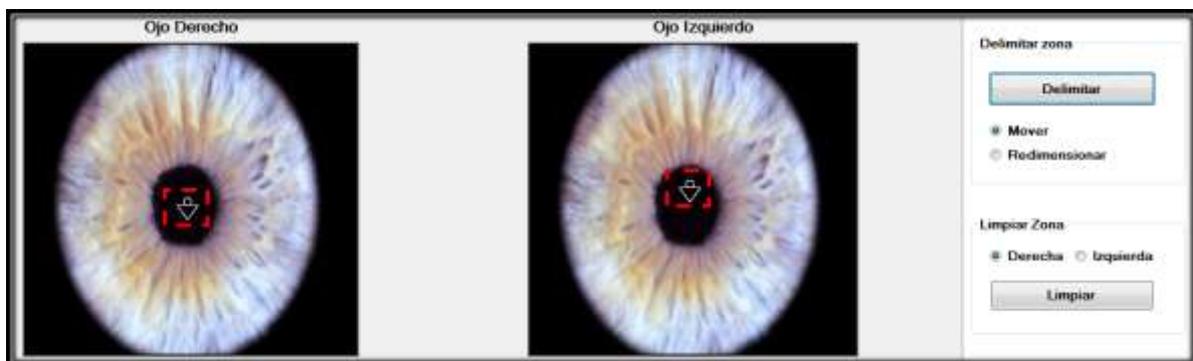


Figura 17. Delimitando zonas

Una vez echo clic en ambas imágenes, deberán visualizarse las **dos plantillas** (ver *Figura 18*).

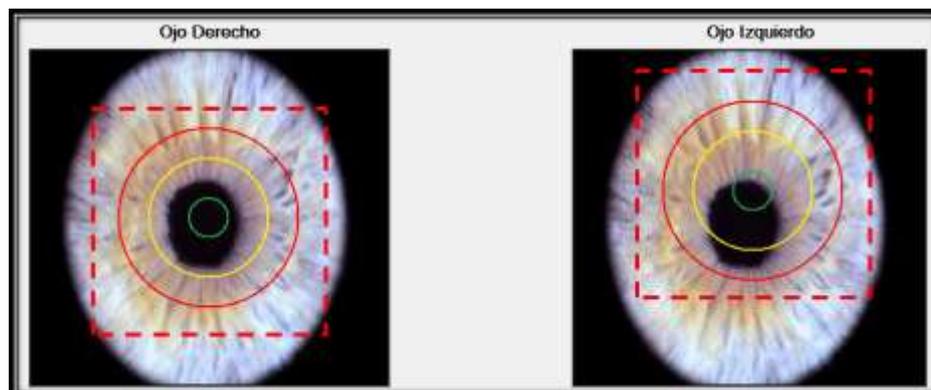


Figura 18. Visualización de plantillas en ambos ojos

### *Mover*

Antes de generar el **Mapa Iridológico**, se deben mover los círculos para después ajustarlos a la imagen (ver *Figura 19*), tanto del **ojo derecho** como del **ojo izquierdo**. Para activar el movimiento sobre cada ojo es necesario seleccionarlo (ojo derecho u ojo izquierdo) y posteriormente elegir la opción **Mover**, localizada en la sección **Delimitar zona**.



Figura 19. Opción Mover

### *Redimensionar*

Como paso final previo a la generación del mapa, es necesario cambiar el tamaño de cada círculo para ajustarlo a la imagen (ver *Figura 20*). Para activar el redimensionamiento sobre cada ojo es necesario seleccionarlo (**ojo derecho** u **ojo izquierdo**) y posteriormente elegir la opción **Redimensionar**, localizada en la sección **Delimitar zona**.



Figura 20. Opción Redimensionar

A continuación, en la *Figura 21* se muestra el ajuste de las plantillas

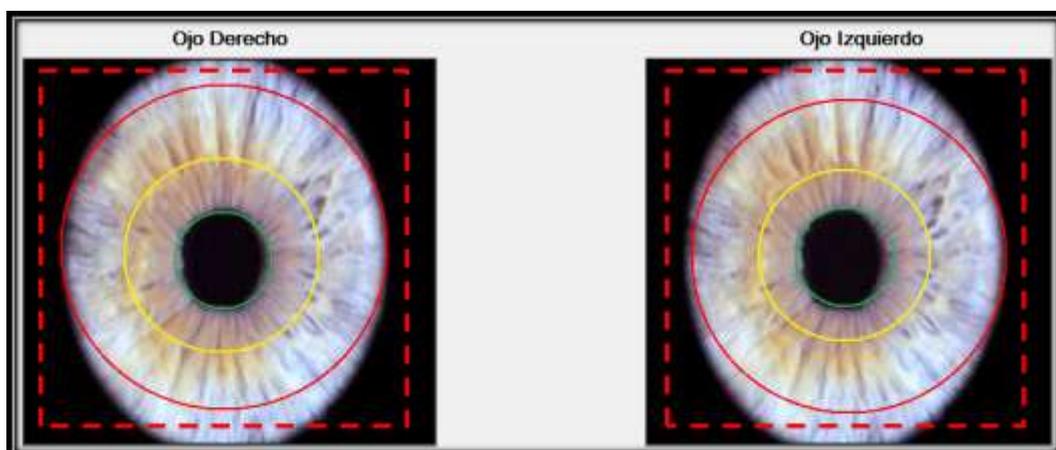


Figura 21. Visualización ajustada de plantillas

## Generar Mapa Iridológico

Para generar los mapas en cada uno de los ojos, basta con hacer clic en el botón **Generar Mapa** (ver *Figura 22*).

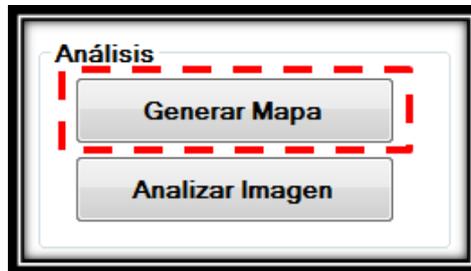


Figura 22. Botón Generar Mapa

Los mapas serán generados en ambos ojos (ver *Figura 23*).

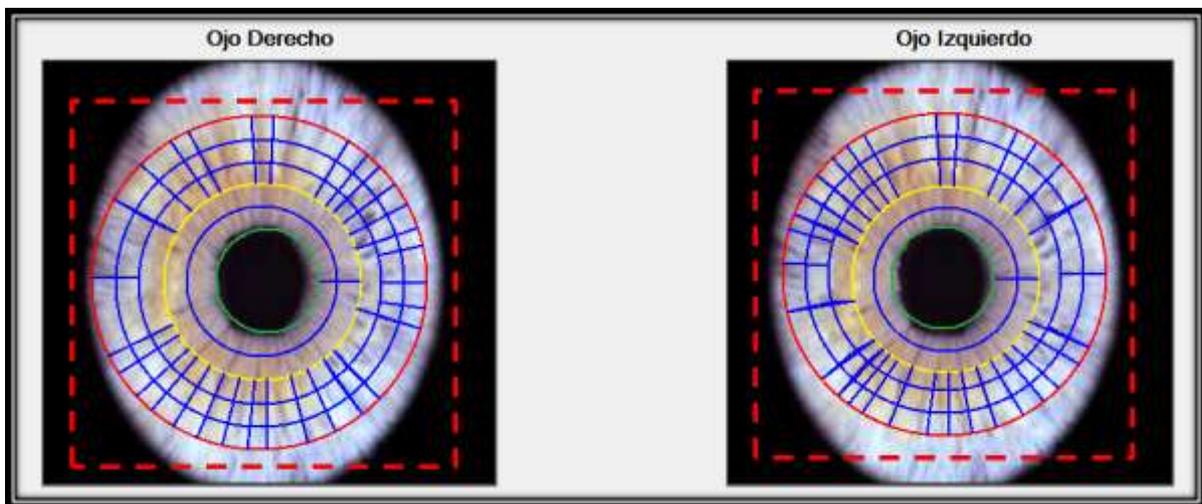


Figura 23. Visualización de Mapa Iridológico en ambos ojos

## Analizar Imagen

Este paso es el más importante, ya que es aquí donde se genera todo el procesamiento **Iridológico**. Para iniciar el análisis se debe hacer clic en el botón **Analizar Imagen** (ver *Figura 24*).

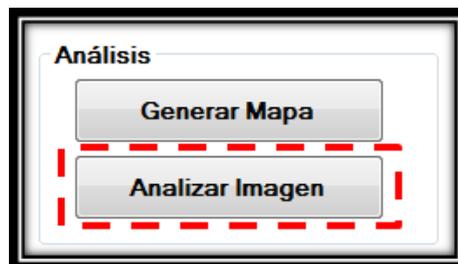


Figura 24. Botón Analizar Imagen

Durante el análisis se mostrará un ícono de progreso (ver *Figura 25*), la cual indica que el análisis está en curso.



Figura 25. Ícono de Progreso

Finalmente en el **Área de Grafica** se observa el resultado de las **10** zonas más afectadas en cada uno de los ojos (ver *Figura 26*).

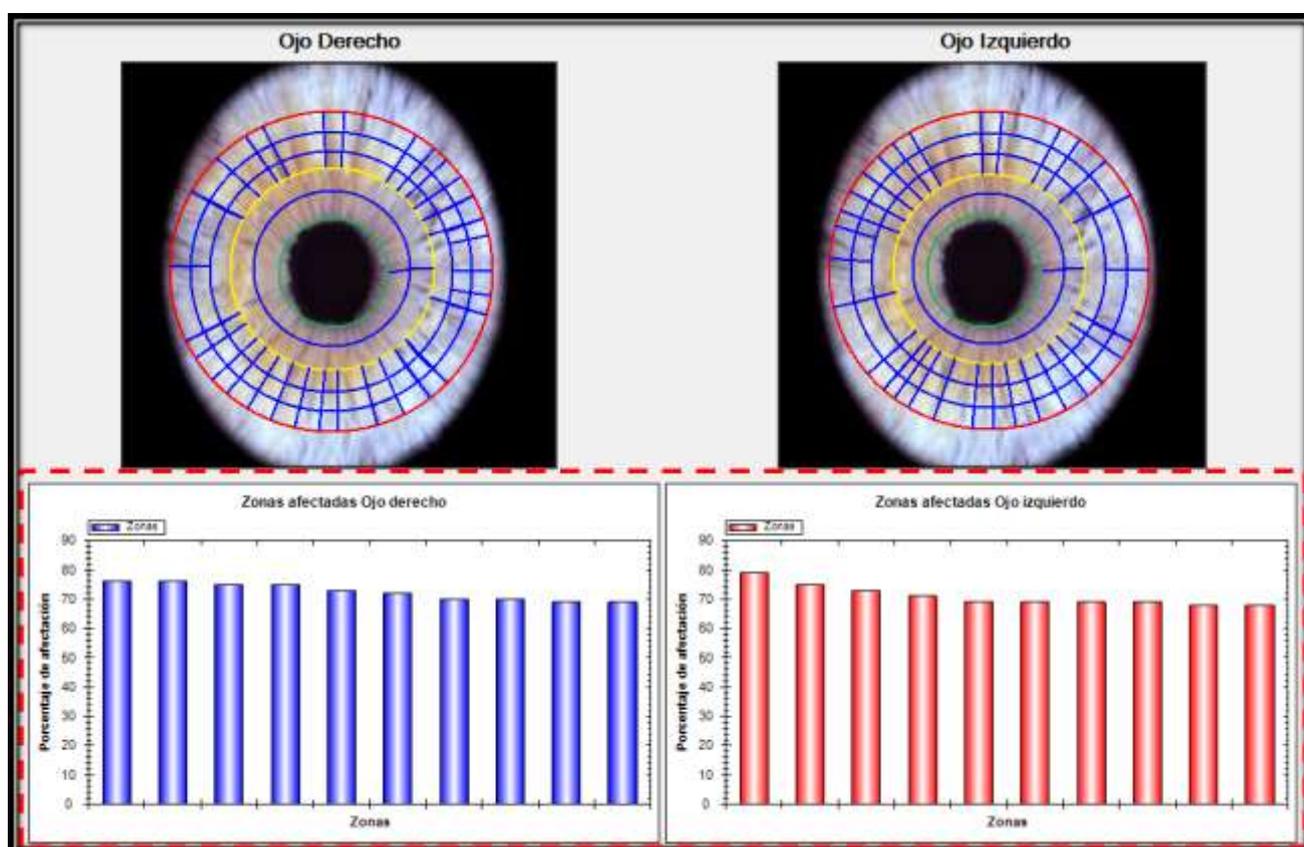


Figura 26. Visualización de las 10 zonas más afectadas en ambos ojos

## Guardar datos del paciente

Para almacenar la información del paciente se debe elegir la pestaña **Resultado** (ver *Figura 27*).



Figura 27. Selección de la pestaña Resultado

Aquí, se deberán capturar los datos marcados con un asterisco (\*) que son los datos mínimos requeridos para el registro de dicho usuario. Al finalizar se debe de dar clic en el botón **Guardar** (ver *Figura 28*).

**Paciente**

**Nombre \***

**Apellido Paterno \***

**Apellido Materno**

**Edad \***

**Sexo \***

Masculino  Femenino

**Domicilio**

**Email**

**Teléfono**

**Guardar**

Figura 28. Campos que deben ser llenados para guardar los datos de un paciente

## Generar reporte

Para obtener el reporte en formato **PDF**, solo se debe hacer clic en el botón **Generar Reporte** (ver *Figura 29*). Posteriormente se asigna un nombre y la ruta del archivo.

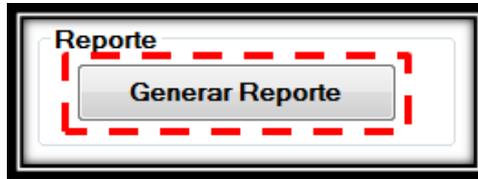


Figura 29. Botón Generar Reporte

El reporte se visualiza de la siguiente manera, (ver *Figura 30*).



### Reporte del Análisis Iridológico

---

Fecha de elaboración: 25/11/2012

Ficha de Identificación		
Nombre: AAAAAA BBBBB BBBBB	Edad: 12	Sexo: Femenino
Domicilio: XXXXX	Teléfono: ( )	
E-mail: XXXXX		

Análisis Iridológico	
Ojo Derecho	Ojo Izquierdo
	
Zonas más afectadas	Zonas más afectadas
<ol style="list-style-type: none"> <li>1.- Pelvis: 76% Afectación, Dañada</li> <li>2.- Psoas/Esfala: 76% Afectación, Dañada</li> <li>3.- Médula: 75% Afectación, Dañada</li> <li>4.- Cuello-Hombro: 75% Afectación, Dañada</li> <li>5.- Boca-Mandíbula: 73% Afectación, Dañada</li> <li>6.- Tumbas: 72% Afectación, Dañada</li> <li>7.- Ovarios-Testículos: 70% Afectación, Dañada</li> <li>8.- Vagina-Pene: 70% Afectación, Dañada</li> <li>9.- Diafragma: 69% Afectación, Dañada</li> <li>10.- Vejiga: 69% Afectación, Dañada</li> </ol>	<ol style="list-style-type: none"> <li>1.- Mesencefalo: 79% Afectación, Dañada</li> <li>2.- Riñón: 75% Afectación, Dañada</li> <li>3.- Pie-Pierna: 73% Afectación, Dañada</li> <li>4.- Diafragma: 71% Afectación, Dañada</li> <li>5.- Mandíbula Superior: 69% Afectación, Dañada</li> <li>6.- Espalda: 69% Afectación, Dañada</li> <li>7.- Recto-Ano: 69% Afectación, Dañada</li> <li>8.- Pared abdominal: 69% Afectación, Dañada</li> <li>9.- Seno: 68% Afectación, Dañada</li> <li>10.- Vagina-Pene: 68% Afectación, Dañada</li> </ol>

---

ANÁLISIS FINAL
Página 1

Figura 30. Reporte del Análisis Iridológico

**Universidad Autónoma Metropolitana**  
**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería  
Ingeniería en Computación

---

---

**Sistema Computarizado Auxiliar en el  
Diagnóstico de Enfermedades mediante el  
Método de la Iridología (SCADE-Iris)**



**UAM – AZCAPOTZALCO | MARIANA BELTRÁN ACOSTA**

**Asesor | DR. ARMANDO JIMÉNEZ FLORES**

**Diciembre 6 de 2012**

### RESUMEN

La Iridología es un método basado en la observación y el análisis del iris para diagnosticar alteraciones en los órganos del cuerpo humano. Este método forma parte de la medicina alternativa y utiliza un mapa iridológico que describe diversas zonas del iris asociadas con los órganos del cuerpo. En él se observan características como son: el color, la forma, la textura y algunas alteraciones que presenta el iris. Cada órgano se refleja en un punto concreto del iris por lo que podremos observar los órganos y las diferentes fases de alteración en la que se encuentra dicho órgano.

Este método de análisis tiene como función interpretar las características del iris, a fin de detectar patologías existentes o prevenir la aparición de daños estructurales al organismo. La Iridología es un método de diagnóstico que es desconocido por muchos y poco confiable para otros. Quienes practican éste técnica en México, son, principalmente los médicos que utilizan métodos alternativos de curación, tales como los naturistas y los homeópatas. Un alto porcentaje de la población desconoce este método debido a que generalmente recurren a la medicina tradicional.

La implementación de este proyecto se debió a la inquietud sobre el tema, ya que se desconocía los beneficios que ofrece la Iridología. Adentrándonos al estudio de este método de diagnóstico se pudo visualizar que no sólo es con el color del iris como se obtienen resultados sobre el grado de afectación de cada órgano, lo más importante son las formas y texturas que presenta el iris. Éste proyecto se basa en el color del iris, por lo que es necesario complementar los resultados de este trabajo con análisis de textura y formas para obtener un diagnóstico con mayor precisión.

En este proyecto se desarrolló un sistema para el diagnóstico de enfermedades por medio de una plantilla que representa el mapa Iridológico, que consiste en la representación gráfica de las áreas iridológicas correspondientes a cada órgano. El sistema se encuentra basado en la topología de Bernard Jensen.

Este documento muestra de manera general y detallada, la forma en que se realizó dicho sistema al que hemos denominado SCADE-Iris. Se detalla el modelado diseñado para la implementación del mismo, la forma gráfica de los algoritmos desarrollados y el código que se implementó para el funcionamiento del sistema.

# ÍNDICE

<b>Prólogo .....</b>	<b>- 7 -</b>
<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>- 8 -</b>
1.1. Antecedentes .....	- 8 -
1.2. Objetivo General .....	- 9 -
1.3. Objetivos Particulares .....	- 9 -
<b>CAPÍTULO 2. MODELADO .....</b>	<b>- 10 -</b>
2.1. Descripción general de bloques.....	- 10 -
2.1.1. Interfaz Gráfica de Usuario .....	- 11 -
2.1.2. Selector de archivo .....	- 11 -
2.1.3. Capturador de imagen .....	- 11 -
2.1.4. Cargador de imagen .....	- 11 -
2.1.5. Ajustador de plantilla .....	- 11 -
2.1.6. Generador de mapa.....	- 11 -
2.1.7. Extractor de datos por zonas .....	- 11 -
2.1.8. Analizador cromático por zonas.....	- 11 -
2.1.9. Analizador Iridológico .....	- 11 -
2.1.10. Base de Datos .....	- 11 -
2.2. Descripción detallada de bloques .....	- 12 -
2.2.1. Interfaz Gráfica de usuario.....	- 12 -
2.2.2. Selector de archivo .....	- 12 -
2.2.3. Capturador de imagen .....	- 12 -
2.2.4. Cargador de imagen .....	- 13 -
2.2.5. Ajustador de plantilla .....	- 13 -
2.2.6. Generador de mapa.....	- 13 -
2.2.7. Extractor de datos por zonas .....	- 14 -
2.2.8. Analizador cromático por zonas.....	- 14 -
2.2.9. Analizador Iridológico .....	- 15 -
2.2.10. Base de Datos .....	- 15 -
<b>CAPÍTULO 3. ESPECIFICACIONES TÉCNICAS .....</b>	<b>- 16 -</b>
3.1. Algoritmos.....	- 16 -
3.1.1. Selector de archivo .....	- 16 -
3.1.2. Capturador de imagen .....	- 17 -
3.1.3. Cargador de imagen .....	- 18 -
3.1.4. Ajustador de plantilla .....	- 18 -
3.1.5. Generador de mapa.....	- 19 -
3.1.6. Extractor de datos por zonas .....	- 19 -
3.1.7. Analizador cromático por zonas.....	- 20 -
3.1.8. Analizador Iridológico .....	- 21 -
<b>CAPÍTULO 4. IMPLEMENTACIÓN DE BLOQUES .....</b>	<b>- 22 -</b>
4.1. Diagramas de Flujo y Código.....	- 22 -
4.1.1. Selector de archivo .....	- 22 -
4.1.1.1. Código (C#) .....	- 23 -
4.1.2. Capturador de imagen .....	- 24 -
4.1.2.1. Código (C#) .....	- 25 -
4.1.3. Cargador de imagen .....	- 26 -
4.1.3.1. Código (C#) .....	- 26 -

# SCADE-Iris

4.1.4.	Ajustador de plantilla .....	- 27 -
4.1.4.1.	Código (C#) .....	- 28 -
4.1.5.	Generador de mapa.....	- 29 -
4.1.5.1.	Código (C#) .....	- 30 -
4.1.6.	Extractor de datos por zonas .....	- 31 -
4.1.6.1.	Código (C#) .....	- 32 -
4.1.7.	Analizador cromático por zonas.....	- 34 -
4.1.7.1.	Código (C#) .....	- 35 -
4.1.8.	Analizador Iridológico .....	- 36 -
4.1.8.1.	Código (C#) .....	- 37 -
4.2.	Arquitectura SCADE-Iris .....	- 38 -
4.3.	Diagrama de Clases del SCADE-Iris .....	- 39 -
4.4.	Diagrama Entidad – Relación (Base de Datos).....	- 40 -
<b>CAPÍTULO 5. PRUEBAS Y RESULTADOS.....</b>		<b>- 41 -</b>
5.1.	Prueba 1.....	- 41 -
5.1.1.	Resultados de la prueba.....	- 47 -
5.2.	Prueba 2.....	- 47 -
5.2.1.	Resultados de la prueba.....	- 50 -
5.3.	Prueba 3.....	- 50 -
5.3.1.	Resultados de la prueba.....	- 52 -
5.4.	Prueba 4.....	- 52 -
5.4.1.	Resultados de la prueba.....	- 55 -
5.5.	Prueba 5.....	- 55 -
5.5.1.	Resultados de la prueba.....	- 57 -
5.6.	Prueba 6.....	- 57 -
5.6.1.	Resultados de la prueba.....	- 59 -
5.7.	Prueba 7.....	- 60 -
5.7.1.	Resultados de la prueba.....	- 62 -
5.8.	Prueba 8.....	- 62 -
5.8.1.	Resultados de la prueba.....	- 64 -
<b>CAPÍTULO 6. CONCLUSIONES .....</b>		<b>- 65 -</b>
6.1.	Lo que se propuso .....	- 65 -
6.2.	Lo que se hizo .....	- 65 -
6.3.	Lo que se obtuvo .....	- 66 -
6.4.	Trabajos futuros.....	- 66 -
<b>REFERENCIAS.....</b>		<b>- 67 -</b>
<b>DOCUMENTOS ANEXOS.....</b>		<b>- 68 -</b>

## Índice de figuras

Figura 1. Diagrama de bloques del SCADE-Iris. ....	10
Figura 2. Entrada-Salida Selector de archivo. ....	12
Figura 3. Entrada-Salida Capturador de imagen. ....	13
Figura 4. Entrada-Salida Cargador de imagen. ....	13
Figura 5. Entrada-Salida Ajustador de plantilla. ....	13
Figura 6. Entrada-Salida Generador de mapa. ....	14
Figura 7. Entrada-Salida Extractor de datos por zona. ....	14
Figura 8. Entrada-Salida Analizador cromático por zonas. ....	15
Figura 9. Entrada-Salida Analizador Iridológico. ....	15
Figura 10. Diagrama de flujo Selector de archivo. ....	22
Figura 11. Diagrama de flujo Capturador de imagen. ....	24
Figura 12. Diagrama de flujo Cargador de imagen. ....	26
Figura 13. Diagrama de flujo Ajustador de plantilla. ....	27
Figura 14. Diagrama de flujo Generador de mapa. ....	29
Figura 15. Diagrama de flujo Extracción de datos por zona. ....	31
Figura 16. Diagrama de flujo Analizador cromático por zonas. ....	34
Figura 17. Diagrama de flujo Analizador Iridológico. ....	36
Figura 18. Arquitectura de SCADE-Iris. ....	38
Figura 19. Diagrama de clases de SCADE-Iris. ....	39
Figura 20. Diagrama Entidad-Relación de Base de Datos. ....	40
Figura 21. Pantalla donde se visualiza la búsqueda del archivo del ojo derecho. ....	41
Figura 22. Pantalla mostrando la primera imagen seleccionada. ....	42
Figura 23. Pantalla mostrando las dos imágenes dentro del sistema. ....	42
Figura 24. Pantalla que muestra los círculos que delimitaran el ojo. ....	43
Figura 25. Pantalla mostrando los ojos delimitados. ....	43
Figura 26. Pantalla mostrando el mapa para delimitar las zonas. ....	44
Figura 27. Pantalla mostrando el proceso de análisis. ....	44
Figura 28. Pantalla mostrando las gráficas generadas para cada ojo. ....	45
Figura 29. Pantalla mostrando el cuadro de dialogo de paciente guardado. ....	45
Figura 30. Pantalla mostrando la ventana de guardado de archivo. ....	46
Figura 31. Reporte correspondiente a la prueba 1. ....	46
Figura 32. Pantalla mostrando los ambos ojos tomados. ....	47
Figura 33. Pantalla mostrando los ojos delimitados por círculos. ....	48
Figura 34. Pantalla mostrando el sistema realizando el análisis. ....	48
Figura 35. Pantalla mostrando el resultado del análisis. ....	49
Figura 36. Reporte correspondiente a la prueba 2. ....	49
Figura 37. Pantalla con las imágenes de los ambos ojos. ....	50
Figura 38. Pantalla con el mapa generado en cada ojo. ....	51
Figura 39. Pantalla con los resultados del análisis desarrollado. ....	51
Figura 40. Reporte correspondiente a la prueba 3. ....	52
Figura 41. Pantalla mostrando las imágenes de ambos ojos. ....	53
Figura 42. Pantalla mostrando la delimitación de los círculos y ojos. ....	53
Figura 43. Pantalla con los resultados del análisis. ....	54
Figura 44. Reporte correspondiente a la prueba 4. ....	54
Figura 45. Pantalla con las imágenes del iris cargadas dentro del sistema. ....	55
Figura 46. Pantalla con el iris delimitado por los círculos. ....	55
Figura 47. Resultados del análisis realizado. ....	56

## SCADE-Iris

---

---

Figura 48. Reporte correspondiente a la prueba 5. ....	- 56 -
Figura 49. Pantalla con las imágenes del iris cargadas dentro del sistema. ....	- 57 -
Figura 50. Ojos delimitados por los círculos. ....	- 58 -
Figura 51. Resultados del análisis realizado. ....	- 58 -
Figura 52. Reporte correspondiente a la prueba 6. ....	- 59 -
Figura 53. Pantalla del sistema SCADE-Iris con las imágenes cargadas. ....	- 60 -
Figura 54. Ojos delimitados por los tres círculos. ....	- 60 -
Figura 55. Análisis realizado. ....	- 61 -
Figura 56. Reporte correspondiente a la prueba 7. ....	- 61 -
Figura 57. Pantalla con la imagen de ambos ojos para su posterior análisis. ....	- 62 -
Figura 58. Ojos delimitados. ....	- 62 -
Figura 59. Resultados del análisis realizado. ....	- 63 -
Figura 60. Reporte correspondiente a la prueba 8. ....	- 63 -

### Prólogo

En el presente documento de Proyecto Terminal se describe el análisis, diseño, implementación y pruebas de un Sistema Computarizado Auxiliar en el Diagnóstico de Enfermedades mediante el Método de la Iridología (SCADE-Iris).

En el primer capítulo se describe la importancia que tiene la Iridología, tomándola como un proceso de análisis Iridológico mediante el cual pretende detectar las zonas afectadas del cuerpo humano. Este capítulo contiene el objetivo general y los objetivos particulares que se alcanzaron en el transcurso de la elaboración del proyecto. Se encontrará también una breve introducción de los antecedentes de este tipo de proyectos y acerca del software que se utiliza actualmente para este tipo de estudios.

Los siguientes tres capítulos describen el modelado, los algoritmos utilizados y la descripción con diagramas de flujo y códigos utilizados para implementar el sistema. En el capítulo dos se expone el modelado del sistema realizado, describiendo cada bloque de manera general y de manera detallada, esto para mejorar el entendimiento de lo que realiza cada módulo. El capítulo tres, es el capítulo de especificaciones técnicas en el cual se muestran los algoritmos desarrollados para cada uno de los módulos. El capítulo cuatro contiene diagramas de flujo así como los códigos de cada uno de los módulos. Adicionalmente, en este capítulo encontramos la arquitectura y la base de datos que se diseñaron para realizar el proyecto.

El capítulo cinco es la parte medular del proyecto ya que en él se muestran las pruebas que se realizaron y los resultados obtenidos en cada uno de los análisis realizados.

El capítulo seis muestra las conclusiones que se obtuvieron al finalizar dicho proyecto una vez realizada la fase de pruebas, aquí se habla de lo que se propuso en un inicio y lo que realmente se desarrolló. Este capítulo concluye con la explicación de las posibles mejoras a implementarse en dicho sistema.

Este trabajo contiene en la parte final una lista de referencias bibliográficas que se utilizaron para el desarrollo del mismo. Finalmente, se incluyen dos anexos: el anexo A, que contiene los reportes realizados durante la etapa de pruebas y el anexo B, que contiene el Manual de usuario.

## CAPÍTULO 1. INTRODUCCIÓN

La utilidad del iris en la medicina alternativa para el diagnóstico de problemas de salud se debe a que el iris es como una huella dactilar; es decir, cada individuo posee un iris único, y en éste se encuentra toda la información de nuestro estado neurológico, químico, orgánico y estructural. Es por ello, que se dice que el diagnóstico mediante el iris, se basa en cualquier alteración del organismo, que llega y se registra por vía nerviosa debido a que contiene un gran número de minúsculos filamentos nerviosos, que reciben impresiones de todo el sistema nervioso del cuerpo.

El correcto examen y diagnóstico del iris realizado por profesionales capacitados, es una de las formas más eficaces, fiables, inofensivas y poco costosas para detectar las deficiencias orgánicas, debido a la estrecha conexión del iris con sus centros nerviosos.

### 1.1. Antecedentes

Con base en una búsqueda bibliográfica realizada en la base de datos de proyectos terminales de la UAM Azcapotzalco, durante los últimos 5 años, se tiene conocimiento de varios proyectos de la carrera de Ingeniería en Computación relacionados con el procesamiento de imágenes [1-7], los cuales no son aplicados en específico al iris. Existe un proyecto llamado “Sistema de identificación de personas a través del iris mediante análisis de texturas por filtrado de wavelets” [8]. Aquí se aplica el procesamiento de imágenes del iris para la identificación de personas. Los objetivos de este proyecto y la técnica de análisis son distintos a los del proyecto SCADE-Iris propuesto.

Se realizó una búsqueda de proyectos en otros planteles y se encontraron dos proyectos relacionados, provenientes de la UAM Iztapalapa. Uno se llama “Autenticación de personas a través del iris” [9], es idéntico al realizado en la UAM Azcapotzalco, pero utiliza otra técnica de análisis. Otro proyecto, llamado “Procesamiento de imágenes de fondo de ojo para la detección temprana de diabetes mellitus e hipertensión arterial” [10] utiliza imágenes del fondo del ojo, pero sólo se aplica a la detección de problemas de diabetes mellitus e hipertensión arterial y no utiliza el método de la Iridología.

Se hizo una búsqueda de proyectos realizados en otras instituciones y se encontró uno que tiene relación con el SCADE-Iris y se llama: “Reconocimiento de imágenes mediante redes neuronales artificiales aplicado a la Iridología” [11]. En este trabajo se desarrolló un sistema para la detección de lesiones por medio del estudio de la Iridología, aplicando redes neuronales. En el SCADE-Iris el análisis se realizará a nivel cromático de las imágenes del iris.

Algunos paquetes comerciales de software que utilizan el método iridológico son:

1. **Aigal** [12]: Es un software que se especializa en el análisis del iris ayudándole al iridólogo a localizar los signos en el iris. No diagnostica ni interpreta, sólo se pueden marcar áreas en el iris, ingresar comentarios y aplicar niveles de zoom para realizar el diagnóstico.
2. **Irisearch Pro 2000** [13]: Este software es más completo con respecto al método de la Iridología. Es muy técnico, por lo que, sólo es apto para usuarios especializados en el tema.

### 1.2. Objetivo General

Diseñar e implementar un sistema computarizado auxiliar para el diagnóstico de enfermedades mediante el análisis de imágenes del iris, aplicando el método de la Iridología (SCADE-Iris).

### 1.3. Objetivos Particulares

1. Diseñar una Interfaz Gráfica de Usuario para el SCADE-Iris.
2. Desarrollar el módulo “Selector de archivo”.
3. Desarrollar el módulo “Capturador de imagen”.
4. Desarrollar el módulo “Cargador de imagen”.
5. Desarrollar el módulo “Ajustador de plantilla”.
6. Desarrollar el módulo “Generador de mapa”.
7. Desarrollar el módulo Extractor de datos por zonas.
8. Desarrollar el módulo “Analizador cromático por zonas”.
9. Desarrollar el módulo “Analizador Iridológico”.
10. Crear una Base de Datos para almacenar información de cada análisis realizado.
11. Realizar pruebas de usabilidad y evaluaciones finales del SCADE-Iris.

## CAPÍTULO 2. MODELADO

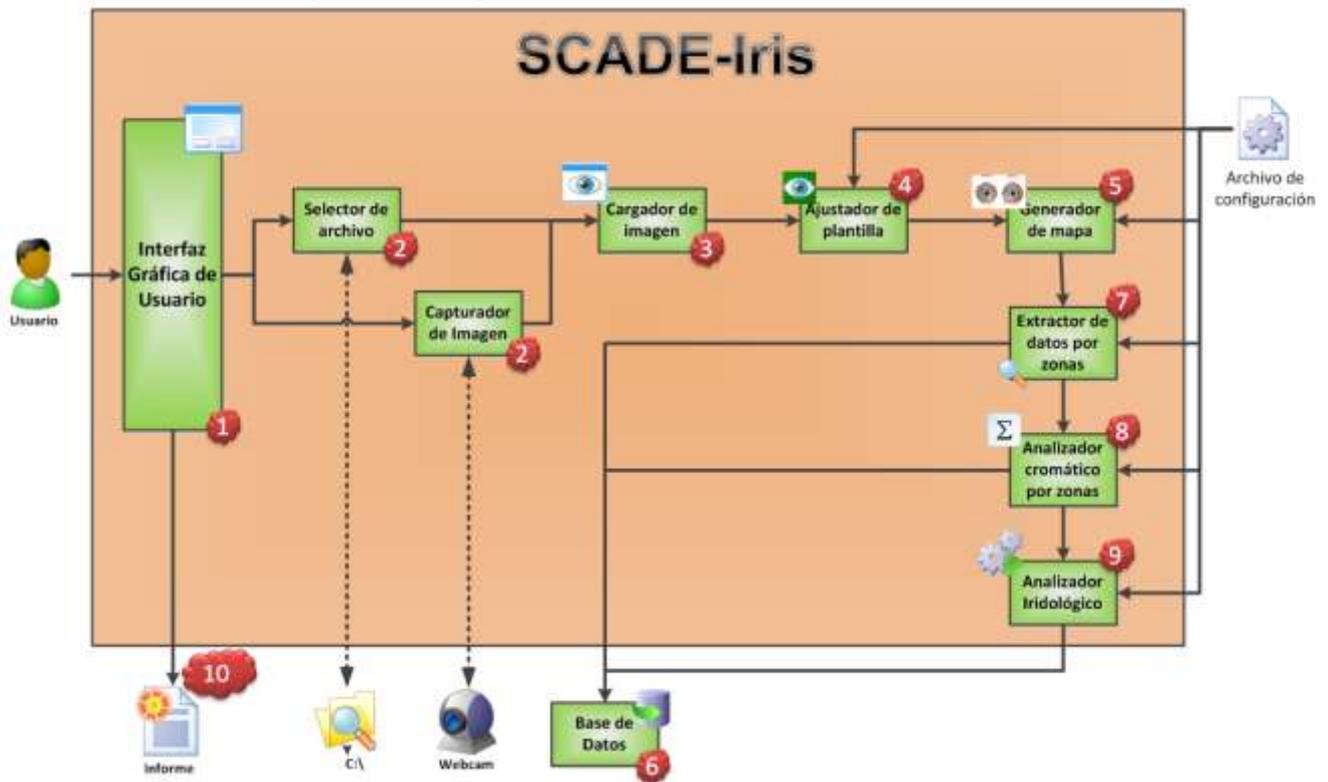
Este capítulo describe el modelado realizado para la implementación del sistema, en el cual se pretende dar a conocer el enfoque global del funcionamiento interno de SCADE-Iris. Se describe el Diagrama general de bloques donde se explica de manera muy general cada módulo, así como su descripción detallada de cada uno de los módulos definiendo las entradas que recibe y las salidas que se obtienen.

### 2.1. Descripción general de bloques

La estructura interna del SCADE-Iris está compuesta por los siguientes bloques:

1. Interfaz Gráfica de usuario
2. Selector de archivo
3. Capturador de imagen
4. Cargador de imagen
5. Ajustador de plantilla
6. Generador de mapa
7. Extractor de datos por zonas
8. Analizador cromático por zonas
9. Analizador Iridológico
10. Base de Datos

La *Figura 1* muestra un diagrama de bloques del SCADE-Iris.



**Figura 1.** Diagrama de bloques del SCADE-Iris.

### **2.1.1. Interfaz Gráfica de Usuario**

Este bloque presenta un formulario que contiene una serie de pestañas gráficas para llevar a cabo operaciones de captura, carga y análisis de una imagen, así como también poder generar un resultado (reporte) final.

### **2.1.2. Selector de archivo**

Este bloque es encargado de seleccionar, de una ruta específica, un archivo de imagen para su análisis, tanto para el ojo derecho como para el ojo izquierdo.

### **2.1.3. Capturador de imagen**

Bloque encargado de capturar la imagen de cada uno de los ojos del paciente mediante el uso de una *webcam*.

### **2.1.4. Cargador de imagen**

Este bloque cargará en el **SCADE-Iris** la imagen seleccionada de una ruta específica o capturada de la *webcam* para su posterior análisis.

### **2.1.5. Ajustador de plantilla**

La función de este bloque consiste en empalmar la imagen de la plantilla Iridológica (3 círculos iniciales) sobre la imagen del paciente.

### **2.1.6. Generador de mapa**

Este bloque genera círculos y líneas que construyen las zonas del mapa Iridológico.

### **2.1.7. Extractor de datos por zonas**

Este bloque se encarga de extraer la información de la imagen a nivel de píxeles y genera todas las zonas del mapa Iridológico.

### **2.1.8. Analizador cromático por zonas**

En este bloque se analiza la imagen de cada zona para etiquetarla con un valor de afectación.

### **2.1.9. Analizador Iridológico**

La función de este bloque es detectar y clasificar zonas afectadas.

### **2.1.10. Base de Datos**

En este bloque se almacena la información generada por cada una de las ejecuciones (análisis) por paciente.

## 2.2. Descripción detallada de bloques

### 2.2.1. Interfaz Gráfica de usuario

En este bloque se diseñó un formulario que fungirá como Interfaz y poder comunicar al usuario final con el sistema. La Interfaz contendrá tres pestañas para llevar a cabo el proceso de **Análisis Iridológico**. Dicho formulario cuenta además con una consola para poder visualizar las imágenes analizadas y posteriormente plasmar en ellas el mapa iridológico.

La Interfaz permite en una de las tres pestañas, poder generar un resultado final (reporte) del análisis ejecutado en ese momento.

### 2.2.2. Selector de archivo

La entrada a este bloque es un archivo de imagen con formato *.bmp* que se encuentra almacenado en alguna ruta.

Una vez que se selecciona esta opción de entrada con la ayuda de la **Interfaz Gráfica de Usuario**, el selector de archivo permite navegar en la computadora del sistema para seleccionar el archivo deseado y poder trabajar con él.

Los archivos seleccionados para ambos ojos (izquierdo y derecho) son almacenados en una carpeta de trabajo que pertenece al **SCADE-Iris** y que tiene referencias al análisis en curso.

El archivo de salida es el que se eligió proveniente de una ruta, y al final, este es enviado al siguiente bloque **Cargador de imagen**. En la *Figura 2* muestra un diagrama de la entrada y salida del módulo.



Figura 2. Entrada-Salida Selector de archivo.

### 2.2.3. Capturador de imagen

Otro modo de entrada de información hacia el **SCADE-Iris** que permite realizar la **Interfaz Gráfica de Usuario**, es la imagen que se recibe a través de la *webcam*.

La resolución típica de una *webcam* es de aproximadamente 1.3 *megapíxeles*, en este caso, no importa cual sea la resolución ya que el sistema permite hacer uso de cualquier tipo de cámara despreciando dicha característica.

Este bloque hace que las imágenes capturadas para ambos ojos, se almacenen en una ruta definida en el espacio de trabajo del **SCADE-Iris** y con referencia al análisis en curso.

Al igual que el bloque **Selector de archivo**, este bloque envía como salida la imagen (*.bmp*) al bloque **Cargador de imagen** para su posterior análisis. En la *Figura 3* muestra un diagrama de la entrada que recibe el módulo y posteriormente la salida.

## SCADE-Iris



Figura 3. Entrada-Salida Capturador de imagen.

### 2.2.4. Cargador de imagen

La entrada de este bloque es el archivo que se recibe del **Selector de imagen** o del bloque **Capturador de imagen**, como se muestra en la *Figura 4*.

Su función es cargar los dos archivos de imagen recibidos en el sistema del **SCADE-Iris** y desplegarlas en pantalla bajo el control de la interfaz de usuario para realizar su análisis.

La salida es la imagen en pantalla que utiliza solo como referencia el siguiente bloque para aplicar el ajuste de plantilla. Esta salida se muestra en la *Figura 4*



Figura 4. Entrada-Salida Cargador de imagen.

### 2.2.5. Ajustador de plantilla

La entrada de este módulo es la imagen en pantalla, la cual sirve para asociar la zona del iris con una plantilla Iridológica predefinida de tres círculos que es leída desde un archivo de configuración.

La función que implementa el **Ajustador de plantilla** permite al usuario definir manualmente la zona del iris; es decir, puede empalmar de manera manual la plantilla Iridológica con el objeto de zonificar el área a estudiar. Dicho procesos se logra mediante dos funciones principales:

- **Mover:** Permite mover de posición cualquiera de los tres círculos generados.
- **Redimensionar:** Permite cambiar de tamaño cualquiera de los tres círculos generados.

En este bloque, la salida es representada por la misma plantilla generada para su uso posterior en la creación del mapa Iridológico. La *Figura 5* muestra la entrada y salida del módulo.



Figura 5. Entrada-Salida Ajustador de plantilla.

### 2.2.6. Generador de mapa

La entrada a este bloque es la plantilla Iridológica generada previamente para tomarla como referencia y poder “pintar” círculos y líneas que delimiten las zonas cargadas desde un archivo de configuración, esto para ambos ojos como se muestra en la *Figura 6*.

La salida de este bloque equivale al número de zonas generadas en el mapa Iridológico; es decir, el número de zonas leídas en el archivo de configuración para su estudio.



Figura 6. Entrada-Salida Generador de mapa.

### 2.2.7. Extractor de datos por zonas

La entrada que utiliza este módulo está constituida por la imagen del iris y la imagen del mapa Iridológico, empalmadas y ajustadas.

El módulo de **Extractor de datos por zonas** obtiene un archivo de imagen de todo el ojo y N número de archivos por cada zona del iris delimitada por la plantilla. Es decir, si desde el archivo de configuración se tienen 25 zonas de estudio clínico, al final será generado un archivo que abarca la totalidad del ojo y 25 archivos de cada zona. Aplica para los dos ojos.

La salida de este bloque son los archivos resultantes de cada zona y del ojo completo (archivo **Base**), los cuales se envían al siguiente bloque para su análisis cromático. La *Figura 7* muestra el diagrama de entrada y salida del módulo de **Extractor de datos por zona**.



Figura 7. Entrada-Salida Extractor de datos por zona.

### 2.2.8. Analizador cromático por zonas

Este bloque recibe como entrada el conjunto de archivos parciales (por zona) de toda el área correspondiente a la imagen del iris así como el archivo que cubre la totalidad del ojo (izquierdo y derecho). La *Figura 8* muestra el diagrama de la entrada que recibe el módulo, así como la salida que genera.

Su función es cuantificar, para cada zona, el número de píxeles existentes para obtener una **media** de dicho conjunto. Posteriormente, se obtiene una **desviación estándar** para poder generar un rango y decidir si la zona está o no, afectada.

Al nivel de Iris (archivo de todo el ojo) se obtiene de igual manera la **media**. Después, cada elemento (*pixel*) de color obtenido por cada zona, es contabilizado de acuerdo con al rango generado por la desviación estándar de su correspondiente zona. Si un *pixel* no entra en el rango generado por la media y la desviación estándar, es contabilizado como fuera de rango, en caso contrario, dicho *pixel* se contabiliza en el conjunto de *píxeles* que caen dentro del rango.

Una vez generado dicho conteo para ambos casos (dentro y fuera de rango), se obtiene su equivalencia de cada uno de ellos en porcentaje tomando como referencia el número total de *píxeles* de la zona en curso.

Como datos de salida, se generan para cada archivo de zona los porcentajes de los distintos rangos. Para el archivo Base se genera la media.



Figura 8. Entrada-Salida Analizador cromático por zonas.

### 2.2.9. Analizador Iridológico

Este bloque recibe como entrada del **Analizador cromático por zonas** dos tipos de archivos: archivo **Base** y archivo **Zona**. De ambos tipos de archivo se recibe el valor de la media.

Para cada archivo de zona se realiza una diferencia entre la media del archivo Base y la media del archivo zona, en donde no debe excederse de 10% para que los *pixeles* que caen dentro del rango de la media ( $\text{media} \pm \text{desviación estándar}$ ) sean etiquetados como no afectados.

En caso de que el conjunto de *pixeles* exceda el 10% de la diferencia entre dichas medias, el conjunto de *pixeles* será etiquetado como afectado. La *Figura 9* muestra a detalle la entrada y salida que genera el módulo.

*No se piensa utilizar una técnica de comparación de dos imágenes similares. La comparación sólo se realiza entre cada zona y su archivo Base, comparando únicamente los valores a nivel de pixel.*



Figura 9. Entrada-Salida Analizador Iridológico.

### 2.2.10. Base de Datos

La **Base de Datos** sirve como medio de almacenamiento de toda aquella información analizada, así como su respectivo paciente asociado a dicho análisis.

La Base de Datos utilizada está diseñada en un ambiente XML.

## CAPÍTULO 3. ESPECIFICACIONES TÉCNICAS

Este capítulo describe los algoritmos que se realizaron e implementaron para la construcción de cada módulo. Para tener un mejor entendimiento, se pretende mostrar a detalle lo que se realizó internamente durante el desarrollo de cada módulo. Además, se muestran los pseudocódigos que representan la solución a los algoritmos desarrollados.

### 3.1. Algoritmos

#### 3.1.1. Selector de archivo

El siguiente algoritmo describe el procedimiento para llevar a cabo la selección de una imagen de tipo .bmp a través de una ruta específica y mediante el explorador de ventanas de Microsoft Windows, mostrando la imagen dentro del área de visualización. Este procedimiento abarca la selección de imágenes para el ojo izquierdo y el ojo derecho.

1. Inicio
2. Seleccionar **tipo** de ojo
  - 2.1. Si **ojo** es **igual** a **Derecho**
    - 2.1.1. ¿Se hace clic en botón “**Buscar...**”?
    - 2.1.2. Si
      - 2.1.2.1. Inicializar variable para formato de archivo “**.bmp**”
      - 2.1.2.2. Inicializar variable con valor igual a “**C:\**” para representar el directorio inicial de búsqueda.
      - 2.1.2.3. Abrir cuadro de dialogo para mostrar archivos locales de tipo “**BMP**” en el directorio “**C:\**”
      - 2.1.2.4. Seleccionar imagen.
      - 2.1.2.5. ¿Se hace clic en el botón “**Abrir**”?
      - 2.1.2.6. Si
        - 2.1.2.6.1. Cargar Imagen en el “**picturebox**” **derecho**
        - 2.1.2.6.2. Cerrar el cuadro de diálogo
        - 2.1.2.6.3. Ir al punto **3**
      - 2.1.2.7. No
        - 2.1.2.7.1. Se cierra el cuadro de dialogo
        - 2.1.2.7.2. Volver al punto **2**
    - 2.1.3. No
      - 2.1.3.1. Esperar a que el usuario haga clic en el botón “**Buscar..**”
      - 2.1.3.2. Volver al punto **2**
  - 2.2. Si **ojo** es **igual** a **Izquierdo**
    - 2.2.1. ¿Se hace clic en botón “**Buscar...**”?
    - 2.2.2. Si
      - 2.2.2.1. Inicializar variable para formato de archivo “**.bmp**”
      - 2.2.2.2. Inicializar variable con valor igual a “**C:\**” para representar el directorio inicial de búsqueda.
      - 2.2.2.3. Abrir cuadro de dialogo para mostrar archivos locales de tipo “**BMP**” en el directorio “**C:\**”
      - 2.2.2.4. Seleccionar imagen.
      - 2.2.2.5. ¿Se hace clic en el botón “**Abrir**”?
      - 2.2.2.6. Si
        - 2.2.2.6.1. Cargar Imagen en el “**picturebox**” **izquierdo**
        - 2.2.2.6.2. Cerrar el cuadro de diálogo
        - 2.2.2.6.3. Ir al punto **3**
      - 2.2.2.7. No
        - 2.2.2.7.1. Se cierra el cuadro de dialogo
        - 2.2.2.7.2. Volver al punto **2**

2.2.3. No

2.2.3.1. Esperar a que el usuario haga clic en el botón “**Buscar.**”

2.2.3.2. Volver al punto **2**

3. Fin

### 3.1.2. Capturador de imagen

En el siguiente algoritmo se describe el procedimiento para llevar a cabo la activación/desactivación de una *webcam* para realizar la captura de la imagen de un ojo y mostrarla en el área de visualización. Este procedimiento abarca la captura de imágenes para el ojo izquierdo y el ojo derecho.

1. Inicio

2. Seleccionar **tipo** de ojo

2.1. Si **ojo** es **igual** a **Derecho**

2.1.1. Seleccionar casilla “**Por dispositivo**”

2.1.2. Se activa la *webcam*

2.1.3. Se visualiza la imagen previa en el “*picturebox*”

2.1.4. ¿Se hace clic en botón “**Capturar**”?

2.1.5. Si

2.1.5.1. Cargar Imagen en el “*picturebox*” **derecho**

2.1.5.2. Cerrar el cuadro de diálogo

2.1.5.3. Ir al punto **3**

2.1.6. No

2.1.6.1. Se cierra el cuadro de dialogo

2.1.6.2. Volver al punto **2**

2.2. Si **ojo** es **igual** a **Izquierdo**

2.2.1. Seleccionar casilla “**Por dispositivo**”

2.2.2. Se activa la *webcam*

2.2.3. Se visualiza la imagen previa en el “*picturebox*”

2.2.4. ¿Se hace clic en botón “**Capturar**”?

2.2.5. Si

2.2.5.1. Cargar Imagen en el “*picturebox*” **izquierdo**

2.2.5.2. Cerrar el cuadro de diálogo

2.2.5.3. Ir al punto **3**

2.2.6. No

2.2.6.1. Se cierra el cuadro de dialogo

2.2.6.2. Volver al punto **2**

3. Fin

### 3.1.3. Cargador de imagen

El siguiente algoritmo describe el procedimiento para llevar a cabo la carga de un archivo seleccionado así como de la *webcam*. Al recibir dicha imagen se redimensiona con respecto al área de visualización de imagen. Este procedimiento abarca la carga de imágenes para el ojo izquierdo y el ojo derecho.

1. Inicio
2. Recibir imagen desde **archivo** o por **webcam**
3. Redimensionar la imagen obtenida al tamaño del “**picturebox**”
4. Cargar la imagen en el “**picturebox**” para ser visualizada
5. Fin

### 3.1.4. Ajustador de plantilla

El siguiente algoritmo describe el procedimiento para generar la plantilla en cada uno de los ojos y cómo poder ajustarla (mover y redimensionar) dependiendo el tamaño de las imágenes que se tengan cargadas en el Área de visualización de imagen. Este procedimiento se realiza de manera independientemente en cada ojo.

1. Inicio
2. Hacer clic en el botón “**Delimitar**”
3. Extraer configuración del archivo “**CirculosMapa.xml**”
4. Hacer clic en “**ojo derecho**” para pintar tres círculos iniciales
5. Hacer clic en “**ojo izquierdo**” para pintar tres círculos iniciales
6. Ajustar plantilla en “**ojo derecho**”
7. ¿Es correcto el tamaño de los **3** círculos?
8. No
  - 8.1. Mientras la plantilla no sea la correcta:
    - 8.1.1. Seleccionar la opción “**Mover**”
      - 8.1.1.1. Colocar los círculos en los lugares correspondientes de la imagen
    - 8.1.2. Seleccionar la opción “**Redimensionar**”
      - 8.1.2.1. Ajustar círculos al tamaño de la imagen
  - 8.2. Ir al punto **10**
9. Si
  - 9.1. Ir al punto **10**
10. Ajustar plantilla en “**ojo izquierdo**”
11. ¿Es correcto el tamaño de los **3** círculos?
12. No
  - 12.1. Mientras la plantilla no sea la correcta:
    - 12.1.1. Seleccionar la opción “**Mover**”
      - 12.1.1.1. Colocar los círculos en los lugares correspondientes de la imagen
    - 12.1.2. Seleccionar la opción “**Redimensionar**”
      - 12.1.2.1. Ajustar círculos al tamaño de la imagen
  - 12.2. Ir al punto **14**
13. Si
  - 13.1. Ir al punto **14**
14. Fin

### 3.1.5. Generador de mapa

El siguiente algoritmo indica el procedimiento para crear el **Mapa Iridológico** en ambas plantillas previamente generadas sobre cada una de las imágenes de ambos ojos, tomando en cuenta los archivos de configuración en los cuales se definen las zonas y sus respectivos nombres (ojo izquierdo y ojo derecho).

1. Inicio
2. Extraer configuración del archivo “**CirculosMapa.xml**”
3. Se obtiene **N** número de círculos con radio **R** diferente
4. En la imagen del “**ojo derecho**”
  - 4.1. Pintar los **N** círculos con radio **R** configurados para el “**ojo derecho**”
5. En la imagen del “**ojo izquierdo**”
  - 5.1. Pintar los **N** círculos con radio **R** configurados para el “**ojo izquierdo**”
6. Extraer configuración del archivo “**MapaIridologico.xml**”
7. Se obtiene **M** número de zonas
8. En la imagen del “**ojo derecho**”
  - 8.1. Pintar las **M** zonas para el “**ojo derecho**”
9. En la imagen del “**ojo izquierdo**”
  - 9.1. Pintar las **M** zonas para el “**ojo izquierdo**”
10. Fin

### 3.1.6. Extractor de datos por zonas

Este algoritmo describe el proceso de obtención de zonas mediante la creación de archivos, en donde cada uno de ellos representa una zona. De igual forma se describe la creación de un archivo que obtiene información a nivel de pixel de todo el ojo. Este procedimiento se lleva a cabo para cada uno de los ojos.

1. Inicio
2. Recibir imagen de **ojo derecho**
3. Recibir mapa iridológico **ojo derecho**
4. Recibir imagen de **ojo izquierdo**
5. Recibir mapa iridológico **ojo izquierdo**
6. Obtener del archivo de configuración el **tipo de extensión** que tendrán los archivos.
7. Para **ojo derecho**:
  - 7.1. Obtener del archivo de configuración la ruta para almacenar archivo **Base**.
  - 7.2. Generar archivo **Base**
  - 7.3. Generar **límite inicial** y **límite final**
  - 7.4. Generar **grado inicial** y **grado final**
  - 7.5. Desde **límite inicial** y **grado inicial**, hasta **grado final** y **límite final**
    - 7.5.1. **Extraer** pixel
    - 7.5.2. **Guardar** pixel en archivo **Base**
  - 7.6. Mientras la zona **i = 0** sea menor a **N**
  - 7.7. Obtener del archivo de configuración la ruta para almacenar archivo **zona**.
  - 7.8. Generar archivo **zona**
  - 7.9. Generar **límite inicial** y **límite final**
  - 7.10. Generar **grado inicial** y **grado final**
  - 7.11. Desde **límite inicial** y **grado inicial**, hasta **grado final** y **límite final**
    - 7.11.1. **Extraer** pixel
    - 7.11.2. **Guardar** pixel en archivo **zona**
  - 7.12. Incrementar **i + 1**
8. Para **ojo izquierdo**:
  - 8.1. Obtener del archivo de configuración la ruta para almacenar archivo **Base**.

- 8.2. Generar archivo **Base**
- 8.3. Generar **límite inicial** y **límite final**
- 8.4. Generar **grado inicial** y **grado final**
- 8.5. Desde **límite inicial** y **grado inicial**, hasta **grado final** y **límite final**
  - 8.5.1. **Extraer** pixel
  - 8.5.2. **Guardar** pixel en archivo **Base**
- 8.6. Mientras la zona **i = 0** sea menor a **N**
- 8.7. Obtener del archivo de configuración la ruta para almacenar archivo **zona**.
- 8.8. Generar archivo **zona**
- 8.9. Generar **límite inicial** y **límite final**
- 8.10. Generar **grado inicial** y **grado final**
- 8.11. Desde **límite inicial** y **grado inicial**, hasta **grado final** y **límite final**
  - 8.11.1. **Extraer** pixel
  - 8.11.2. **Guardar** pixel en archivo **zona**
- 8.12. Incrementar **i + 1**
9. Fin

### 3.1.7. Analizador cromático por zonas

En este algoritmo se describe el procedimiento de análisis de cada zona, de donde se obtiene la media (promedio) de los *pixeles*. A partir de estos valores se cuantifican los *pixeles* que están o no dentro de un rango generado, gracias a la obtención de la desviación estándar. Este procedimiento se realiza en ambos ojos.

1. Inicio
2. Obtener promedio de los colores del archivo **Base**
3. Inicializar variable con numero de zonas generadas (**N zonas**)
4. Mientras la zona **i=0** sea menor a **N**
  - 4.1. Calcular número total de datos del archivo (**total**)
  - 4.2. Sumar el total de los datos (**suma**)
  - 4.3. Obtener media de los datos (**media = suma/total**)
  - 4.4. Obtener **desviación estándar** con datos anteriores
  - 4.5. Obtener límite superior (**lim sup = media + desviación estándar**)
  - 4.6. Obtener límite inferior (**lim inf = media - desviación estándar**)
  - 4.7. Mientras el numero de datos sea menor a **total**
    - 4.7.1. Comparar con rango generado por los límites **inferior** y **superior**
    - 4.7.2. Contabilizar **datos dentro de rango**
    - 4.7.3. Contabilizar **datos fuera de rango**
  - 4.8. Incrementar **i + 1**
5. Convertir el valor **datos dentro de rango** en porcentaje del valor **total**
6. Convertir el valor **datos fuera de rango** en porcentaje del valor **total**
7. Fin

### 3.1.8. Analizador Iridológico

En el siguiente algoritmo se describe el proceso de clasificación de cada una de las zonas, en donde lo que se obtiene es el grado de afectación de cada una de ellas, en donde se consideran 3 grados de afectación que son: “En buen estado”, “En mal estado” y “Dañada”.

1. Inicio
2. Inicializar variable con la **media del archivo Base**
3. Inicializar variable con numero de zonas generadas (N zonas)
4. Mientras la zona **i=0** sea menor a N
  - 4.1. Restar: **media** de la base – **media** de la zona
  - 4.2. Obtener el valor de la resta en la variable **resultado**
  - 4.3. ¿El **resultado** es menor o igual a **10**?
  - 4.4. Si
    - 4.4.1. Marcar la zona como **No Afectada**
    - 4.4.2. Incrementar **i + 1**
  - 4.5. No
    - 4.5.1. Marcar la zona como **Afectada**
    - 4.5.2. Incrementar **i + 1**
5. Mientras la zona **i=0** sea menor a N
  - 5.1.1. Comparar el valor **resultado**
  - 5.1.2. **Case1:** ¿**0** <= **resultado** <= **10**?
  - 5.1.3. Etiquetar la zona como **En buen estado**
  - 5.1.4. **Case2:** ¿**11** <= **resultado** <= **50**?
  - 5.1.5. Etiquetar la zona como **En mal estado**
  - 5.1.6. **Case3:** ¿**51** <= **resultado** <= **100**?
  - 5.1.7. Etiquetar la zona como **Dañada**
  - 5.1.8. Incrementar **i + 1**
6. Fin

## CAPÍTULO 4. IMPLEMENTACIÓN DE BLOQUES

Este capítulo pretende una mejor comprensión de la solución de cada módulo tomando los algoritmos representados con anterioridad. Los diagramas representados a continuación separaran la solución lógica de programación de la parte de reglas y sintaxis de codificación por lo que se muestra el código que se desarrollo durante la implementación de los módulos. También se detalla la Arquitectura y la Base de Datos que se diseñó para la implementación del proyecto.

### 4.1. Diagramas de Flujo y Código

#### 4.1.1. Selector de archivo

El diagrama de selector de archivo muestra el proceso para la selección de una imagen de tipo .bmp a través del explorador de ventanas de Windows, mostrando la toma de decisiones para abrir y seleccionar dicho archivo. Este procedimiento se ejecuta en ambos ojos. La *Figura 10* muestra el flujo para la implementación del módulo **Selector de archivo**.

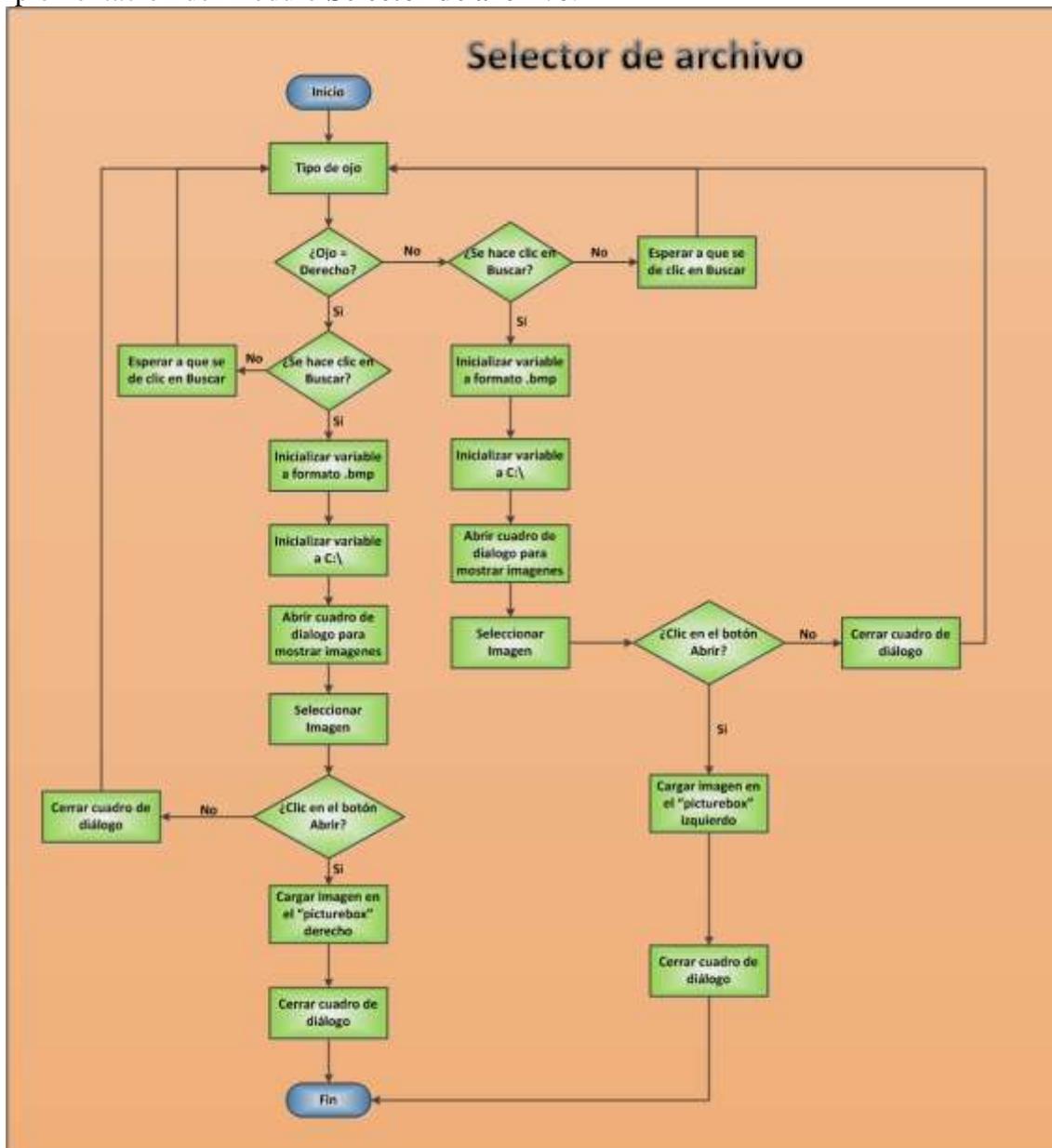


Figura 10. Diagrama de flujo Selector de archivo.

### 4.1.1.1. Código (C#)

```
private void lnkBuscarImagen_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    OpenFileDialog opd = new OpenFileDialog();
    opd.Filter = "Archivos BMP (*.bmp)|*.bmp";
    opd.InitialDirectory = "C:";

    if (opd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        if (rbtnOjoIzquierdo.Checked)
            pictureBoxOjoIzquierdo.Image = resizeImage((Image)(new Bitmap(opd.FileName)),
            pictureBoxOjoIzquierdo.Size);
        else if (rbtnOjoDerecho.Checked)
            pictureBoxOjoDerecho.Image = resizeImage((Image)(new Bitmap(opd.FileName)),
            pictureBoxOjoDerecho.Size);
    }
}
```

4.1.2. Capturador de imagen

El diagrama muestra el módulo Capturador de imagen, en el cuál se detalla el procedimiento para llevar a cabo la activación/desactivación de la *webcam* y realizar la captura de ambos ojos (izquierdo y derecho), mostrando la toma de decisiones para la ejecución de la captura de imagen. La *Figura 11* muestra el flujo para la implementación del módulo **Capturador de imagen**.

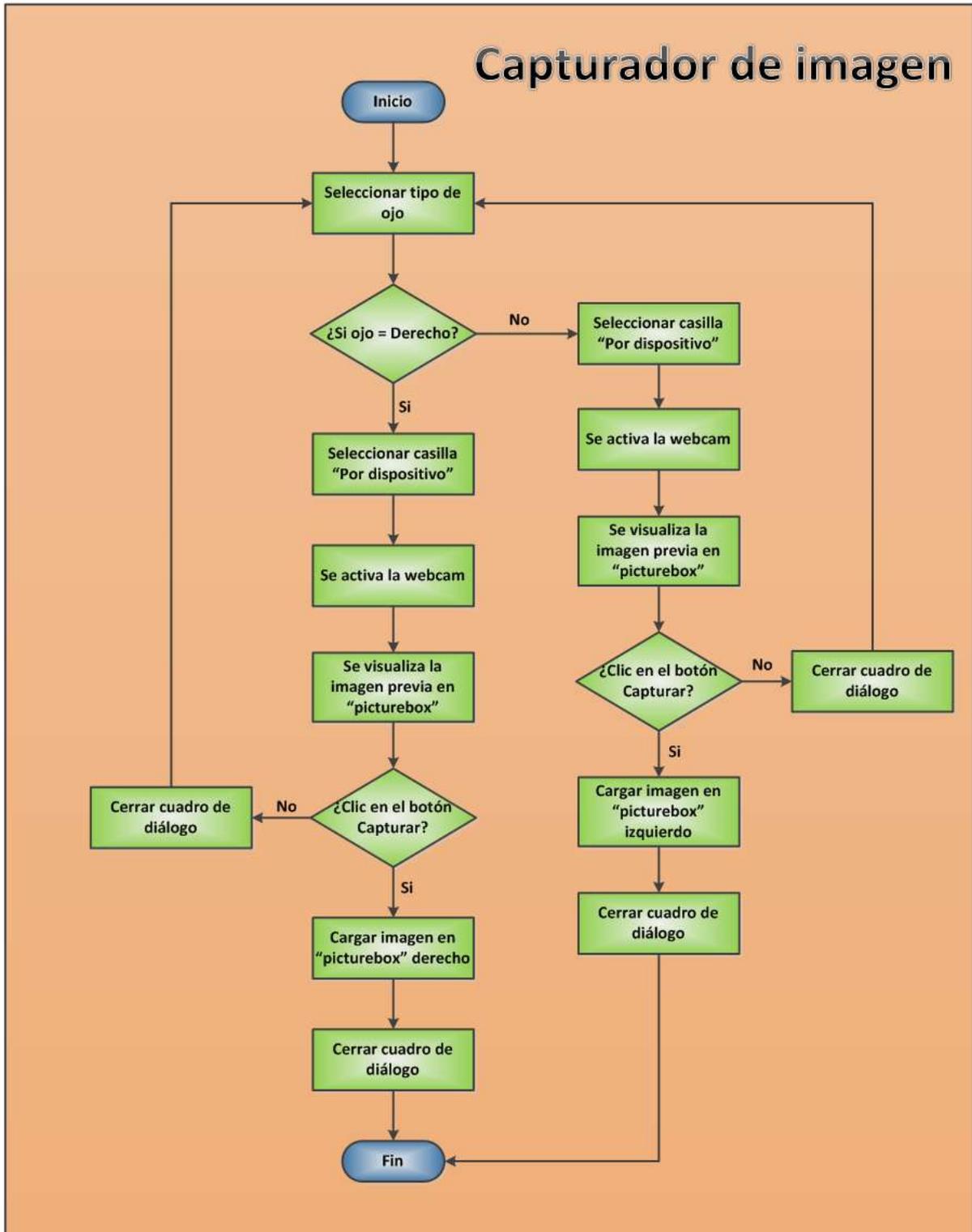


Figura 11. Diagrama de flujo Capturador de imagen.

### 4.1.2.1. Código (C#)

```
private void chkWebCam_CheckedChanged(object sender, EventArgs e)
{
    if (chkWebCam.Checked)
    {
        webCam.TimeToCapture_milliseconds = 1;
        webCam.Start(0);
    }
    else
    {
        webCam.Stop();
        pictureBoxWebCam.Image = null;
    }

    btnCapturarImagen.Enabled = chkWebCam.Checked;
}

private void webCam_ImageCaptured(object source, WebCam_Capture.WebcamEventArgs e)
{
    pictureBoxWebCam.Image = e.WebCamImage;
}

private void btnCapturarImagen_Click(object sender, EventArgs e)
{
    if (rbtnOjoIzquierdo.Checked)
        pictureBoxOjoIzquierdo.Image = resizeImage(pictureBoxWebCam.Image,
pictureBoxOjoIzquierdo.Size);
    else if (rbtnOjoDerecho.Checked)
        pictureBoxOjoDerecho.Image = resizeImage(pictureBoxWebCam.Image,
pictureBoxOjoDerecho.Size);
}
```

### 4.1.3. Cargador de imagen

El diagrama representa el módulo de Cargador de imagen, y muestra el procedimiento que lleva a cabo el sistema para cargar las imágenes de ambos ojos dentro del Área de visualización, en primera estancia redimensionando las imágenes para la colocación de la imagen dentro del área. La *Figura 12* muestra el diagrama de flujo para la implementación del módulo **Cargador de imagen**.

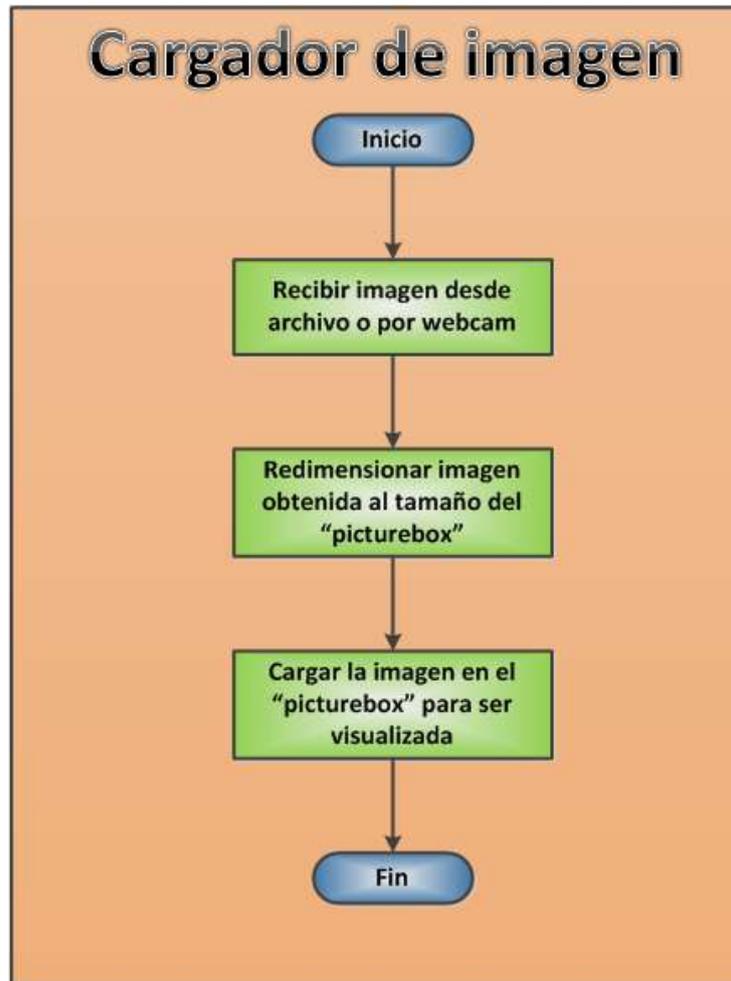


Figura 12. Diagrama de flujo Cargador de imagen.

#### 4.1.3.1. Código (C#)

```
if (rbtnOjoIzquierdo.Checked)
pictureBoxOjoIzquierdo.Image = resizeImage((Image)(new Bitmap(opd.FileName)),
pictureBoxOjoIzquierdo.Size);
else if (rbtnOjoDerecho.Checked)
pictureBoxOjoDerecho.Image = resizeImage((Image)(new Bitmap(opd.FileName)),
pictureBoxOjoDerecho.Size);

if (rbtnOjoIzquierdo.Checked)
pictureBoxOjoIzquierdo.Image = resizeImage(pictureBoxWebCam.Image,
pictureBoxOjoIzquierdo.Size);
else if (rbtnOjoDerecho.Checked)
pictureBoxOjoDerecho.Image = resizeImage(pictureBoxWebCam.Image, pictureBoxOjoDerecho.Size);
```

#### 4.1.4. Ajustador de plantilla

El diagrama para ajustar la plantilla muestra el proceso que lleva para aplicar los círculos que se encuentran definidos en un archivo de configuración dependiendo de cada ojo y posteriormente tomar la decisión de mover o redimensionar dichos círculos al tamaño del ojo, este proceso se realizará independientemente en ambos ojos. La *Figura 13* muestra el diagrama de flujo para la implementación del módulo Ajuste de plantilla.

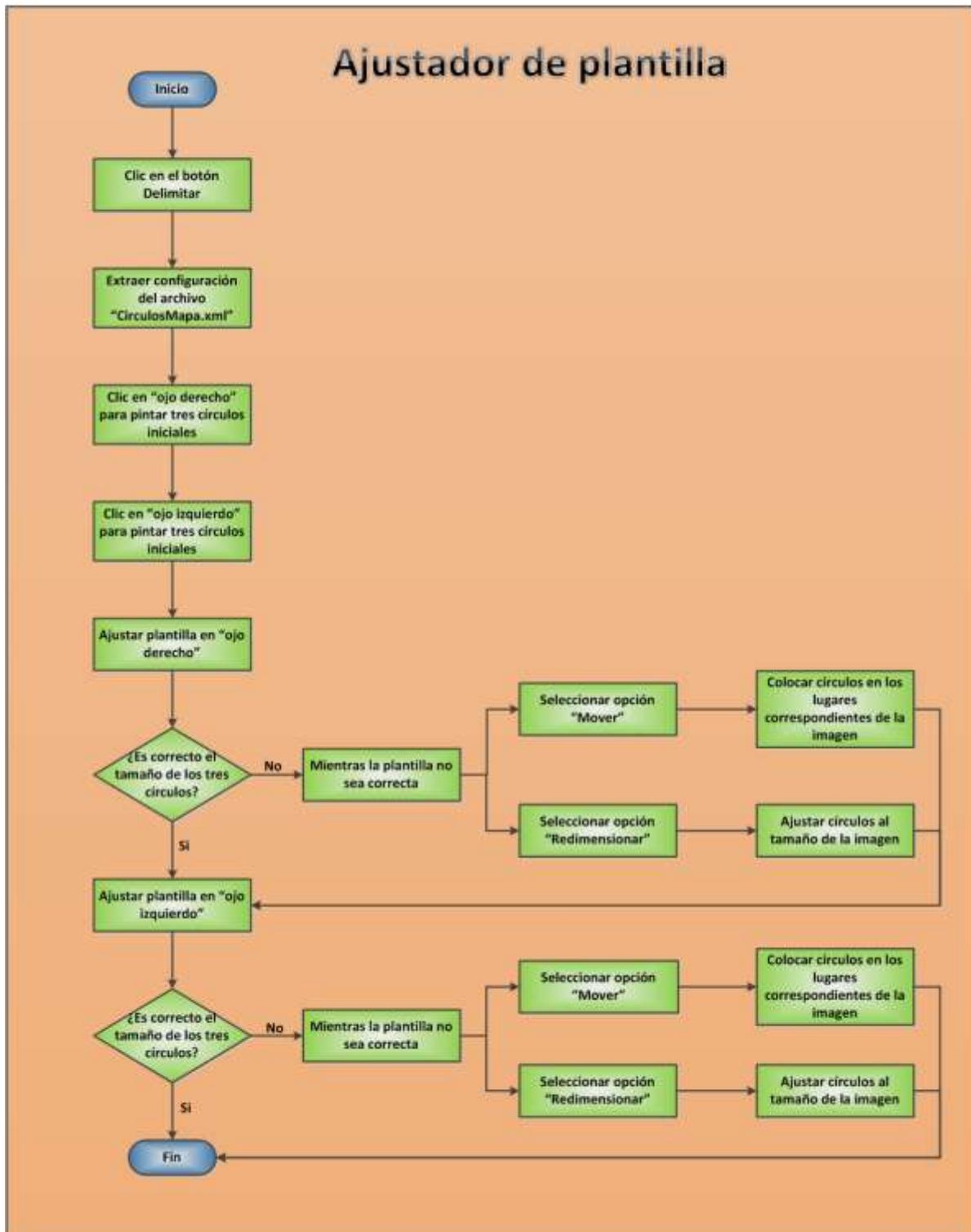


Figura 13. Diagrama de flujo Ajustador de plantilla.

### 4.1.4.1. Código (C#)

```
private void crearZona(PictureBox picture, int x, int y, ConfiguracionZona config)
{
    config.CirculosZona.ForEach((circulo) =>
    {
        if (circulo.LimiteInicial == null && circulo.LimiteFinal == null)
        {
            Circle c = new Circle();
            c.Name = circulo.Nombre;
            c.BorderColor = Color.FromName(circulo.ColorBorde);
            c.Width = c.Height = circulo.Diametro;
            c.Location = new Point(x - (c.Width / 2), y - (c.Height / 2));
            picture.Controls.Add(c);
        }
    });

    Circle.IsMoving = true;
}

protected override void OnMouseMove(System.Windows.Forms.MouseEventArgs e)
{
    if (isDragging)
    {
        if (IsMoving)
        {
            this.Top = this.Top + (e.Y - oldY);
            this.Left = this.Left + (e.X - oldX);
        }
        else if (isClimbing)
        {
            this.Width = e.X;
            this.Height = e.X;
        }
    }
    this.Cursor = Cursors.SizeAll;
}
```

## 4.1.5. Generador de mapa

El diagrama Generador de mapa, muestra el proceso para la creación del mapa Iridológico descrito en los archivos de configuración tanto de los círculos que delimitan dichas zonas, como de las zonas que se encuentran dentro del mapa. Este proceso se genera para ambos ojos de forma paralela. La *Figura 14* muestra el diagrama de flujo para la implementación del módulo **Generador de mapa**.

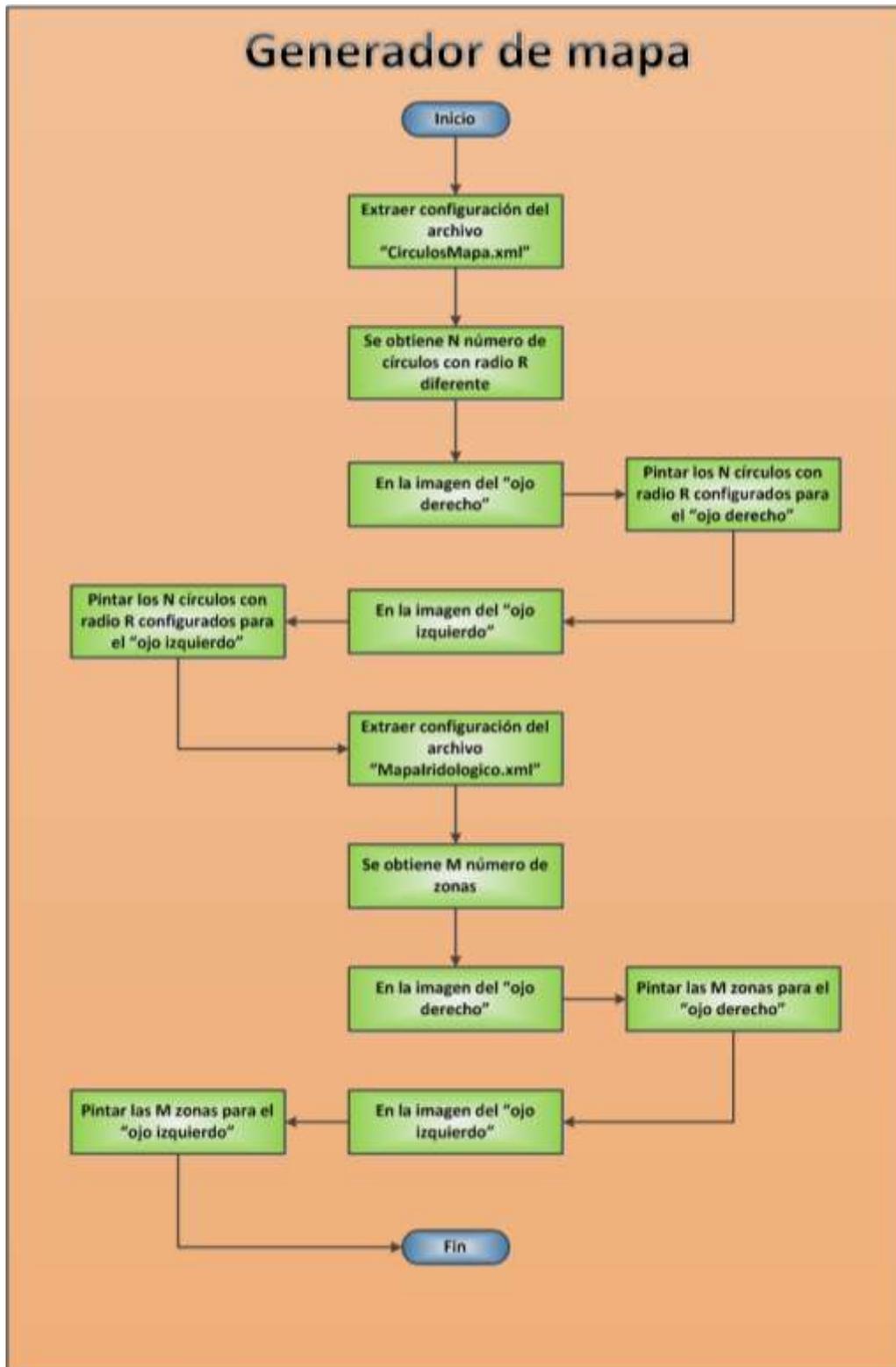


Figura 14. Diagrama de flujo Generador de mapa.

### 4.1.5.1. Código (C#)

```
private void generarMapaCirculos(PictureBox picture, ConfiguracionZona config)
{
    var distribucionCirculos = obtenerDistribucionCirculos(config);

    foreach (string key in distribucionCirculos.Keys)
    {
        string[] nombres = key.Split(new[] { "-" },
StringSplitOptions.RemoveEmptyEntries);
        Circle circuloInicial = (Circle)picture.Controls[nombres[0]];
        Circle circuloFinal = nombres.Length > 1 ?
(Circle)picture.Controls[nombres[1]] : (Circle)null;
        var diametroInicial = circuloInicial.Width;
        var diametroFinal = circuloFinal == null ? 0 : circuloFinal.Width;
        var factorProporcion = distribucionCirculos[key].NumeroCirculos + 1;
        var proporcion = Math.Abs((diametroFinal - diametroInicial) /
factorProporcion);
        var diametro = circuloFinal == null ? proporcion : diametroInicial +
proporcion;
        foreach (var nombre in distribucionCirculos[key].NombresCirculo)
        {
            Circle c = new Circle();
            var circuloZona = config.CirculosZona.Find(cir =>
cir.Nombre.Equals(nombre));
            c.Name = nombre;
            c.BorderColor = Color.FromName(circuloZona.ColorBorde);
            c.Width = c.Height = diametro;
            circuloFinal = circuloFinal == null ? circuloInicial : circuloFinal;
            c.Location = new Point(circuloFinal.Center.X - (c.Width / 2),
circuloFinal.Center.Y - (c.Height / 2));
            picture.Controls.Add(c);
            diametro += proporcion;
        }
    }
}

private void generarMapa(PictureBox picture, Mapa mapa, System.Windows.Forms.Timer
timerMapa, ConfiguracionZona config)
{
    List<Circle> circles = new List<Circle>();
    config.Zonas.ForEach((zona) =>
    {
        ZonaMapa zonaMapa = new ZonaMapa();
        zona.CirculosLimite.ForEach((circulo) =>
        {
            var c = (Circle)picture.Controls[circulo.Nombre];
            zonaMapa.CirculosLimite.Add(c);
        });
        zonaMapa.LineasConfig = zona.Lineas;
        zonaMapa.Nombre = zona.Nombre;
        mapa.ZonasMapa.Add(zonaMapa);
    });
    mapa.Generar(picture.Width, picture.Height);
    if (mapa.Lineas.Count > 0)
        timerMapa.Start();
}
```

4.1.6. Extractor de datos por zonas

El diagrama Extractor de datos por zonas, muestra el proceso que se lleva a cabo para la obtención de las mismas. En primera instancia, extrae la información de cada ojo y crea archivos de cada zona y de la totalidad del ojo obteniendo los pixeles. Este procedimiento se desarrolla en ambos ojos. La *Figura 15* muestra el flujo de implementación del módulo **Extractor de datos por zonas**.

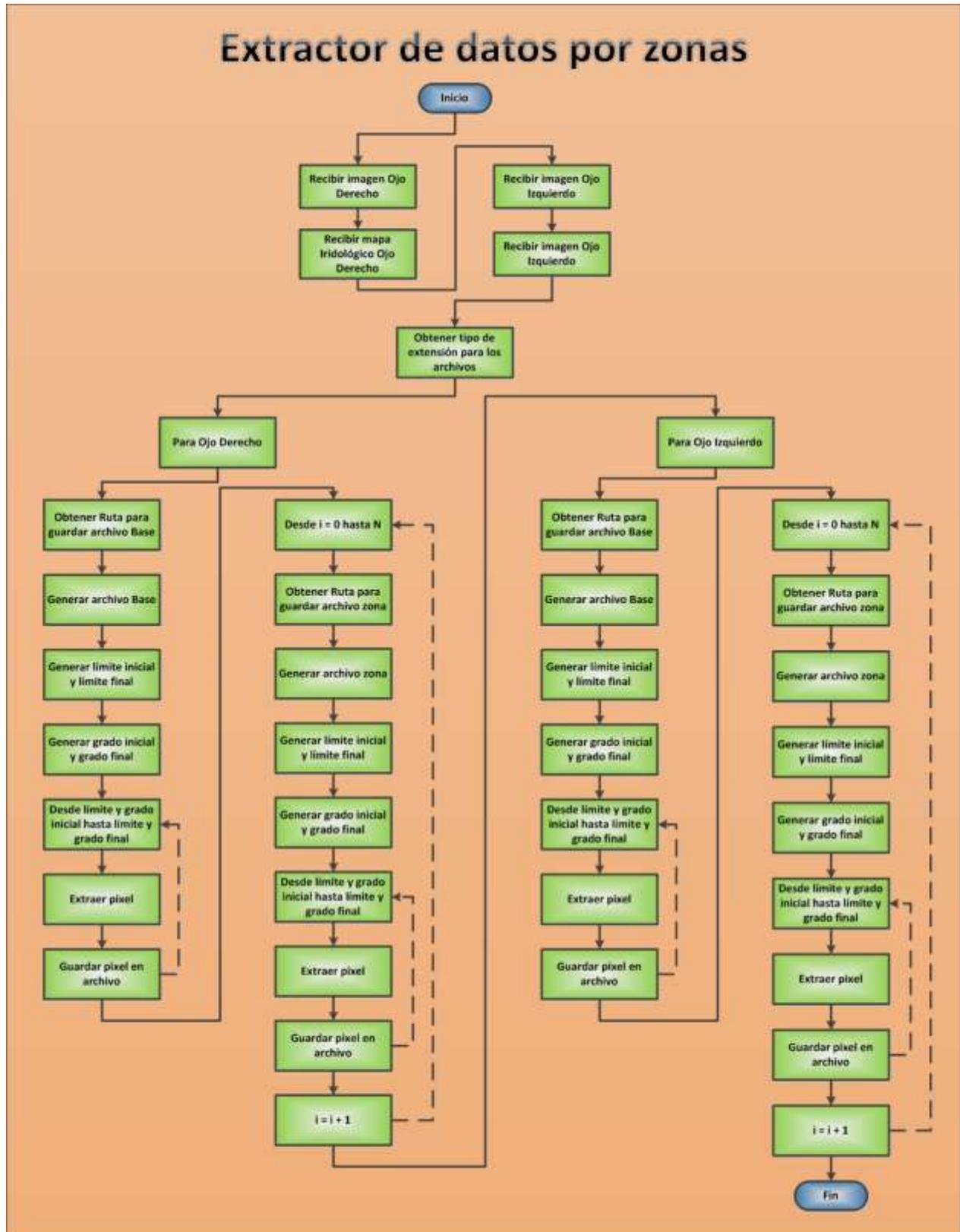


Figura 15. Diagrama de flujo Extracción de datos por zona.

### 4.1.6.1. Código (C#)

```
public void AnalizarImagen(int anchoZona, int alturaZona, Image imagenOjo, Estudio estudio)
{
    List<Zona> zonas = new List<Zona>();
    try
    {
        foreach (ZonaMapa zonaMapa in ZonasMapa)
        {
            if (zonaMapa.CirculosLimite.TrueForAll(c => c == null))
                continue;
            string strOjo = _ojo.ToString("G");
            StringBuilder nombreArchivo = new StringBuilder();
            var ruta = ConfigurationManager.AppSettings["Ruta"].ToString() +
estudio.IDEstudio.ToString() + "\\\" + strOjo + "\\\";
            var extension =
ConfigurationManager.AppSettings["ExtensionArchivo"].ToString();
            nombreArchivo.Append(ruta);
            nombreArchivo.Append(strOjo);
            nombreArchivo.Append(zonaMapa.Nombre);
            nombreArchivo.Append(extension);
            StreamWriter archivo = new StreamWriter(nombreArchivo.ToString());
            int gradosIniciales, gradosFinales;
            var lineaInicial = zonaMapa.LineasConfig[0];
            var lineaFinal = zonaMapa.LineasConfig[1];
            if (lineaFinal.Grados > lineaInicial.Grados)
            {
                gradosIniciales = lineaInicial.Grados;
                gradosFinales = lineaFinal.Grados;
            }
            else
            {
                gradosIniciales = lineaFinal.Grados;
                gradosFinales = lineaInicial.Grados;
            }

            for (int grados = gradosIniciales; grados <= gradosFinales; grados++)
            {
                LineaMapa lineaMapa = new LineaMapa(anchoZona, alturaZona,
zonaMapa.CirculosLimite, grados, lineaInicial.ColorBorde);
                zonaMapa.LineasParaAnalizar.Add(lineaMapa);
            }
            getPixeles(zonaMapa, imagenOjo, archivo);
            archivo.Close();
            Zona zona = new Zona()
            {
                ArchivoPixeles = nombreArchivo.ToString(),
                Descripcion = zonaMapa.Nombre,
            };
            zonas.Add(zona);
        }
        ProcesoAnalisis procesoAnalisis = new ProcesoAnalisis();
        procesoAnalisis.AnalizarZonas(zonas, estudio, _ojo, imagenOjo);
    }
    catch (Exception ex)
    {
        throw new Exception("No se puede generar completamente el análisis de la
imagen", ex);
    }
}
```

```
private void getPixeles(ZonaMapa zona, Image imagenOjo, StreamWriter archivo)
{
    Bitmap bitmapOjo = new Bitmap(imagenOjo);
    foreach (LineaMapa lineaPorAnalizar in zona.LineasParaAnalizar)
    {
        int xInicio, xFin;
        Point A = lineaPorAnalizar.A;
        Point B = lineaPorAnalizar.B;
        if (lineaPorAnalizar.B.X > lineaPorAnalizar.A.X)
        {
            xInicio = A.X;
            xFin = B.X;
        }
        else
        {
            xInicio = B.X;
            xFin = A.X;
        }
        for (int x = xInicio; x <= xFin; x++)
        {
            if (B.X == A.X)
                continue;
            var m = (B.Y - A.Y) / (B.X - A.X);
            var y = (m * (x - A.X)) + A.Y;

            int argb = bitmapOjo.GetPixel(x, y).ToArgb();
            archivo.Write(argb);
            archivo.Write("|");
        }
        archivo.WriteLine();
    }
}
```

4.1.7. Analizador cromático por zonas

El diagrama del módulo **Analizador cromático por zonas**, muestra el procedimiento para obtener la media de los colores de cada zona, convirtiéndolo en promedio y obteniendo la desviación estándar, posteriormente contabiliza los datos que están dentro y fuera del rango. Finalmente se calculan promedios de los rangos obtenidos anteriormente. La *Figura 16* muestra el flujo de implementación del módulo **Analizador cromático por zonas**.

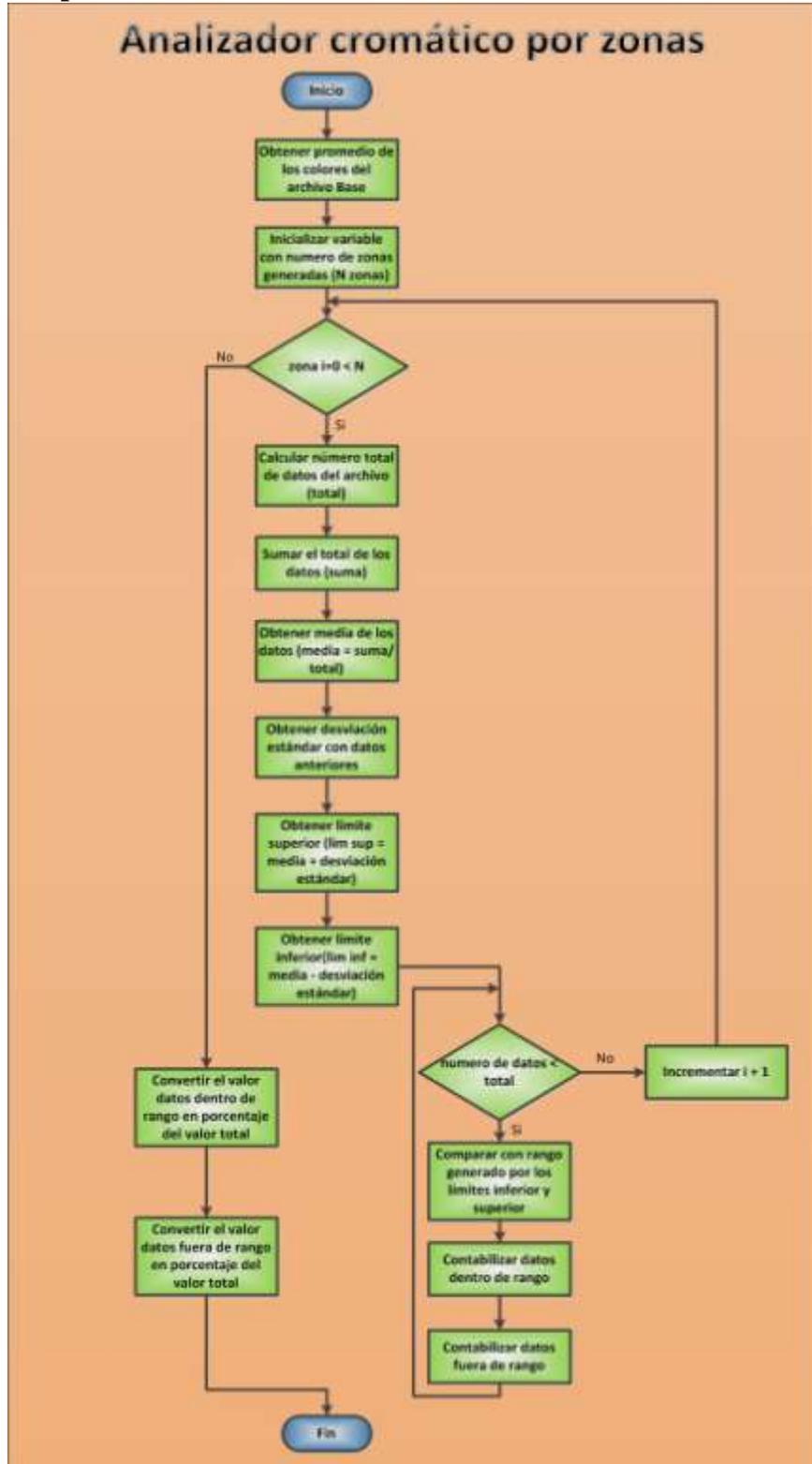


Figura 16. Diagrama de flujo Analizador cromático por zonas.

## 4.1.7.1. Código (C#)

```
public void ObtenerResultadoPorCalculo(Zona zona)
{
    try
    {
        var lines = File.ReadAllLines(zona.ArchivoPixeles);
        double n = 0;
        double suma = 0;
        double sumaVarianza = 0;
        lines.ToList().ForEach((l) =>
        {
            var data = l.Split(new[] { "|" }, StringSplitOptions.RemoveEmptyEntries);
            n += data.Length;
        });
        lines.ToList().ForEach((l) =>
        {
            var data = l.Split(new[] { "|" }, StringSplitOptions.RemoveEmptyEntries);
            data.ToList().ForEach((d) =>
            {
                long number = (-1) * long.Parse(d);
                suma += number;
            });
        });
        var media = suma / n;
        lines.ToList().ForEach((l) =>
        {
            var data = l.Split(new[] { "|" }, StringSplitOptions.RemoveEmptyEntries);
            data.ToList().ForEach((d) =>
            {
                long number = (-1) * long.Parse(d);
                sumaVarianza += Math.Pow(number - media, 2);
            });
        });
        var varianza = sumaVarianza / n;
        var desviacionEstandar = Math.Sqrt(varianza);
        var rangoInicial = media - desviacionEstandar;
        var rangoFinal = media + desviacionEstandar;
        double numerosFueraDeRango = 0;
        double numerosDentroDeRango = 0;
        lines.ToList().ForEach((l) =>
        {
            var data = l.Split(new[] { "|" }, StringSplitOptions.RemoveEmptyEntries);
            data.ToList().ForEach((d) =>
            {
                long number = (-1) * long.Parse(d);
                if (number >= rangoInicial && number <= rangoFinal)
                    numerosDentroDeRango++;
                else
                    numerosFueraDeRango++;
            });
        });
        ResultadoZona resultadoZona = new ResultadoZona()
        {
            PorcentajeDentroDeRango = (numerosDentroDeRango * 100) / n,
            PorcentajeFueraDeRango = (numerosFueraDeRango * 100) / n,
            Media = media
        };
        this.InsertarResultadoZona(resultadoZona);
        zona.IDResultadoZona = resultadoZona.IDResultadoZona;
    }
    catch (Exception ex)
    {
        throw new Exception(string.Format("Error al calcular el resultado de zona"), ex);
    }
}
```

4.1.8. Analizador Iridológico

El diagrama del **Analizador Iridológico** muestra el proceso para la elección del tipo de afectación que existe en cada zona, dependiendo de los rangos agregados. Para saber el rango de afectación se debe de comparar la media generada por cada ojo contra la media de cada zona. Este proceso se realiza para cada ojo con sus respectivas zonas. La *Figura 17* muestra el diagrama de flujo del módulo Analizador Iridológico.

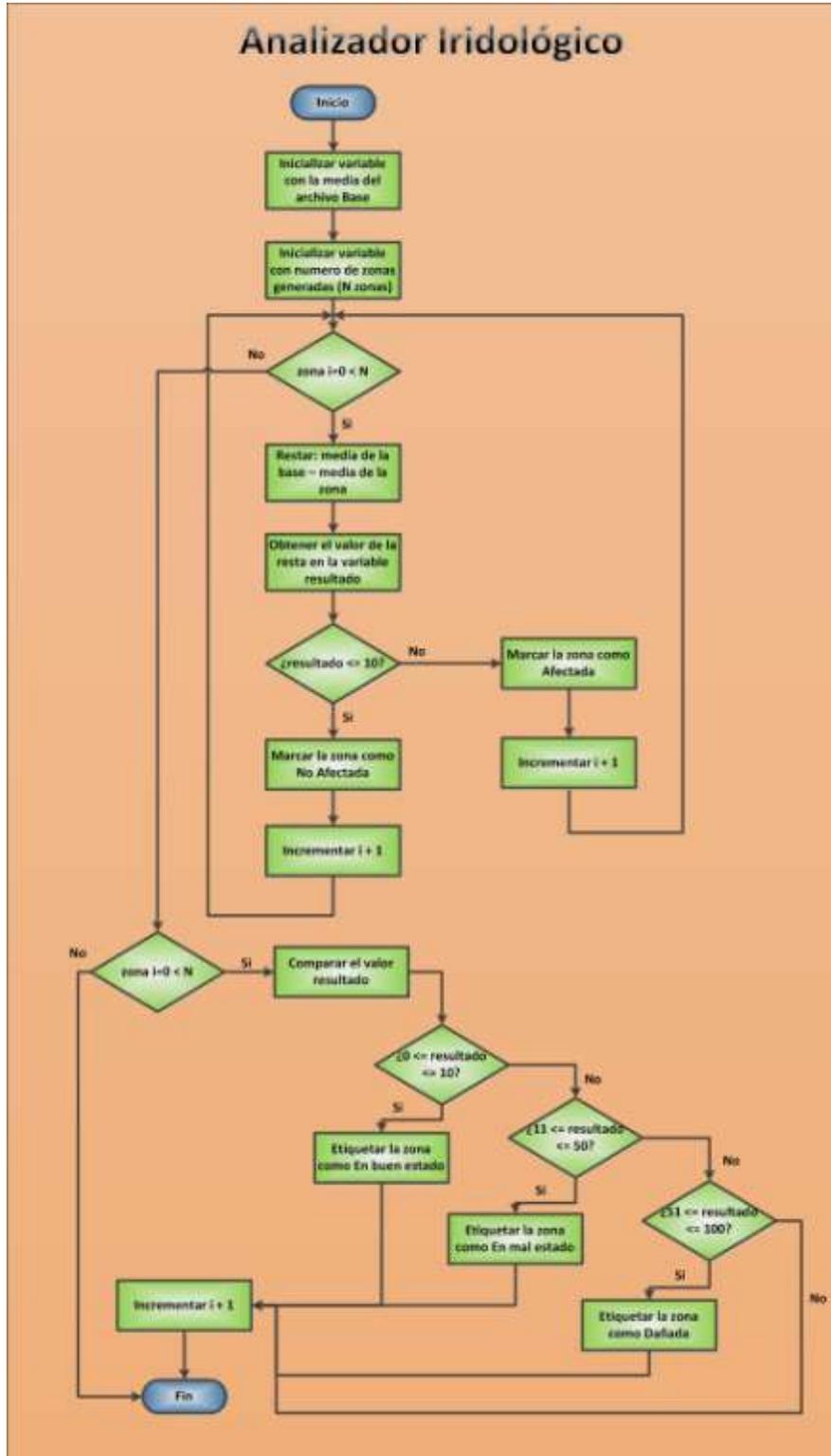


Figura 17. Diagrama de flujo Analizador Iridológico.

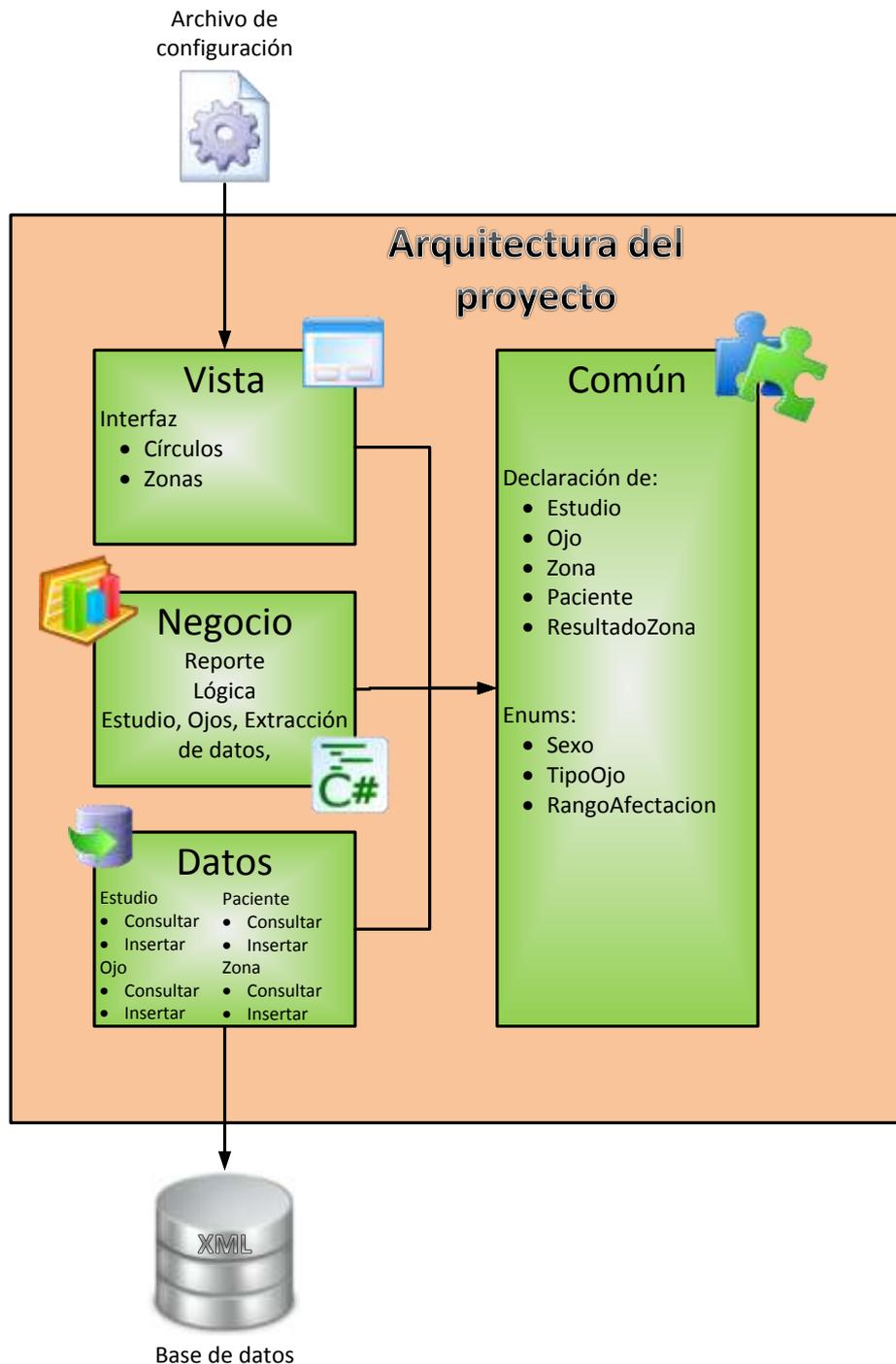
### 4.1.8.1. Código (C#)

```
private void analizarColor(List<Zona> zonas)
{
    var resultadoZonaDao = new ResultadoZonaDAO();
    var rangoAfectacionDao = new RangoAfectacionDAO();
    var zonaBase = zonas.FirstOrDefault(zona => zona.Descripcion.Equals("BASE"));
    var porcentajeDiferenciaMedia =
Convert.ToInt32(ConfigurationManager.AppSettings["PorcentajeDiferenciaMedia"]);
    var resultadoBase =
resultadoZonaDao.ObtenerResultadoZona(zonaBase.IDResultadoZona);
    var rangosAfectacion = rangoAfectacionDao.ObtenerRangosAfectacion();
    zonas.ForEach((zona) =>
    {
        if(!zona.Descripcion.Equals("BASE"))
        {
            var resultadoZona =
resultadoZonaDao.ObtenerResultadoZona(zona.IDResultadoZona);
            var diferencia = 100 - ((resultadoZona.Media * 100) /
resultadoBase.Media);
            if(diferencia <= porcentajeDiferenciaMedia)
            {
                rangosAfectacion.ForEach((rango) =>
                {
                    if (resultadoZona.PorcentajeFueraDeRango >=
rango.LimiteInicial &&
                    resultadoZona.PorcentajeFueraDeRango <=
rango.LimiteFinal)
                        zona.IDRangoAfectacion = rango.IDRangoAfectacion;
                });
                zona.PorcentajeAfectacion =
(int)resultadoZona.PorcentajeFueraDeRango;
            }
            else
            {
                rangosAfectacion.ForEach((rango) =>
                {
                    if (resultadoZona.PorcentajeDentroDeRango >=
rango.LimiteInicial &&
                    resultadoZona.PorcentajeDentroDeRango <=
rango.LimiteFinal)
                        zona.IDRangoAfectacion = rango.IDRangoAfectacion;
                });
                zona.PorcentajeAfectacion =
(int)resultadoZona.PorcentajeDentroDeRango;
            }
        }
    });
}
```

## 4.2. Arquitectura SCADE-Iris

La Arquitectura desarrollada para SCADE-Iris esta conformada por 4 bloques, el primero es la Vista, el cual contiene toda la interfaz de usuario, la definición de los círculos y las zonas, esto, alimentado por unos archivos de configuración. Posteriormente se tiene el bloque de Negocio, el cuál contiene toda la lógica utilizada para la extracción de las zonas y la información que se necesita para plasmarla en la base de datos. Existe además la capa de Datos, donde se almacena la información que se va a ingresar a la Base de Datos. Finalmente se creó una capa Común, donde se declaran los objetos que harán el mapeo de información hacia la Base de Datos.

La *Figura 18* muestra la Arquitectura diseñada para la implementación del SCADE-Iris:



**Figura 18.** Arquitectura de SCADE-Iris.

### 4.3. Diagrama de Clases del SCADE-Iris

El Diagrama de clases describe la estructura a nivel de clases (objetos) del SCADE-Iris, atributos y las relaciones entre ellos, su objetivo de este diagrama es mostrar el diseño conceptual de la información que maneja el sistema. La *Figura 19* muestra el Diagrama de Clases utilizado para el funcionamiento del SCADE-Iris:

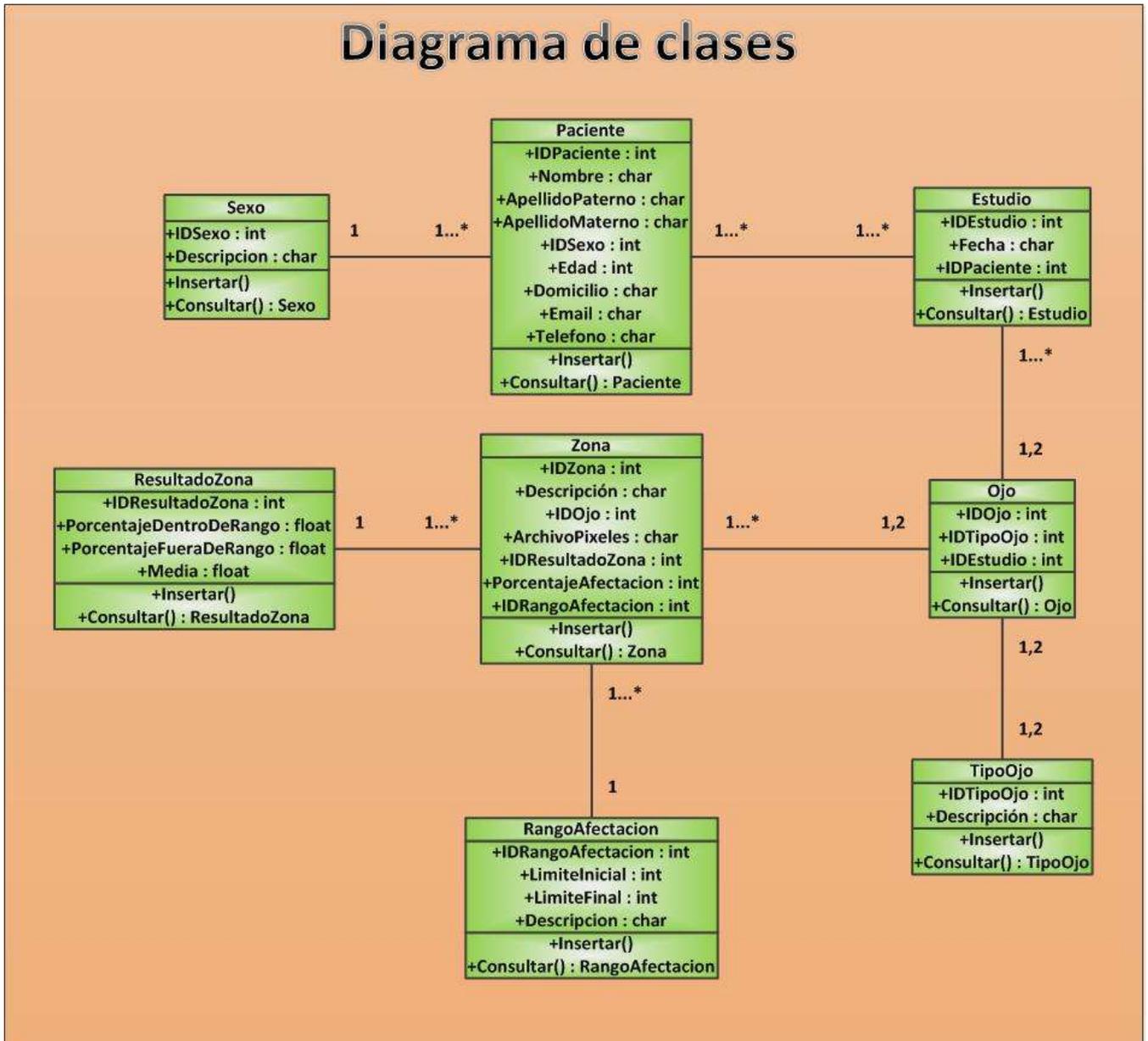


Figura 19. Diagrama de clases de SCADE-Iris.

4.4. Diagrama Entidad – Relación (Base de Datos)

El modelo Entidad - Relación representa las entidades relevantes del sistema SCADE-Iris, así como su interrelación entre ellas y sus propiedades que se almacenan en la base de datos.

La *Figura 20* muestra el Diagrama Entidad – Relación que almacena la información generada por el SCADE-Iris:

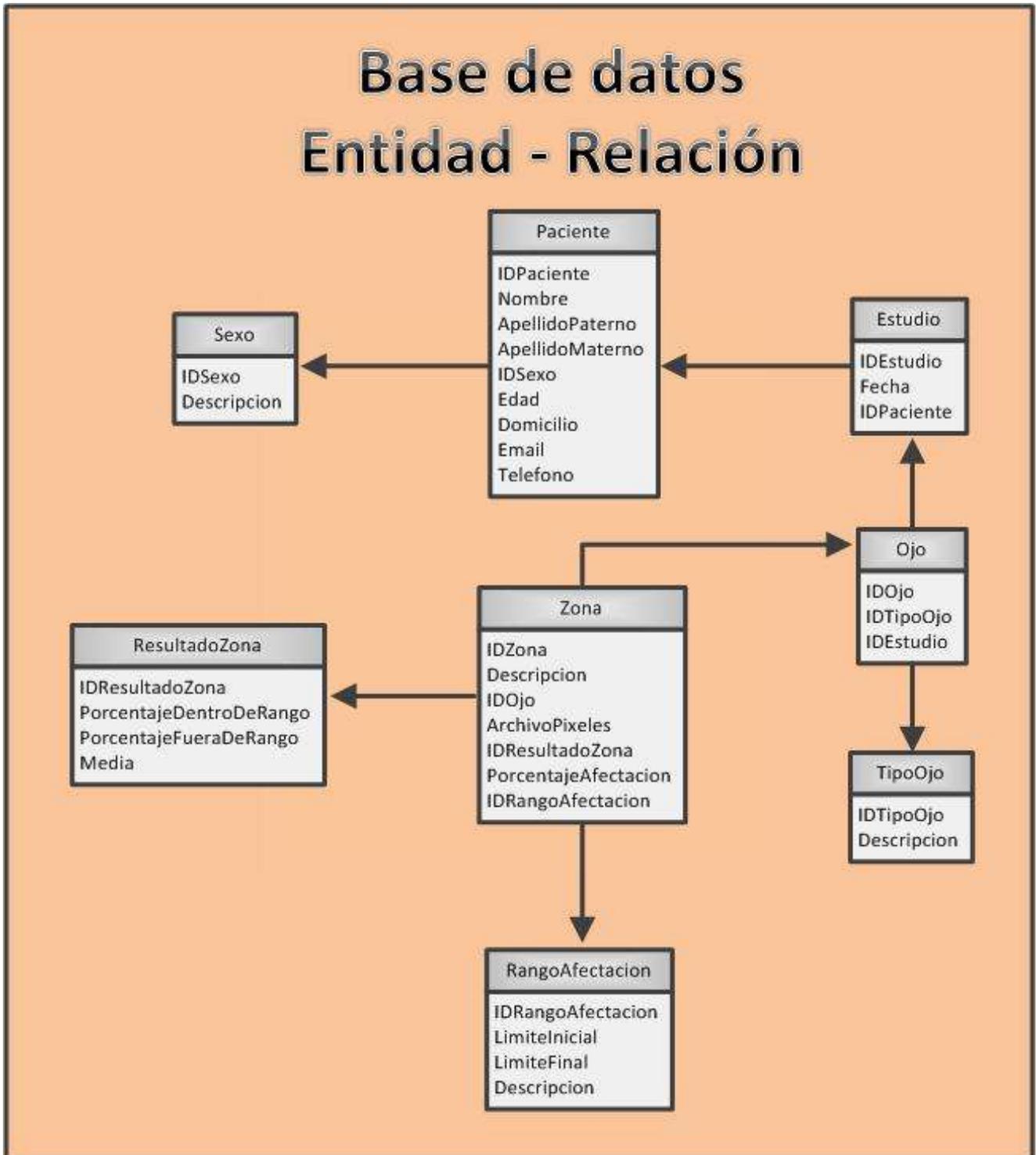


Figura 20. Diagrama Entidad-Relación de Base de Datos.

## CAPÍTULO 5. PRUEBAS Y RESULTADOS

Este capítulo describe las pruebas realizadas y los resultados obtenidos durante el desarrollo de las mismas. Lo que se pretende es mostrar los resultados de la implementación representando lo que se definió durante la elaboración de la propuesta. Cada prueba será representada con imágenes que fueron obtenidas durante la ejecución del sistema.

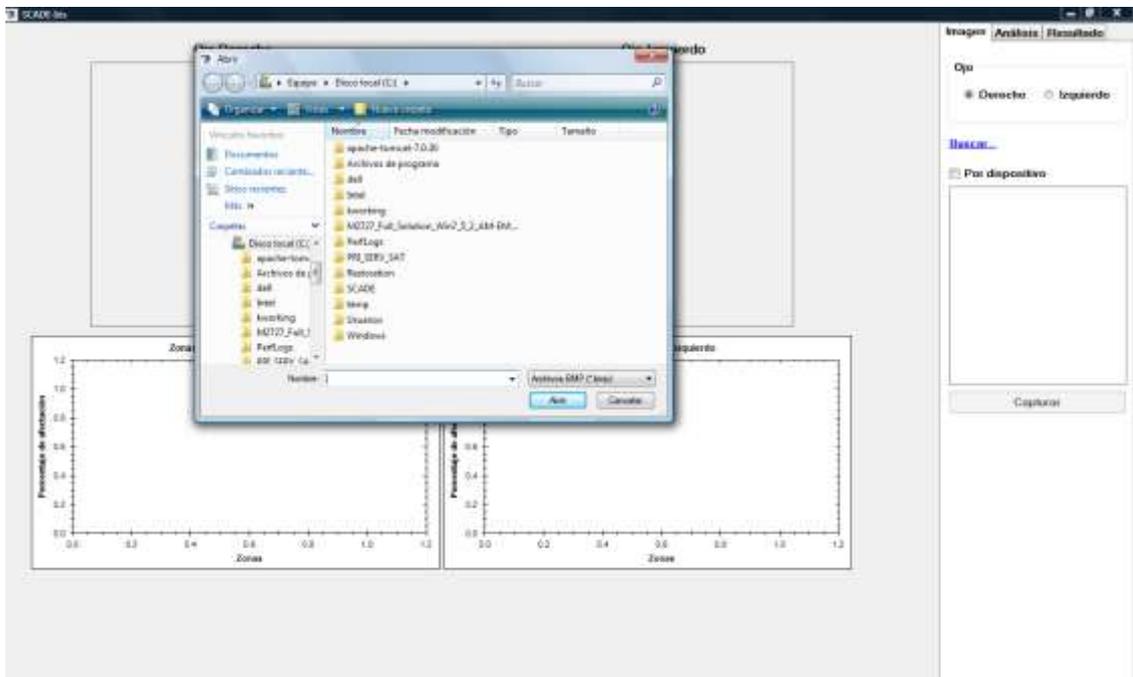
Las pruebas que se muestran a continuación, cabe aclarar que se tomaron con una cámara de 12 mega *pixeles*, por lo que la definición de las imágenes no es la ideal para obtener un resultado exacto. Para ello se recomienda utilizar una cámara profesional especializada en la toma de fotografías del iris.

Se realizaron 8 pruebas a distintas personas incluyendo 2 niños.

### 5.1. Prueba 1

Al usuario (mujer de 57 años) a quien se le realizó la primera prueba, se le complicó inicialmente operar el sistema. Posteriormente dicho usuario comenzó a explorarlo y se percató que no era muy difícil su manipulación. Después de un tiempo y gracias al uso del Manual de Usuario comenzó a colocar la imagen del ojo derecho y posteriormente la imagen del ojo izquierdo.

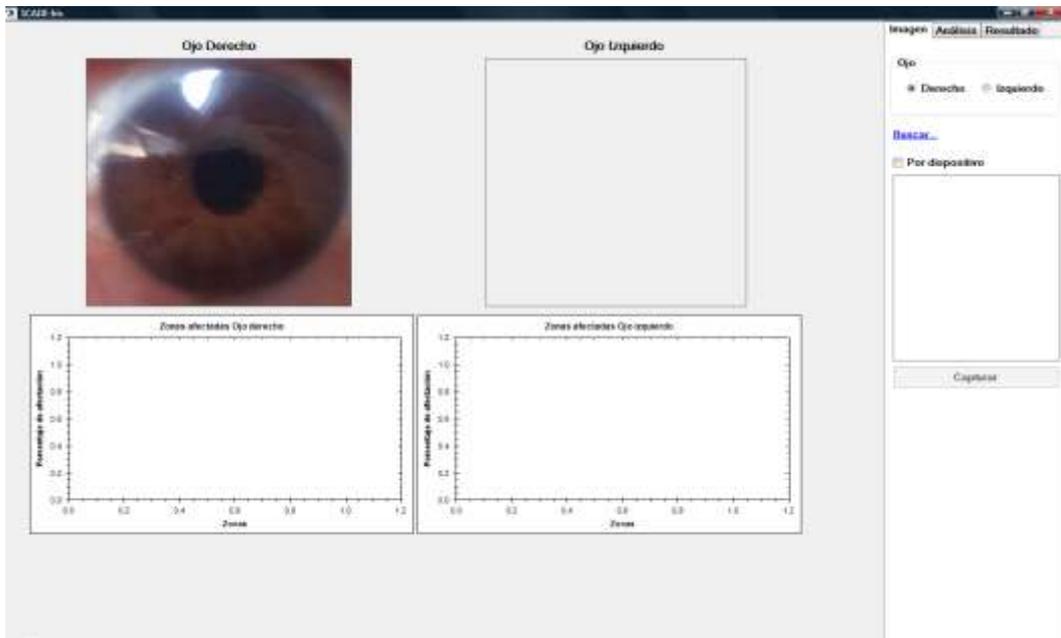
La *Figura 21* muestra el proceso de buscar el archivo del ojo derecho para poderla visualizar dentro del área de visualización.



**Figura 21.** Pantalla donde se visualiza la búsqueda del archivo del ojo derecho.

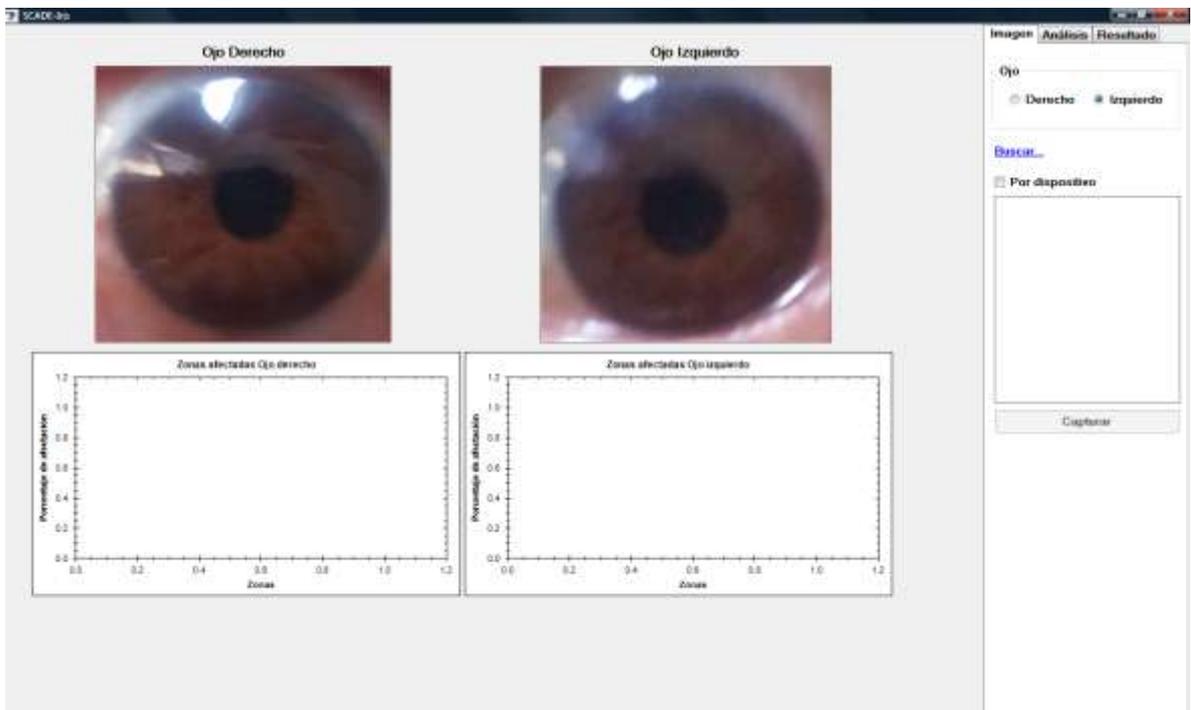
## SCADE-Iris

Al localizar su archivo y seleccionarlo, el usuario dio *clic* en Abrir y se mostró la imagen dentro del Área de visualización y análisis. La *Figura 22* muestra dicho proceso.



**Figura 22.** Pantalla mostrando la primera imagen seleccionada.

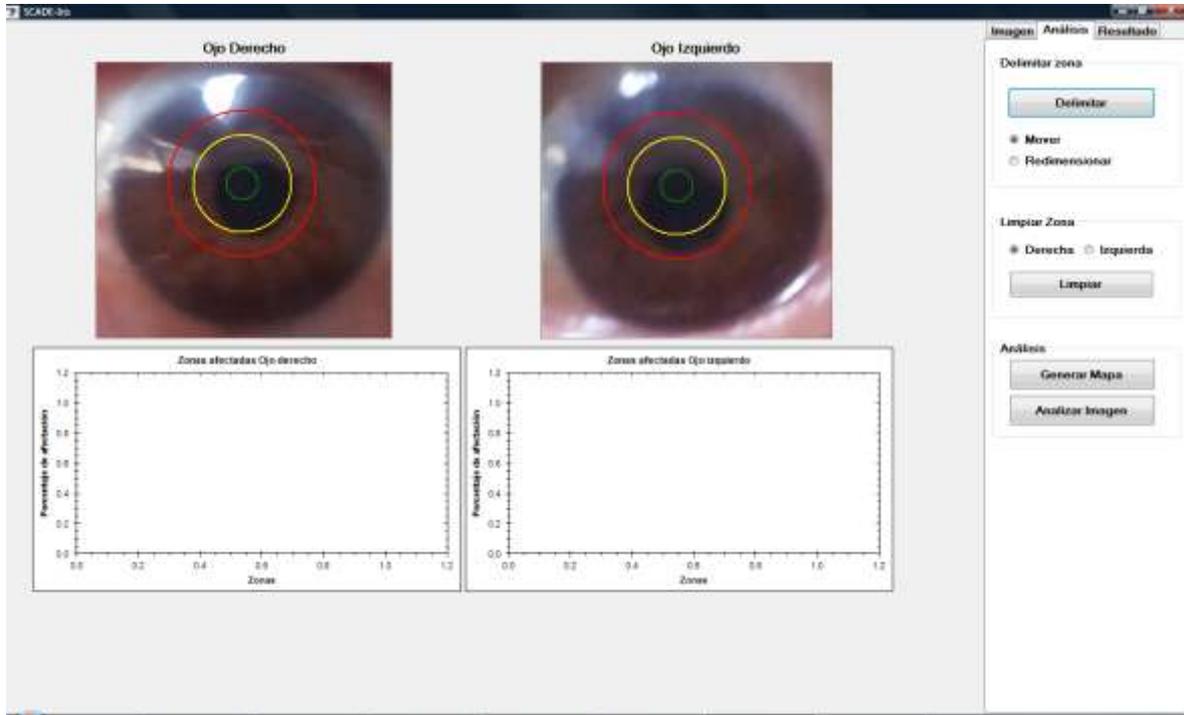
Ya que realizó el primer proceso del programa, continuó realizando lo mismo para el ojo izquierdo, seleccionando primero la opción de ojo izquierdo. La *Figura 23* muestra ya las dos imágenes posicionadas dentro del sistema.



**Figura 23.** Pantalla mostrando las dos imágenes dentro del sistema.

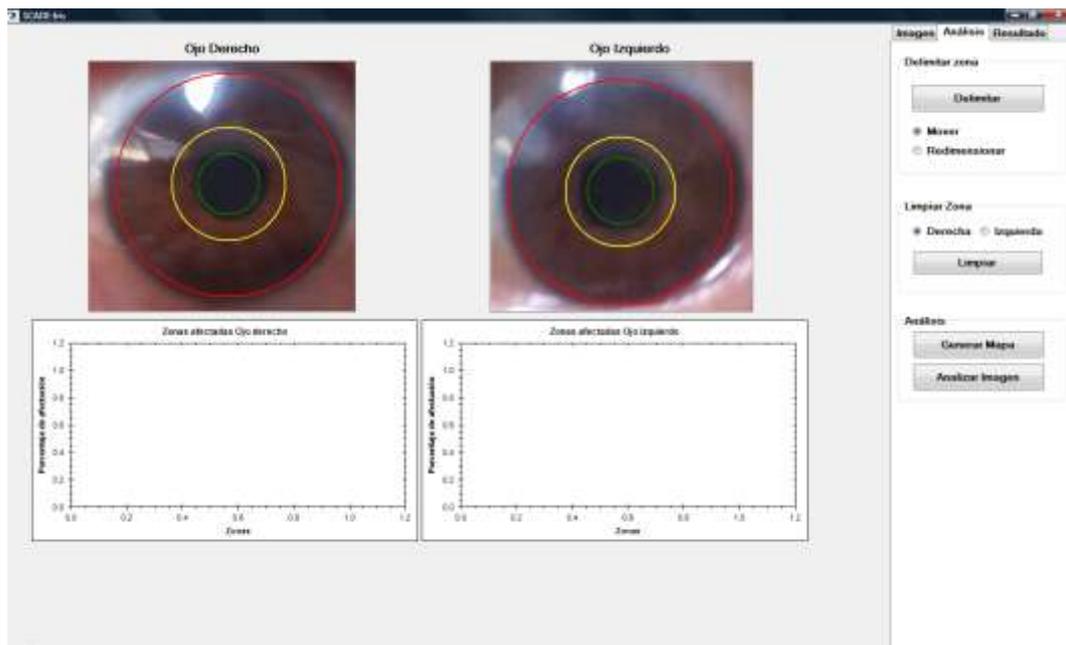
## SCADE-Iris

Después el usuario siguió con la pestaña de Análisis donde en la sección Delimitar zonas, se le explicó como utilizarla, ya que surgieron muchas dudas en esa etapa. Se le explicó que primero se debe hacer *clic* en el botón Delimitar y posteriormente hacer *clic* sobre cada una de las imágenes para que aparezcan los 3 círculos requeridos. La *Figura 24* muestra los círculos generados a partir del clic sobre la imagen.



**Figura 24.** Pantalla que muestra los círculos que delimitaran el ojo.

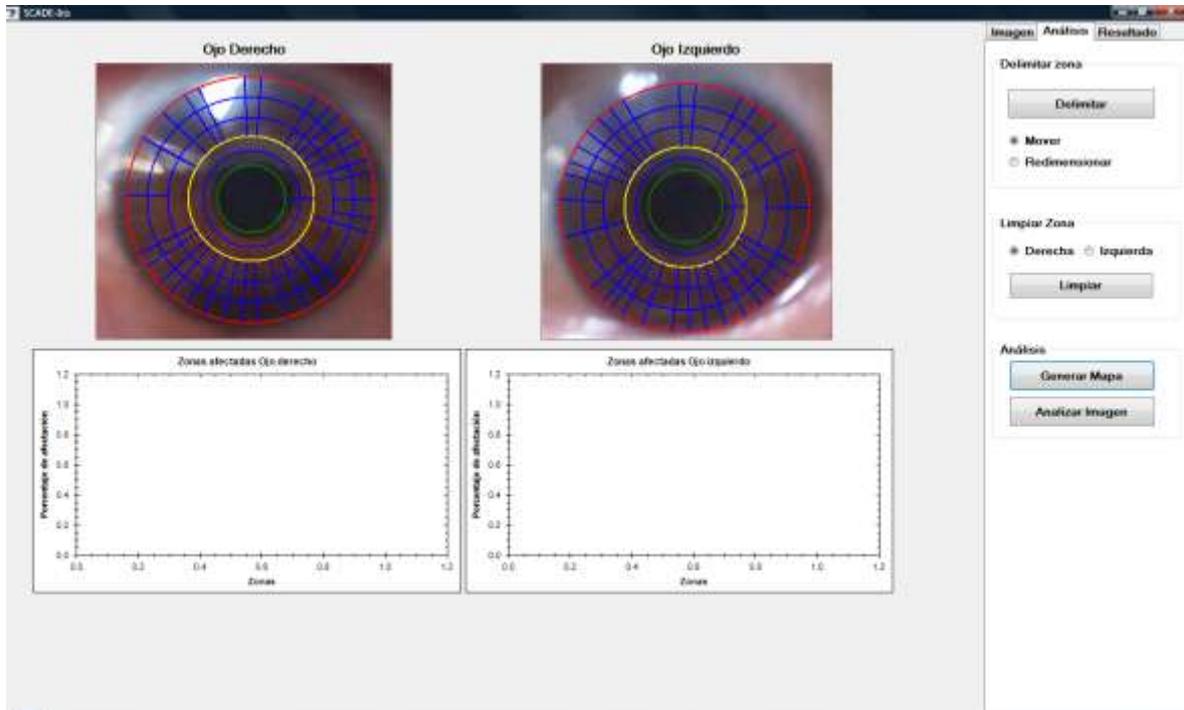
Continuando, se le explicó al usuario cómo mover y redimensionar dichos círculos. Se logró que al usuario le fuera fácil el acomodo de dichas plantillas mediante la práctica. La *Figura 25* muestra los ojos delimitados.



**Figura 25.** Pantalla mostrando los ojos delimitados.

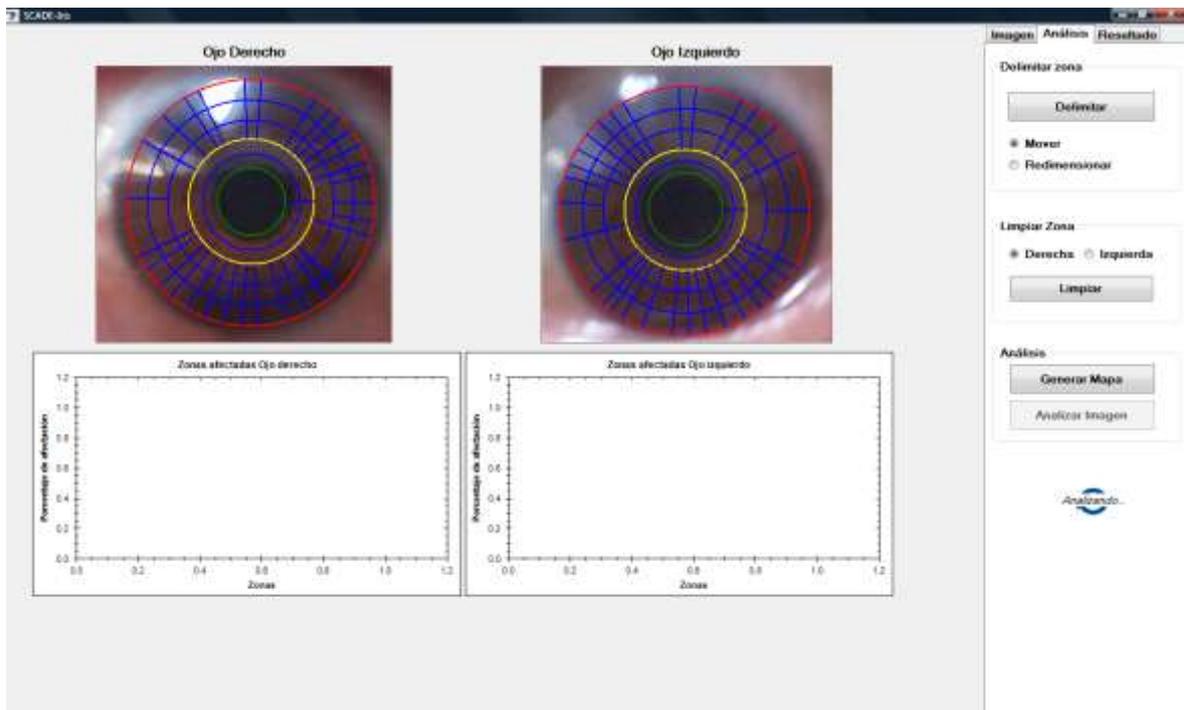
## SCADE-Iris

Una vez que el usuario concluyó con la delimitación, hizo mención si lo que continuaba era la generación del mapa, y se indicó que en efecto era el paso siguiente, y bastaba con hacer *clic* en el botón Generar Mapa. La *Figura 26* muestra el mapa generado sobre los ojos.



**Figura 26.** Pantalla mostrando el mapa para delimitar las zonas.

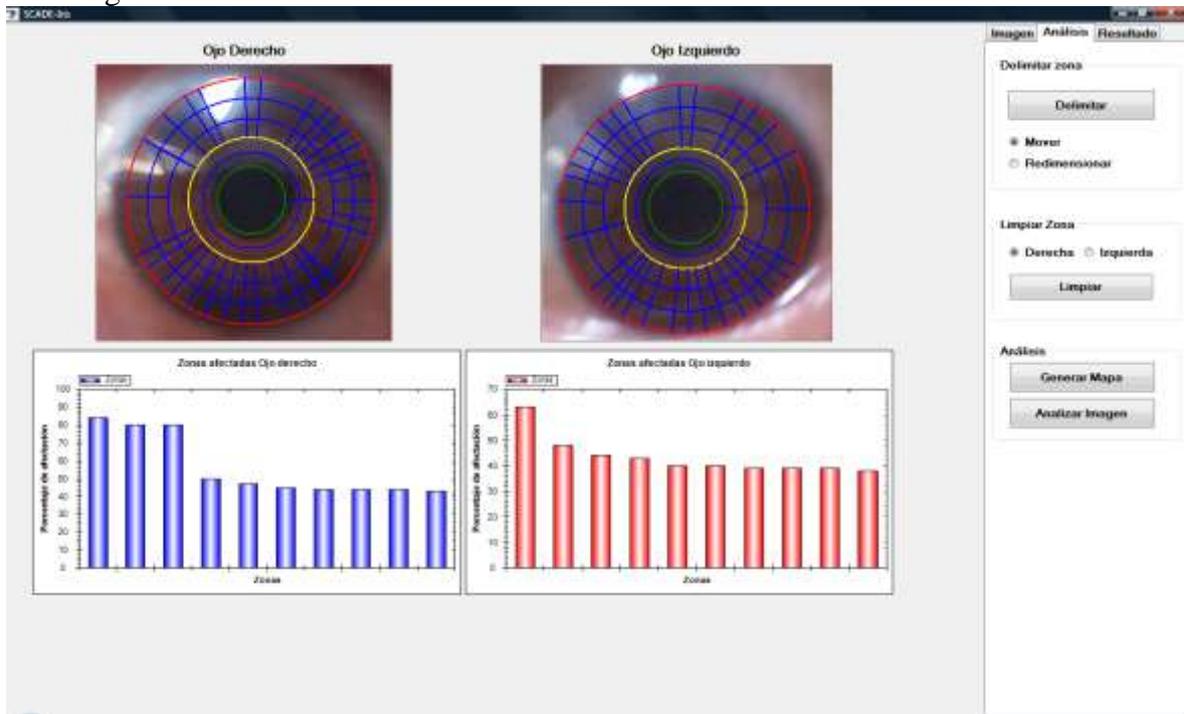
Mientras el mapa se generaba, hubo un tiempo de espera. Al finalizar, se hizo *clic* sobre el botón Analizar Imagen. La *Figura 27* muestra la imagen del proceso de análisis.



**Figura 27.** Pantalla mostrando el proceso de análisis.

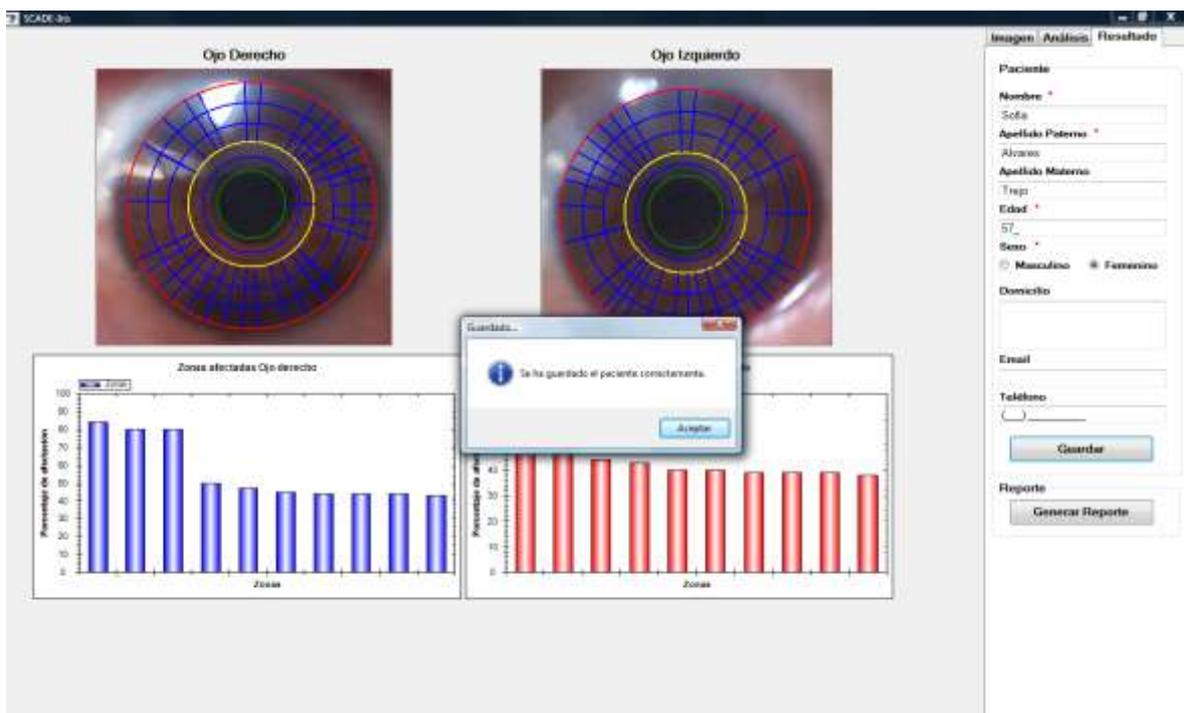
## SCADE-Iris

Una vez que se tuvieron los mapas generados, las gráficas de resultados fueron mostradas en cada uno de los ojos. Para observar los resultados, se indicó al usuario que debía colocar el puntero del *mouse* sobre cada una de las barras para poder visualizar los órganos que presentan mayor afectación. La *Figura 28* muestra las gráficas dentro del sistema.



**Figura 28.** Pantalla mostrando las gráficas generadas para cada ojo.

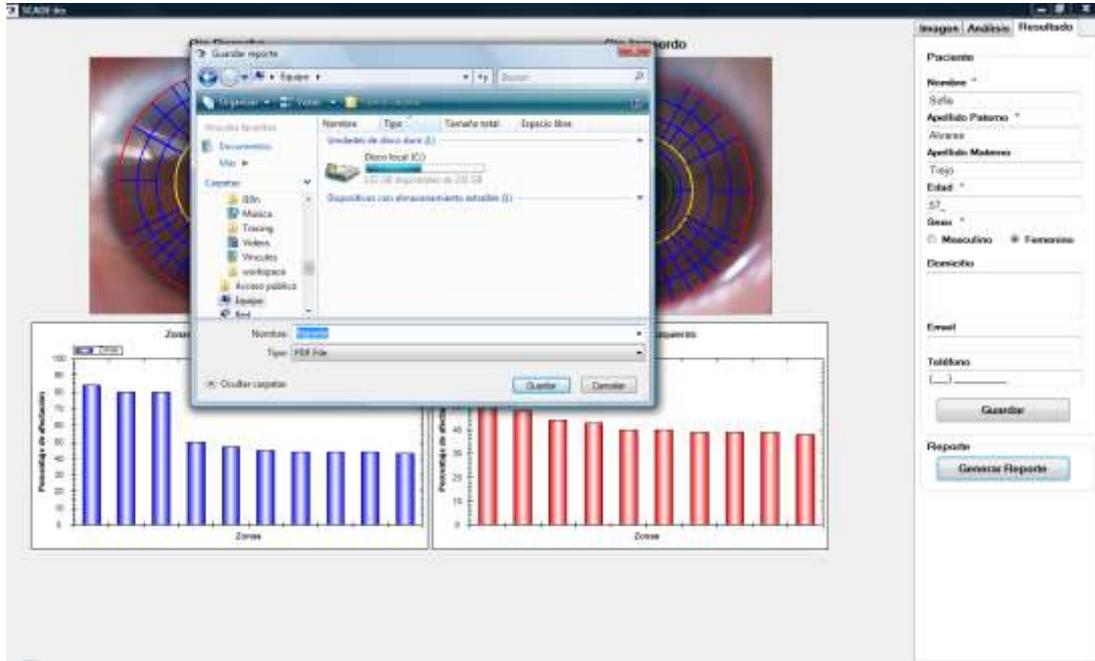
El usuario cuestionó a cerca del uso de la pestaña Resultados, y se explicó que esta es utilizada para guardar la información del paciente con el respectivo análisis ejecutado. Se pidió hiciera *click* en la pestaña Resultado para ingresar su información personal. Al finalizar, se pidió que hiciera *click* sobre el botón Guardar. La *Figura 29* muestra el cuadro de dialogo de paciente guardado.



**Figura 29.** Pantalla mostrando el cuadro de dialogo de paciente guardado.

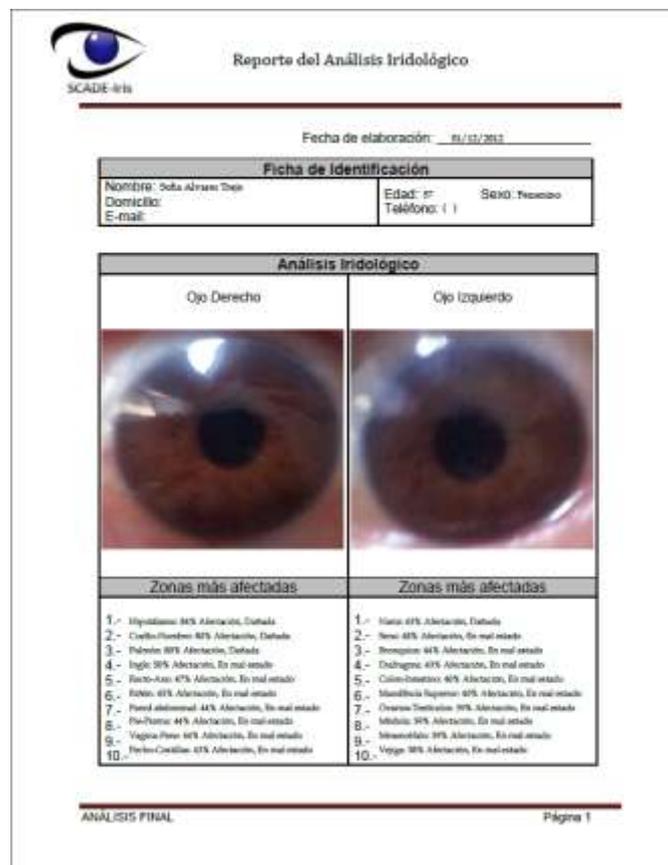
## SCADE-Iris

El paso siguiente fue hacer *clic* sobre el botón de Generar Reporte para mostrar el explorador de ventanas de Windows y elegir la ruta para guardar dicho reporte. La *Figura 30* muestra la ruta donde se desea guardar el archivo y asignarle un nombre.



**Figura 30.** Pantalla mostrando la ventana de guardado de archivo.

La *Figura 31* muestra el reporte creado por el sistema con los datos e imágenes analizadas.



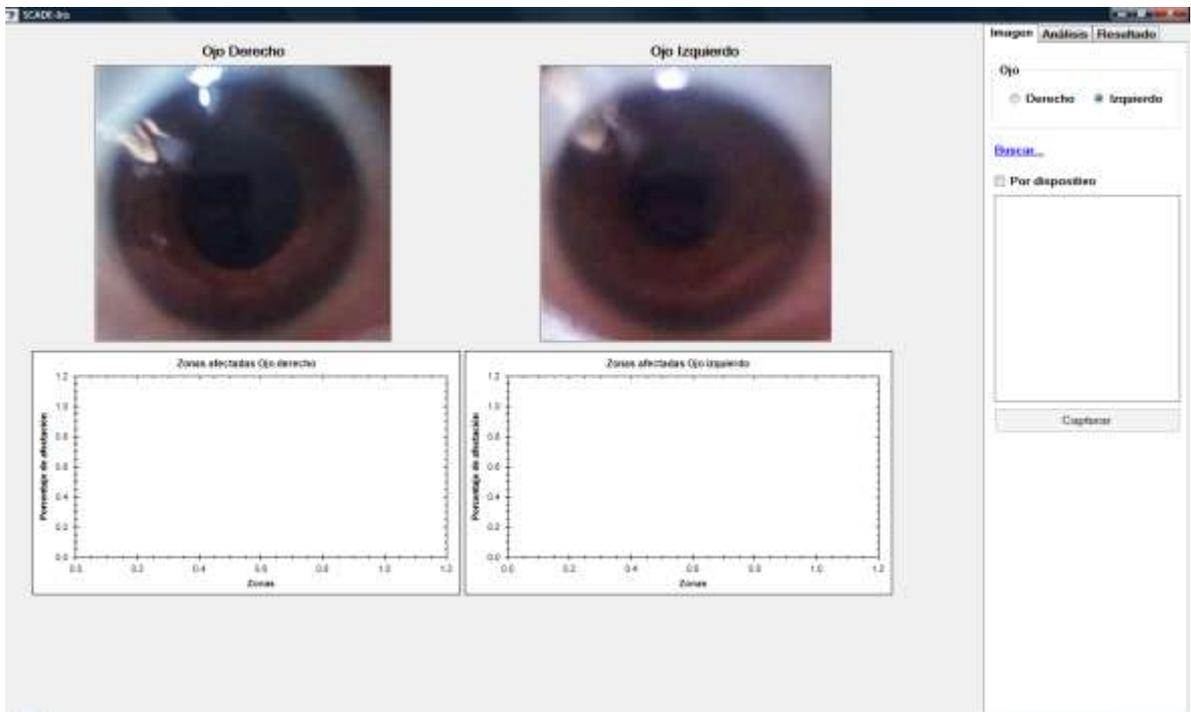
**Figura 31.** Reporte correspondiente a la prueba 1.

## 5.1.1. Resultados de la prueba

Los resultados obtenidos fueron exitosos y sorprendentes para el usuario, ya que no estaba acostumbrado a este tipo de sistemas ni análisis. Con los resultados presentados dentro del reporte en la parte de zonas más afectadas, el usuario se sorprendió debido a que en efecto presentaba esos órganos con afectación.

## 5.2. Prueba 2

La prueba 2 presentó menos complicación, ya que se le realizó a un usuario (mujer de 26 años) que tiene más conocimiento del manejo de una computadora. La mayor parte del proceso de análisis la dedujo el usuario. Sólo al principio se le comentó una pequeña introducción sobre el uso del sistema. La *Figura 32* muestra las imágenes ya cargadas dentro del sistema.



**Figura 32.** Pantalla mostrando los ambos ojos tomados.

Posteriormente se colocó en la pestaña de Análisis, aquí pidió que se le explicará que hacer. Se le indicó el proceso de Delimitación para poder acomodar la plantilla con los 3 círculos en cada ojo, tomando en cuenta el movimiento y la redimensión del ojo. La *Figura 33* muestra la delimitación de los círculos respecto a los ojos.

## SCADE-Iris

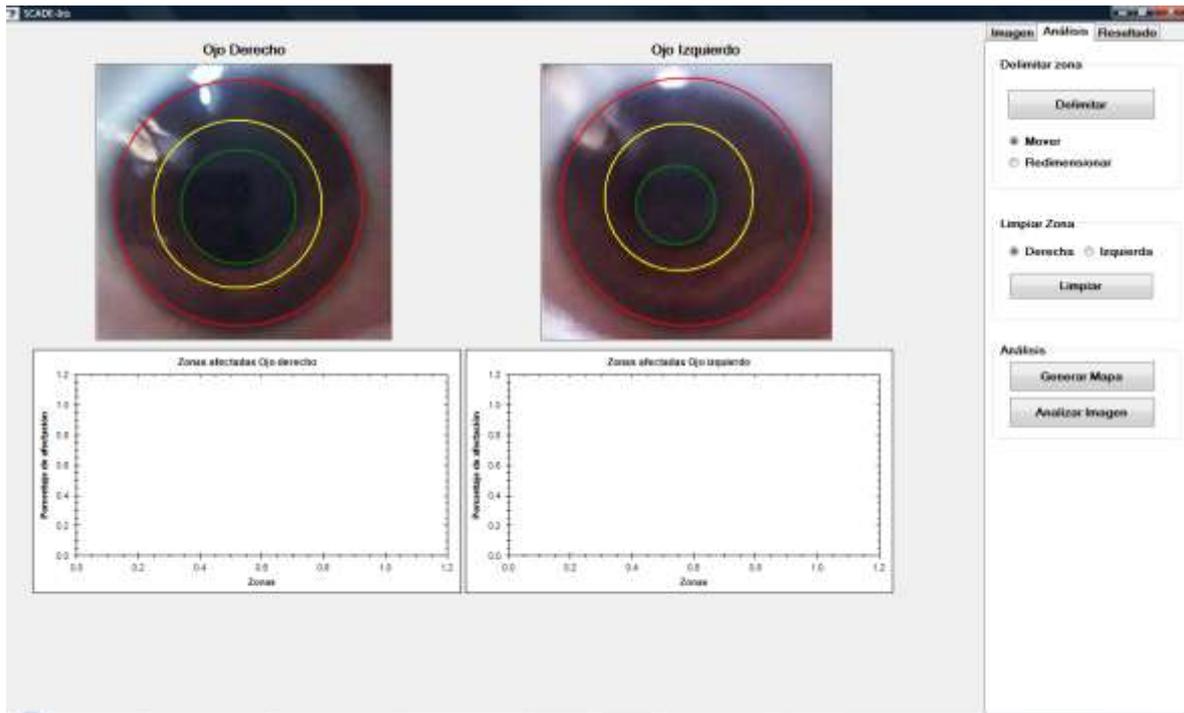


Figura 33. Pantalla mostrando los ojos delimitados por círculos.

Inmediatamente se hizo *clic* en el botón Generar mapa y al finalizar el usuario también hizo *clic* en el botón Analizar Imagen, mostrando la imagen de Analizando. La Figura 34 muestra al sistema realizando el análisis.

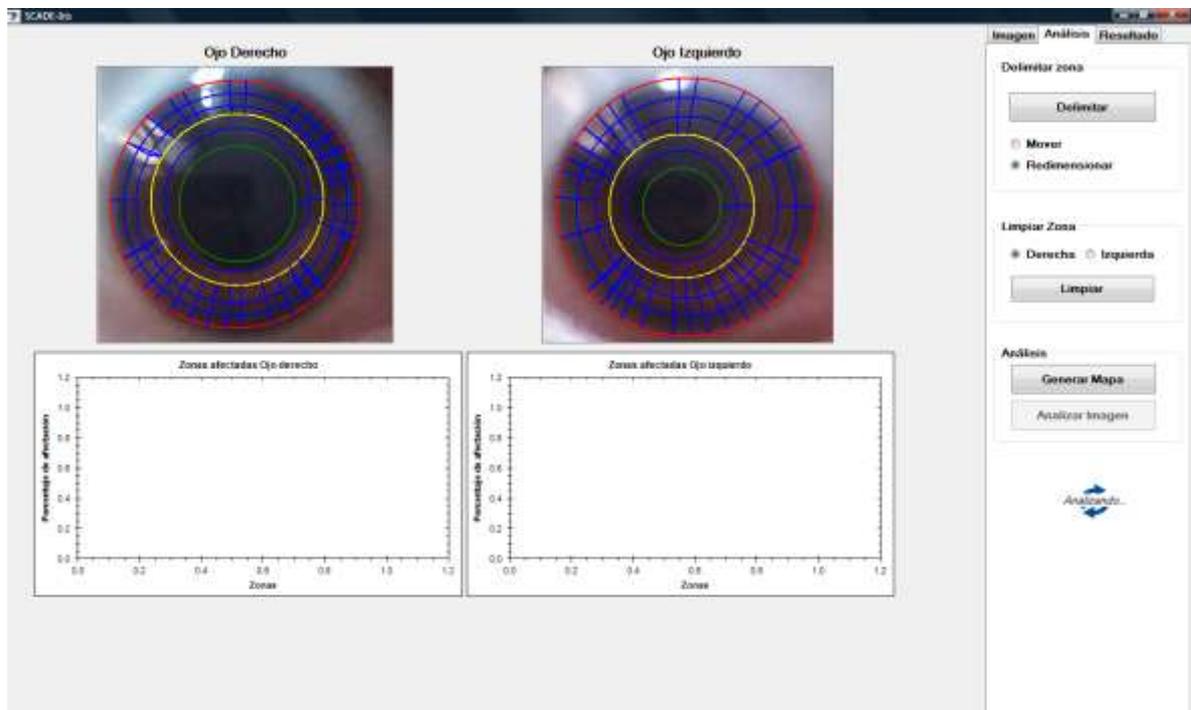


Figura 34. Pantalla mostrando el sistema realizando el análisis.

Al finalizar el análisis el sistema mostró las gráficas con los resultados. La Figura 35 muestra las gráficas con los resultados obtenidos.

# SCADE-Iris

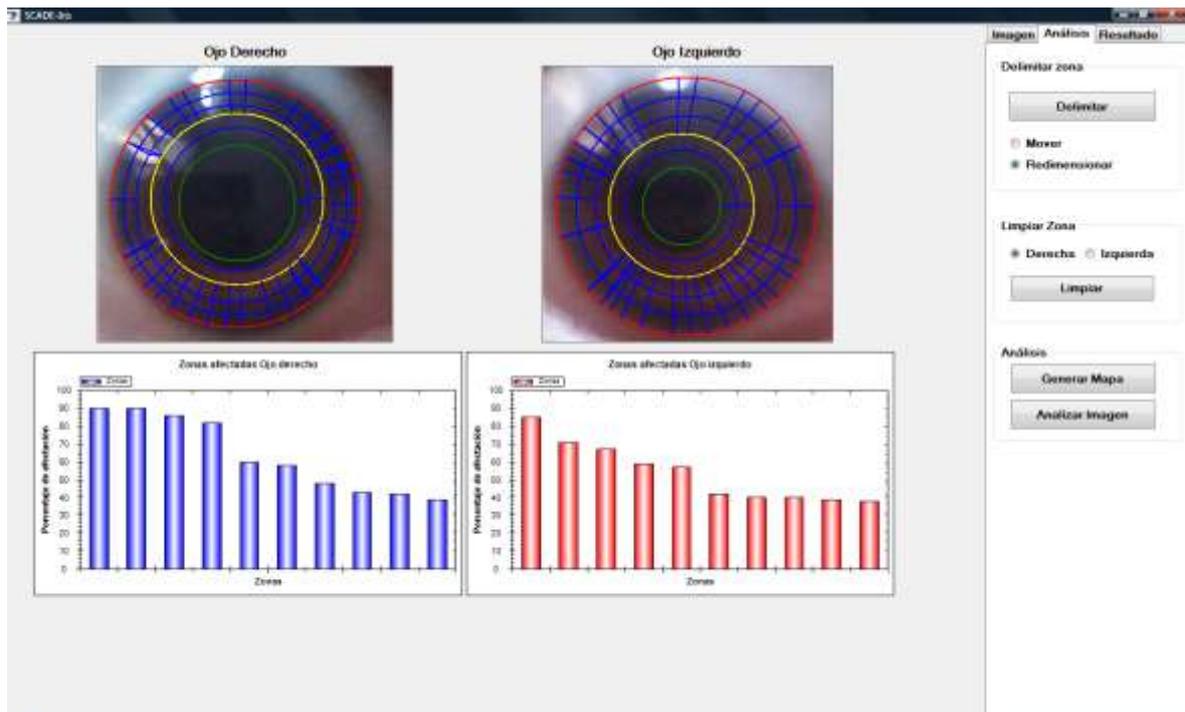


Figura 35. Pantalla mostrando el resultado del análisis.

Para concluir, el usuario se ubicó en la pestaña de Resultado e ingresó la información necesaria para el guardado del paciente y poder generar el reporte correspondiente. La Figura 36 muestra el reporte generado con la información del análisis realizado.

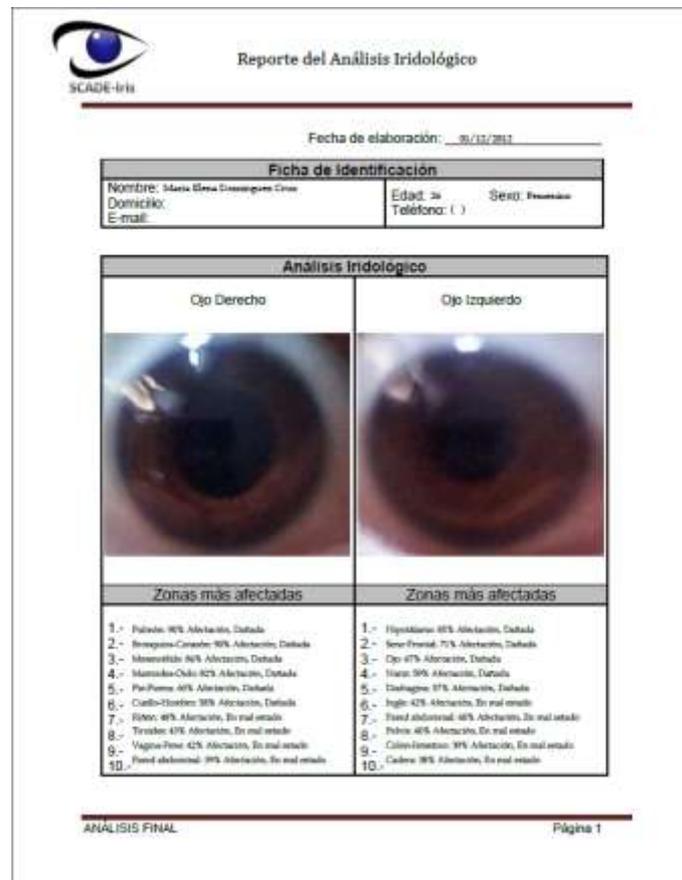


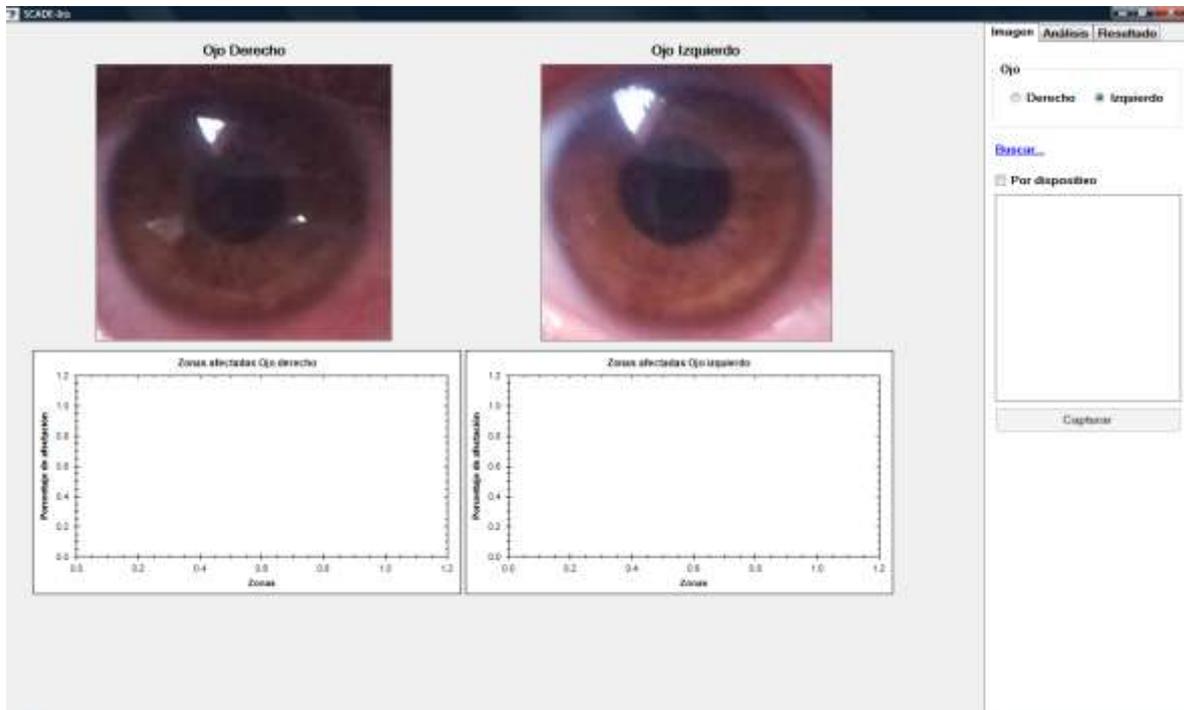
Figura 36. Reporte correspondiente a la prueba 2.

### 5.2.1. Resultados de la prueba

Los resultados fueron tomados de la mejor forma, ya que esta persona no cree en este tipo de análisis, se le comentó que solo fue una prueba para ver el funcionamiento de dicho sistema, por lo que esta prueba solo cumplió con los requisitos de navegabilidad sobre el mismo, sin tener alguna complicación.

### 5.3. Prueba 3

El siguiente usuario, al igual que el anterior, también es una persona de 26 años (hombre), que de igual manera no presentó problemas para ejecutar el análisis completo. Este usuario en específico, fue un poco distinto ya que se le comentó para qué era la prueba y posteriormente solo realizó el análisis sin la ayuda de otras personas. La *Figura 37* muestra las imágenes plasmadas dentro del área del sistema.



**Figura 37.** Pantalla con las imágenes de los ambos ojos.

El usuario se posicionó en la pestaña de Análisis y delimitó las zonas del ojo e hizo *click* sobre el botón Generar Mapa. La *Figura 38* muestra el mapa generado respectivamente de cada ojo.

# SCADE-Iris

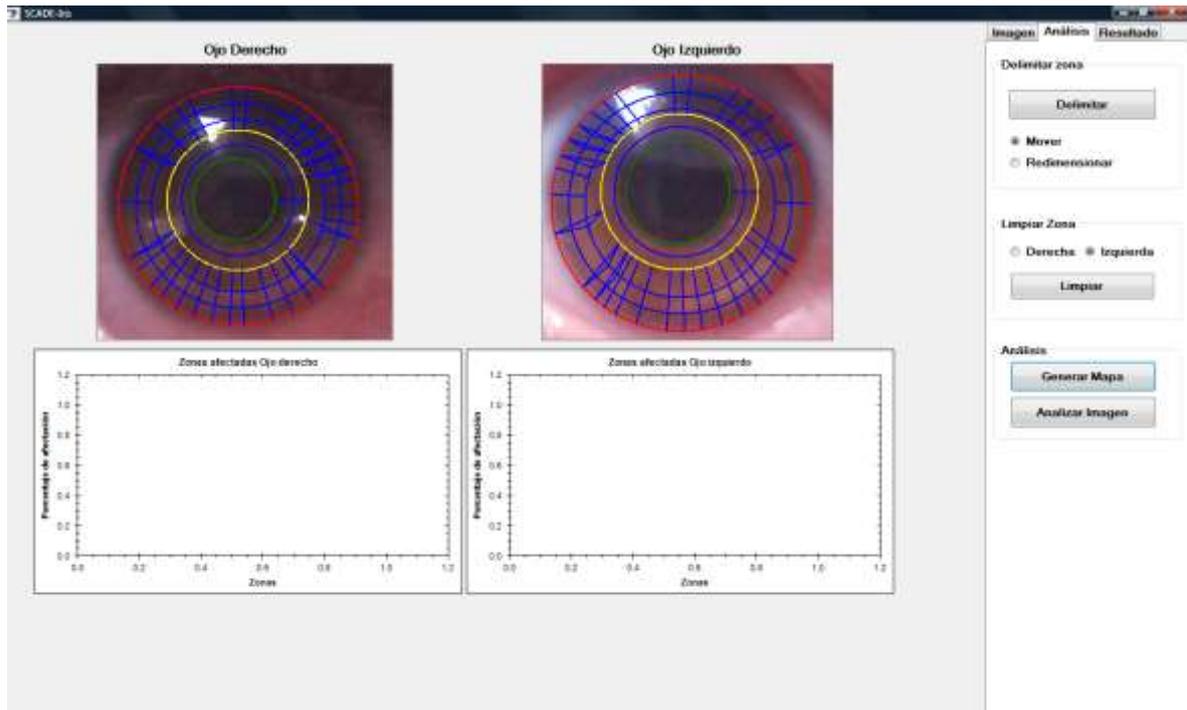


Figura 38. Pantalla con el mapa generado en cada ojo.

Esperó a que finalizara el proceso de generación del mapa y prosiguió a hacer *click* sobre el botón Analizar Imagen, mostrando los resultados dentro de las gráficas. La Figura 39 muestra los resultados del análisis mostrado dentro de las gráficas.

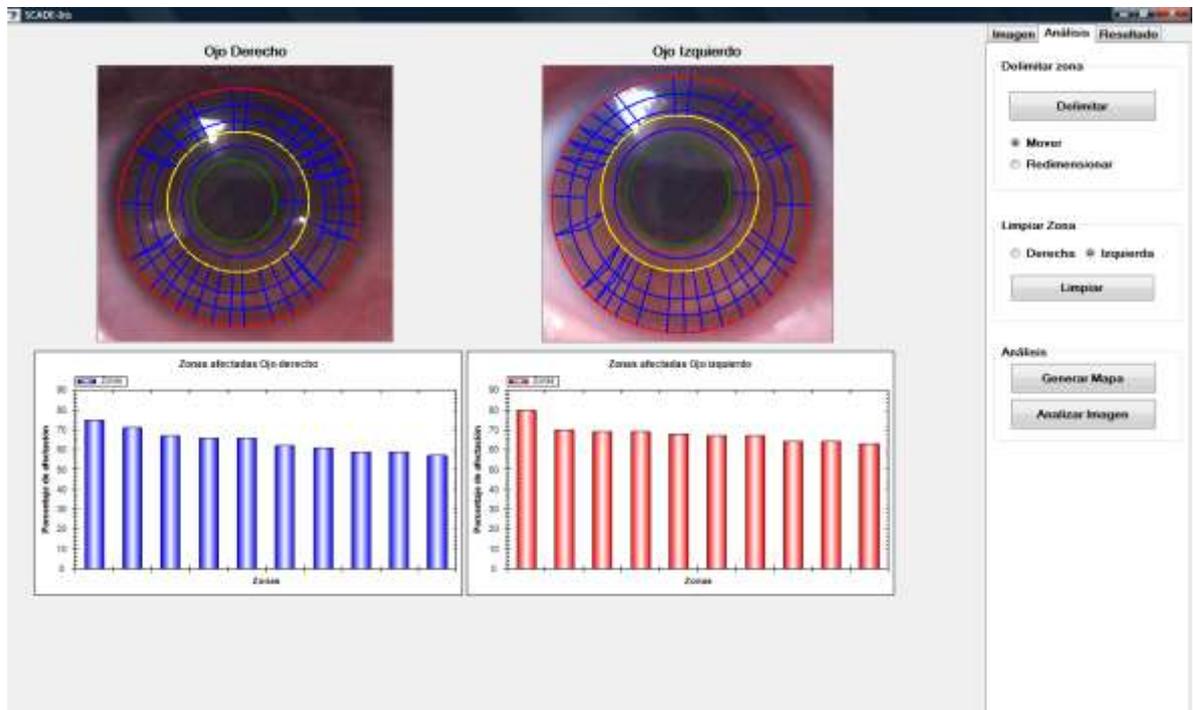


Figura 39. Pantalla con los resultados del análisis desarrollado.

Finalmente guardó su información y generó el reporte. La Figura 40 muestra el reporte generado durante el análisis.



Reporte del Análisis Iridológico

---

Fecha de elaboración: 01/12/2012

Ficha de Identificación		
Nombre: Ehecatl Hernández Guzmán	Edad: 26	SEXO: Masculino
Domicilio:	Teléfono: ( )	
E-mail:		

Análisis Iridológico	
Ojo Derecho	Ojo Izquierdo
	
Zonas más afectadas	Zonas más afectadas
<ol style="list-style-type: none"> <li>1.- Hígado-Vitícula: 75% Afectación, Dañada</li> <li>2.- Diafragma: 71% Afectación, Dañada</li> <li>3.- Mesencefalo: 67% Afectación, Dañada</li> <li>4.- Riñón: 66% Afectación, Dañada</li> <li>5.- Cadera: 66% Afectación, Dañada</li> <li>6.- Vejiga: 62% Afectación, Dañada</li> <li>7.- Ingle: 61% Afectación, Dañada</li> <li>8.- Mano-Brazo: 59% Afectación, Dañada</li> <li>9.- Pelvis: 59% Afectación, Dañada</li> <li>10.- Vagina-Pene: 57% Afectación, Dañada</li> </ol>	<ol style="list-style-type: none"> <li>1.- Prosenocefalo: 80% Afectación, Dañada</li> <li>2.- Bazo-Hígado: 70% Afectación, Dañada</li> <li>3.- Espalda: 69% Afectación, Dañada</li> <li>4.- Recto-Ano: 69% Afectación, Dañada</li> <li>5.- Pelvis: 68% Afectación, Dañada</li> <li>6.- Scápula: 67% Afectación, Dañada</li> <li>7.- Pared abdominal: 67% Afectación, Dañada</li> <li>8.- Riñón: 64% Afectación, Dañada</li> <li>9.- Ingle: 64% Afectación, Dañada</li> <li>10.- Vejiga: 63% Afectación, Dañada</li> </ol>

---

ANÁLISIS FINAL
Página 1

Figura 40. Reporte correspondiente a la prueba 3.

### 5.3.1. Resultados de la prueba

Los resultados fueron satisfactorios ya que ciertos problemas que el paciente padece se reflejo en el análisis realizado. Esto sorprendió al paciente ya que no se esperaba ese resultado. Al igual el manejo del sistema le fue muy fácil y en ningún momento tuvo la necesidad de consultar el Manual de usuario.

## 5.4. Prueba 4

La cuarta prueba fue realizada a una mujer de 34 años, a quién de igual forma se le explicó de qué se trataba dicha prueba. En este caso, fue necesario explicarle cada paso para la ejecución del análisis. Esto se debió al poco conocimiento del uso del ordenador. La *Figura 41* muestra las imágenes dentro del sistema.

## SCADE-Iris

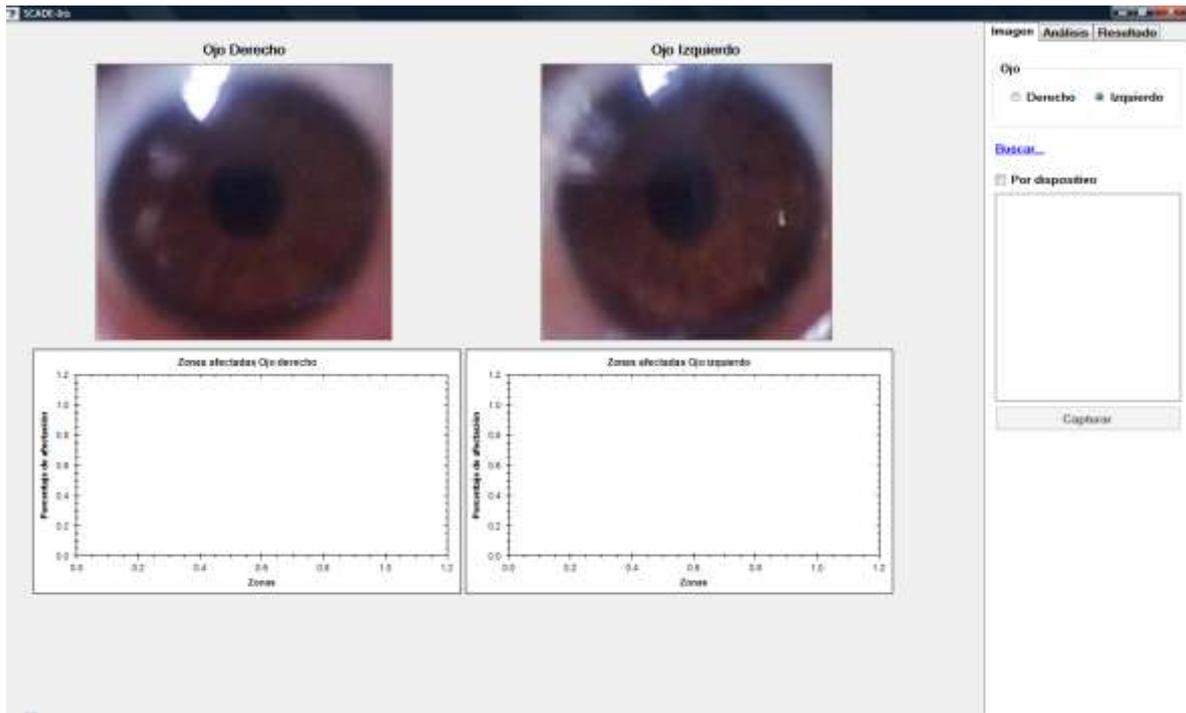


Figura 41. Pantalla mostrando las imágenes de ambos ojos.

La Figura 42 muestra la delimitación de los ojos con los círculos, este proceso se le dificultó al usuario ya que no lograba ajustar dichos círculos, por lo que fue necesario completar dicha tarea.

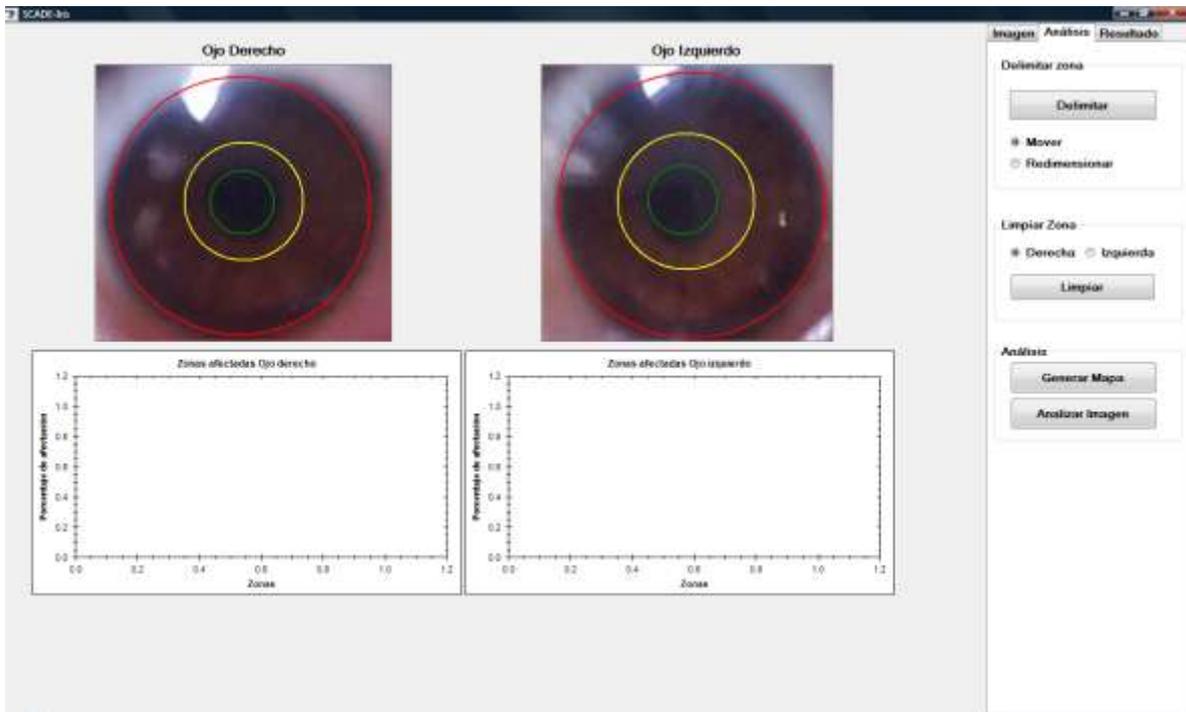


Figura 42. Pantalla mostrando la delimitación de los círculos y ojos.

En seguida se hizo *click* en el botón Generar Mapa y al dejar pasar 1 minuto se hizo *click* sobre el botón Analizar Imagen para visualizar los resultados. La Figura 43 muestra los resultados del análisis realizado para éste paciente.

# SCADE-Iris

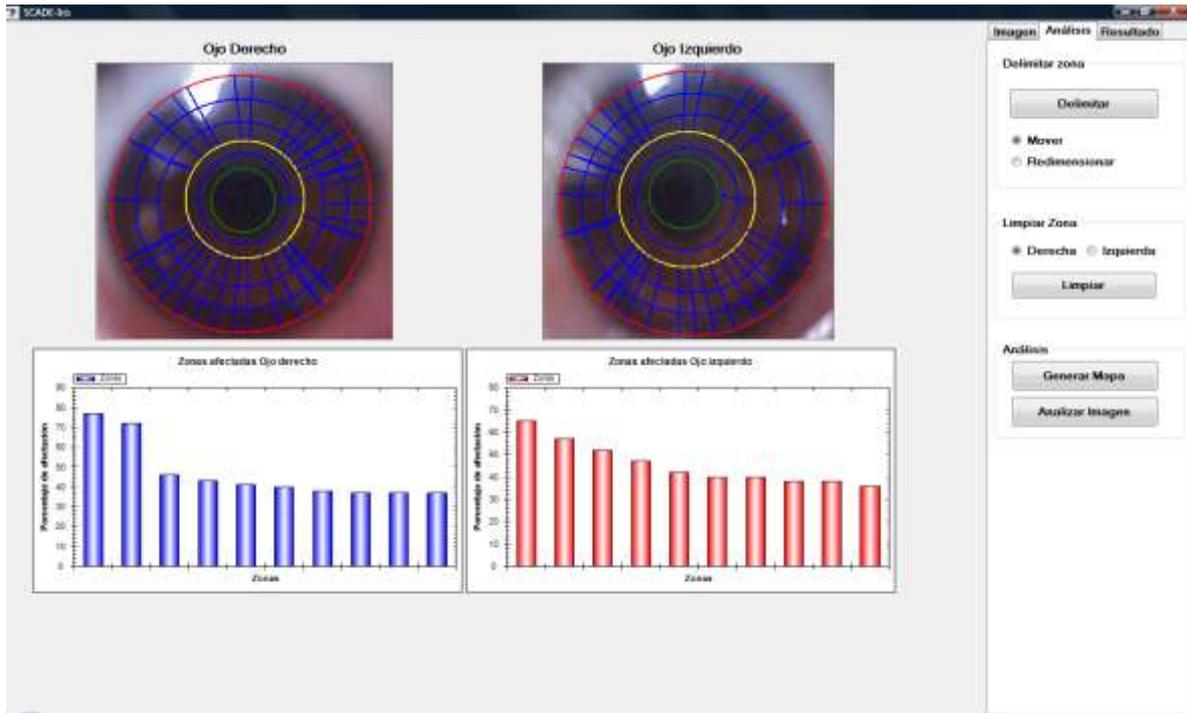


Figura 43. Pantalla con los resultados del análisis.

Al finalizar se le pidió al usuario que ingresara sus datos dentro de la pestaña de Resultados y que los guardara. Se concluyó con el reporte generado. La Figura 44 muestra el reporte generado para este análisis.

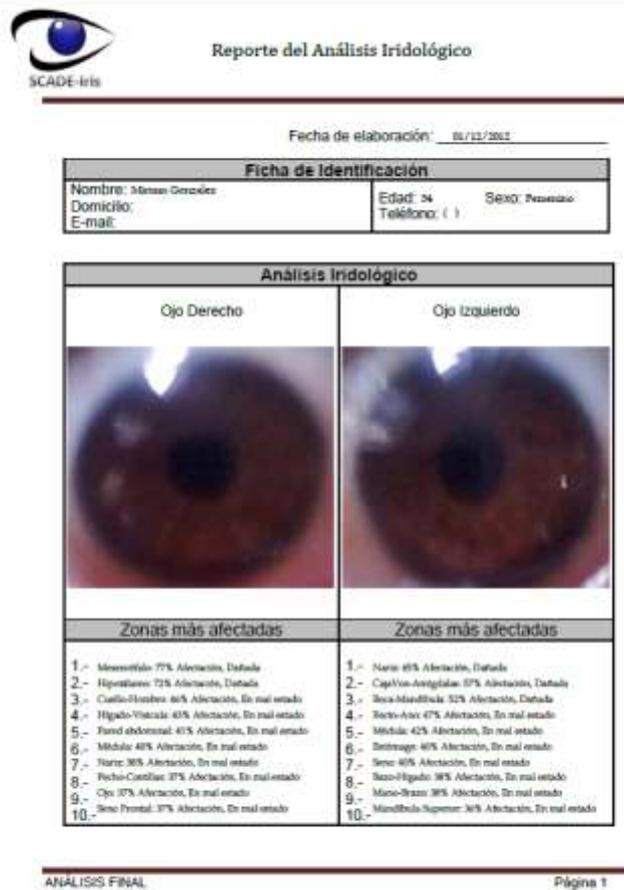


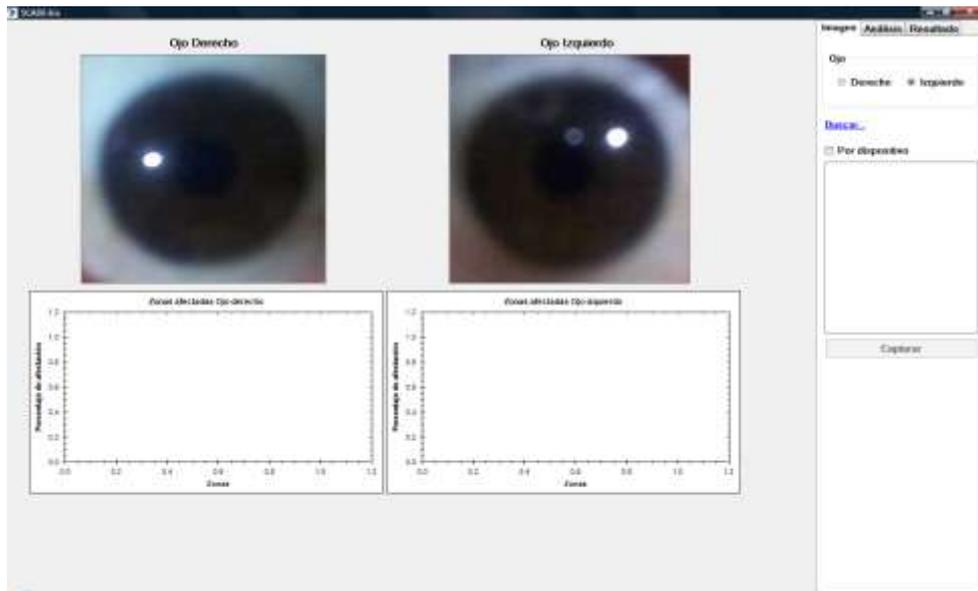
Figura 44. Reporte correspondiente a la prueba 4.

## 5.4.1. Resultados de la prueba

Los resultados de este análisis fueron sorprendentes para el paciente ya que era algo innovador porque nunca había escuchado sobre este método de diagnóstico. Se le cuestionó que opinión tenía a cerca del SCADE-Iris, y su respuesta fue que era muy interactivo, que a primera vista le fue complicado el entendimiento, pero que posiblemente un segundo análisis lo ejecutaría con menores dificultades.

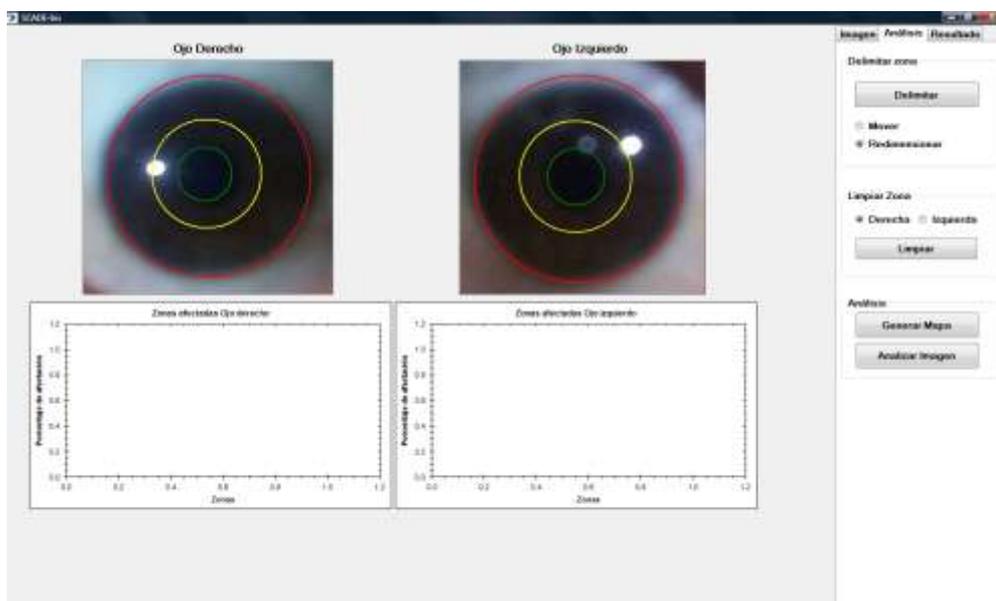
## 5.5. Prueba 5

El paciente que se utilizó en esta prueba, tiene 38 años, e inicio realizando el cargado de imágenes al sistema SCADE-Iris. La *Figura 45* muestra las imágenes cargadas en el sistema.



**Figura 45.** Pantalla con las imágenes del iris cargadas dentro del sistema.

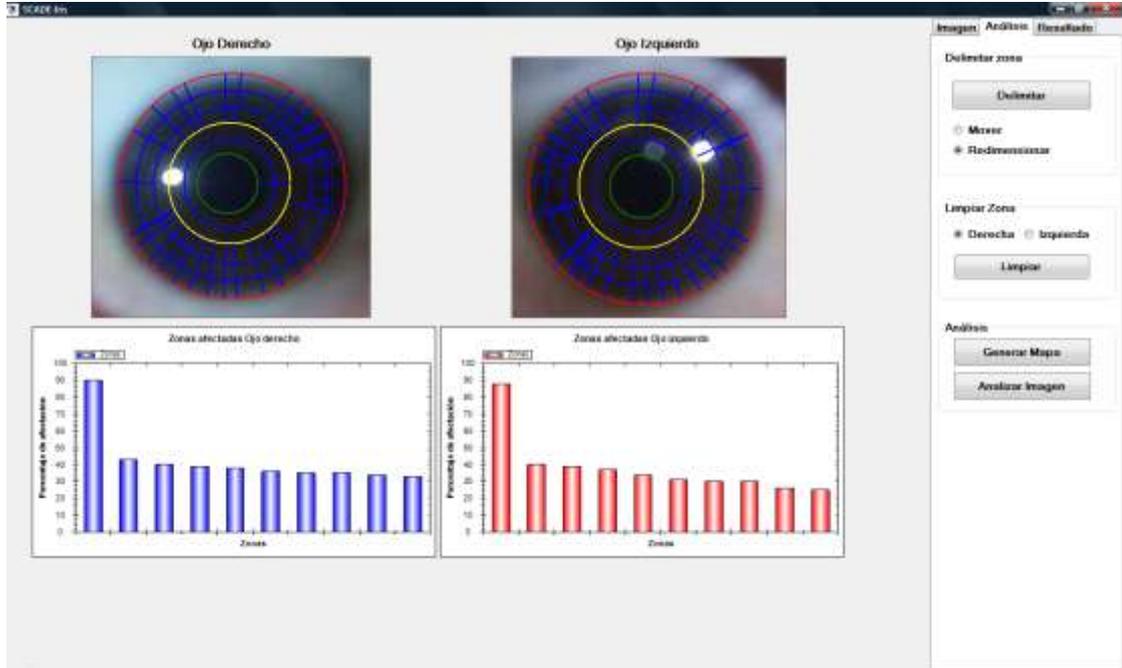
En este paso (Delimitación de ojos) se le explicó como es que debía ejecutarse, ya que desconocía el procedimiento. Una vez comprendido el procedimiento para realizar dichas ejecuciones, el usuario fue capaz de concluir con la tarea. La *Figura 46* muestra las imágenes de los respectivos ojos delimitados.



**Figura 46.** Pantalla con el iris delimitado por los círculos.

# SCADE-Iris

Se continuó haciendo *clic* sobre el botón Generar Mapa y al finalizar el proceso nuevamente fue necesario hacer *clic* sobre el botón Analizar Imagen. La *Figura 47* muestra el resultado del análisis realizado.



**Figura 47.** Resultados del análisis realizado.

Se finalizó con el guardado de la información de dicho paciente, así como la generación del reporte. La *Figura 48* muestra el reporte generado.



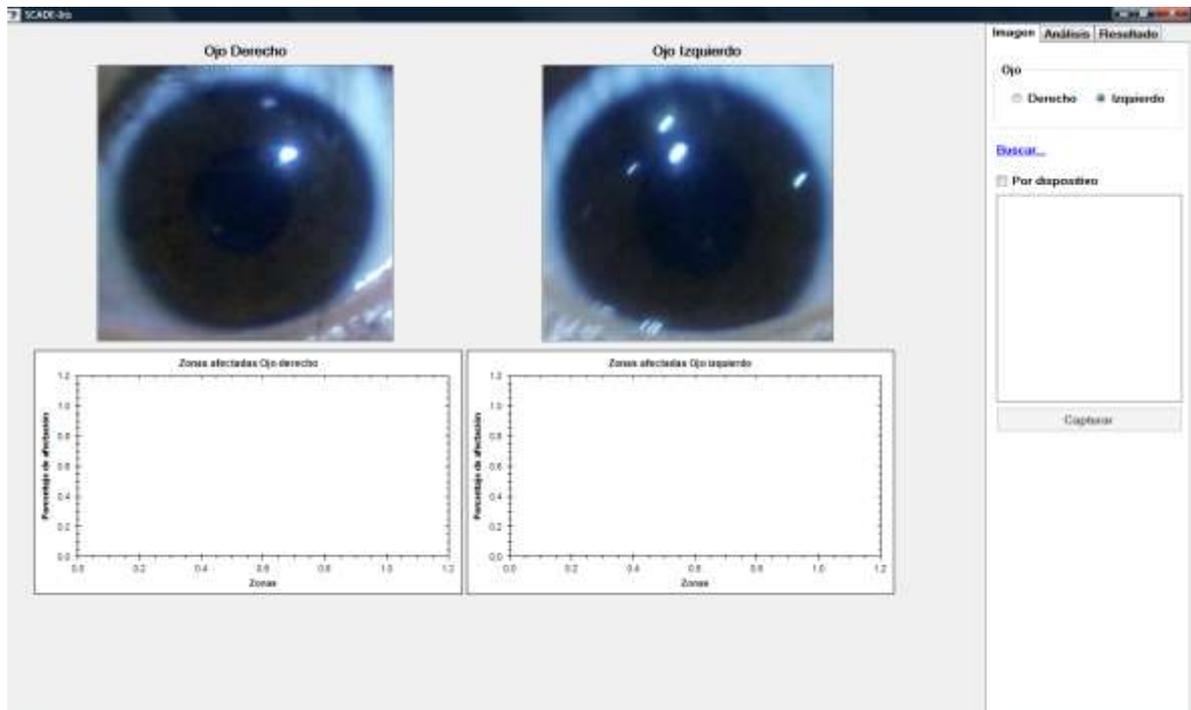
**Figura 48.** Reporte correspondiente a la prueba 5.

### 5.5.1. Resultados de la prueba

Los resultados de este análisis fueron importantes para el proyecto, ya que en las gráficas muestra una afectación en ambos ojos, esto se debe a la mancha de luz presentada en cada ojo. Donde se encuentra dicha mancha es el órgano que se representa dentro de la gráfica y de la cual muestra un grado de afectación muy elevado, lo que nos lleva a la conclusión de que el programa realiza lo que desde un principio se propuso, que es sacar los colores y promediarlos para que posteriormente sean comparados con las zonas.

### 5.6. Prueba 6

El paciente que se utilizó para esta prueba fue un usuario (mujer) de 11 años, la cual fue una prueba más fácil de lo que se pronosticaba, ya que a pesar de la poca edad, supo cómo utilizar el ordenador para el uso del SCADE-Iris. Antes de iniciar la prueba, se le explicó como funcionaba el sistema, por lo que se le facilitó aún más el manejo del mismo. Primero el usuario cargó las imágenes de sus ojos en el área de visualización. La *Figura 49* muestra la pantalla con las imágenes del iris.



**Figura 49.** Pantalla con las imágenes del iris cargadas dentro del sistema.

Se le comentó a detalle como realizar el procedimiento para la delimitación del ojo, ya que en las pruebas anteriores se detectó que era el paso con mayor dificultad. La *Figura 50* muestra los ojos delimitados con los círculos.

# SCADE-Iris

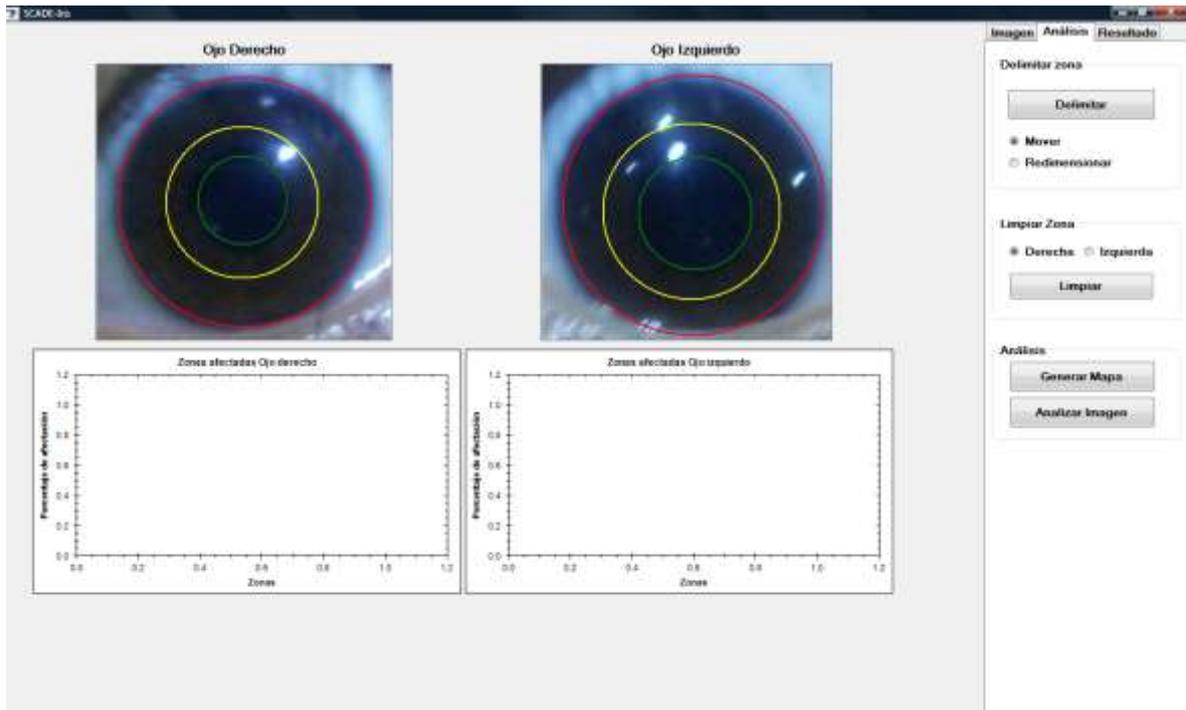


Figura 50. Ojos delimitados por los círculos.

Como siguiente paso se generó el mapa y se hizo *click* sobre el botón Analizar Imagen. La Figura 51 muestra los resultados del análisis realizado, mostrados dentro del área de las gráficas.

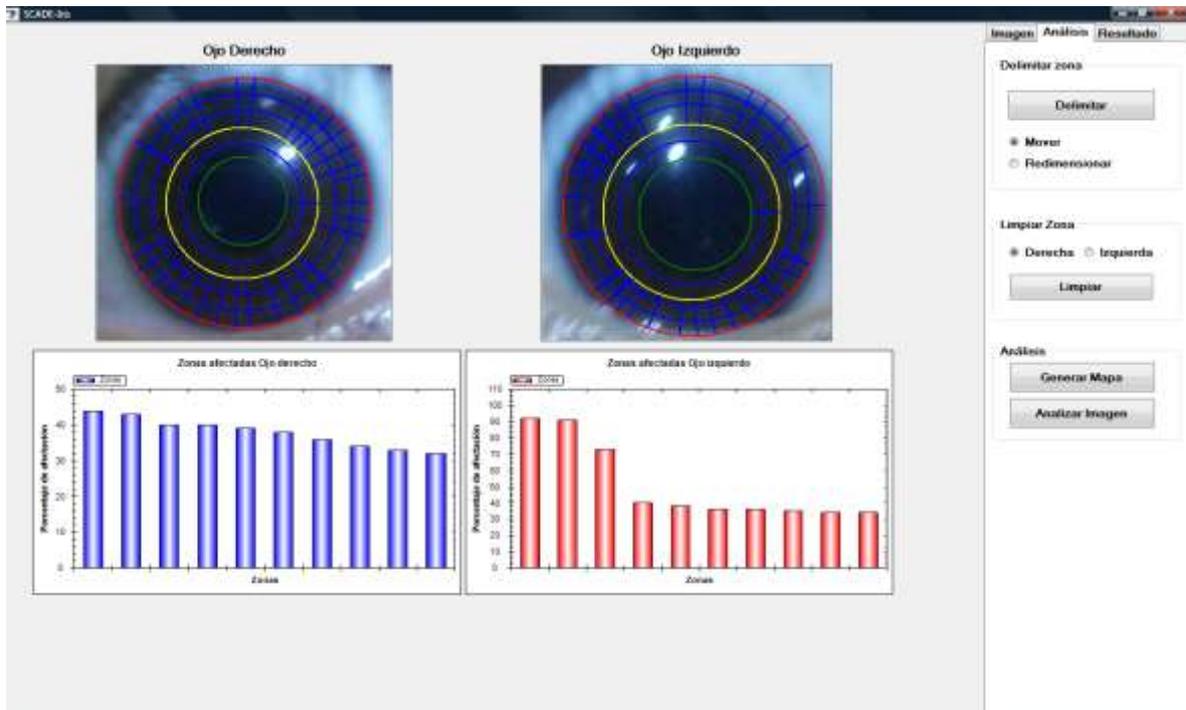


Figura 51. Resultados del análisis realizado.

Finalmente agregó sus datos y guardo la información para poder generar el reporte del análisis. La Figura 52 muestra el reporte que se generó.



Reporte del Análisis Iridológico

Fecha de elaboración: 02/12/2012

Ficha de Identificación		
Nombre: Jimena Cortes Bautista	Edad: 11	SEXO: Femenino
Domicilio:	Teléfono: ( )	
E-mail:		

Análisis Iridológico	
Ojo Derecho	Ojo Izquierdo
	
Zonas más afectadas	Zonas más afectadas
<ol style="list-style-type: none"> <li>1.- Pie-Pierna: 44% Afectación, En mal estado</li> <li>2.- Diafragma: 43% Afectación, En mal estado</li> <li>3.- Riñón: 40% Afectación, En mal estado</li> <li>4.- Recto-Ano: 40% Afectación, En mal estado</li> <li>5.- Ingle: 39% Afectación, En mal estado</li> <li>6.- Pared abdominal: 38% Afectación, En mal estado</li> <li>7.- Mano-Brazo: 36% Afectación, En mal estado</li> <li>8.- Vagina-Pene: 34% Afectación, En mal estado</li> <li>9.- Bronquios-Corazón: 33% Afectación, En mal estado</li> <li>10.- Pelvis: 32% Afectación, En mal estado</li> </ol>	<ol style="list-style-type: none"> <li>1.- Pulmón: 92% Afectación, Dañada</li> <li>2.- Prosenófalo: 91% Afectación, Dañada</li> <li>3.- Pie-Pierna: 73% Afectación, Dañada</li> <li>4.- Ojo: 40% Afectación, En mal estado</li> <li>5.- Ovarios-Testículos: 38% Afectación, En mal estado</li> <li>6.- CajaVoz-Amígdalas: 36% Afectación, En mal estado</li> <li>7.- Espalda: 36% Afectación, En mal estado</li> <li>8.- Bronquios: 35% Afectación, En mal estado</li> <li>9.- Seno: 34% Afectación, En mal estado</li> <li>10.- Mano-Brazo: 34% Afectación, En mal estado</li> </ol>

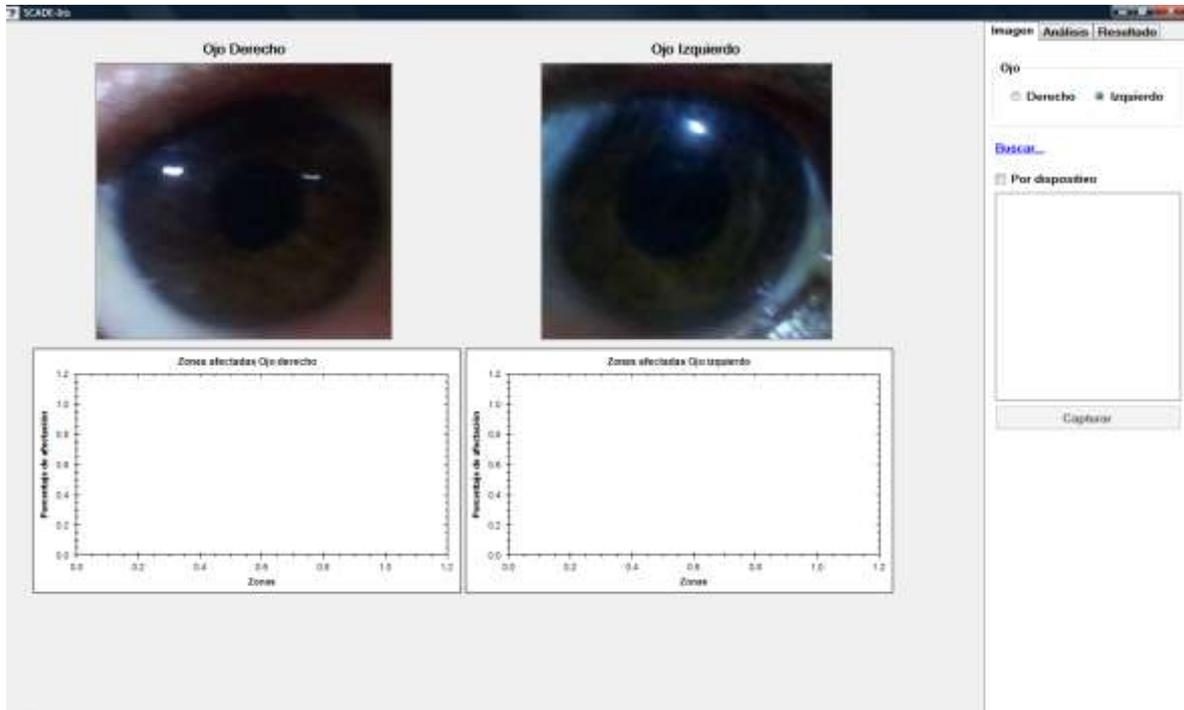
Figura 52. Reporte correspondiente a la prueba 6.

5.6.1. Resultados de la prueba

Esta prueba fue importante para ver la usabilidad de dicho proyecto, ya que el análisis fue realizado por un menor de edad, y por lo que se deduce que el sistema es fácil de usar solo con la explicación que se de al usuario de cómo utilizar dicho sistema. Los resultados no fueron tan interesantes para el paciente ya que su interés no es saber este tipo de análisis.

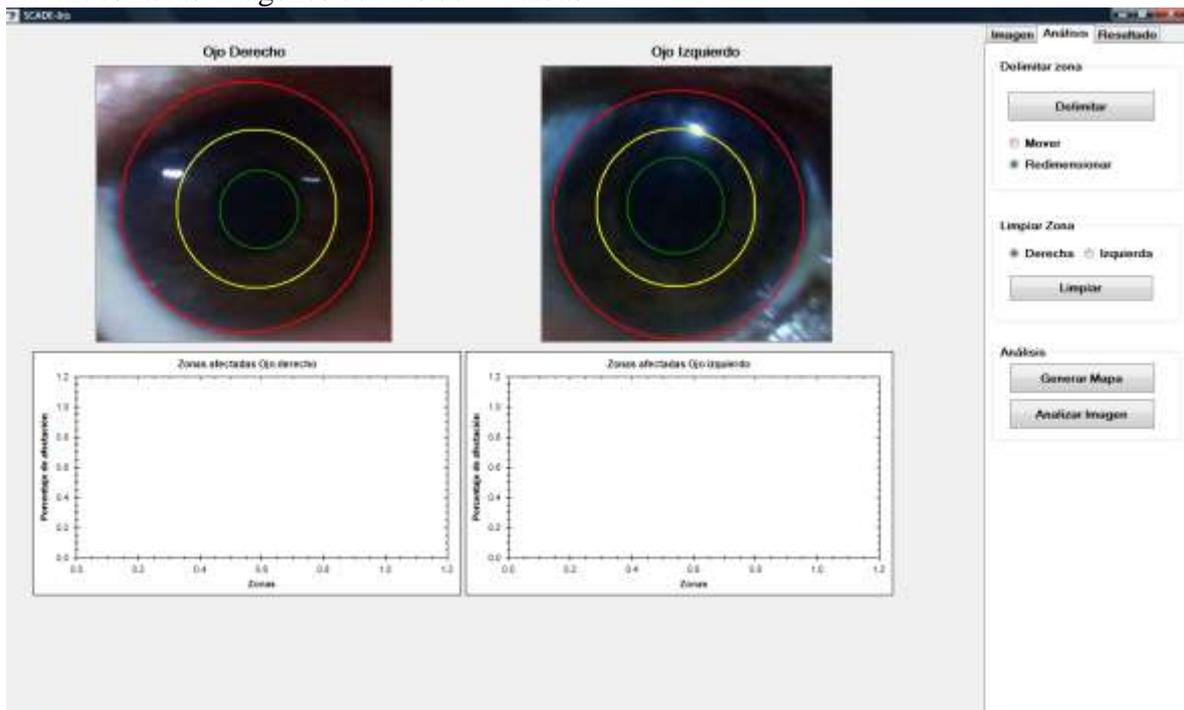
## 5.7. Prueba 7

El siguiente paciente es un usuario (mujer) de 45 años, a la cuál se le explicó cómo realizar la prueba. Lo complicado fue hacer que interactuara con el sistema SCADE-Iris ya que se le dificultó entender el funcionamiento del SCADE-Iris, y para ello fue necesario brindarle asesoría y que pudiera realizar dicho análisis. La *Figura 53* muestra las imágenes cargadas en el sistema.



**Figura 53.** Pantalla del sistema SCADE-Iris con las imágenes cargadas.

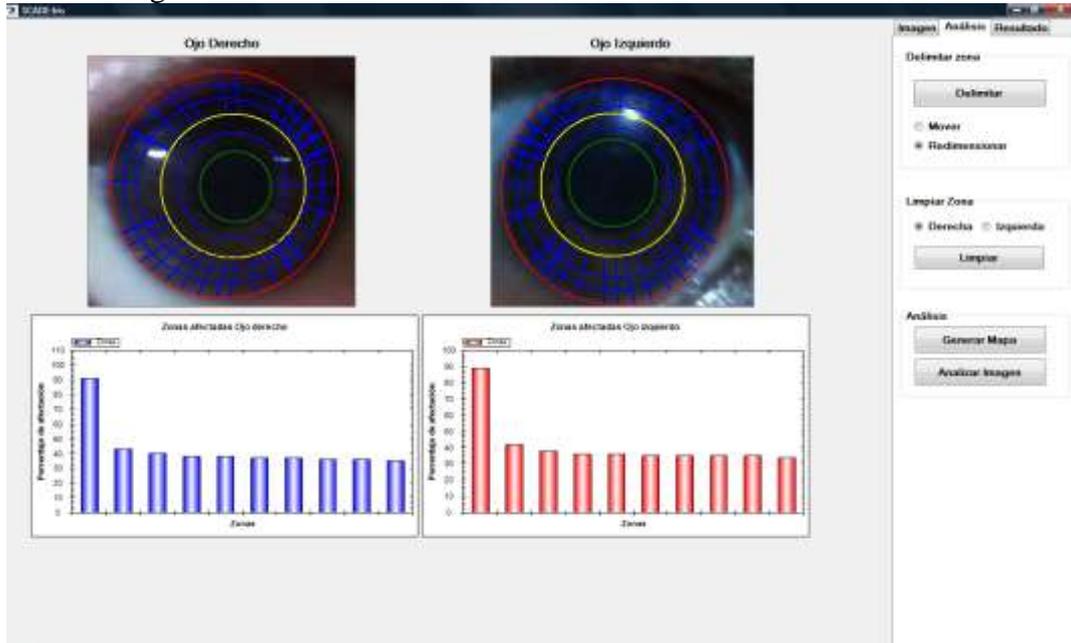
A continuación se le ayudó a delimitar el ojo con los tres círculos que muestra el sistema. La *Figura 54* muestra las imágenes del iris delimitadas.



**Figura 54.** Ojos delimitados por los tres círculos.

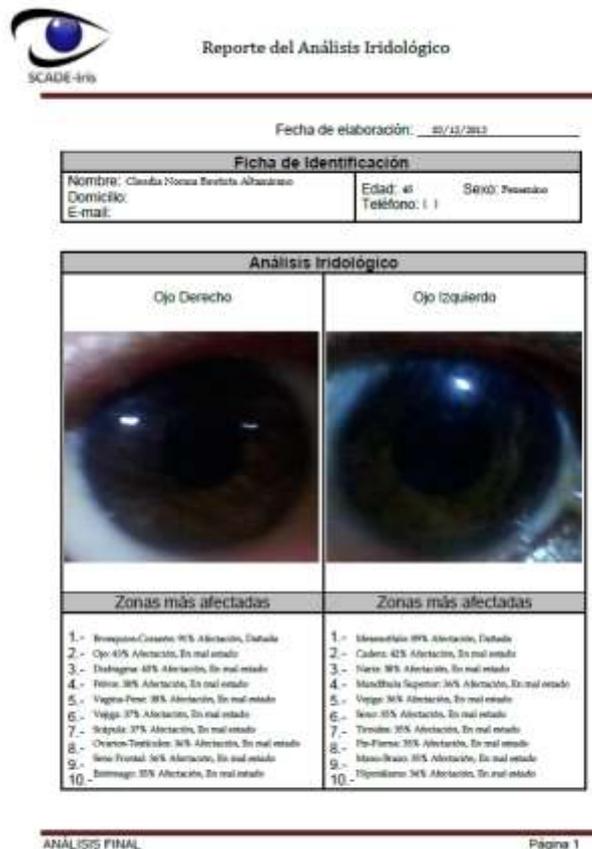
## SCADE-Iris

Después, el paciente hizo *clic* sobre el botón Generar Mapa y después de 1 minuto hizo también *clic* sobre el botón Analizar Imagen. La *Figura 55* muestra el análisis realizado de cada iris, representándolo en las gráficas.



**Figura 55.** Análisis realizado.

Se finalizó con el agregado de la información personal para guardarla dentro de la Base de Datos y generar el reporte. La *Figura 56* muestra el reporte realizado.



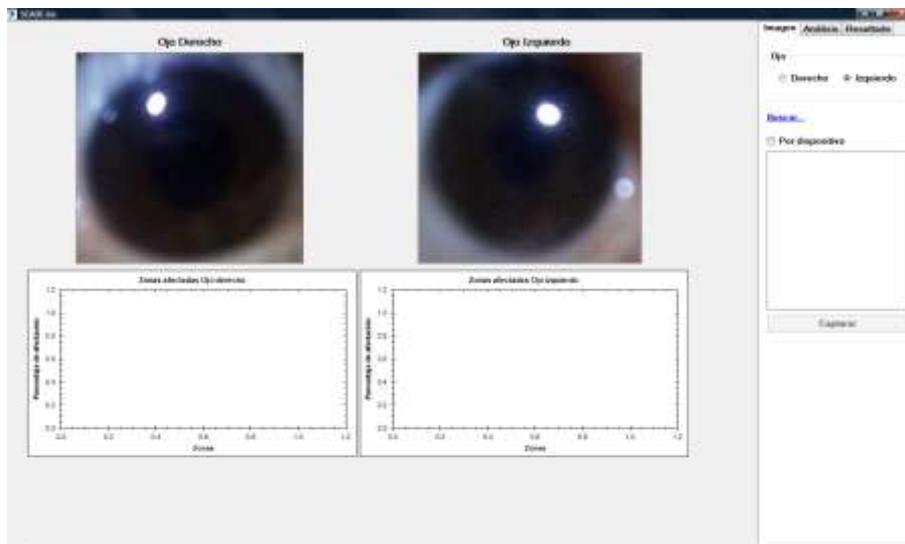
**Figura 56.** Reporte correspondiente a la prueba 7.

## 5.7.1. Resultados de la prueba

El resultado de este análisis nos reporta la afectación de un órgano, el cual fue afectado por la toma de la foto, ya que la mancha que se presentó indicaba la zona afectada. Nuevamente se confirma el correcto funcionamiento del SCADE-Iris.

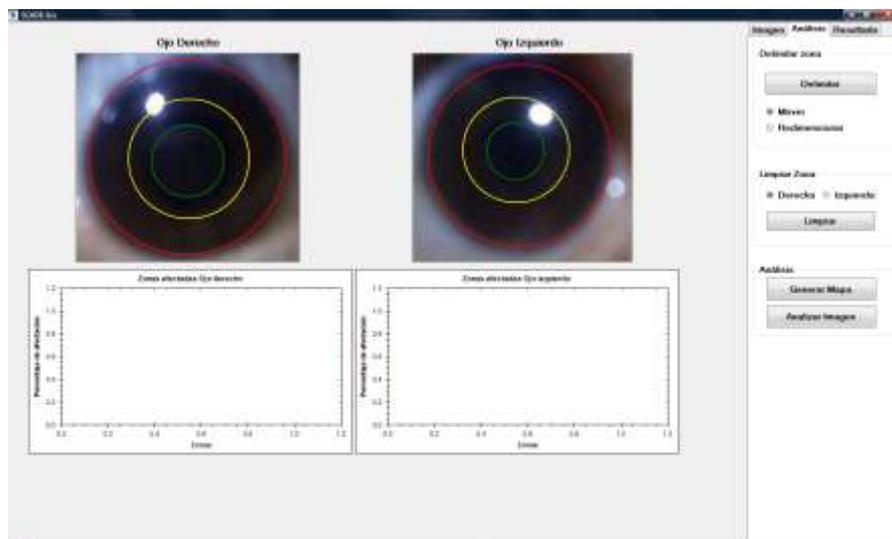
## 5.8. Prueba 8

Esta prueba resultó interesante y difícil porque el paciente fue un usuario de 9 años, sobre el cuál es difícil mantener la atención y mantenerlo quieto. Por consiguiente se le ayudó a realizar dicha prueba. Se inició tomando la foto de ambos ojos, para posteriormente cargarlas en la herramienta. La *Figura 57* muestra las imágenes del iris cargadas en el sistema SCADE-Iris.



**Figura 57.** Pantalla con la imagen de ambos ojos para su posterior análisis.

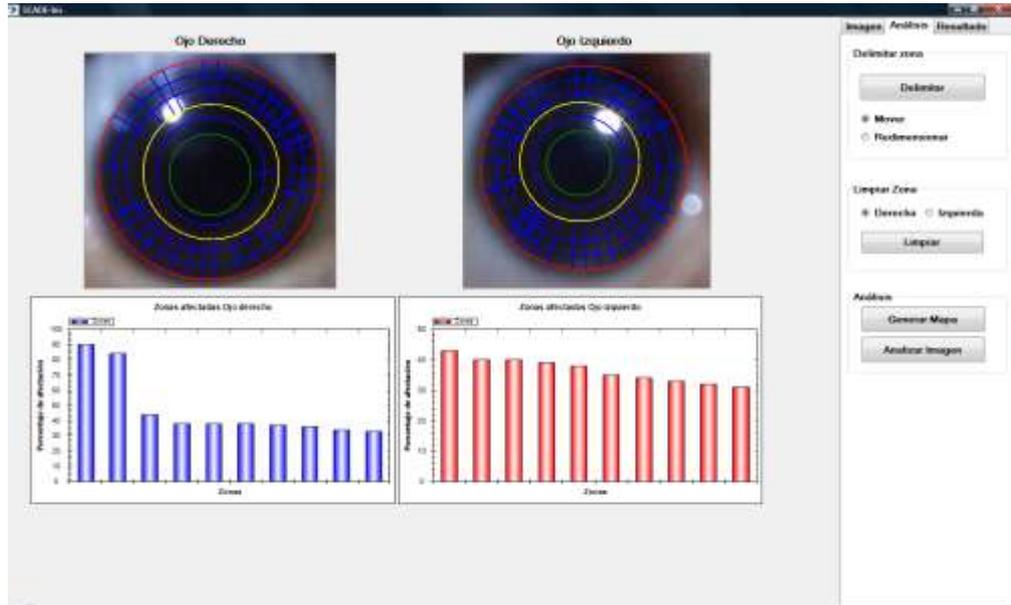
El siguiente paso, fue a ayudarlo a delimitar las áreas del iris. La *Figura 58* muestra las imágenes delimitadas.



**Figura 58.** Ojos delimitados.

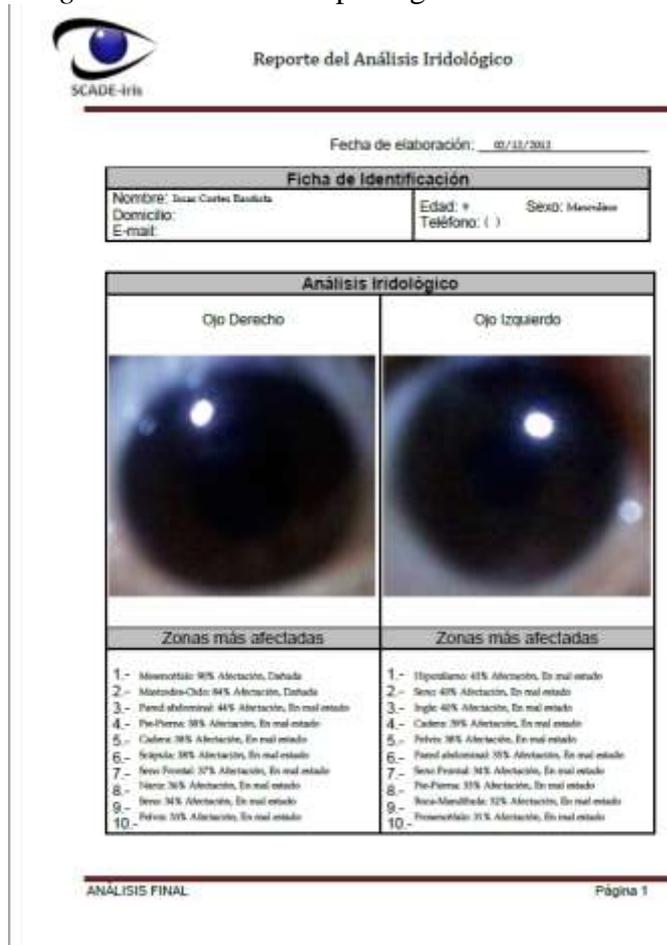
## SCADE-Iris

Posteriormente el usuario hizo *clic* sobre el botón Generar Mapa y seguidamente en el botón de Analizar Imagen. La *Figura 59* muestra el análisis realizado con la demostración de las gráficas de ambos ojos.



**Figura 59.** Resultados del análisis realizado.

Para finalizar se le ayudó a capturar la información requerida para la generación del reporte así como el guardado del paciente. La *Figura 60* muestra el reporte generado.



**Figura 60.** Reporte correspondiente a la prueba 8.

### **5.8.1. Resultados de la prueba**

Con esta prueba se identifico que el sistema esta desarrollado para pacientes que cumplen la mayoría de edad o con una edad en la que entiendan un poco mas a cerca de lo que se les esta hablando, como con el usuario de 11 años mencionado en las pruebas anteriores. Esta afirmación se basa en los resultados obtenidos en esta prueba, ya que un usuario de 9 años no le interesa este tipo de análisis y posiblemente no le corresponde conocer este tipo de medicina alternativa.

## CAPÍTULO 6. CONCLUSIONES

Las conclusiones de este trabajo hacen referencia a los objetivos inicialmente propuestos y se comparan con los resultados finalmente obtenidos. Se agregan algunos comentarios acerca de posibles trabajos futuros que le den continuidad a este proyecto.

### 6.1. Lo que se propuso

Inicialmente se propuso crear una herramienta computacional que ayude al iridólogo en el proceso de diagnóstico para detectar problemas de salud, y en la detección de indicios que permitan prevenir problemas posteriores, dicha herramienta también se propuso que fuera útil para cualquier persona interesada en saber si es necesario consultar a un médico o no.

En cuanto a la metodología que se pretendía utilizar de manera inicial, fue crear una Base de Datos para llevar un registro detallado de cada uno de los historiales de los pacientes, lo cual facilitaría la visualización de la evolución de los problemas diagnosticados. Además, esta herramienta debía permitir crear en el futuro importantes bases de conocimiento, debido a que se almacenarían imágenes con sus respectivos análisis y diagnósticos Iridológicos.

La solución que se propuso para el análisis de la imagen del iris debía reconocer ciertos tipos de colores o manchas ubicadas en determinadas zonas del iris sin importar su forma precisa, esto, mediante el uso de librerías OpenCV. Por consiguiente, la forma de análisis basada en colores y zonas es la que se pretendía adoptar como solución al problema del análisis del iris.

El objetivo funcional planteado es, permitir al médico hacer un diagnóstico más completo y rápido, complementando la experiencia indispensable del médico y permitirá generar, de inmediato, un reporte impreso sobre los resultados.

### 6.2. Lo que se hizo

Ya en la práctica y durante el desarrollo de todo el proyecto, se diseñó un sistema que permitiera generar el análisis iridológico de la imagen de ambos ojos (izquierdo y derecho) mediante el uso de fórmulas estadísticas y descartando el uso de librerías de OpenCV. Dicho sistema (SCADE-Iris), permite analizar una imagen que provenga de un archivo en formato .bmp o de una imagen tomada por una *webcam* almacenada también en formato .bmp.

Una vez que se tiene la imagen, el sistema es capaz de ajustarle el mapa Iridológico para que mediante una serie de formulas y reglas de análisis se pueda generar la detección de zonas y poder diferenciar entre una zona afectada y una zona que se encuentra sana. Al final, el sistema permite asociarle al estudio iridológico los datos del paciente que se está analizando y las imágenes que se tienen de él, así como poder generar un reporte impreso de dicho análisis.

También, el sistema cuenta con una Base de Datos para el almacenamiento de la información estudiada para que a futuro pueda ser utilizado como se propuso en un inicio, una fuente de conocimiento.

### **6.3. Lo que se obtuvo**

Con el Proyecto Terminal SCADE-Iris se obtuvieron importantes resultados durante el desarrollo de las pruebas; se obtuvieron resultados tanto positivos como negativos. Los resultados negativos se debieron al grado de credibilidad acerca del tema de la iridología, ya que los pacientes que se tomaron al inicio, se les comento el tipo de prueba que iban a realizar, por lo que al principio no deseaban formar parte de las pruebas.

Los resultados positivos de estas pruebas, fueron la facilidad del uso del sistema, a la mayoría de los pacientes se les explicó el funcionamiento del sistema y al momento de sentarlos a realizar dicha prueba se les hizo más fácil el desarrollo del análisis. Algunos tuvieron ciertos problemas, ya que no tenían mucho conocimiento de como usar una computadora. Al finalizar algunos análisis la respuesta emocional fue de sorpresa ya que en ciertos casos los resultados coincidieron con lo que nos comentaron los pacientes que padecían.

### **6.4. Trabajos futuros**

Este trabajo de Proyecto Terminal no termina aquí, ya que su método de análisis diseñado se puede adaptar a diversas áreas que no fueran las imágenes de un ojo y posiblemente se puede mejorar con métodos más exactos en la obtención de colores de una imagen que discriminen la luz y la sombra que pueden contener en la mayoría de los casos. Otro método que se puede implementar es la búsqueda de patrones con respecto a las texturas del iris, ya que es más exacto dar un diagnóstico de los órganos afectados y el grado de afectación.

Este proyecto puede ampliar sus funciones mostrando el nombre de cada órgano asociado con las zonas del mapa, esto para un mejor manejo del sistema; no sólo para el uso médico si no para el uso personal de quien quiera realizar el análisis sin la necesidad de saber sobre esta metodología de análisis.

## REFERENCIAS

- [1] C. J. Castán, “Reconocimiento de patrones de mamografías usando redes neuronales artificiales para detectar cáncer”, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2008.
- [2] C. M. Aguilar, “Sistema de reconocimiento y contabilización de objetos en línea de producción, a través de imágenes digitales”, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [3] L. Marín, “Sistema de reconocimiento del alfabeto dactilológico utilizando procesamiento de imágenes”, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [4] M. A. Rojas, “Algoritmo para acoplamiento automático por similitudes de imágenes digitales”, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [5] A. Pacheco, “Procesamiento de imágenes utilizando mapas autoorganizados”, Propuesta de proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [6] E. A. García, “Reconocimiento de huellas digitales”, Propuesta de proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [7] M. E. García, “Software para validación antropométrica por procesamiento digital de imágenes”, Propuesta de proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [8] T. D. Santiago y A. Hernández, “Sistema de identificación de personas a través del iris mediante análisis de texturas por filtrado de wavelets”, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.
- [9] H. Cervantes y A. Márquez, “Autenticación de personas a través del iris”, Tesis de Licenciatura, Universidad Autónoma Metropolitana Iztapalapa, D.F., México, 2004.
- [10] G. E. López y R. A. Martín, “Procesamiento de imágenes de fondo de ojo para la detección temprana de diabetes mellitus e hipertensión arterial”, Tesis de Licenciatura, Universidad Autónoma Metropolitana Iztapalapa, D.F., México, 2007.
- [11] A. E. Romo, “Reconocimiento de imágenes mediante redes neuronales artificiales aplicado a la iridología”, Tesis de Maestría, Centro de Investigación y Estudios Avanzados del I.P.N, México, 2005.
- [12] “Iridología – Software de Localización de los Signos en el Iris”. [En línea]. Disponible: <http://aigaliris.com/es/node/1> [Consultada: 26 Mayo 2011].
- [13] “Home of Globally Distributed Iridology Software & Iriscope Systems”. [En línea]. Disponible: <http://www.healthline.com.au/products/irisearch2000.htm> [Consultada: 26 Mayo 2011].

## DOCUMENTOS ANEXOS

Anexo A – Reportes realizados durante la ejecución de las pruebas.



Prueba-1-Paciente\_S  
ofia.pdf



Prueba-2-Paciente\_  
MariaElena.pdf



Prueba-3-Paciente\_E  
hecatl.pdf



Prueba-4-Paciente\_  
Miriam.pdf



Prueba-5-Paciente\_AP  
braham.pdf



Prueba-6-Paciente\_Ji  
mena.pdf



Prueba-7-Paciente\_C  
laudia.pdf



Prueba-8-Paciente\_I  
ssac.pdf

Anexo B - Manual de Usuario - SCADE-Iris.pdf



Manual de Usuario -  
SCADE-Iris.pdf