

*UNIVERSIDAD DE LA REPÚBLICA*  
*FACULTAD DE INGENIERÍA*

# **Analizador de Imágenes Médicas**

**PROYECTO DE TALLER 5**  
**INFORME FINAL**

*A/C Pablo Flores*  
*A/C Marcelo Rodríguez*

# INDICE

<b><u>1</u></b>	<b><u>PREFACIO</u></b> .....	<b>5</b>
<b><u>2</u></b>	<b><u>ORGANIZACIÓN DEL INFORME</u></b> .....	<b>6</b>
<b><u>3</u></b>	<b><u>PRESENTACIÓN DE LAS PARTES INVOLUCRADAS</u></b> .....	<b>8</b>
3.1	<u>ORGANIZACIÓN PARA LA QUE SE TRABAJÓ</u> .....	8
3.2	<u>INVOLUCRADOS POR PARTE DEL HOSPITAL DE CLÍNICAS</u> .....	8
3.3	<u>TUTORES EN FACULTAD DE INGENIERÍA</u> .....	8
3.4	<u>INTEGRANTES DEL GRUPO</u> .....	8
<b><u>4</u></b>	<b><u>INTRODUCCIÓN</u></b> .....	<b>9</b>
4.1	<u>CONCEPCIÓN INICIAL DEL PROYECTO</u> .....	9
4.2	<u>ESPECIALIZACIÓN EN IMÁGENES MÉDICAS</u> .....	10
4.3	<u>OBJETIVOS DEL PROYECTO</u> .....	11
4.4	<u>CONCEPCIÓN DEL SISTEMA</u> .....	11
4.5	<u>RESEÑA DE LAS TECNOLOGÍAS UTILIZADAS</u> .....	13
4.6	<u>ADQUISICIÓN Y TRATAMIENTO DE IMÁGENES</u> .....	13
4.7	<u>DIVISIÓN DE IMÁGENES EN REGIONES (SEGMENTACIÓN)</u> .....	13
4.8	<u>EXTRACCIÓN DE DESCRIPTORES DE IMÁGENES Y REGIONES Y ALMACENAMIENTO DE LOS MISMOS EN UNA BASE DE DATOS</u> 14	14
4.9	<u>COMPARACIÓN DE IMÁGENES Y REGIONES A PARTIR DE LA COMPARACIÓN DE SUS DESCRIPTORES</u> .....	15
4.10	<u>MECANISMOS PARA AMPLIAR EL ESPECTRO DE ALGORITMOS DE TRATAMIENTO DE IMÁGENES Y DE EXTRACCIÓN DE DESCRIPTORES</u> .....	15
<b><u>5</u></b>	<b><u>PROCESO DE DESARROLLO</u></b> .....	<b>16</b>
5.1	<u>TAMAÑO DEL SISTEMA</u> .....	16
5.2	<u>DISTRIBUCIÓN DEL ESFUERZO A LO LARGO DEL PROYECTO</u> .....	17

<b><u>6</u></b>	<b><u>ANÁLISIS DEL SISTEMA</u></b>	<b>18</b>
6.1	<u>CASOS DE USO</u>	18
6.2	<u>MODELO CONCEPTUAL</u>	20
6.3	<u>PREPROCESAMIENTO</u>	21
6.4	<u>CLASIFICACIÓN DE IMÁGENES</u>	21
6.5	<u>DEFINICIÓN DE PATOLOGÍAS</u>	23
6.6	<u>BÚSQUEDA DE PATOLOGÍAS</u>	24
6.7	<u>BÚSQUEDA DE PATOLOGÍA EN IMÁGENES</u>	24
<b><u>7</u></b>	<b><u>DEFINICIÓN DE ALCANCE</u></b>	<b>25</b>
7.1	<u>BASES DE DATOS</u>	25
7.2	<u>MANEJO DE PERFILES</u>	26
7.3	<u>ASOCIACIÓN DE SCRIPTS A LOS CASOS PATOLÓGICOS</u>	26
7.4	<u>DESCRIPTORES</u>	26
7.5	<u>MULTIPLICIDAD DE CASOS PATOLÓGICOS POR PATOLOGÍA</u>	26
<b><u>8</u></b>	<b><u>DISEÑO DEL SISTEMA</u></b>	<b>28</b>
8.1	<u>ARQUITECTURA DE 3 CAPAS</u>	28
8.2	<u>ARQUITECTURA GENERAL</u>	30
8.3	<u>CLASIFICACIÓN DE IMÁGENES</u>	33
<b><u>9</u></b>	<b><u>IMPLEMENTACIÓN</u></b>	<b>35</b>
9.1	<u>LENGUAJES UTILIZADOS</u>	35
9.2	<u>MANEJO DE STRINGS</u>	35
9.3	<u>CLASES AUXILIARES</u>	36
9.4	<u>WRAPPER</u>	36
9.5	<u>INTERFAZ DE LÍNEA DE COMANDOS</u>	36
9.6	<u>CONSTANTES Y ARCHIVO DE PARÁMETROS</u>	37

9.7	<u>TRATAMIENTO DE IMÁGENES</u> .....	37
9.8	<u>CLASE DESCRIPTOR</u> .....	38
9.9	<u>DESCRIPTORES PROGRAMADOS</u> .....	38
9.10	<u>MANEJO DE XML</u> .....	40
9.11	<u>REUSABILIDAD</u> .....	41
9.12	<u>DEFINICIÓN DE LOS FORMATOS DE LOS DESCRIPTORES EN XML</u> .....	41
<b>10</b>	<b><u>AIM Y EL ANÁLISIS DE IMÁGENES MÉDICAS</u></b> .....	<b>44</b>
10.1	<u>TOMOGRAFÍA COMPUTARIZADA</u> .....	44
10.2	<u>PATOLOGÍAS IDENTIFICADAS EN LA TC CRANEAL</u> .....	45
10.3	<u>IMÁGENES UTILIZADAS</u> .....	46
10.4	<u>IDENTIFICACIÓN DE HEMATOMAS</u> .....	47
<b>11</b>	<b><u>EJEMPLO DE USO DEL SISTEMA</u></b> .....	<b>48</b>
<b>12</b>	<b><u>CONCLUSIONES</u></b> .....	<b>54</b>
12.1	<u>ASPECTOS POSITIVOS A DESTACAR</u> .....	55
12.2	<u>ASPECTOS NEGATIVOS A DESTACAR</u> .....	55
12.3	<u>POSIBLES MEJORAS</u> .....	57
12.4	<u>CONCLUSIÓN FINAL</u> .....	60
<b>13</b>	<b><u>LISTA DE ANEXOS</u></b> .....	<b>61</b>
<b>14</b>	<b><u>REFERENCIAS BIBLIOGRÁFICAS</u></b> .....	<b>62</b>
14.1	<u>REFERENCIAS EN INTERNET</u> .....	62
14.2	<u>LIBROS</u> .....	68
14.3	<u>HERRAMIENTAS UTILIZADAS</u> .....	69
<b>15</b>	<b><u>AGRADECIMIENTOS</u></b> .....	<b>70</b>

# Capítulo 1

## 1 PREFACIO

---

Durante el período que va desde Mayo de 2001 a Mayo de 2002 trabajamos, en el marco del proyecto de grado de la carrera Ingeniería en Computación de la Universidad de la República Oriental del Uruguay, en una idea que, por diversas circunstancias a nivel personal nos resultaba de particular interés: La elaboración de un motor de búsqueda de imágenes. Bajo esta idea fue que realizamos una investigación, definimos un alcance razonable para un proyecto de grado, y desarrollamos un producto y la presente documentación.

Durante nuestra investigación confirmamos la necesidad de un producto de estas características, pensando sobre todo en poder buscar imágenes de acuerdo a distintos criterios en la web, reduciendo la brecha de dificultad actualmente existente en este medio, entre lo que significa buscar texto y buscar imágenes. Nuestro anhelo es aportar a la construcción de un producto que el mercado necesita, y aún no se ha desarrollado, al menos con el nivel de madurez que lo requieren los usuarios.

Esperamos que la documentación de este proyecto reúna toda la información necesaria como base para la elaboración de motores de búsqueda de imágenes más inteligentes, y que el producto desarrollado, pese a que sabemos que a la fecha tiene una utilidad principalmente académica, siga siendo elaborado y creciendo.

# Capítulo 2

## 2 ORGANIZACIÓN DEL INFORME

Este capítulo tiene como objetivo brindar al lector una descripción de la estructura general del informe para que se adquiera una visión global, previo al desarrollo del mismo. El informe se estructura en diferentes capítulos, al comienzo de cada uno de los cuales se presenta una breve introducción a su contenido para que el lector conozca de antemano qué aspectos tratará. A continuación se resume el contenido de cada uno de los capítulos.

<b><u>1</u></b>	<b><u>PREFACIO</u></b> .....	<b>5</b>
<b><u>2</u></b>	<b><u>ORGANIZACIÓN DEL INFORME</u></b> .....	<b>6</b>
<b><u>3</u></b>	<b><u>PRESENTACIÓN DE LAS PARTES INVOLUCRADAS</u></b> .....	<b>8</b>
	Se presentan los distintos actores que participaron en la elaboración del proyecto.	
<b><u>4</u></b>	<b><u>INTRODUCCIÓN</u></b> .....	<b>9</b>
	Se realiza una introducción al proyecto, y una presentación de los objetivos que para el mismo se plantearon.	
<b><u>5</u></b>	<b><u>PROCESO DE DESARROLLO</u></b> .....	<b>16</b>
	Contiene una breve reseña de algunos aspectos que entendemos valen la pena destacar relativos al proceso de desarrollo.	

<b><u>6</u></b>	<b><u>ANÁLISIS DEL SISTEMA</u></b> .....	<b>18</b>
	Se explican las distintas etapas que abarcó el proceso de análisis del sistema, y se detalla el modelo conceptual al que se arribó.	
<b><u>7</u></b>	<b><u>DEFINICIÓN DE ALCANCE</u></b> .....	<b>25</b>
	Se describe la definición realizada respecto al alcance del proyecto, explicando las causas e implicancias de la misma.	
<b><u>8</u></b>	<b><u>DISEÑO DEL SISTEMA</u></b> .....	<b>28</b>
	Se describen los aspectos que consideramos más relevantes respecto a las decisiones de diseño en el sistema y la arquitectura finalmente alcanzada.	
<b><u>9</u></b>	<b><u>IMPLEMENTACIÓN</u></b> .....	<b>35</b>
	Se detallan aspectos que consideramos relevantes respecto a cómo se implementó el sistema y la algoritmia que éste utilizó.	
<b><u>10</u></b>	<b><u>AIM Y EL ANÁLISIS DE IMÁGENES MÉDICAS</u></b> .....	<b>44</b>
	Se describen algunas utilidades del sistema en cuanto al análisis de imágenes médicas, fruto de la investigación realizada en el área.	
<b><u>11</u></b>	<b><u>EJEMPLO DE USO DEL SISTEMA</u></b> .....	<b>48</b>
	Se mostrará el producto obtenido a través de un ejemplo práctico de uso del mismo.	
<b><u>12</u></b>	<b><u>CONCLUSIONES</u></b> .....	<b>54</b>
	Se listan los anexos que se adjuntan al presente informe.	
<b><u>13</u></b>	<b><u>LISTA DE ANEXOS</u></b> .....	<b>61</b>
	Se listan los anexos que se adjuntan al presente informe.	
<b><u>14</u></b>	<b><u>REFERENCIAS BIBLIOGRÁFICAS</u></b> .....	<b>62</b>
	En este capítulo detallaremos las referencias bibliográficas utilizadas tanto en este informe final como en los anexos.	
<b><u>15</u></b>	<b><u>AGRADECIMIENTOS</u></b> .....	<b>70</b>
	Nombraremos a tan solo algunas de las personas que recordamos como apoyo importante en distintas etapas del proyecto.	

# Capítulo 3

## 3 PRESENTACIÓN DE LAS PARTES INVOLUCRADAS

**En este capítulo se presentarán los distintos actores que participaron en la elaboración del proyecto.**

### 3.1 ORGANIZACIÓN PARA LA QUE SE TRABAJÓ

El proyecto fue realizado con la ayuda de personal del Hospital de Clínicas, en el marco de un conjunto de proyectos que buscan ayudar a la aplicación de sistemas informáticos en el Hospital.

### 3.2 INVOLUCRADOS POR PARTE DEL HOSPITAL DE CLÍNICAS

Dra. Cristina Balbiani  
Dr. Gustavo Stellardi

### 3.3 TUTORES EN FACULTAD DE INGENIERÍA

Ing. Alvaro Pardo  
Ing. Raúl Ruggia

### 3.4 INTEGRANTES DEL GRUPO

Pablo Flores  
Marcelo Rodríguez

C.I. 3.340.337-0  
C.I. 1.840.211-5

[pflores@netlabs.com.uy](mailto:pflores@netlabs.com.uy)  
[msrr@montevideo.com.uy](mailto:msrr@montevideo.com.uy)

# Capítulo 4

## 4 INTRODUCCIÓN

---

**En este capítulo se realizará una introducción al proyecto, y una presentación de los objetivos que para el mismo se plantearon.**

Analizador de Imágenes Médicas (en adelante AIM) es el nombre que le hemos dado al producto de este Taller 5. Sin embargo, el proyecto sigue una propuesta más ambiciosa que el trabajo exclusivo con imágenes médicas, la cual describiremos a continuación y servirá como punto de partida para explicar la concepción y los objetivos del proyecto.

### 4.1 CONCEPCIÓN INICIAL DEL PROYECTO

Actualmente, la búsqueda de imágenes en bases de datos se realiza principalmente mediante consultas en bases de datos relacionales tradicionales, las cuales contienen conjuntos de palabras claves asociadas a las imágenes, ingresadas por humanos. Básicamente este es el método que utilizan Google, Yahoo e ImageBank, entre otros. Nuestra propuesta inicial tenía como objetivo investigar mecanismos que mejoren los motores de búsqueda de imágenes en bases de datos, permitiendo que el ingreso de las palabras claves antes mencionado pueda ser sustituido por la ejecución de procesos automáticos de clasificación de imágenes y extracción de *descriptores*.

La propuesta inicial se encuentra documentada en Acta 0 (ver Anexo I: Actas de reuniones), y en ella se pone de manifiesto nuestra intención inicial de investigar el estado del arte en tratamiento de imágenes digitales y, a partir de él, definir e implementar un motor de búsqueda que logre identificar distintos tipos de imágenes y reconocer patrones, de modo de poder realizar consultas sobre bases de datos de imágenes a partir de otras imágenes. La investigación realizada se encuentra documentada en el Anexo A: Administración y proceso de desarrollo, Capítulo Investigación.

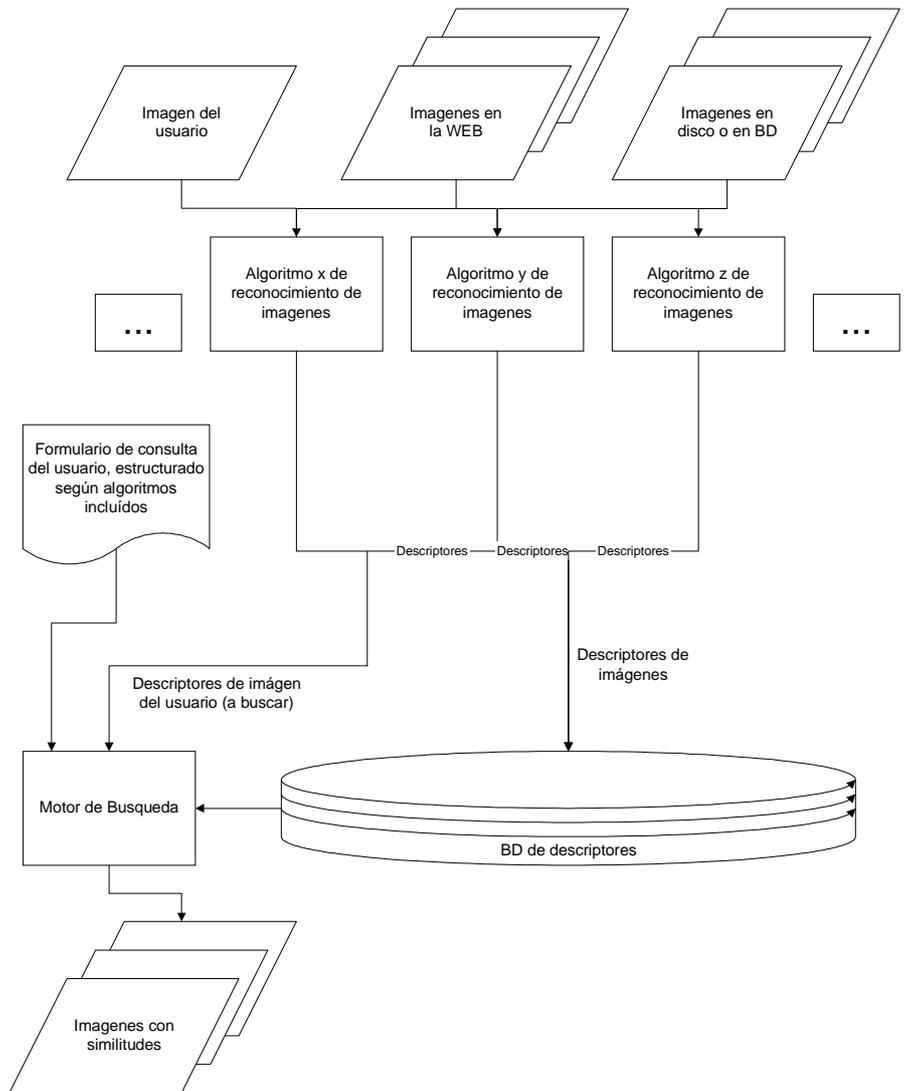
Además del trabajo de investigación, al inicio del proyecto planteamos una idea general del sistema, que pasamos a describir.

Como entradas, utilizar imágenes en la WEB y/o en una base de datos en disco, que sirvan como fuente de búsqueda y, por otra parte, proporcionar una imagen de la cual se desea buscar sus similares.

Todas estas imágenes son procesadas por distintos algoritmos que extraen sus descriptores, los cuales son bloques de información extraída a partir de las imágenes, según distintos criterios.

Estos se guardan en una BD de descriptores para servir de entrada al motor de búsqueda, el cual realiza los matcheos entre los descriptores presentes en la BD y los de la imagen a buscar.

El usuario puede seleccionar mediante el formulario de consulta los descriptores que desea incluir y los que no, en la búsqueda que realiza el motor. Las imágenes que cumplan con cierto nivel de similitud en sus descriptores, son retornadas como salida del proceso.



#### 4.2 ESPECIALIZACIÓN EN IMÁGENES MÉDICAS

Esta idea original, a la hora de la presentación de los proyectos de final de carrera, varió ligeramente, enfocándose a la búsqueda de imágenes radiológicas, en un proyecto enmarcado dentro de los dos que se realizarían con el Hospital de Clínicas (ver Anexo G: Proyecto planteado). Esto llevó a los siguientes cambios sobre la definición del proyecto:

- Tratamiento de imágenes únicamente en blanco y negro

- Posibilidad de trabajar con imágenes en formato DICOM [7]
- Nomenclatura utilizada
- Necesidad de investigación acerca de análisis médicos imagenológicos.

### 4.3 OBJETIVOS DEL PROYECTO

Dado que el proyecto abarca una amplia gama de campos de trabajo de diferente naturaleza, debimos decidir en qué puntos concentrar nuestra atención para que el proyecto no se sobrepasara del alcance razonable para un Taller 5, por lo cual los objetivos principales planteados fueron:

- Estudiar el estado del arte acerca de las tecnologías actualmente utilizadas para búsquedas de imágenes
- Diseñar un sistema con una arquitectura extensible, que permita realizar búsquedas en imágenes en base a la extracción de descriptores
- Implementar un programa que sirva como prueba de la posibilidad de realizar un sistema del estilo, y sirva como base de un sistema más amplio e inteligente de búsqueda y reconocimiento de imágenes.

Queremos resaltar de los objetivos antes mencionados, la importancia que le dimos a la definición de una arquitectura extensible. Dado que, a partir de imágenes, se pueden obtener descriptores de los más variados tipos (valores numéricos, valores discretos, ecuaciones, conjuntos de distintos elementos, etc.), resultó fundamental diseñar un sistema que brindara una interfaz potente pero no restringida para la creación de descriptores y la definición a partir de ellos de métodos de búsqueda.

Por otra parte, al concentrar el esfuerzo en los objetivos antes mencionados, somos concientes de haber dejado otros puntos de lado, los cuales pueden ser mejorados en futuras versiones del sistema, como ser:

- Tratamiento de bases de datos de imágenes
- Tratamiento de bases de datos XML
- Investigación de otros algoritmos avanzados para preprocesamiento, segmentación y generación de descriptores a partir de imágenes.

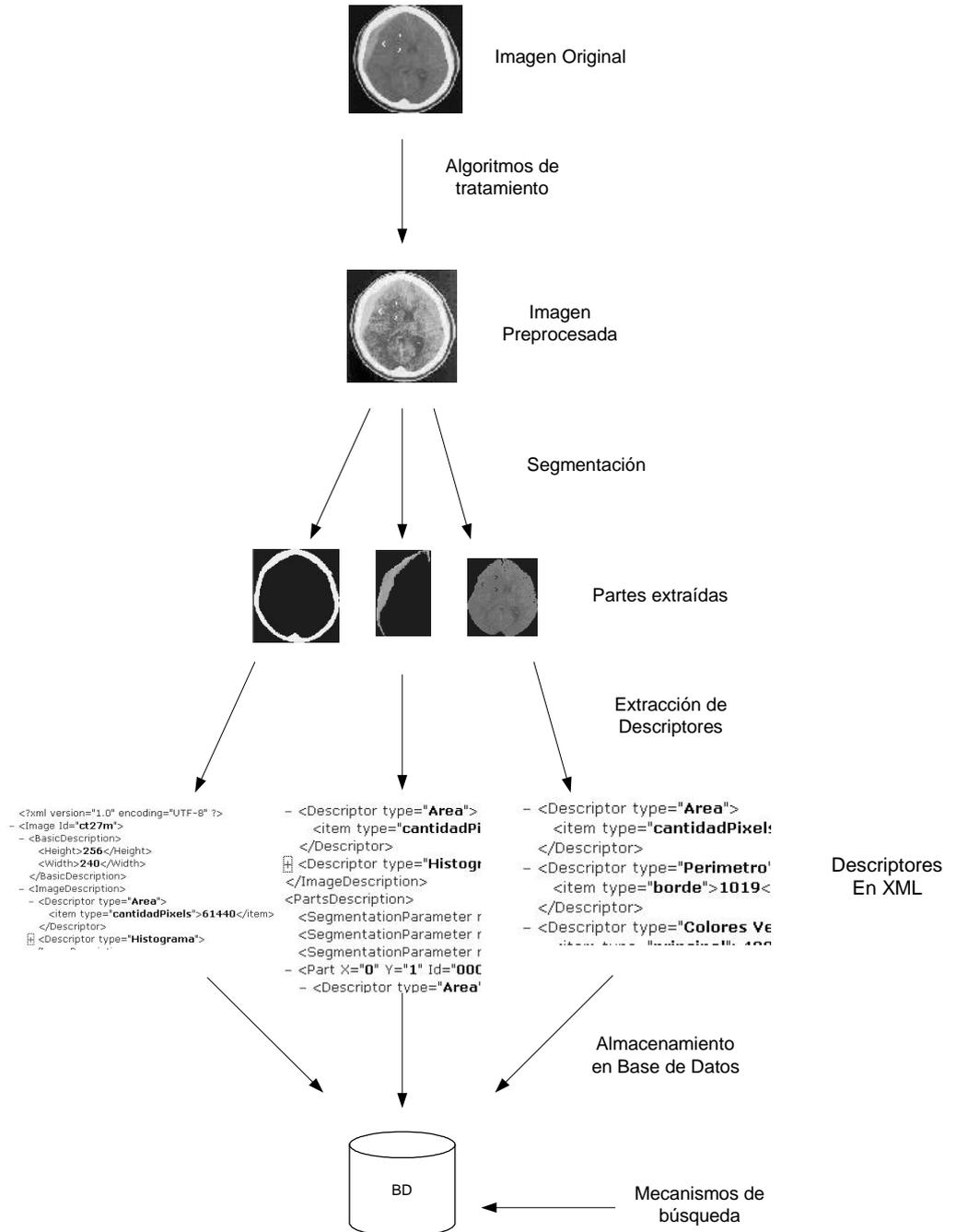
En el capítulo Conclusiones, sección Posibles mejoras se incluye una reseña de áreas en las cuales el proyecto puede ser enriquecido.

### 4.4 CONCEPCIÓN DEL SISTEMA

El proceso de desarrollo sacó a la luz varios detalles sobre el funcionamiento que debía tener el sistema, pero a grandes rasgos el producto desarrollado puede ser dividido en los siguientes grandes bloques:

- Adquisición y tratamiento de imágenes
- Proceso de clasificación:
  - División de imágenes en regiones (segmentación)

- Extracción de descriptores de imágenes y regiones, y almacenamiento de los mismos en una base de datos
- Comparación de imágenes y regiones a partir de la comparación de sus descriptores
- Mecanismos para ampliar el espectro de algoritmos de tratamiento de imágenes y de extracción de descriptores



Más adelante en este informe se detallarán las consideraciones tomadas en cuenta a lo largo del proceso de desarrollo. A continuación, se detalla una breve reseña de las tecnologías utilizadas, para luego describir más en detalle cada uno de los bloques antes mencionados.

#### 4.5 RESEÑA DE LAS TECNOLOGÍAS UTILIZADAS

Con el objetivo de brindar un panorama general que ayude a comprender mejor los puntos que siguen, se detallarán brevemente las tecnologías utilizadas. La discusión y explicación completa de las tecnologías evaluadas y usadas, se encuentra en el Anexo F: Analisis de las tecnologías usadas.

- El motor del sistema fue programado totalmente en Visual C++ 6.0, encapsulado como un objeto COM, utilizando funciones pertenecientes al Standard C++, a no ser por las específicas de la tecnología COM, que se encuentran encapsuladas en una clase Wrapper.
- La interfaz de usuario fue programada en Visual Basic 6.0.
- Para el tratamiento de imágenes se utilizó la librería ImageMagick (free).
- Los descriptores fueron guardados en archivos XML, utilizándose para su tratamiento la tecnología DOM y la librería Xerces (free).
- La base de datos consiste en archivos guardados en una estructura de directorios.

#### 4.6 ADQUISICIÓN Y TRATAMIENTO DE IMÁGENES

Lo primero que debe realizar un sistema de tratamiento de imágenes es adquirirlas. Dado que el tomógrafo del Hospital de Clínicas tiene un sistema para guardar directamente las imágenes en disco, el mecanismo provisto por el sistema es un simple procedimiento que levanta de disco imágenes en cualquiera de los formatos tratados por ImageMagick, entre los cuales se encuentra el formato DICOM [7] que graba el tomógrafo (ver Anexo F: Analisis de las tecnologías usadas, capítulo Librerías de tratamiento de imágenes).

Una vez adquirida la imagen, la pasaremos a llamar “**Imagen Objeto de Estudio**” (IOE), y sobre ella es que trabajaremos.

El primer procedimiento que podemos realizar sobre la IOE es la aplicación de algoritmos de tratamiento de imágenes (preprocesamiento). Esto consiste en modificar la imagen, ya sea para visualizarla mejor, como para mejorar el resultado de los algoritmos que luego se correrán en el proceso de clasificación.

#### 4.7 DIVISIÓN DE IMÁGENES EN REGIONES (SEGMENTACIÓN)

La segmentación consiste en realizar una partición de la imagen original en partes constituyentes [30, pág 447] o regiones disjuntas entre ellas, y cada una con una única componente conexas.

El proceso está separado en dos etapas:

- Umbralización (thresholding) de la imagen: Obtención de conjuntos de pixels con colores similares.
- Etiquetado (labeling): Definición de partes detectando las componentes conexas.

Luego de este proceso, quedan asociadas a la imagen un conjunto de partes, que se reconocen por los siguientes datos:

- (x,y) de comienzo
- (ancho,alto) de la región
- máscara que indica dentro del rectángulo definido por (x,y, ancho,alto), cuáles pixels pertenecen a la región y cuáles no.

Hay más detalles acerca de los algoritmos usados para la segmentación en el capítulo Implementación.

#### 4.8 EXTRACCIÓN DE DESCRIPTORES DE IMÁGENES Y REGIONES Y ALMACENAMIENTO DE LOS MISMOS EN UNA BASE DE DATOS

Para la presente versión del sistema, se obtienen los siguientes descriptores de imágenes y regiones:

- **Área.** Medida como la cantidad de pixels total de una región.
- **Perímetro.** Cantidad de pixels que componen el contorno de la imagen. Se utilizó un algoritmo propio (ver Capítulo Implementación en este informe, y Anexo E: Documentación técnica) para realizar una recorrida por el borde de la región. A diferencia del propuesto por González [30 pag. 574], este algoritmo de recorrida de la región sirve para la obtención de otros descriptores.
- **Histograma.** Indica el porcentaje de los pixels de la imagen que tienen cada nivel de gris. Para resumir el resultado, se agrupan los pixels en rangos de intervalos constantes.
- **Color promedio.** Indica nivel de gris promedio correspondiente a una región.
- **Colores vecinos.** Indica el nivel de gris promedio correspondiente a las dos regiones vecinas con más pixels bordeando determinada región.
- **Factor de forma.** Indicador que se obtiene del cociente entre perímetro al cuadrado y el área. Indica, en forma invariable respecto al tamaño de la imagen, qué tanto se aproxima la región a un disco (en el caso de un disco, tiene valor mínimo)[30, pág 545]

Los detalles acerca de los algoritmos usados para el cálculo de Descriptores se encuentran en el capítulo Implementación.

Una vez segmentada la imagen, se recorre cada región aplicándole los algoritmos correspondientes a cada descriptor, para luego unirlos en una “descripción de la parte” y luego en una “descripción de la imagen”, la cual une las descripciones de todas las partes.

El formato en que se almacenan estos descriptores es XML. En el transcurso de la investigación, descubrimos información muy alentadora sobre los objetivos de nuestro proyecto, los cuales se encontraban incluidos en un proyecto de gran magnitud, llamado MPEG-7, que consiste en la definición de un estándar para la definición de descriptores a partir de todo tipo de contenido multimedia (imágenes, sonido y video). La investigación acerca de la definición de dicho estándar (que a la fecha aún sigue en curso) se encuentra documentada en el Anexo F: Análisis de tecnologías utilizadas, sección MPEG7. Debido a que el mismo aún se encuentra en fase de elaboración, fue imposible obtener la especificación de los descriptores utilizados, por lo cual nos limitamos a definir una sintaxis “MPEG-7-like” para los descriptores. La especificación de la sintaxis de estos descriptores se detalla más adelante, en el capítulo Implementación.

#### 4.9 COMPARACIÓN DE IMÁGENES Y REGIONES A PARTIR DE LA COMPARACIÓN DE SUS DESCRIPTORES

Esta es una componente fundamental para el motor de búsqueda, y por ello es también que deja campo abierto para muchísimas extensiones del sistema. En la presente versión consiste en:

- Mecanismo de comparación de descriptores uno a uno, devolviendo un porcentaje de similitud
- Procedimiento para calcular un porcentaje de similitud en base a la similitud obtenida de la comparación de cada uno de los descriptores, y la relevancia que se le da a cada descriptor.

Se estudió durante el análisis de requerimientos y la definición del modelo conceptual la posibilidad de obtener descriptores que impliquen varias regiones, pero en la definición de alcance del sistema (más detallado en el capítulo Definición de alcance) se decidió no incluir dicha característica, con el objetivo de acotar tiempos de implementación.

Hay más detalles acerca de los métodos de comparación de Descriptores en el capítulo Implementación.

#### 4.10 MECANISMOS PARA AMPLIAR EL ESPECTRO DE ALGORITMOS DE TRATAMIENTO DE IMÁGENES Y DE EXTRACCIÓN DE DESCRIPTORES

La arquitectura del sistema tomó especial cuidado en permitir ampliar las siguientes secciones del sistema:

- Algoritmos de preprocesamiento de imágenes
- Algoritmos de segmentación
- Algoritmos de extracción de descriptores

En el Anexo G: Documentación técnica se encuentran detallados algunos procedimientos a seguir para enriquecer el sistema, en particular en la documentación de las clases Descriptor e Imagen.

# Capítulo 5

## 5 PROCESO DE DESARROLLO

---

**En este capítulo se realizará una breve reseña de algunos aspectos que valen la pena destacar relativos al proceso de desarrollo.**

Hasta aquí se realizó una reseña de los aspectos más importantes del proyecto. En adelante, se describirán en detalle los aspectos relativos a las distintas etapas del proceso de desarrollo, comenzando por algunos aspectos relativos a la administración y el proceso de desarrollo en si mismo.

La descripción completa del proceso se encuentra en el Anexo A: Administración y proceso de desarrollo. Aquí simplemente haremos un resumen de datos recabados, e información importante para comprender el resto del informe principal.

### 5.1 TAMAÑO DEL SISTEMA

#### Programación en VB

- 8 formularios
- 2 módulos
- 3 clases
- 4.260 líneas de código (no incluye generadas automáticamente por VB a partir de formularios diseñados)

#### Programación en VC++

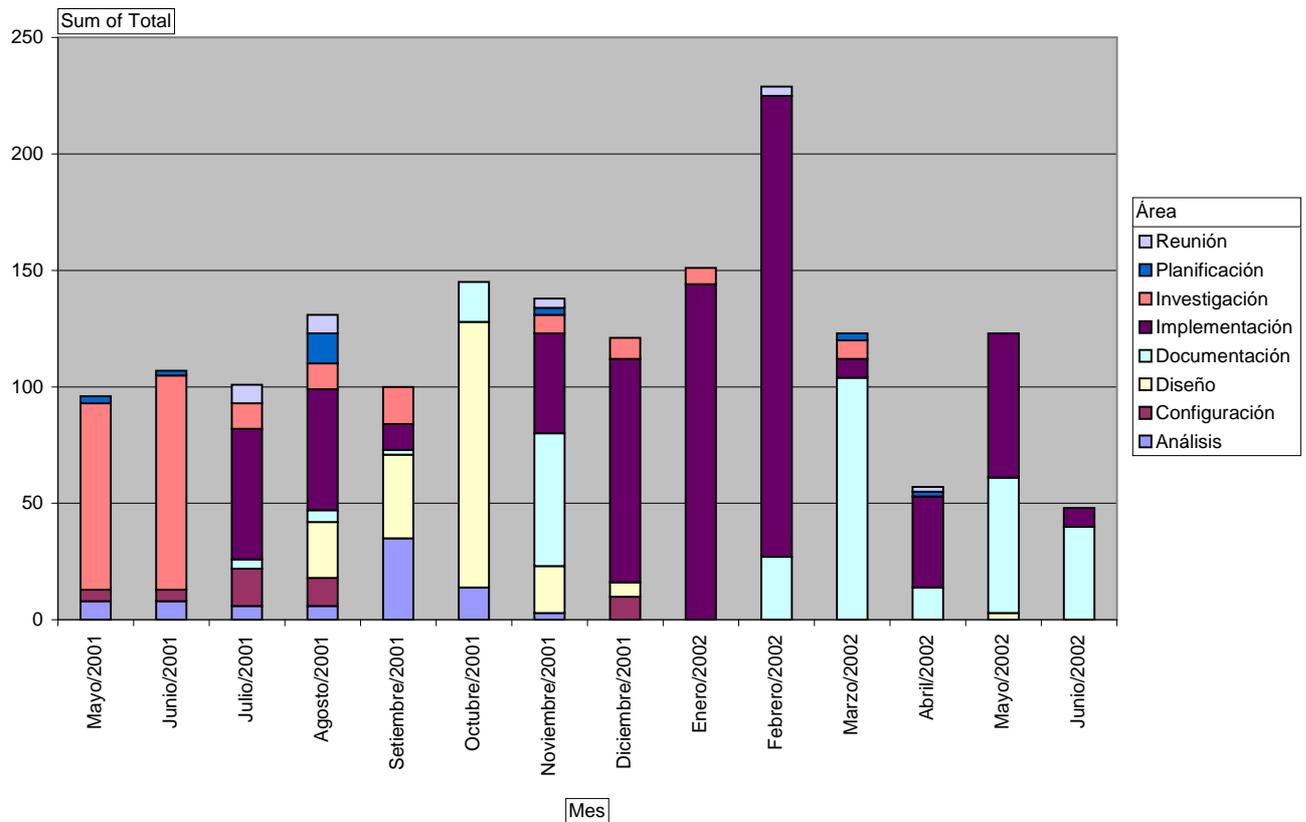
- 45 clases
- 14.468 líneas de código

#### Cantidad de líneas de código total del sistema

- 18.728 líneas de código

5.2 DISTRIBUCIÓN DEL ESFUERZO A LO LARGO DEL PROYECTO

El proyecto comenzó en Mayo de 2001, y se extendió hasta Junio de 2002. El siguiente gráfico muestra la distribución de la carga de trabajo a lo largo de dicho período:



El proyecto llevó un total de 1670 horas/hombre.

# Capítulo 6

## 6 ANÁLISIS DEL SISTEMA

---

**En este capítulo se explicarán las distintas etapas que abarcó el proceso de análisis del sistema, y se detallará el modelo conceptual al que se arribó.**

Esta tarea no fue sencilla dado que, por razones detalladas en el Anexo A: Administración y proceso de desarrollo, gran parte del tiempo del proyecto transcurrió sin tener claramente definido el producto que en el Hospital de Clínicas estaban interesados en obtener.

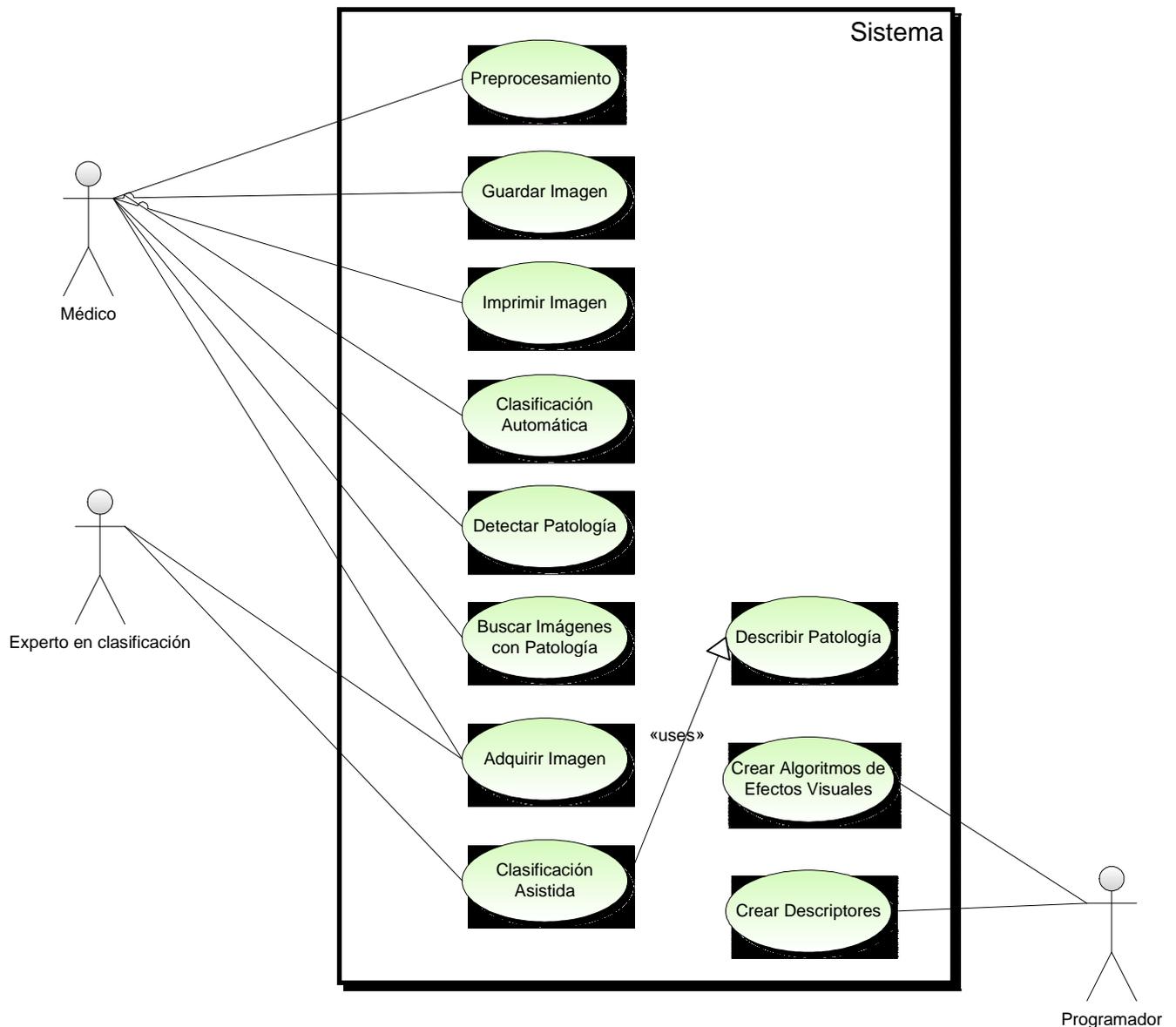
Una vez definido el sistema a desarrollar, surgieron muchísimos desafíos. No solo por el hecho de haber puesto nuestro máximo esfuerzo en guiarnos por el estándar propuesto por Craig Larman de documentación UML y utilización de patrones de diseño [28], sino sobre todo por la gran cantidad de detalles que traía consigo el sistema, que en apariencia era sencillo: Al comenzar a hilar fino en todos los conceptos que tiene implícitos, y el alto nivel de diversificación que requiere, surgieron muchos detalles, cada uno de los cuales fue haciendo aumentar el tamaño del sistema.

Durante el proceso de desarrollo, se realizaron ajustes al alcance. Sin embargo, dado que resulta de gran importancia el permitir la ampliación del mismo, para el análisis de requerimientos se realizó la especificación sin tomar en cuenta los recortes, de modo que la arquitectura definida contempla futuras extensiones al sistema. A continuación explicaremos los pasos llevados a cabo en la etapa de análisis, explicando la modelación del sistema completo. En la próxima sección detallaremos el alcance que decidimos implementar en el taller. La documentación completa del análisis de requerimientos se encuentra en el Anexo B: Análisis de requerimientos.

### 6.1 CASOS DE USO

Detectamos los siguientes agentes externos al sistema (actores):

- **Médico:** Es el que se encarga de utilizar el programa para observar imágenes, realizar procedimientos automáticos y, sobre todo, para llevar a cabo búsquedas.
- **Experto en clasificación:** Este tipo de usuario se dedica a asistir al sistema a clasificar imágenes patológicas y definir los elementos que señalan la existencia de patologías.
- **Programador:** Surgen tomando en cuenta la posibilidad de agregar generadores de descriptores y algoritmos de segmentación y preprocesamiento (explicados más adelante) al sistema. Éste programa módulos que se incorporan al sistema para que cuente con mejores elementos de búsqueda.



A grandes rasgos, el sistema debe permitir las siguientes funcionalidades:

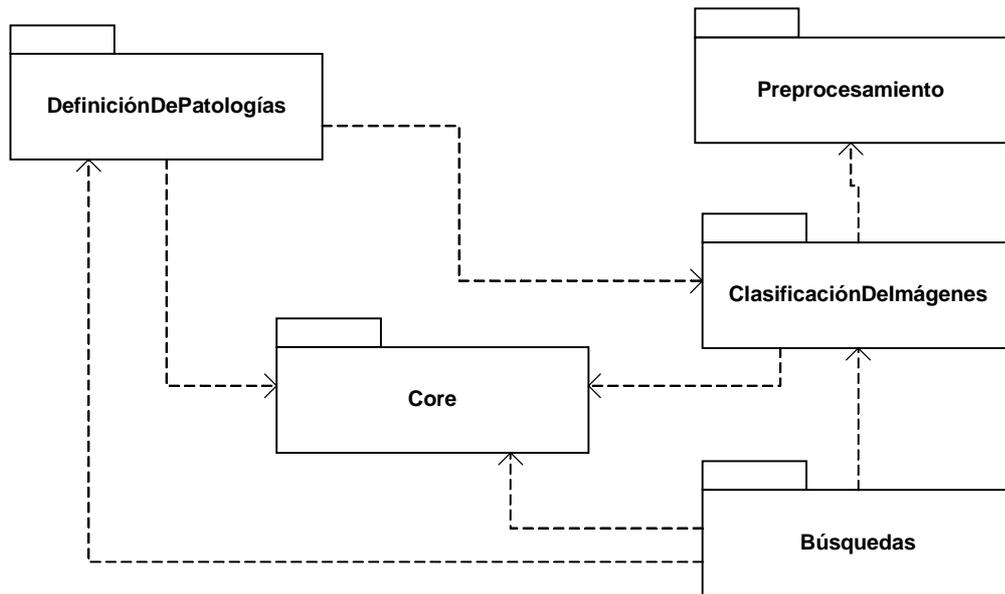
- **Adquirir imagen:** Trabajar con una imagen generada por el tomógrafo (El tomógrafo del Hospital de Clínicas brinda la posibilidad de guardarlas en disco en formato digital en forma nativa, generando archivos DICOM).
- **Preprocesamiento:** Aplicar algoritmos que permitan mejorar la visualización, clasificación o búsqueda de imágenes.
- **Clasificación Automática/Asistida:** Realización de un proceso, automático o asistido por un experto clasificador, sobre una imagen previamente adquirida que realice un análisis completo de la imagen extrayendo de esta forma descriptores, los cuales deben permanecer almacenados para una posterior búsqueda. Esto se puede realizar desde una interfaz interactiva o de línea de comandos.
- **Describir Patología:** Enseñar al sistema los aspectos relevantes de una patología a partir de una imagen patológica de ejemplo, para luego realizar búsquedas.
- **Detectar Patología:** En una imagen adquirida, buscar descriptores que se asemejen a los correspondientes en la descripción de una o varias patologías seleccionadas por el usuario, para así identificar regiones con determinado porcentaje de probabilidad de corresponder a una patología.
- **Buscar imágenes con patología:** A partir de una patología, buscar dentro de un conjunto de imágenes, regiones que tengan descriptores con cierto nivel de probabilidad de corresponder a la patología seleccionada.
- **Agregar descriptores:** El sistema debe tener abierta la posibilidad de incluir nuevos algoritmos de extracción de descriptores a partir de imágenes.
- **Agregar algoritmos de preprocesamiento y segmentación:** El sistema debe tener abierta la posibilidad de incluir nuevos algoritmos de preprocesamiento y de segmentación sobre las imágenes a partir de nuevos algoritmos desarrollados.

En el Anexo B: Análisis de requerimientos se detalla el funcionamiento de cada uno de ellos.

## 6.2 MODELO CONCEPTUAL

El estudio del sistema lo dividimos en los siguientes grandes módulos

- Core
- Preprocesamiento
- Clasificación de imágenes
- Definición de patologías
- Búsquedas



En las próximas secciones describiremos cada uno de ellos.

### 6.3 PREPROCESAMIENTO

Son algoritmos aplicables a una imagen con el objetivo de prepararla mejor para su visualización y/o clasificación. Entre éstos algoritmos están los de llevar imágenes color a blanco y negro, aumento de contraste, etc.

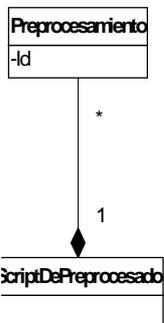
Dado que la clasificación que se realice de una imagen depende de los algoritmos de preprocesamiento que se le hayan aplicado, se incluye la posibilidad de guardar un “script”, el cual contendrá los algoritmos a aplicar y los parámetros necesarios para cada uno de ellos. A la hora de la implementación, éste quedará generado en formato XML como forma de mantener un estándar para todo el sistema.

### 6.4 CLASIFICACIÓN DE IMÁGENES

El sistema debe proveer una opción para clasificar automáticamente una imagen. Esto consiste en:

- Asociarle un script que detalle el preprocesamiento que se le aplicó
- Segmentarla
- Guardar los descriptores de todas sus partes en el repositorio de descriptores, con el objetivo de usarlas para posteriores búsquedas.

Por otra parte, en el caso de que un médico experto se presente como usuario del sistema, éste debe poder hacer una clasificación más precisa de las imágenes. En este caso, al igual que en la clasificación automática, se asociará a la imagen el script de efectos visuales aplicados y se segmentará





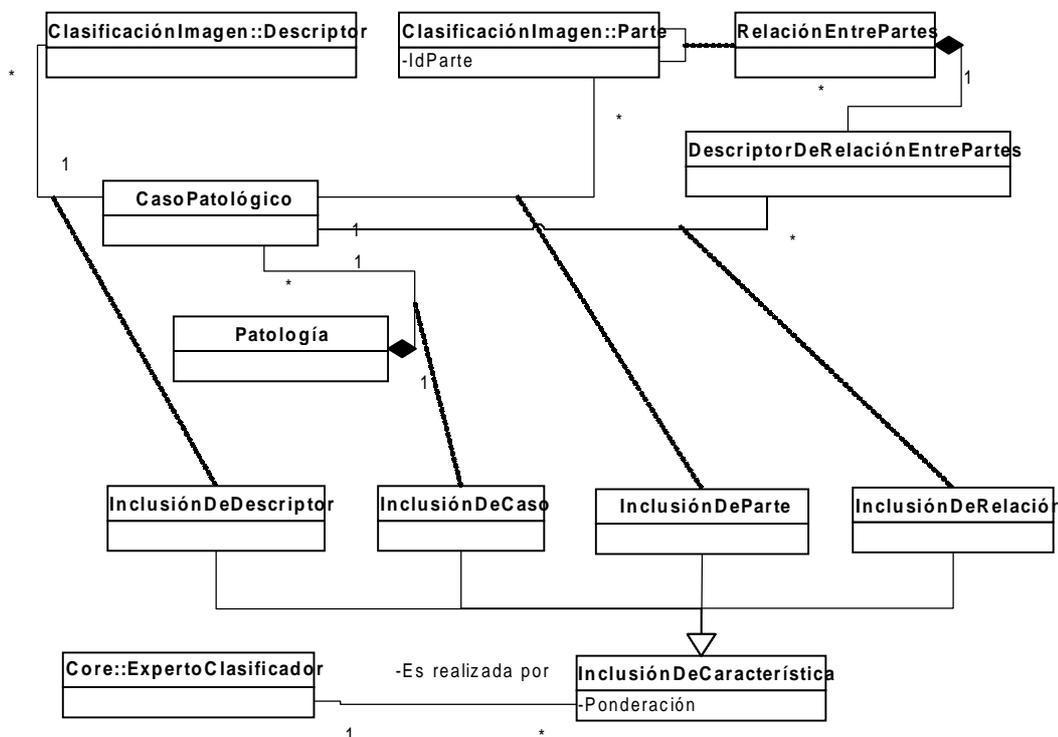
- En este último caso, el experto clasificador también puede seleccionar las partes relevantes para ser guardadas en el repositorio.

## 6.5 DEFINICIÓN DE PATOLOGÍAS

En caso que una imagen clasificada contenga patologías, se le debe “enseñar” al sistema las mismas, de modo que luego las pueda identificar en forma automática. A este proceso le llamamos **definición de patologías**.

Este proceso consiste en marcar las partes de una imagen que definen la patología, ponderar su presencia en la misma (% de relevancia de que aparezca), indicar la relación entre ellas y ponderarla (“la parte 005 debe estar limitando con la parte 014 y es un 70% importante que lo esté”) e indicar la ponderación de cada descriptor de cada parte seleccionada (ejemplo: ColorPromedio 65%, Área 20%, etc.). Luego de la definición de cada caso patológico, también se debe indicar cuán representativo de la patología lo es, también ponderándolo respecto a otros casos patológicos (si los hubiere). Todas estas definiciones y ponderaciones generaran descriptores que serán almacenados en el repositorio de patologías.

Veamos un extracto del modelo conceptual asociado:



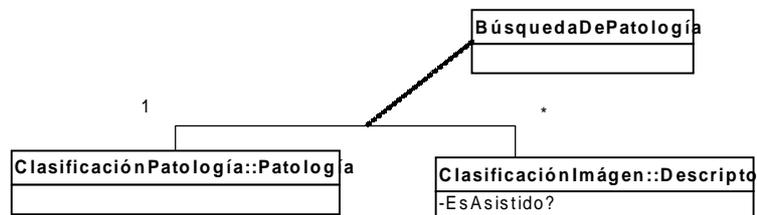
- Hay 4 características que se toman en cuenta para definir una patología:
  1. Cada patología se define por un conjunto de casos patológicos

2. Dentro de cada caso, solo algunas de sus partes indican la presencia de patologías
  3. En algunos casos, para que realmente exista una patología, se deben tomar en cuenta varias partes de la imagen, las cuales deben cumplir cierta relación entre ellas (de posición, color, etc.)
  4. Finalmente, no todos los descriptores de una parte indican por igual la presencia de una patología. En algunos casos, por ejemplo, puede ser importante el color, pero no la forma.
- El experto clasificador realiza el proceso de incluir cada uno de los rasgos que componen la patología, dándole una ponderación para mejorar la precisión de las búsquedas posteriores.

### 6.6 BÚSQUEDA DE PATOLOGÍAS

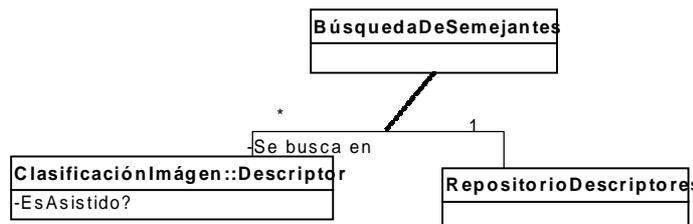
Es la principal funcionalidad del sistema. Aportando una imagen e indicando una o varias de las patologías clasificadas, el sistema busca en el repositorio si la imagen presenta alguna de las patologías: cuando encuentra las partes “sospechosas” en la imagen las remarcará, indicará la patología encontrada, y el porcentaje de probabilidad de que se trate de ésta.

Para ello el proceso a seguir es aplicarle los scripts que indica cada imagen almacenada en el repositorio, segmentarla, extraer los descriptores y compararlos con los de las partes clasificadas de la imagen que corresponde a cada caso patológico, retornando la presencia o no de la patología indicando en caso de corresponder en qué sectores de la imagen.



### 6.7 BÚSQUEDA DE PATOLOGÍA EN IMÁGENES

Esta funcionalidad consiste en, dada una patología, buscar entre un conjunto de imágenes clasificadas, casos “sospechosos”. Básicamente esta opción es análoga a la de búsqueda de patologías, pero se realiza sobre varias imágenes.



# Capítulo 7

## 7 DEFINICIÓN DE ALCANCE

**En este capítulo se describirá la definición realizada respecto al alcance del proyecto, explicando las causas e implicancias de la misma.**

Una vez culminado el análisis de requerimientos detectamos que el sistema se mostraba bastante ambicioso en cuanto a su alcance: la implementación de todas las funcionalidades planteadas puede llevar un tiempo de programación bastante superior al alcance del proyecto de Taller 5, por lo cual el recorte de la funcionalidad se convirtió en una necesidad. Detallamos a continuación cada uno de los aspectos en que se decidió restringir el alcance del sistema.

### 7.1 BASES DE DATOS

Los descriptores son guardados en formato XML. Esto lleva a que los procesos de búsqueda, para ser óptimos, deberían contar con un DBMS que soporte en forma nativa XML y una estructuración especial para este tipo de búsquedas. Se realizó una pequeña investigación de posibles DBMSs que cubran esta necesidad: hasta ese momento, de libre distribución, no había ningún DBMS que soporte nativamente XML, ni manejo de imágenes. Existen sí opciones comerciales como Tamino u Oracle, pero pensando en la posibilidad de ponerlos en producción para el Hospital de Clínicas, estas opciones son inviables. Por otra parte, dado que encontramos que el caso de uso más importante para el sistema es la detección de patologías ya clasificadas en una imagen, consideramos que se trataba de un aspecto que no era central para nuestros objetivos del taller, por lo cual decidimos utilizar directamente archivos XML en disco. Sin embargo, tomamos especial preocupación en el diseño en separar una capa de persistencia que pueda ser fácilmente reemplazada por otra que sí trate con un DBMS.

En resumen, no usamos un DBMS, pero programamos el producto de modo de poder incluirlo fácilmente, modificando solo las clases de la capa de persistencia en los accesos puntuales.

## 7.2 MANEJO DE PERFILES

En el modelo conceptual surge la idea de un manejo de perfiles (experto clasificador, médico, administrador). No nos propusimos como objetivo primario la implementación de los mismos ni de técnicas de seguridad de acceso a los datos; esto no es un problema para la institución ya que no existen problemas de seguridad asociados a la aplicación, ni se piensa en que el sistema tenga un uso exhaustivo en producción, sino que se utilizará a efectos académicos y de consulta puntual.

## 7.3 ASOCIACIÓN DE SCRIPTS A LOS CASOS PATOLÓGICOS

El preprocesamiento de la imagen la haría el médico previo a la búsqueda. No consideramos esencial toda la definición y almacenamiento de los scripts de algoritmos de preprocesamiento, por lo cual decidimos recortarlo en esta versión.

## 7.4 DESCRIPTORES

La relación entre las partes es una de los ítems más costosos de implementar, como también el que puede dar más exactitud al sistema. Limitamos los descriptores únicamente a los correspondientes a las partes en sí.

Para superar esta limitante, se debe agregar a la arquitectura diseñada el concepto estudiado en el análisis del sistema de “relación entre partes”. Esto llevaría a agregar complejidad a las clases Parte, DescripcionParte y Descriptor (referidas más adelante, en el capítulo Diseño del sistema).

Otro cambio en el alcance respecto al uso de descriptores, es que no se permiten generar descriptores asistidos, creados por el experto clasificador para las partes de una imagen. Consideramos que no era de gran importancia, por lo cual su relación costo-beneficio no lo justificaba.

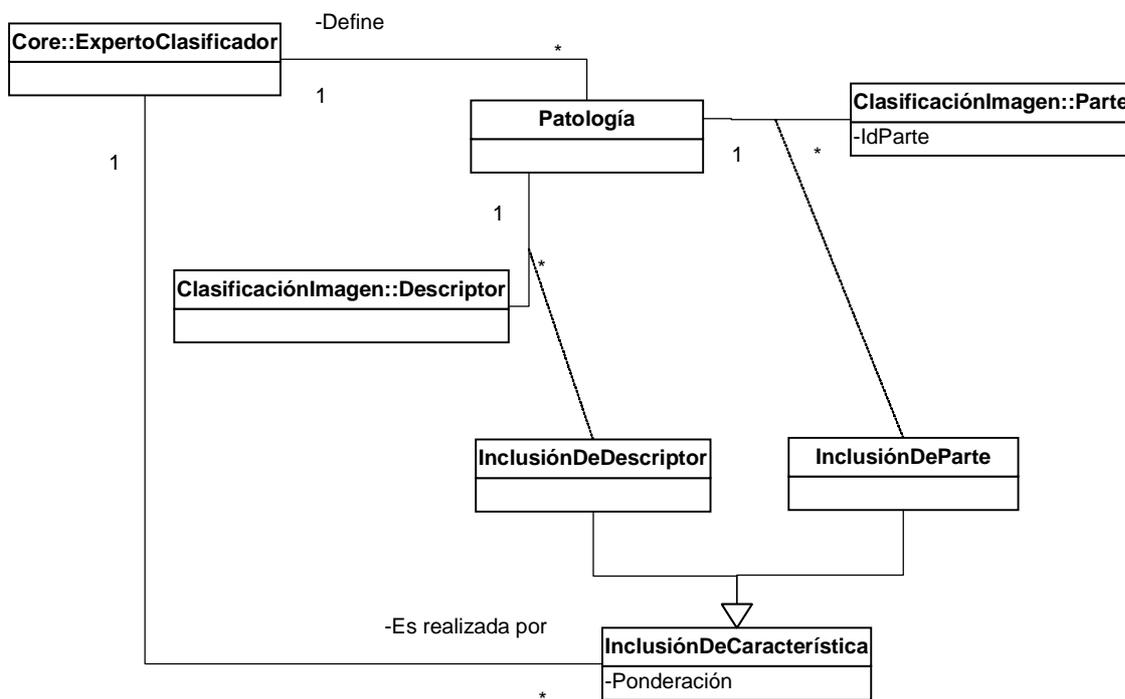
En resumen, resulta claro que para dar más complejidad a los descriptores se debe realizar un importante esfuerzo de ampliación del sistema, pero dado que en la etapa de análisis de requerimientos se tuvo en cuenta brindar estas posibilidades, esto no resultará incompatible con la arquitectura obtenida para la presente versión.

## 7.5 MULTIPLICIDAD DE CASOS PATOLÓGICOS POR PATOLOGÍA.

Existirá sólo un caso patológico por patología, lo cual también elimina su ponderación dentro de cada patología.

En futuras versiones del sistema, esta no parece ser una limitante importante. De hecho, la multiplicidad de casos patológicos es solo agregar un nivel más en la definición de la patología. Por otra parte, tal como se describirá en la sección Posibles Mejoras, el desafío más interesante consiste en que el sistema automáticamente detecte los patrones que corresponden a las patologías a partir del análisis de un conjunto suficientemente amplio de casos patológicos.

Con los cambios antes detallados en los descriptores y la definición de patologías, el modelo conceptual del sistema implementado para el módulo de clasificación queda de la siguiente manera:



La funcionalidad total con que cuenta finalmente el sistema se describe en el Manual de usuario.

# Capítulo 8

## 8 DISEÑO DEL SISTEMA

---

**En este capítulo se describirán los aspectos que consideramos más relevantes respecto a las decisiones de diseño en el sistema y la arquitectura finalmente alcanzada.**

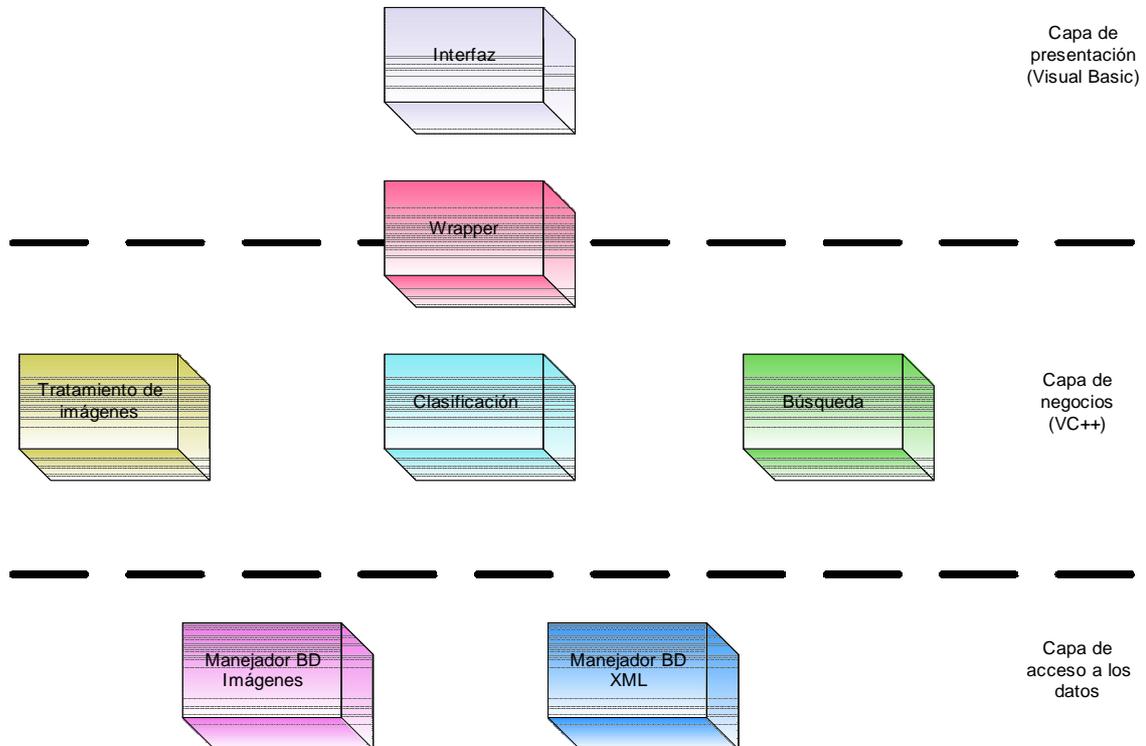
El diseño del sistema implicó tomar en cuenta una gran cantidad de factores, como por ejemplo:

- Permitir la extensibilidad del sistema en general, y en particular de los generadores de descriptores
- Independizar el motor de la interfaz y de la forma en que se manejen las bases de datos
- Cuidar la performance
- Ser compatible con la librería de tratamiento de imágenes ImageMagick.

Como guía utilizamos los patrones de diseño sugeridos en los libros “Applying UML and Patterns” [28] y “Design Patterns” [29]. Concentramos nuestro mayor esfuerzo en utilizarlos para tomar las decisiones de diseño.

### 8.1 ARQUITECTURA DE 3 CAPAS

Utilizamos la clásica arquitectura de 3 capas, como se muestra en el siguiente esquema:



El sistema fue diseñado pensando en una implementación en distintos lenguajes: Como lo mencionamos anteriormente, el motor se encuentra desarrollado en Visual C++, dentro de un objeto COM, y la interfaz en Visual Basic. De todos modos, es flexible en cuanto a que permite fácilmente cambiar ésta última, por ejemplo, a una interfaz web. Como se puede apreciar, entre la capa de presentación y la capa de negocios incluimos un wrapper, que permite la comunicación entre ambas, estando estas desarrolladas en distintos lenguajes. Como método de pasaje de imágenes, dada la complejidad de los datos que tendrían que subir de la capa de negocios hacia la interfaz y las facilidades que brinda Visual Basic para trabajar con archivos de imágenes, resolvimos utilizar directamente archivos BMP. Cabe resaltar que el manejo de los mismos es una tarea exclusiva del Wrapper.

La capa de negocios se divide en 3 grandes módulos:

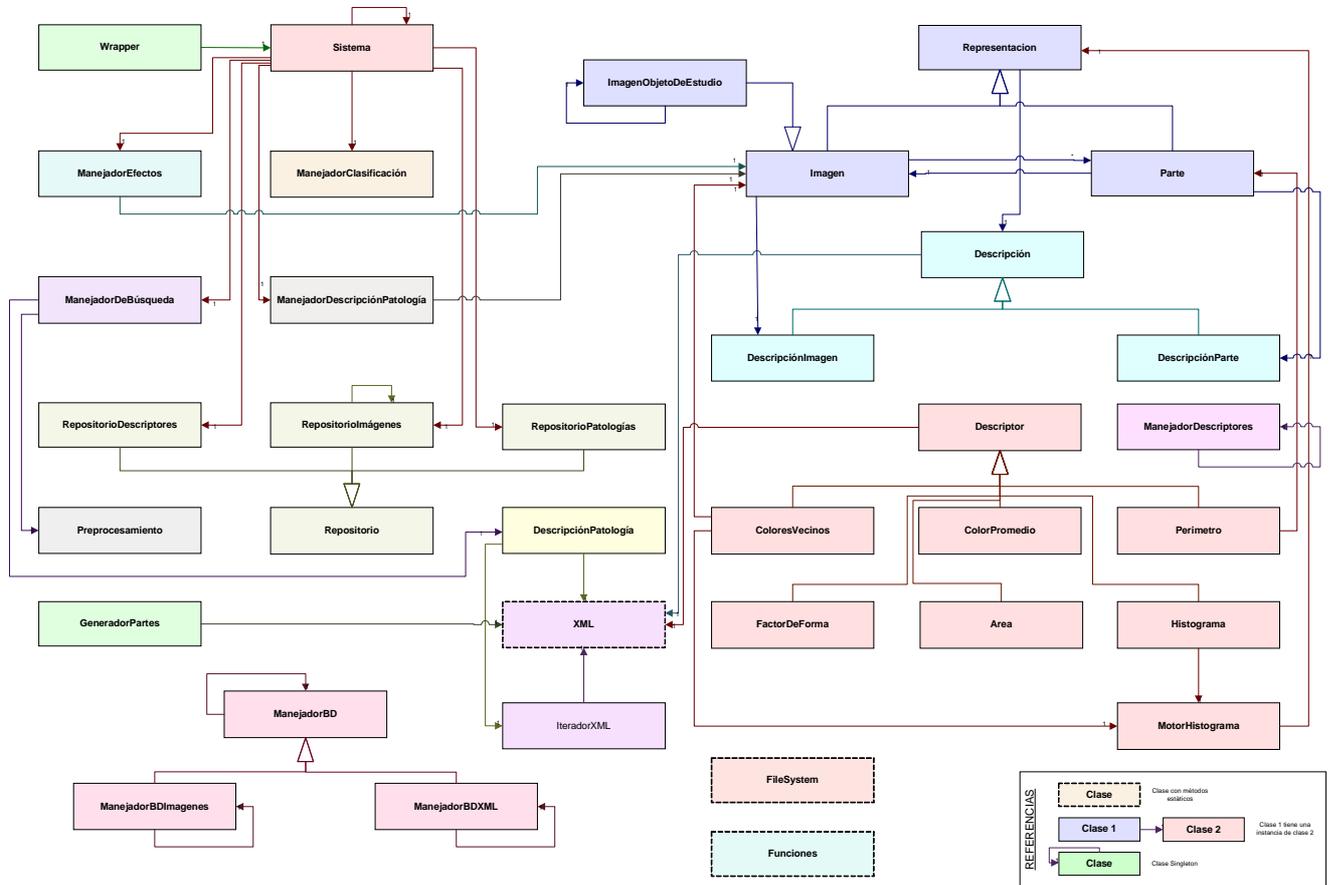
- **Tratamiento de imágenes.** Básicamente, son las funciones que brinda la librería ImageMagick, con la que se realizan efectos visuales sobre las imágenes. Es posible agregar mpas algoritmos con solo agregar los métodos correspondientes a las clases Imagen y Wrapper (ver Anexo G: Documentación técnica).
- **Clasificación.** Contiene todas las funcionalidades que se encargan del “aprendizaje” del sistema: La lógica de generación de descriptores y descripciones de patologías.
- **Búsqueda.** En este módulo se incluyen los algoritmos que realizan el matching de imágenes y descriptores, en busca de ayudar a los médicos a detectar patologías.

En cuanto al manejo de bases de datos, como ya se explicó, no fue nuestra intención concentrar el esfuerzo de desarrollo en este punto: decidimos trabajar directamente con el File System, guardando los XML y las imágenes en directorios separados. Sin embargo, esto puede ser cambiado en forma sencilla gracias a la aislación de la capa de acceso a los datos, que puede ser modificada para utilizar cualquier DBMS, incluso pudiendo utilizar distintos manejadores para las imágenes y los descriptores (XML).

## 8.2 ARQUITECTURA GENERAL

Antes de ir a la explicación de la estructura de clases en sí, queremos hacer una aclaración acerca de los diagramas que mostraremos. Durante la primera etapa de diseño del sistema, se utilizó notación UML para todos diagramas que realizamos, siguiendo el modelo de desarrollo propuesto por Larman [28]. En este informe final no incluimos los métodos ni los atributos en el diagrama de clases por la simple razón de que es demasiado grande para ser impreso en hojas comunes y que se pueda apreciar su contenido. En cuanto a los otros diagramas utilizados, incluiremos en las próximas páginas algunos diagramas generados por nosotros en Visio, y otros generados automáticamente a partir de los códigos fuentes por el programa Doxygen [38] (ver más información en el Anexo F: Análisis de las tecnologías utilizadas).

Veamos ahora cómo se refleja esta arquitectura en la estructura de clases del sistema.



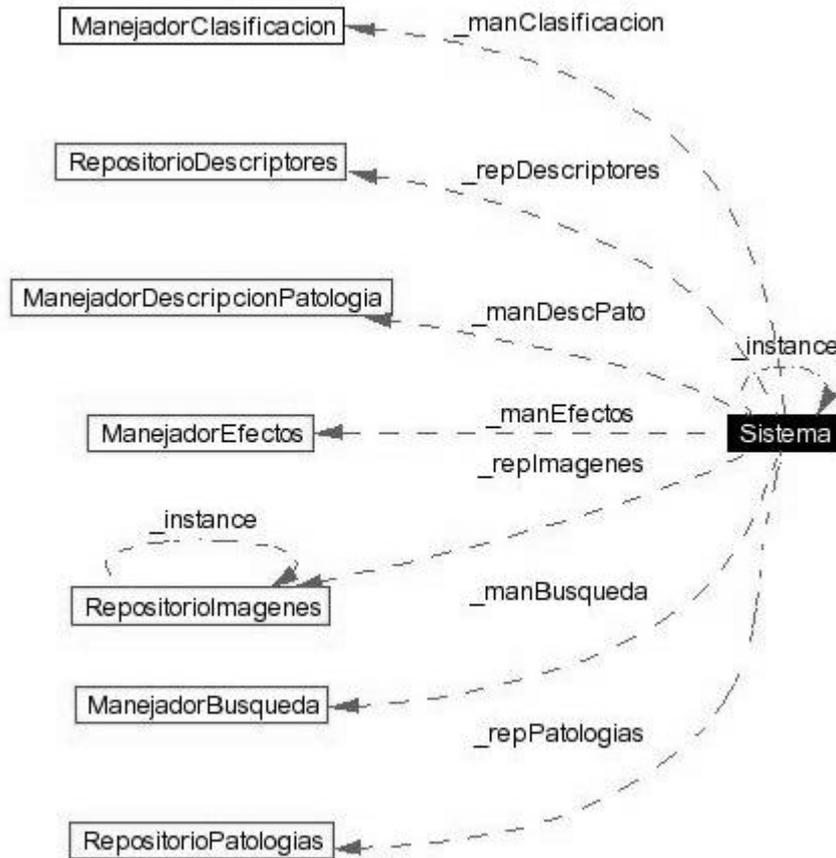
(En el Anexo C: Diseño se encuentra una versión ampliada de este diagrama).

Esta estructura fue deducida a partir de la realización y el estudio de los diagramas de colaboración para cada uno de los casos de uso (ver Anexo C: Diseño), como veremos más adelante. Comenzaremos por ampliar las clases relacionadas a la clase Sistema.

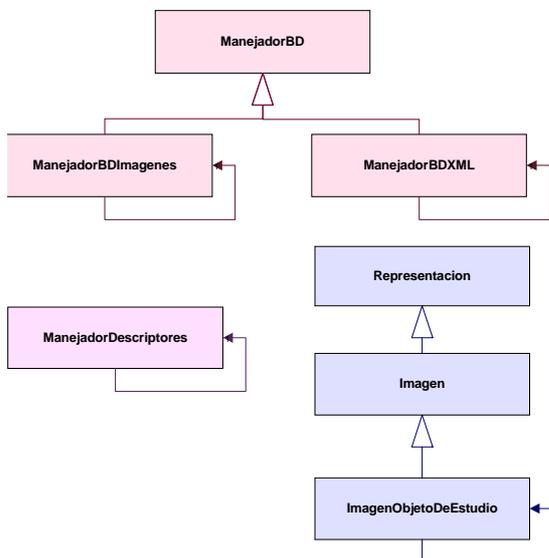


(Las líneas punteadas significan los atributos de las clases)

Hay una única clase que oficia de wrapper con la interfaz. Ésta traduce y traspasa las peticiones exclusivamente a la clase **Sistema** (actúa según el patrón de diseño *Facade*, como punto único de entrada al sistema [28]).



Las peticiones llegadas desde **Wrapper** a **Sistema** son delegadas a manejadores (según patrón de diseño *Pure Fabrication* [28]) que se encargan de cada uno de los casos de uso. Estos manejadores guardan el estado interno de cada caso de uso, y siempre que necesiten acceder a los datos, lo harán a través de peticiones al sistema, para que éste las delegue a los objetos que se encargan de manejar cada uno de los repositorios.

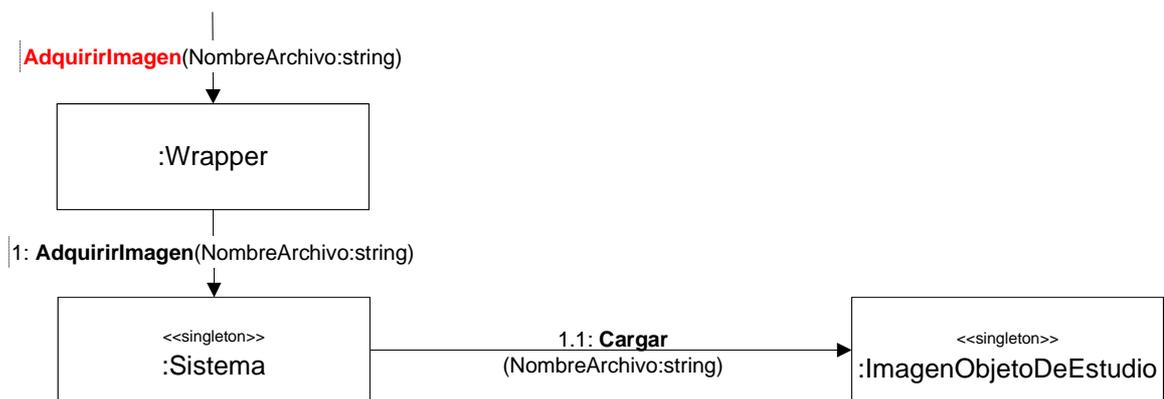


El sistema utiliza algunos objetos *singleton* (o sea, que mantiene una única instancia de la clase, accesible desde todo punto [28]). La clase **Sistema** propiamente dicha cumple esta característica, pero hay otras:

- **ManejadorBDImágenes** y **ManejadorBDXML**. Estos son accedidos desde los repositorios, para acceder propiamente a los datos. Resolvimos utilizar manejadores de BD genéricos de modo de independizarlos al máximo del resto del sistema, en particular de las clases **Repositorio**, que tienen como responsabilidad interpretar los objetos a los cuales les brindan persistencia generando las primitivas adecuadas para las clases **ManejadorBD**.

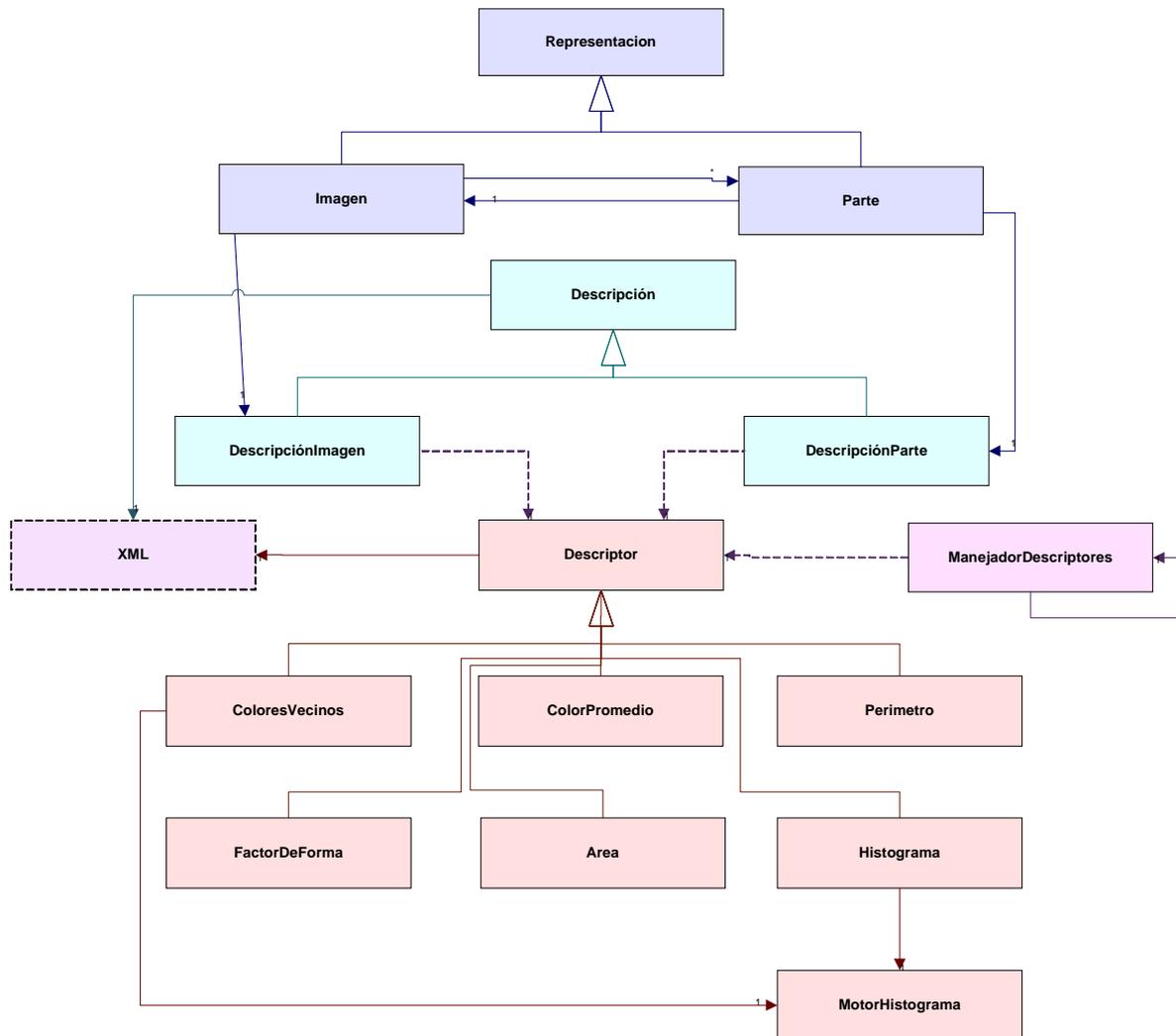
- **ManejadorDescriptores.** Como mencionamos antes, el sistema genera descriptores, los cuales contienen rasgos de una imagen, o de una parte de ella. Para una misma imagen se pueden obtener varios descriptores, y esta clase se encarga precisamente de conocer y generar todos los descriptores que soporta el sistema, a partir de objetos **Repositorio: Imagen** o **Parte** (cumple con el patrón de diseño *Factory* [28]). El funcionamiento de los descriptores se describe más adelante y en el Anexo C: Diseño.
- **ImagenObjetoDeEstudio.** El sistema trabaja con una clase **Imagen**, pero en todo momento se está trabajando con una imagen en particular, que es la que verá el usuario en la pantalla, y que debe poder ser accedida desde cualquier otro objeto.

El siguiente diagrama de colaboración muestra cómo se carga el único objeto **ImagenObjetoDeEstudio** al adquirir una imagen:



### 8.3 CLASIFICACIÓN DE IMÁGENES

Para comprender mejor el mecanismo de funcionamiento de los descriptores, veamos el siguiente extracto del diagrama de clases:



En primer lugar, se da el proceso de Segmentación, el cual crea los objetos **Parte** a partir de la **IOE**. Luego, se da el proceso de generación de descriptores: **ManejadorDescriptores** crea objetos **Descriptor**. A partir de varios de estos objetos se genera una **Descripción**. Esta tiene características distintas dependiendo de si se trata de la imagen completa o de una parte de ella. A nivel interno, tanto **Descriptor** como **Descripción** se guardan en forma de XML.

Hasta aquí se expuso una reseña a alto nivel de la arquitectura del sistema. La explicación de las decisiones de diseño más importantes tomadas para cada uno de los casos de uso diseñados, se encuentra en el Anexo C: Diseño.

# Capítulo 9

## 9 IMPLEMENTACIÓN

---

**En este capítulo se detallarán aspectos que consideramos relevantes respecto a cómo se implementó el sistema y la algoritmia que éste utilizó.**

### 9.1 LENGUAJES UTILIZADOS

Como se mencionó anteriormente, la implementación se realizó en dos lenguajes: la interfaz grafica en Visual Basic y el motor del sistema en Visual C++, encapsulado en un objeto COM. Tal como se describe en el Anexo F: Análisis de las tecnologías usadas, esta decisión trajo grandes ventajas a la hora de programar una interfaz de usuario atractiva.

### 9.2 MANEJO DE STRINGS

Dados los problemas para asignación de memoria para cadenas que tiene C++, se decidió usar el tipo String bajo el nombre Cadena (typedef) para evitarnos el manejo directo de memoria. Por otra parte, el encapsulado dentro del objeto COM exige el uso del tipo BSTR para los String, en la interfaz externa del objeto; para ello, desarrollamos funciones y macros especiales que transforman entre los distintos tipos de cadenas de caracteres (char\*, BSTR y Cadena) de forma transparente y sin preocuparse por la asignación de memoria y el acceso a un carácter determinado.

### 9.3 CLASES AUXILIARES

Se crearon dos clases con métodos estáticos accedidos desde todas las otras clases, con el objetivo de agrupar funciones:

- **Clase Funciones:** contiene un pequeño conjunto de funciones miselánicas del sistema.
- **Clase FileSystem:** contiene funciones para manejo de archivos y directorios.

### 9.4 WRAPPER

La clase Wrapper actúa de traductor entre los tipos de datos manejados por Visual Basic y los de Visual C++. Hasta ella llegan solicitudes desde la interfaz visual, las cuales son traducidas y pasadas a la clase Sistema. Desde esta última se realizan las llamadas a los métodos necesarios, y se devuelven al Wrapper objetos de distintas clases. Por ejemplo, en caso que Sistema devuelva un objeto Imagen, Wrapper se encarga de invocar el método correspondiente para grabar la Imagen en un archivo temporal (BMP), el cual es levantado por Visual Basic para ser mostrado en pantalla.

En todos los casos, los parámetros pasados entre los dos lenguajes fueron de tipos enteros o string, evitando usar estructuras complejas de VC++, de modo de ganar independencia del lenguaje que trabaja la interfaz. Por ello, en caso de querer cambiar la interfaz, por ejemplo por una interfaz web, alcanza con utilizar esta clase en caso de poder utilizar COM, o adaptar solo esta clase para que “traduzca” los parámetros y resultados hacia la interfaz, permaneciendo el resto del sistema (es decir, todo el sistema propiamente dicho) inalterado.

### 9.5 INTERFAZ DE LÍNEA DE COMANDOS

Un aprovechamiento del wrapper antes descripto permitió agregarle al sistema la funcionalidad de ser accedido a través de una línea de comandos fácilmente, de modo de poder correr procesos batch. Esta interfaz, básicamente permite:

- Clasificar imágenes
- Definir automáticamente patologías
- Detectar patologías en imágenes
- Buscar imágenes con determinada patología

La interfaz programada en Visual Basic determina cómo debe levantar el sistema, dependiendo de si se recibieron parámetros. En caso negativo, levanta la interfaz gráfica interactiva, mientras que en caso contrario (ejecutado con parámetros desde una línea de comandos DOS) ejecuta lo solicitado según los parámetros, mostrando en pantalla si los mismos fueron correctos, y guardando en un archivo de log los detalles del proceso ejecutado.

## 9.6 CONSTANTES Y ARCHIVO DE PARÁMETROS

Todas las constantes del sistema son cargadas al inicializarse, leyendo de un archivo XML que contiene los parámetros del sistema.

La primera vez que se ejecuta la aplicación, el sistema requiere ciertos parámetros que puede definir el usuario, o aceptar sus valores por defecto.

## 9.7 TRATAMIENTO DE IMÁGENES

La clase Imagen mantiene un atributo `_image`, que contiene el objeto que maneja imágenes de ImageMagick. Los algoritmos de preprocesamiento de la imagen, en su mayoría, son invocaciones a este objeto. Otros algoritmos debieron ser implementados, como ser el convertir una imagen de color a blanco y negro, dado que el que traía ImageMagick no funcionaba correctamente.

La segmentación de la imagen se realiza en dos etapas: La umbralización y el etiquetado de las regiones.

### **Umbralización**

A su vez, la umbralización requiere dos pasos. En primer lugar se corre sobre la imagen la función `Segment` de ImageMagick, la cual utiliza Fuzzy c-Means: analiza el histograma de color, identificando unidades homogéneas (descrito en la documentación de ImageMagick [36], archivos `api/segment.html` y `magick/segment.c`). Para ello recibe dos parámetros, los cuales no se encuentran muy bien explicados en la documentación de ImageMagick:

- El “nivel de similitud” indica el porcentaje mínimo de pixels que puede contener una región para ser considerada válida.
- La “suavidad” indica el nivel de suavidad esperado de la segunda derivada del histograma de color.

Luego, con el fin de que las regiones no contengan “huecos”, o sea, que no falten conjuntos pequeños de puntos en el interior de una parte, se corre el siguiente algoritmo, que une las regiones pequeñas a las grandes (entendiendo por “pequeñas” a aquellas en que la cantidad de pixels que la componen es menor que el parámetro `PixelsAEliminar` pasado por el usuario):

1. Se realiza un primer etiquetado, con el algoritmo descrito más adelante (siguiente ítem: “**Etiquetado**”). Este proceso devuelve dos resultados:
  - 1.1. Un vector que para cada región etiquetada indica:
    - 1.1.1. el conjunto de pixels que la conforman
    - 1.1.2. su ubicación respecto a la imagen original
    - 1.1.3. si se trata de una región “pequeña”
  - 1.2. Un mapa que indica las correspondencias entre cada pixel y la región que lo contiene.
2. Mientras queden regiones pequeñas, para cada una de ellas

- 2.1. Reviso los pixels de alrededor de su perímetro con el fin de obtener cuál de ellos contiene el color más cercano (esta revisión se realiza sobre la imagen ya umbralizada por ImageMagick)
- 2.2. Reviso en el mapa de pixels a qué región corresponde dicho pixel.
- 2.3. Uno las dos regiones

El resultado de este proceso es una nueva Imagen, que es guardada como atributo de la Imagen original (atributo \_imagenUmbralizada).

### **Etiquetado**

Con el fin de hallar componentes conexas, se implementó el siguiente algoritmo:

1. Para cada pixel no enmascarado
  - 1.1. Creo una nueva Parte
  - 1.2. Agrego el pixel a la Parte y lo enmascaro
  - 1.3. Para cada uno de los 8 pixels vecinos no enmascarados que tengan el mismo color, repito el procedimiento desde el punto 1.2

Con este algoritmo, se van detectando los pixels que tienen igual nivel de gris alrededor de cada pixel que comienza una región.

Otros algoritmos aplicados sobre las imagenes se describen a continuación, en la explicación de los descriptores implementados.

### 9.8 CLASE DESCRIPTOR

Esta clase fue creada para que hereden de ella todos los descriptores que se puedan crear, tanto para Imagenes como para Partes. Además de definir su interfaz, brinda a los descriptores un conjunto de métodos que ayudan a su implementación, como ser:

- la recorrida lineal (pixel a pixel, de izquierda a derecha, de arriba abajo, saltando los pixels enmascarados en caso de tratarse de una Parte)
- la recorrida por el contorno
- la lectura de la estructura principal del XML asociado

Todos estos métodos de asistencia a la programación de descriptores invocan a métodos abstractos que deben ser implementados por cada descriptor en particular.

### 9.9 DESCRIPTORES PROGRAMADOS

El mayor énfasis del proyecto lo ubicamos en la definición de una arquitectura extensible, que permitiera la creación sencilla de descriptores. Es por ese motivo que le implementamos gran funcionalidad a la clase Descriptor. Los descriptores particulares que se programaron fueron realizados

con algoritmos propios, que permitieran probar de manera sencilla el sistema. A continuación describiremos brevemente dichos algoritmos.

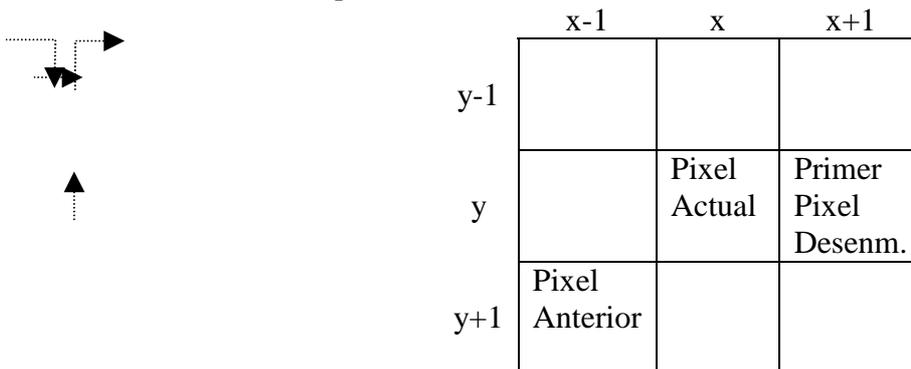
**Área**

Para las Imágenes, simplemente calcula ancho\*alto. Para las Partes, se vale del método RecorridaLineal (de Descriptor) para recorrer los pixels no enmascarados, contando cuántos son estos.

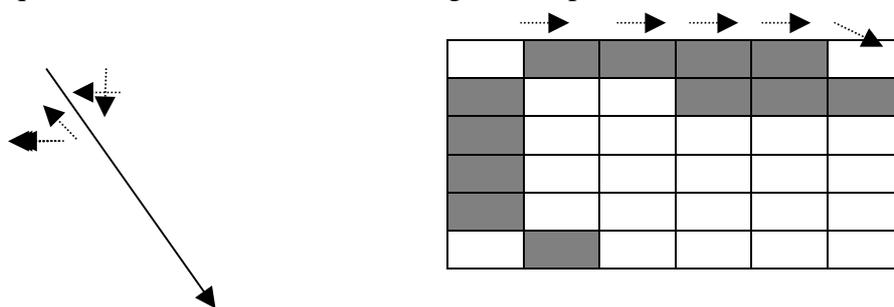
**Perímetro**

Descriptor solo aplicable a Partes. Utiliza la misma lógica de cálculo del descriptor Área, pero llamando en este caso a RecorridaBorde (de Descriptor). Describiremos a continuación el algoritmo que éste método usa:

1. Busca el primer pixel desenmascarado de la región con una recorrida lineal.
2. Mientras no vuelva al primer pixel
  - 2.1. Busco el primer pixel no enmascarado alrededor del pixel actual, comenzando en sentido horario desde el pixel anterior.



Con este sistema aseguramos la recorrida por los pixels del contorno de la región, en todos los casos en que no suceda una situación del siguiente tipo:



Aquí se regresa al pixel inicial

Para estos casos es necesario comenzar a ejecutar nuevamente el algoritmo desde el primer punto de la segunda fila.

## **Histograma**

Tanto para este descriptor, como para ColoresVecinos, se utiliza una clase auxiliar MotorHistograma, la cual se encarga de almacenar la cantidad de pixels que tiene cada nivel de gris, a partir de las llamadas al método AgregarPixel.

El método para obtener el descriptor, consiste simplemente en recorrer la Imagen/Parte, y pasar cada pixel al MotorHistograma, con la salvedad de que se agrupan cada intervalos constantes (para evitar descriptores con 65536 niveles de color distintos). Una vez recorrida toda la imagen, se graba el histograma, indicando en el descriptor el porcentaje de los pixels correspondientes a cada nivel de gris.

## **ColorPromedio**

Dado que el color promedio de una región es el obtenido en el proceso de umbralización, el cálculo de este descriptor consiste sencillamente en obtener el color correspondiente a cualquier pixel de la región en la imagen umbralizada.

## **ColoresVecinos**

ColoresVecinos obtiene el color promedio de las dos regiones vecinas con más pixels “cercaños” a la región actual. El algoritmo consiste en recorrer la región por el borde, y crear un histograma de colores al que se le ingresan los pixels a una distancia menor que cierta constante del pixel actual (de borde), sin incluir los correspondientes a la propia región. Con este procedimiento, se logra un histograma de los pixels “vecinos” a la región. Para lograr que este histograma guarde los colores promedio de las regiones vecinas, la recorrida anteriormente descrita se realiza sobre la imagen umbralizada. De este modo, el descriptor se calcula obteniendo los colores más comunes en el histograma.

## **FactorDeForma**

Este descriptor calcula  $\text{perímetro}^2 / \text{área}$ . Sin embargo, en este caso el cálculo de área no puede ser el mismo utilizado para obtener el descriptor área, dado que se necesita incluir los huecos que la región pueda tener. Por ello, el cálculo del área se obtiene sumando las distancias entre el primer y el último pixel de cada línea de la región, a diferencia del cálculo de área antes mencionado que contabilizaba estrictamente la cantidad de pixels que componen la región.

## 9.10 MANEJO DE XML

El manejo de documentos XML se realizó utilizando tecnología DOM y la librería Xerces, tal como se encuentra explicado en el Anexo F: Análisis de las tecnologías usadas. Sin embargo, esta librería no alcanzó para el manejo que se necesitaba de XML, y debió ser extendida por una clase XML propia para añadir las siguientes funcionalidades, que no traía en forma nativa:

- Grabar documentos XML (esta funcionalidad no se encuentra implementada en la versión para C++ de la librería)
- Extraer de documentos XML sub-documentos con el árbol debajo de cierto elemento

Por otra parte, la utilización de esta clase permite independizar totalmente el resto del sistema de la tecnología y la clase que se usen para manejar XML, por lo cual se puede cambiar Xerces por otra librería, e incluso la tecnología de acceso a los datos podría ser SAX, con solo modificar esta clase.

### 9.11 REUSABILIDAD

Dado que el proyecto buscó construir un motor de búsqueda genérico, la mayor parte del código puede considerarse reusable. Destacamos las siguientes clases:

- **Representación, Imagen, Parte:** Brindan funcionalidades aplicables a todo tipo de proyectos que utilicen imágenes segmentadas.
- **XML:** Aplicable siempre que se necesite independizar una implementación de la tecnología utilizada para representar internamente XML, así como para aprovechar la implementación de los métodos no existentes en la librería Xerces para C++ anteriormente mencionados.
- **Funciones, FileSystem:** Conjuntos de funciones generales aplicables a varios sistemas.
- **ManejadorBDImagenes, ManejadorBDXML:** Aplicables a todo proyecto que necesite almacenar imágenes y XML en forma de archivos.
- **MotorHistograma:** Brinda la posibilidad de manejar histogramas de colores.

### 9.12 DEFINICIÓN DE LOS FORMATOS DE LOS DESCRIPTORES EN XML

Se crearon dos archivos DTD, que definen la estructura de:

- Descripción completa de una imagen (incluyendo los descriptores para cada una de sus regiones)
- Descripción de una patología

Consideramos importante incluir estos DTDs en el informe, dado que explican el formato de los descriptores.

#### **Imagen.DTD**

```
<!ELEMENT Image (BasicDescription, ImageDescription, PartsDescription)>
<!ATTLIST Image id CDATA #REQUIRED>

<!ELEMENT BasicDescription (Height, Width)>
<!ELEMENT Height (#PCDATA)>
<!ELEMENT Width (#PCDATA)>

<!ELEMENT ImageDescription (Descriptor*)>
<!ELEMENT PartsDescription (SegmentationParameter*, Part*)>
<!ELEMENT SegmentationParameter (#PCDATA)>
<!ATTLIST SegmentationParameter name CDATA #REQUIRED>

<!ELEMENT Part (Descriptor*)>
<!ATTLIST Part id CDATA #REQUIRED
             x CDATA #REQUIRED
             y CDATA #REQUIRED
             height CDATA #REQUIRED
```

width CDATA #REQUIRED >

```
<!ELEMENT Descriptor (item*)>
<!ATTLIST Descriptor type CDATA #REQUIRED>
<!ELEMENT Item (#PCDATA)>
<!ATTLIST Item type CDATA #REQUIRED>
```

Como se puede apreciar, la descripción de una imagen se distribuye en tres grandes elementos: BasicDescription, ImageDescription y PartsDescription.

BasicDescription está destinado a guardar la información básica sobre la imagen, información que no es extraída por ningún Descriptor, sino que se trata de características intrínsecas de la imagen. Actualmente consta tan solo de dos subelementos: alto y ancho (height y width). Por ejemplo:

```
- <BasicDescription>
  <Height>228</Height>
  <Width>217</Width>
</BasicDescription>
```

ImageDescription agrupa todos los descriptores obtenidos a partir del análisis de la imagen completa y PartsDescription agrupa los correspondientes a cada una de las Partes resultantes de la segmentación. Dado que las Partes se extraen de la Imagen a partir del proceso de segmentación, PartsDescription incluye elementos SegmentationParameter, los cuales indican los parámetros a través de los cuales se llegó a la partición descripta.

Cada descriptor, ya sea de la Imagen o de sus Partes, tiene un nombre, y un conjunto de Items. Estos últimos se usan para indicar cada uno de los resultados que devuelven los descriptores. Por ejemplo, el

descriptor Histograma genera un Item para cada nivel de gris que haya contabilizado pixels. El valor del elemento es el porcentaje de pixels que se encuentran dentro de este nivel de gris. Para los descriptores que arrojan un único valor, se necesita tan solo un Item.

```
- <ImageDescription>
  - <Descriptor type="Area">
    <item type="cantidadPixels">49476</item>
  </Descriptor>
</ImageDescription>

- <PartsDescription>
  <SegmentationParameter name="nivelSimilitud">
    1,000
  </SegmentationParameter>
  <SegmentationParameter name="suavidad">
    0,000
  </SegmentationParameter>
  ...
  ...
  - <Part X="15" Y="31" Id="001" Width="174"
    Height="185">
    - <Descriptor type="Area">
      ...
    </Descriptor>
    ...
  </Part>
  ...
</PartsDescription>
```

### **Patologia.DTD**

```
<!ELEMENT Pathology (Source*, SegmentationParameter*, Descriptor*)>
<!ATTLIST Pathology id CDATA #REQUIRED>

<!ELEMENT Source (#PCDATA)>
<!ATTLIST Source partId CDATA #REQUIRED>

<!ELEMENT SegmentationParameter (#PCDATA)>
<!ATTLIST SegmentationParameter name CDATA #REQUIRED>

<!ELEMENT Descriptor (Item*)>
<!ATTLIST Descriptor type CDATA #REQUIRED
                relevance CDATA #REQUIRED >
<!ELEMENT Item (#PCDATA)>
<!ATTLIST Item type CDATA #REQUIRED>
```

La descripción de Patologías sigue la misma sintaxis que la de las Imágenes, solo que aquí también se incluye el Source, que indica la imagen y el número de parte a partir de los cuales se definió la Patología.

# Capítulo 10

## 10 AIM Y EL ANÁLISIS DE IMÁGENES MÉDICAS

**En este capítulo se describirán algunas utilidades del sistema en cuanto al análisis de imágenes médicas, fruto de la investigación realizada en el área.**

Parte del trabajo de investigación consistió en aprender acerca de las imágenes con las que tratamos (tomografías computadas) y definir un conjunto de patologías detectables a partir de un conjunto sencillo de descriptores. Para ello contamos con la ayuda de dos doctores del Hospital de Clínicas, y sobre todo de amigos estudiantes de medicina.

A continuación, a modo de introducción al tema, y también con el objetivo de brindar una base para quien quiera profundizar en la mejora del sistema para el reconocimiento de patologías, describiremos algunos datos obtenidos de nuestra incursión en temas médicos.

### 10.1 TOMOGRAFÍA COMPUTARIZADA

El primer elemento con que cuenta un médico para realizar un diagnóstico, además de su contacto directo con el paciente, es la historia clínica, que es el registro cronológico de datos médicos. La tomografía computarizada (TC o TAC, por “Tomografía Axial Computarizada”, o CT por su sigla en inglés) es el método neurorradiológico complementario de mayor utilización y ha llegado a ser el primer examen diagnóstico que se realiza después de la historia clínica y la exploración física en la mayoría de los pacientes con síntomas o signos neurológicos.

La TC se lleva a cabo generalmente en dos modalidades:

- Con contraste intravenoso
- Sin contraste intravenoso

La sustancia de contraste aumenta los coeficientes de atenuación de las zonas patológicas y permite su identificación con mayor precisión.

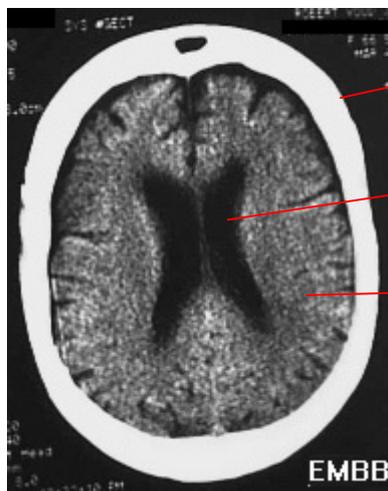
En las imágenes obtenidas por el tomógrafo se distinguen distintos niveles de densidad, los cuales se traducen en distintos niveles de gris en la imagen. Las zonas más hipodensas corresponden a líquidos o aire, y se ven como regiones oscuras, mientras que las hiperdensas corresponden a partes más “macizas” del cuerpo, como por ejemplo los huesos, y se muestran en tonalidades claras.



Las imágenes generadas por el tomógrafo se almacenan en un formato especial, llamado DICOM [7], el cual no solo almacena los distintos cortes axiales, sino que también guarda otras informaciones médicas, como ser la altura a la que se realizó cada uno de ellos. En el tomógrafo del Hospital de Clínicas, las TC en este formato pueden ser transferidas a un PC con un sencillo mecanismo de grabado, desde la consola del tomógrafo.

## 10.2 PATOLOGÍAS IDENTIFICADAS EN LA TC CRANEAL

En primer lugar, vamos a definir muy básicamente los términos usados para referirse a las principales áreas de la cabeza:



Cráneo

Ventriculos

Encéfalo

La TC craneal está indicada en el estudio de la mayoría de las entidades neurológicas. A continuación describiremos breve e informalmente los casos más comunes de uso de la TC craneal:

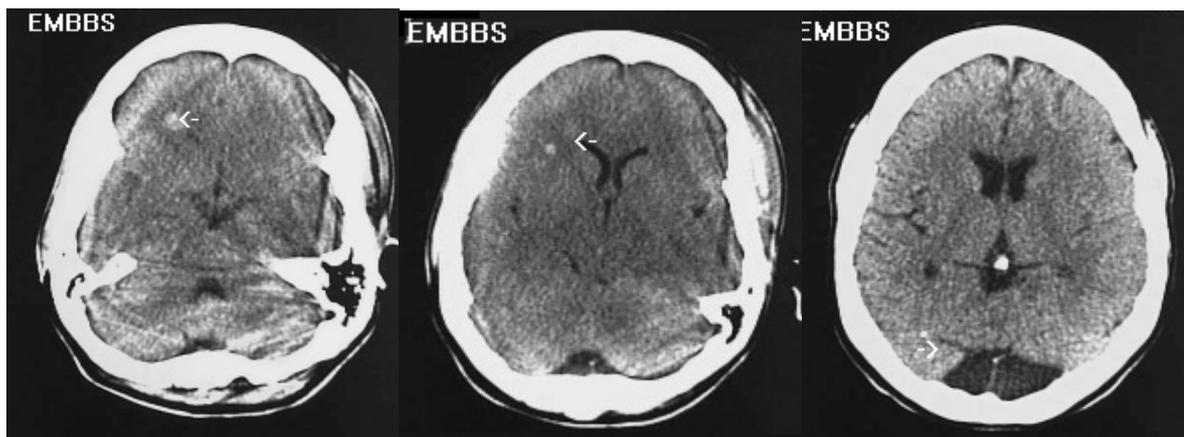
- Traumatismos craneales. Golpes en la cabeza, de distinta índole. Se describirá con detalle más adelante los hematomas, que serán el principal caso de estudio que describiremos.
- Accidente cerebro vascular establecido. Disfunciones establecidas en el interior del cerebro por problemas vasculares, o sea de irrigación de sangre al cerebro. En este grupo, principalmente se encuentran los infartos (producidos por falta de irrigación de sangre a zonas del cerebro) y las hemorragias cerebrales y subaracnoideas (las hemorragias se producen por accidentes o “estallidos” vasculares, que producen sangrados en el interior del encéfalo).

- Tumores cerebrales. Se trata de distintas formas de cancer, ya sea primario o metastático, o sea, surgido como efecto secundario de un tumor primario surgido en otra parte del cuerpo. La presencia de imágenes únicas sugiere tumores primarios, y las imágenes múltiples suelen corresponder a metástasis.
- Hidrocefalia. Patología debida a malformaciones o tumores, en la que el líquido que normalmente fluye por la subaracnoide o los ventrículos desborda en el cerebro.
- Enfermedades demenciales. Por ejemplo, la atrofia cerebral, en la que el encéfalo se ve reducido en su tamaño.
- Enfermedades inflamatorias intracraneales.

### 10.3 IMÁGENES UTILIZADAS

No resultó sencillo aplicar en forma exhaustiva el sistema a imágenes médicas, debido a la dificultad existente en obtener conjuntos grandes de estas imágenes digitalizadas. Por otra parte, no nos fue posible conseguir que los médicos del Hospital de Clínicas archivaran en formato digital las tomografías sacadas a los pacientes. De todos modos se obtuvo, de varios sitios en Internet, un set representativo de imágenes radiológicas y tomográficas. Las principales dificultades que brindan las mismas son:

- La mayoría de las imágenes presentan patologías poco comunes
- Muy pocas imágenes muestran órganos “sanos”
- No se cuenta con información importante como ser la altura del cuerpo a la que pertenecen los cortes tomográficos. Esto, por ejemplo en las tomografías de cerebro, resulta muy importante a la hora de asociar descriptores a patologías, como se puede apreciar en las siguientes imágenes.

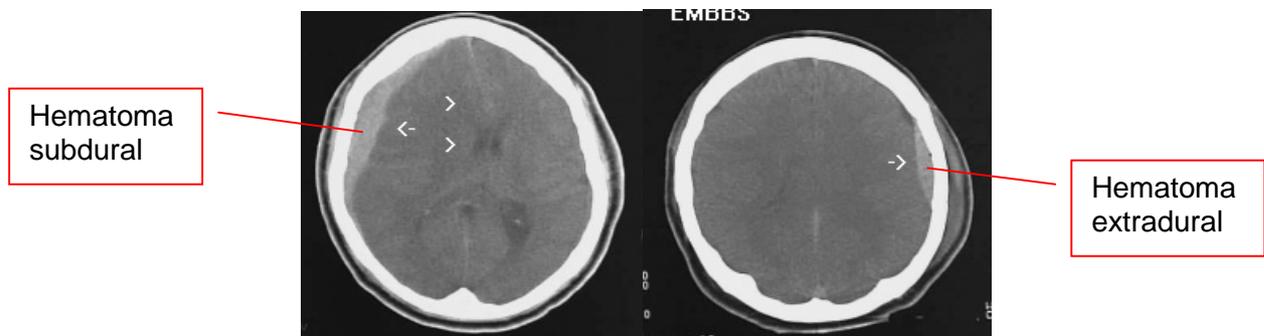


Estos son tres cortes del mismo cerebro, con pocos centímetros de distancia entre ellos. Se nota claramente como las texturas, las tonalidades y las formas cambian notablemente. Por otra parte, algunos rasgos que en ciertos cortes indican presencia de patologías, en otros cortes su presencia es normal.

De todos modos, con las imágenes recabadas (se encuentran en el CD entregado), se pudieron identificar algunas patologías con las cuales probar el sistema. Veremos a continuación, a modo de ejemplo el método de definición de una patología en particular.

#### 10.4 IDENTIFICACIÓN DE HEMATOMAS

Hay dos tipos principales de hematomas cerebrales: el hematoma extradural (o epidural) agudo y el hematoma subdural.



A nivel médico, el hematoma extradural significa un sangrado entre el hueso y la duramadre, mientras que el subdural representa sangrado a un nivel más interior, entre la duramadre y la aracnoides.

Cualquiera de los dos casos se distinguen en las TC por presentar una región hipodensa entre el encéfalo y el cráneo. El sistema AIM utiliza precisamente esta información para identificarlos. A través de los descriptores ColorPromedio, y ColoresVecinos puede detectar en nivel de densidad correspondientes al hematoma, el cráneo y el encéfalo.

Por otra parte, se puede distinguir el hematoma extradural del subdural debido a la forma “redondeada” o de luna que el primero presenta. El sistema puede reconocer esto a través del descriptor FactorDeForma, que es menor cuanto más redonda sea una región.

# Capítulo 11

## 11 EJEMPLO DE USO DEL SISTEMA

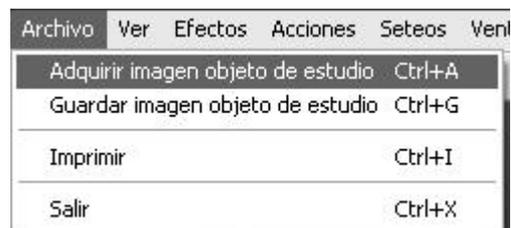
**En este capítulo se mostrará el producto obtenido a través de un ejemplo práctico de uso del mismo.**

El producto final brinda un amplio espectro de posibilidades de utilización. A continuación, con el objetivo de mostrar el resultado de nuestro trabajo, mostraremos un sencillo ejemplo de algunas de las funcionalidades que brinda el sistema.

El ejemplo a desarrollar consiste en la definición de una patología y su posterior detección en una imagen. Por su sencillez de definición y por las imágenes con que contamos, consideramos que es un buen ejemplo la detección de hematomas.

Para ello, en primer lugar abriremos una imagen que servirá como ejemplo, dado que sabemos que tiene la patología a definir. Esto lo podemos hacer a través de:

- la opción correspondiente en el menú:



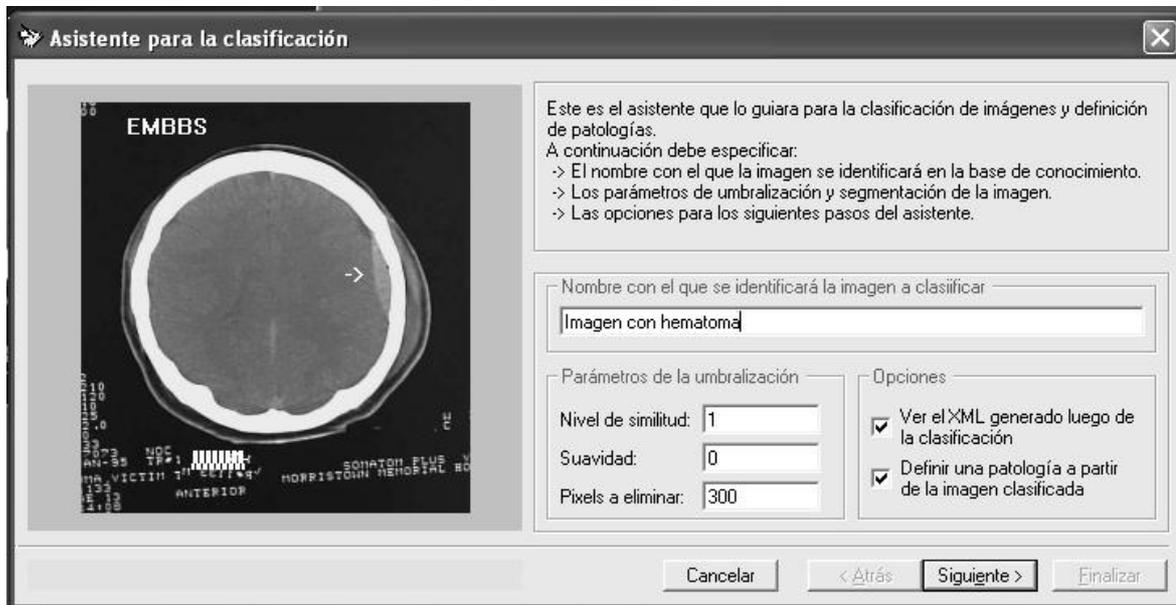
- utilizando el botón correspondiente:



Con esto, podremos seleccionar la imagen desde el archivo correspondiente en disco:

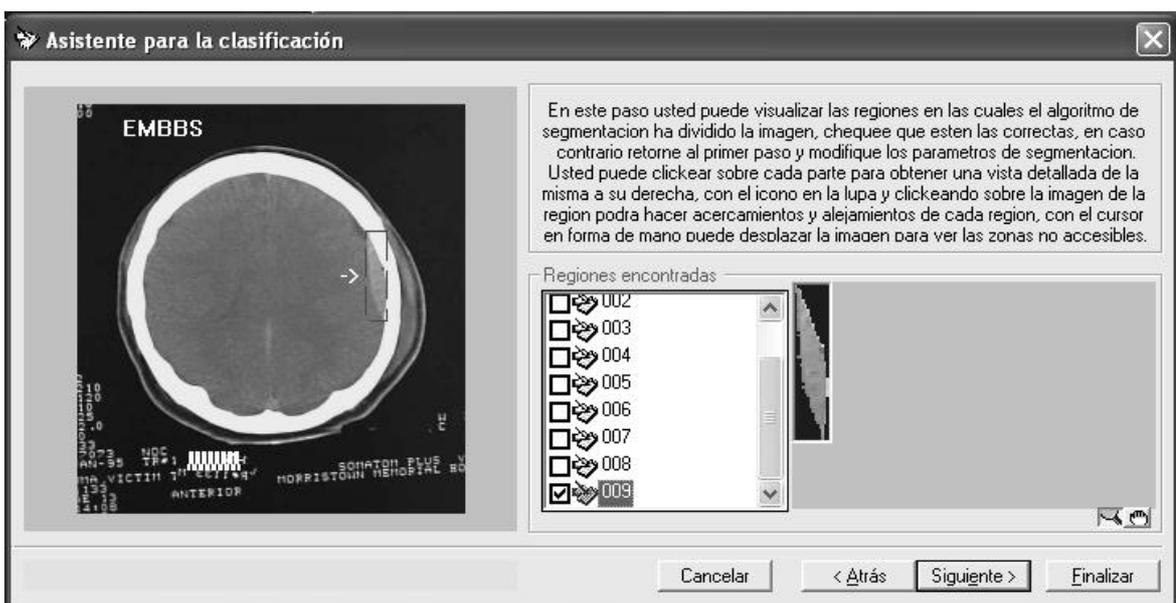


Una vez seleccionada la imagen, ésta nos aparecerá en el escritorio de trabajo, como Imagen Objeto de Estudio (IOE). A continuación, el próximo paso será clasificarla mediante el Asistente para clasificación asistida, presionando el botón .



Aquí asignamos como nombre de la imagen en la base de datos “Imagen con hematoma”, dejamos los parámetros por defecto para la segmentación e indicamos al sistema que deseamos definir una patología a partir de la imagen una vez que esta se encuentre clasificada. Para continuar, presionaremos el botón de “Siguiete”.

El segundo paso de la clasificación asistida consiste en observar la imagen umbralizada para así comprobar que la zona que contiene al hematoma se encuentra correctamente diferenciada de las demás regiones. Luego, presionaremos nuevamente el botón de “Siguiete”, con lo cual accederemos a la siguiente ventana:



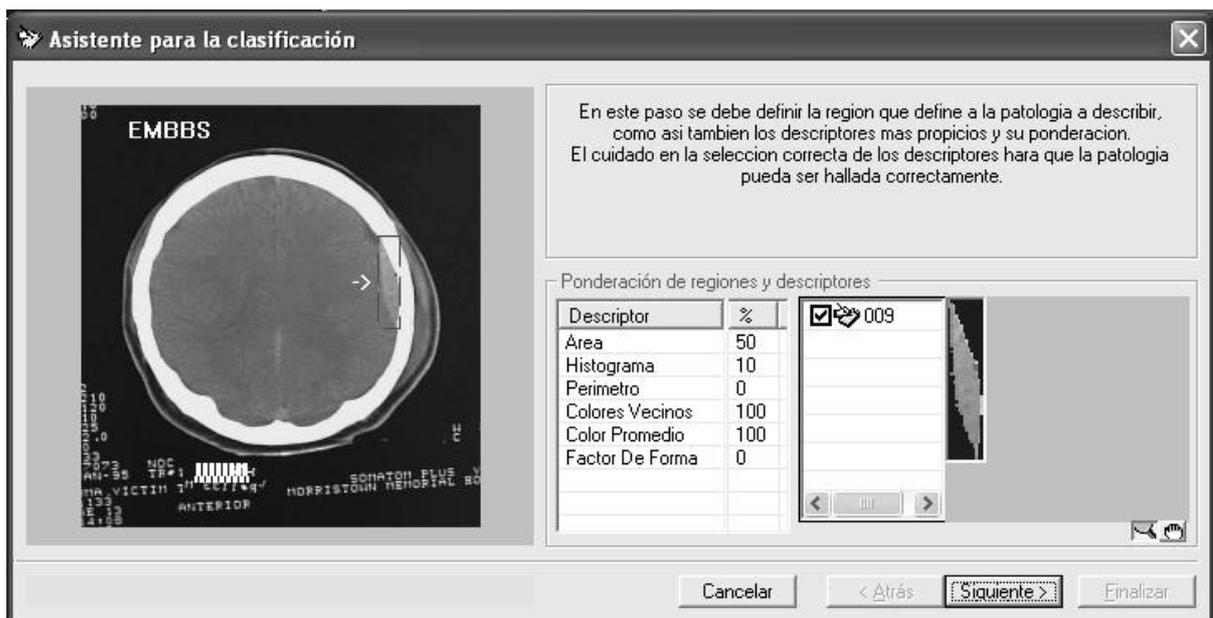
Aquí seleccionamos, de entre todas las regiones reconocidas por el sistema, las que deseamos que se mantengan en la base de datos de regiones clasificadas. En este ejemplo, seleccionamos únicamente a la correpondiente a la patología. Luego, hacemos click en “Siguiente”. Con esto se termina el proceso de clasificación, y el sistema da la posibilidad de ver el XML generado, el cual tiene el siguiente aspecto:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Image Id="Imagen con hematoma">
<BasicDescription>
<Height>256</Height>
<Width>240</Width>
</BasicDescription>
<ImageDescription>
<Descriptor type="Area">
<item type="cantidadPixels">61440</item>
</Descriptor>
<Descriptor type="Histograma">
<item type="4000">1,769</item>
<item type="5000">12,459</item>
<item type="6000">13,792</item>
<item type="7000">12,446</item>
<item type="8000">9,246</item>
<item type="24000">1,229</item>
<item type="25000">4,019</item>
<item type="26000">6,659</item>
<item type="27000">5,216</item>
<item type="28000">5,465</item>
<item type="29000">3,612</item>
<item type="30000">2,163</item>
<item type="31000">1,304</item>
<item type="60000">6,170</item>
<item type="61000">1,309</item>
```

```
</Descriptor>
</ImageDescription>
<PartsDescription>
<SegmentationParameter
name="nivelSimilitud">1,000</Segmentati
onParameter>
<SegmentationParameter
name="suavidad">0,000</SegmentationPar
ameter>
<SegmentationParameter
name="pixelsAEliminar">300</Segmentati
onParameter>
<Part X="178" Y="79" Id="009" Width="14"
Height="57">
<Descriptor type="Area">
<item type="cantidadPixels">372</item>
</Descriptor>
<Descriptor type="Perimetro">
<item type="borde">113</item>
</Descriptor>
<Descriptor type="Colores Vecinos">
<item type="principal">60652</item>
<item type="secundario">27242</item>
</Descriptor>
<Descriptor type="Color Promedio">
<item type="promedio">44461</item>
```

```
</Descriptor>
<Descriptor type="Factor De Forma">
<item type="factor">34,325</item>
</Descriptor>
<Descriptor type="Histograma">
<item type="33000">3,495</item>
<item type="34000">6,183</item>
<item type="35000">9,409</item>
<item type="36000">7,258</item>
<item type="37000">11,828</item>
<item type="38000">16,667</item>
<item type="39000">13,172</item>
<item type="40000">9,409</item>
<item type="41000">8,333</item>
<item type="42000">3,495</item>
<item type="44000">1,075</item>
<item type="45000">1,344</item>
<item type="46000">1,075</item>
<item type="48000">1,344</item>
<item type="53000">1,344</item>
</Descriptor>
</Part>
</PartsDescription>
</Image>
```

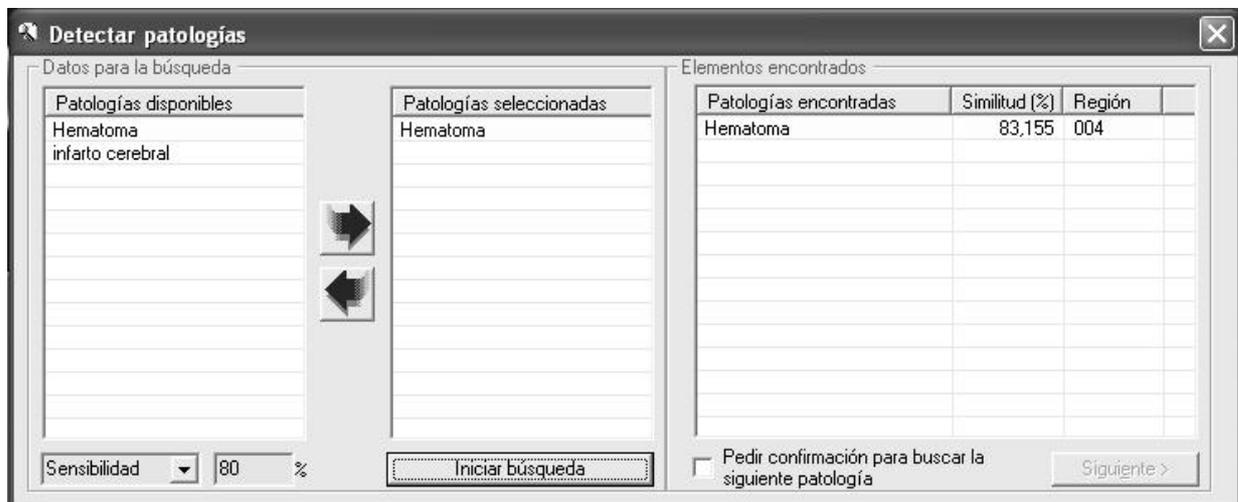
Presionando una vez más el botón de “Siguiente”, podremos comenzar a definir la patología.



En este paso del asistente nos aparecen las regiones guardadas en la base de datos, por lo cual aparece tan solo la que utilizaremos para definir la patología. Sobre ella, definimos los descriptores que son relevantes, ponderándolos. En el ejemplo, se toman como rasgos de mayor relevancia para definir el hematoma a los “colores vecinos” (dado que es importante que los principales colores de alrededor sean los correspondientes al cráneo y al encéfalo) y “color promedio” (dado que todos los hematomas tienen una densidad similar). Consideramos como rasgos menos relevantes el área (50% de importancia) y el histograma de color (10%). Presionando el botón de “Siguiente”, el sistema nos pide un nombre con el cual registrar la patología en la base de datos. La llamaremos “Hematoma”. El sistema nos ofrece también aquí la posibilidad de ver el XML generado, esta vez a partir de la patología definida. Luego, podremos presionar el botón de “Finalizar” para dar por culminados los procesos de clasificación de imagen y definición de patología.

A continuación, abriremos otra imagen con el fin de que el sistema busque en ella la patología recién definida. Esto lo haremos con el mismo procedimiento que para la imagen anterior. Esta vez, seleccionaremos el archivo “ct27m”. Una vez abierto, iremos detectar si existe la patología en la imagen, presionando el ícono correspondiente: 

Luego, seleccionaremos la patología “Hematoma” como única a buscar en la imagen, indicaremos que queremos buscar las regiones que tengan más de 80% de similitud e iniciaremos la búsqueda, para llegar a la siguiente pantalla:



El sistema detectó en una región de la nueva imagen un 83,155% de similitud con la patología antes definida. Si deseamos ver a qué región corresponde, podemos hacer doble-click sobre el elemento encontrado, y seleccionar la opción “Marcar coincidencia en la IOE actual”, pudiendo así ver el recuadro correspondiente:



El producto cuenta con muchas otras funcionalidades, entre las que destacamos el poder aplicar efectos de preprocesamiento a las imágenes con el fin de mejorar las búsquedas, poder realizar búsquedas de patologías en conjuntos de imágenes, o ser utilizado en forma batch desde una interfaz de línea de comandos para, por ejemplo, hacer clasificaciones masivas. Estas funcionalidades se encuentran más extensamente detalladas en el Manual de usuario.

# Capítulo 12

## 12 CONCLUSIONES

---

**En este capítulo se buscará arribar a conclusiones acerca de los distintos aspectos que implicó el proyecto y mirar a futuro respecto a las posibilidades que éste puede ofrecer.**

El sistema, que a primera vista no exhibe la gran complejidad que encierra, trajo varios desafíos. El tema que atañe es muy rico y se presta para una amplia gama de extensiones, lo cual trajo las siguientes consecuencias:

- El proceso de investigación, tomando en cuenta que inicialmente los requerimientos del sistema no estaban bien definidos, llevó un gran esfuerzo, tal como se encuentra en el Anexo A: Administración y proceso de desarrollo.
- El análisis de requerimientos se realizó sobre ideas concretas, pensadas por los integrantes del Taller, acerca de la utilidad que se le pensaba dar al producto. No obstante, durante la definición del modelo conceptual surgió una amplia gama de conceptos relacionados con el sistema, los cuales se encuentran documentados en el Anexo B: Análisis de requerimientos
- Se debió acotar el diseño del sistema a un subconjunto de los conceptos surgidos del modelo conceptual, para llegar a un programa de tamaño razonable para el taller.
- El punto en que resultó necesario focalizar el esfuerzo de diseño y desarrollo fue el lograr un sistema extensible, que permita incluir todas las partes del modelo conceptual, y que fuera posible extenderlo para añadirle descriptores, y otros elementos que hacen a la inteligencia del motor de búsqueda.

En este marco, analizaremos los aspectos positivos y negativos del proyecto.

### 12.1 ASPECTOS POSITIVOS A DESTACAR

A nivel general, se puede mencionar el desafío de enfrentar temas inicialmente desconocidos por los integrantes del grupo. Fue necesario para ello aprender mucho en distintas áreas, a saber:

- Tratamiento digital de imágenes
- Documentación de las distintas fases del proceso de desarrollo, con notación UML
- Lenguajes de programación Visual Basic y Visual C++ (contabamos con poca experiencia C++ bajo entorno Unix)
- Otras tecnologías que trajo la fase de implementación, como objetos COM y XML.

Consideramos como un aspecto muy positivo del proyecto el haber incorporado todos estos conocimientos.

Por otra parte, aprendimos también a utilizar una gran variedad de herramientas actuales, como ser Visual SourceSafe (un aporte fundamental para el trabajo en equipo), Xerces, ImageMagick, doxygen (muy buen sistema de generación de documentación técnica), entre otras.

El proceso de investigación del estado del arte en búsqueda de imágenes fue sumamente enriquecedor. Entendimos que el mejor medio para buscar información al respecto era la Internet, y ahí pasamos muchas horas navegando en busca de algún sistema que tuviera un enfoque similar al que nosotros encaramos. Para nuestra sorpresa, nos encontramos con varios sistemas de búsqueda, pero todos especializados en aspectos particulares (búsquedas basadas en colores [23], motores especializados en manejo de gran cantidad de imágenes, sin gran poder en la investigación del contenido de las mismas [24], sistemas basados en búsquedas especiales [25] o bancos de datos basados en información extraída de las imágenes en base a métodos alternativos [26, 27]), y ninguno que realmente cubriera una gama realmente amplia de tipos de búsqueda. Finalmente, encontramos que el tratamiento digital de imágenes se encuentra en un momento clave, en el cual se encuentran en proceso de definición estándares que seguramente cambiarán la forma de buscar contenido multimedia en la web, como promete el estándar MPEG-7 [18].

El haber llegado a a la realización concreta de un producto de tal envergadura es quizás el logro que más satisfacción produce, ya que se ve plasmado el esfuerzo en algo concreto y tangible. Pero el mayor logro puede haber sido, y el futuro lo dirá, el haber iniciado una nueva línea de investigación en este tema en el INCO, por parte de otros proyectos de grado o de otras materias relacionadas, proporcionando una base sólida, tanto a nivel de conceptos como de diseño, junto a piezas de software funcionando. Creemos que el sistema fue desarrollado en el momento correcto, y tiene un largo camino por recorrer.

### 12.2 ASPECTOS NEGATIVOS A DESTACAR

En primer lugar, vale la pena destacar las dificultades y la gran cantidad de trabajo poco productivo que trajo el no tener desde el comienzo bien definido el sistema a implementar. Otro aspecto negativo que no ayudó a solucionar el punto anteriormente mencionado, fue la falta de

involucramiento por parte de los doctores del Hospital de Clínicas. A modo de ejemplo, cabe destacar que se definió con ellos, en la segunda reunión que se mantuvo, recién el 28 de Agosto, la idea del sistema a desarrollar. En todo el período que transcurrió desde el comienzo del proyecto hasta entonces, realizamos un gran trabajo de investigación y pruebas sin tener una línea clara de estudio, tanto en el área de tratamiento de imágenes, como en la teoría acerca de las imágenes radiológicas.

Respecto a la investigación sobre tratamiento digital de imágenes, esta se fue realizando de forma informal, buscando al mismo tiempo:

- ejemplos de sistemas de búsqueda de imágenes
- algoritmos para preprocesar, segmentar e interpretar imágenes
- técnicas de almacenamiento y búsquedas en bases de datos de imágenes
- librerías en distintos lenguajes para tratar imágenes

Este trabajo se fue realizando durante los primeros meses del proyecto, y a la hora de documentarlo resultó muy difícil hacerlo. De hecho gran parte de la investigación en Internet no quedó documentada.

La falta de involucramiento de parte de los doctores del Hospital de Clínicas trajo, como otra consecuencia negativa, el problema de no contar con muchas imágenes médicas, como fue explicado anteriormente. Este tema llevó a no estar seguros del tipo de imágenes que iba a tratar el sistema y el tipo de patologías que iba a reconocer, hasta luego de empezado el período de implementación, cuando logramos encontrar en Internet un set razonablemente amplio de imágenes médicas y obtuvimos el apoyo de amigos estudiantes de medicina.

Otro aspecto que produjo pérdidas de tiempo fue la falta de definición del lenguaje en que se iba a implementar la aplicación. Las razones que llevaron a buscar alternativas para no implementar el sistema en Visual C++ se encuentran explicadas en el Anexo A: Administración y proceso de desarrollo. Lo que vale la pena destacar, es que esto llevó a la implementación de varios prototipos en Visual Basic, y al análisis de otras posibilidades, como ser Java, lo cual implicó una importante inversión de tiempo poco productivo en cuanto a los objetivos finales del proyecto.

En cuanto al manejo de bases de datos, nos hubiera gustado desarrollar un sistema que corriera con un DBMS "real", y no el file system como lo hicimos. Sin embargo, esto no fue posible debido a que no encontramos ningún DBMS de libre distribución que manejara en forma nativa XML y/o Imágenes. Consideramos que, de haberlo tenido, podríamos haberlo utilizado sin necesidad de añadir trabajo en el proyecto, dado que la confección de la clase FileSystem también trajo algún trabajo adicional.

Finalmente, quisieramos destacar el problema de la enorme carga de trabajo que trajo el proyecto, sobre todo en la segunda mitad, que incluyó varios fines de semana enteros, jornadas extensas de trabajo durante tiempos prolongados y continuos, que resultaron desgastantes.

Todos los aspectos mencionados anteriormente no tienen el ánimo de desmerecer el proyecto. Por el contrario, son parte de lo que significa enfrentarse a un problema complejo y que requiere gran carga de investigación sobre temas actuales, como demostró ser este taller. Todas estas dificultades se convierten en logros a la hora de sentir que culminamos el proyecto con una solución satisfactoria

y con un sistema funcionando que prueba la aplicabilidad del enfoque de búsqueda de imágenes que planteamos desde el inicio del proyecto.

### 12.3 POSIBLES MEJORAS

Intentaremos resumir las ideas de extensiones al sistema más importantes manejadas durante el desarrollo del proyecto, dado que entendemos que esta sección bien podría ser la más larga del informe, debido al amplísimo campo de posibilidades que se abre a partir de la idea de un motor de búsqueda de imágenes.

#### **Preprocesamiento y segmentación de imágenes**

Actualmente, el sistema cuenta con un único algoritmo de segmentación de imágenes, basado en fuzzy c-means. Pese a que la segmentación trae intrínsecamente varias dificultades [27, pág 447], sería muy útil extenderlo, agregándole métodos a la clase Imagen que llamen a distintos algoritmos de segmentación y guardando esta información en la descripción (elemento SegmentationParameter). Luego, se podrían obtener descriptores distintos de acuerdo al algoritmo que haya sido usado, e incluso combinar descriptores obtenidos con distintos algoritmos de segmentación.

En cuanto a los algoritmos de preprocesamiento, está incluido dentro del diseño del sistema la posibilidad agregar nuevos, tal como se mencionó anteriormente. Otra extensión importante se encuentra planteada en el modelo conceptual, y consiste en guardar dentro de la descripción de una patología los algoritmos de preprocesamiento utilizados para llegar a la misma en un script XML.

Finalmente, sería conveniente que, dentro de la descripción de una única patología, se pudieran combinar descriptores obtenidos con distinto trabajo de preprocesamiento y segmentado, incluso permitiendo aplicar efectos sobre regiones luego de segmentadas, para obtener mejor algunos descriptores. Para implementar esto correctamente, se debería tener en cuenta que todos estos algoritmos deberán luego ser corridos sobre cada imagen en que se realice la búsqueda de la patología, por lo cual el motor de búsqueda debería optimizar este proceso detectando caminos de preprocesamiento y segmentación comunes entre los descriptores, y así correrlos la menor cantidad de veces posible.

#### **Tipos de imágenes a tratar**

El sistema implementado, por los algoritmos que utiliza, trabaja exclusivamente con imágenes en blanco y negro. Sin embargo, su concepción es aplicable a todo tipo de imágenes, solo con la adaptación de algunos de sus algoritmos, lo cual enriquecería el rango de posibilidades que brinda el sistema.

Por otra parte, pese a que el sistema levanta correctamente imágenes en formato DICOM, no aprovecha al máximo toda la riqueza que el mismo ofrece, como ser el texto descriptivo acerca de la imagen que incluye. Estos datos podrían generar descriptores especiales en el sistema, con los cuales

realizar comparaciones más precisas entre imágenes, por ejemplo comparando únicamente cortes axiales realizados a la misma altura del cuerpo.

### **Relación entre regiones**

En el modelo conceptual se detectó la necesidad, para obtener un motor de búsqueda de imágenes más poderoso, de trabajar con datos acerca de la relación entre distintas regiones de una imagen, e incluso agrupar conjuntos de imágenes relacionadas entre sí (por ejemplo distintos cortes axiales de una tomografía) trabajar con datos acerca de la relación entre ellas. La descripción acerca de los conceptos relacionados a esta funcionalidad ya fueron analizados en este informe, en el Capítulo Análisis del sistema. Por el simple hecho de limitar el trabajo a realizar al tiempo de desarrollo de un Taller 5, esto fue recortado en la definición de alcance, y no se encuentra dentro de las funcionalidades del sistema final. Entre los aspectos a mejorar del sistema, consideramos que es uno de los más importantes.

### **Unir información médica al sistema**

Es mucha la información médica que se puede unir al sistema.

Un aspecto que surge de los planteos de mejoras anteriores, es completar el manejo de imágenes médicas con otras informaciones acerca del paciente, como ser la historia clínica u otros datos personales. El sistema podría acceder desde los descriptores a información clínica que influya en la definición y comparación de patologías. También sería conveniente que el sistema manejara imágenes de un mismo paciente en distintos momentos a lo largo del tiempo, y detecte patrones de evolución, y patologías relacionadas a estos. Para ello se puede aprovechar el sistema SIREI, desarrollado para el Hospital de Clínicas por un grupo anterior de Taller, el cual administra información sobre pacientes. Sin embargo, la integración de ambos sistemas no resulta trivial, dado que esta información debería pasar a formar parte de los descriptores y se deberían establecer criterios de comparación, entre otras tareas.

En la misma línea de unir información médica al sistema, también vale la pena considerar la posibilidad de que el proceso de segmentación identifique rangos de tonalidades de gris con el tipo de textura que los mismos representan, de modo de mostrar en forma más clara las distintas componentes físicas (tipos de materia) que representan la imagen.

### **Implementación de mayor cantidad de descriptores**

El conjunto de algoritmos de extracción de descriptores actualmente desarrollado solo alcanza para probar el sistema con un número reducido de patologías. Consideramos que con la programación de nuevos descriptores, la incorporación del concepto “descriptores asistidos” planteado en el modelo conceptual, junto con el punto antes propuesto de generar descriptores respecto a la relación entre regiones e imágenes, se podrían lograr búsquedas de gran precisión en imágenes médicas y no médicas.

### **Paralelización del sistema**

Se podría mejorar la performance del sistema llevandolo a una estructura distribuida. Este punto en sí es todo un tema, pero lo vemos posible, y sería muy beneficioso para el sistema, ya que hay algoritmos que son paralelizables, y actualmente consumen una gran cantidad de procesamiento, y por tanto de tiempo.

### **Búsqueda de imágenes en Internet y manejo de grandes bases de datos**

Dado que el motor de búsqueda es aplicable a todo tipo de imágenes, es lógico pensar en utilizarlo como una herramienta para buscar imágenes en grandes bases de datos a través de la web.

Esto requeriría principalmete de:

- Una herramienta de extracción de imágenes de la web
- Un manejador de bases de datos que trate las descripciones XML de las imágenes en forma óptima, dado que sobre ellas se realizarán las búsquedas
- Un front-end web, que permita ingresar una imagen o una descripción (al estilo de la descripción de patologías) de modo amigable, y permita navegar entre los resultados obtenidos.

Consideramos que esta sería una gran utilidad del sistema, aunque entendemos que para ello primero debe madurar en cuanto a los puntos anteriores, la optimización al máximo de los algoritmos, y la cantidad de descriptores que maneje.

### **Mejor estructuración de los datos**

Actualmente, el sistema tiene un único repositorio para las imágenes que clasificó, sobre las cuales se realizan las búsquedas. Ayudaría al uso exhaustivo del sistema el poder contar con repositorios diferentes, en los que se guarden distintos grupos de imágenes clasificadas, así como repositorios diferentes para conjuntos de definiciones de patologías.

### **Algoritmos heurísticos de detección de patrones comunes en patologías**

Del análisis de grandes conjuntos de imágenes relacionadas a una patología, el sistema podría implementar algoritmos de data-mining que detecten elementos comunes entre las imágenes, lo cual permitiría enriquecer la descripción de patologías.

Respecto a este punto, existe un muy interesante campo de investigación, que consideramos en trabajar sobre el comienzo del taller, sobre todo a raíz del curso de tratamiento digital de imágenes con enfoque lógico-combinatorio [32], en el cual se planteó el reconocimiento y uso de “testores”, que son conjuntos de “rasgos” (en este sistema se pueden considerar como rasgos a los descriptores) que definen gurpos de imágenes.

Este tema abre un muy interesante campo de investigación, de aprovechamiento de los descriptores para detectar información no detectable a simple vista en bancos de datos de imágenes.

#### 12.4 CONCLUSIÓN FINAL

El principal objetivo del proyecto era llegar a un producto extensible, que sirva de base para otros proyectos de tratamiento de imágenes que profundicen en los descriptores obtenibles, la facilidad de uso e inteligencia del sistema. Creemos que el mismo está más que cumplido, dado que se encuentra construido y funcionando un núcleo que muestra la viabilidad de proyectos de mayor envergadura en el área, y que es un puntapié inicial en el manejo de descripciones de imágenes alineado en la idea de MPEG-7, un estándar que consideramos será muy utilizado en un futuro cercano.

# Capítulo 13

## 13 LISTA DE ANEXOS

---

**En este capítulo se listarán los anexos que se adjuntan al presente informe.**

Se incluyen los siguientes anexos al presente documento:

- Anexo A: Administración y proceso de desarrollo
- Anexo B: Análisis de requerimientos
- Anexo C: Diseño
- Anexo D: Actas de reuniones
- Anexo E: Documentación técnica
- Anexo F: Análisis de las tecnologías y herramientas utilizadas
- Anexo G: Proyecto planteado
- Anexo H: Primera presentación

# Capítulo 14

## 14 REFERENCIAS BIBLIOGRÁFICAS

**En este capítulo detallaremos las referencias bibliográficas utilizadas tanto en este informe final como en los anexos.**

### 14.1 REFERENCIAS EN INTERNET

#### **Sobre tratamiento de imágenes y segmentación**

- [1] <http://www.cvc.uab.es/shared/teach/a20383/practiques/practica3/enunciado.htm>  
Detección de Cuts: esta página trata de detección de cambios en secuencias de video, nos sirvió la parte que habla de los histogramas, su definición tridimensional y las funciones de distancia.
- [2] <http://www.infomed.sld.cu/instituciones/cecam/revista/R0100A02.pdf>  
Algoritmo de segmentación topológico para imágenes adquiridas de la tomografía computada y la resonancia magnética, Revista Cubana de Informática Médica. Resultaba interesante el método de segmentación, aunque excesivamente complicado para nuestros propósitos. A tener en cuenta para versiones de “afinamiento” del producto que se realicen.
- [3] <http://yupana.autonoma.edu.co/porta/asignaturas/investigacion/proyd/ne.../preproce.htm>  
Pre-Procesamiento digital de imágenes  
Overview para comprender el procesamiento de imágenes, nos fue muy útil en el comienzo para ver el proceso que se sigue normalmente para procesar una imagen.

- [4] <http://www.ctr.columbia.edu/~jrsmith/html/pubs/abtfsfir/bintexture.html>  
Automated binary texture feature sets for image retrieval  
Overview sobre extracción de texturas para el matcheo de imágenes por este criterio.
- [5] <http://www.iie.edu.uy/investigación/grupos/gti/imagina/Manual97.htm>  
Imagina 1.0 (Proyecto del Instituto de Ingeniería Eléctrica)  
Software de tratamiento de imágenes. Realiza una cantidad de “efectos visuales” sobre imágenes. Nosotros tenemos resuelto este tema a través de la librería ImageMagick.

### **Sobre formato de archivos de imágenes**

- [6] <http://www.medicalconnections.co.uk/html/dicomobjects.html>  
Producto para la lectura, impresión, conversión, etc. de archivos DICOM, enfocado como librerías para desarrolladores. Como dijimos antes el ImageMagick ya posee estas funcionalidades.
- [7] <http://www.hcu-lblesa.es%20/mane/%20noticia/%20didcom3.htm>  
El estándar DICOM 3.0, overview general.

### **Sobre búsqueda y almacenamiento de gran cantidad de imágenes**

- [8] <http://citeseer.nj.nec.com/175002.html>  
Hierarchical Browsing and Search of Large Image Database.
- [9] <http://citeseer.nj.nec.com/chen98similarity.html>  
Similarity Pyramids for Browsing and Organization of Large Image Databases
- [10] <http://citeseer.nj.nec.com/chen99active.html>  
Active browsing using Similarity Pyramids
- [11] [http://www.ece.purdue.edu/~bouman/publications/pdf/thesis\\_jauyuen.pdf](http://www.ece.purdue.edu/~bouman/publications/pdf/thesis_jauyuen.pdf)  
Image database management using similarity pyramids

Estos cuatro artículos fueron de los mas interesantes. Describían formas de clasificación de las imágenes en una base de datos grande para una búsqueda eficiente mediante sus descriptores. Sin embargo, por las características del producto que finalmente estamos desarrollando, no se hace necesaria la aplicación de estas técnicas.

### **Sobre implementación**

- [12] [http://msdn.microsoft.com/library/en-es/dncomg/html/msdn\\_house1.asp?frame=true](http://msdn.microsoft.com/library/en-es/dncomg/html/msdn_house1.asp?frame=true)  
MFC/COM Objects 1:Create a Simple Object

- [13] <http://club.idecnet.com/~chavesj/dcom/client.htm>  
Cliente DCOM  
Ejemplo de creación de un cliente DCOM
- [14] [http://members.es.tripod.de/adromeda\\_studios/paginas/tutoriales/articulo02.htm](http://members.es.tripod.de/adromeda_studios/paginas/tutoriales/articulo02.htm)  
COM / DICOM en C++  
Clara explicación del COM con ejemplos, es el que nos fue mas útil para las pruebas del tema que llevamos acabo.
- [15] <http://www.monografias.com/trabajos5/visualcurso/visualcurso.shtml>  
Curso Visual C++
- [16] <http://guille.costasol.net/colabora/vb2vc.htm>  
Conexión entre Visual Basic y Visual C++  
Reseña de cómo intercambiar datos entre los dos lenguajes, no a nivel de objetos sino a nivel de funciones.
- [17] [http://www.lafacu.com/apuntes/informatica/compi\\_visual\\_c/default.htm](http://www.lafacu.com/apuntes/informatica/compi_visual_c/default.htm)  
El compilador de Visual C++ versión 6  
Ayuda para el editor de Visual C++

### **Sobre estándares**

- [18] <http://www.mpeg7.com>  
Página principal sobre el estándar
- [19] <http://www.w3.org/TR/xmlschema-0/>  
XML schemas.  
Sirve como introducción al tema.
- [20] <http://viplab.hanyang.ac.kr/~hhlee/reference/ip/mpeg7/w3445.html>  
El estándar MPEG7  
Draft completo del estándar.
- [21] <http://archive.dstc.edu.au/mpeg7-ddl/>  
MPEG7: DDL Working draft 4.0  
Draft sobre el lenguaje de definición de descriptores (Description Definition Lenguaje).
- [22] <http://www.darmstadt.gmd.de/mobile/MPEG7/Documents.html>  
MPEG7: Study of ISO/IECFCD 15938-6 information tecnologia multimedia content description interface part6: reference software

## Productos de búsqueda y tratamiento de imágenes

[23] [http://zomax.wins.uva.nl/ret\\_user](http://zomax.wins.uva.nl/ret_user)

**PicToSeek:** A content-based image search system for the world wide web

De los productos más interesantes, dada una imagen busca similares, pero lo hacia básicamente por color.

PicToSeek trabaja con imágenes obtenidas por web-crawlers de internet, manteniendolas catalogadas automáticamente en primer lugar por métodos del análisis de imágenes básicos como ser: JFIF-GIF, gris-color, fotografía-dibujo, tamaño, fecha de creación, y profundidad de color. Una segunda clasificación se realiza según características invariantes del color de la imagen. En base a estos datos indexa las imágenes, de modo de poder detectar principalmente:

- distintas “fotografías” (por ejemplo, de distintos ángulos) de las mismas imágenes
- variaciones fotométricas en las imágenes debido a los cambios en sombras, brillos e iluminación
- fragmentación, obstrucción o distorción de la imagen original por la presencia de otros objetos en la escena



El servidor (motor) está implementado en C++ y el cliente (interfase) en Java.

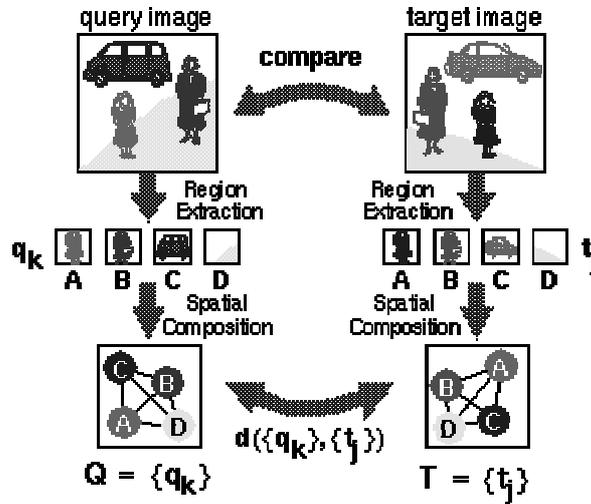
[24] <http://www.qbic.almaden.ibm.com/>

**QBIC: IBM's Query By Image Content**

Motor de búsqueda para realizar consultas sobre grandes bases de datos de imágenes, basándose principalmente en porcentajes y disposición de colores y texturas. Claramente, el mayor objetivo de este motor está en el poder indexar características de grandes cantidades de imágenes, basándose en las posibilidades que brinda el motor DB2 de IBM.

[25] <http://disney.ctr.columbia.edu/safe>

**SaFe**, permite al usuario separar una imagen en objetos, y detectar en una segunda imagen a los mismos, identificando los cambios en la distribución espacial.



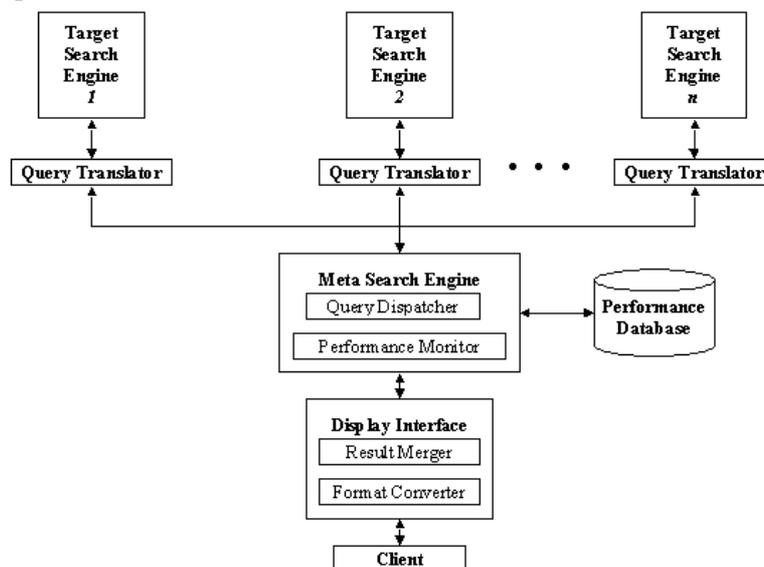
[26] <http://disney.ctr.columbia.edu/webseek/> -- [www.virage.com](http://www.virage.com) -- [www.google.com](http://www.google.com)

**WebSeek / Virage / Google:** Motores de búsqueda basados en clasificación por categorías.

Estos motores de búsqueda recolectan información acerca de las imágenes que componen sus bases de datos por mecanismos alternativos al análisis de la imagen en si misma: información de la página web que las contiene, unión de bancos de datos especializados o clasificación asistida por humanos.

[27] <http://ana.ctr.columbia.edu/metaseek/MetaSEEk.html>

**Metaseek**, meta-buscador, que realiza búsqueda de similitudes en imágenes en varios motores de búsqueda, realizando luego una clasificación. El siguiente diagrama muestra el esquema de trabajo que utiliza:



## 14.2 LIBROS

[28] “APPLYING UML AND PATTERNS” DE CRAIG LARMAN

Una descripción muy detallada de todos los pasos a seguir en el proceso de desarrollo de software y su documentación. Resultó una guía fundamental durante todo el trayecto de desarrollo del producto.

[29] “DESIGN PATTERNS, ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE” DE ERICH GAMMA, RICHARD HELM, RALPH JOHNSON Y JOHN VLISSIDES

Explica los patrones de diseño más utilizados, incluyendo cómo pasar de la etapa de análisis al diseño y de ahí a la implementación. Intentamos utilizarlo al máximo de modo de lograr un buen diseño para el sistema.

[30] “TRATAMIENTO DIGITAL DE IMÁGENES” DE GONZÁLEZ/WOODS

Edición 1996, Addison-Wesley Iberoamericana / Diaz de Santos, 773 páginas.

Fue extremadamente útil en un comienzo como la primera aproximación al tema de tratamiento de imágenes. De allí sacamos los primeros algoritmos que implementamos a modo de prueba. Abarca todos los pasos del procesamiento de la imágenes, “efectos visuales”, mejoramiento de la imagen, segmentación y extracción de descriptores.

[31] “INTRODUCCIÓN AL PROCESAMIENTO Y ANÁLISIS DE IMÁGENES DIGITALES” DE R. MOLINA

Incluye también un panorama general del tratamiento de imágenes digitales, pero con un enfoque más matemático, que nos ha resultado útil a la hora de estudiar el tratamiento de descriptores.

[32] “ENFOQUE LÓGICO-COMBINATORIO AL RECONOCIMIENTO DE PATRONES” DE J. RUÍZ SHULCLOPER, A. GUZMÁN ARENAS Y J. F. MARTÍNEZ TRINIDAD

Libro correspondiente al curso “Reconocimiento de patrones con enfoque lógico-combinatorio”, dictado por el Dr. Manuel Lazo Cortéz, del Instituto de Cibernética, Matemática y Física de Cuba.

El curso abarcó varios aspectos básicos en el reconocimiento de características de imágenes digitales, a partir de conjuntos de rasgos distintivos, y la creación a partir de ellos de “testores”. Fue un importante aporte a la hora de definir un modelo de análisis de imágenes, y la utilización de descriptores.

[33] “CURSO DE PROGRAMACIÓN C++, PROGRAMACIÓN ORIENTADA A OBJETOS” DE FCO. JAVIER CEBALLOS

Edición 1991, Addison-Wesley Iberoamericana, 780 páginas

Libro de referencia para la implementación. No nos fue muy útil.

- [34] “VISUAL C++ 6.0, MANUAL DE REFERENCIA” DE CHRIS H. PAPPAS Y WILLIAM H. MURRAY  
Edición 1999, McGraw-Hill, 944 páginas  
Libro de referencia para la implementación. Fue útil para comprender algunos puntos específicos de Visual C++, pero al no implementar la interface gráfica en Visual C++ y tampoco usar funciones específicas de VC++, sino usar del estandar C++, nos cayó en desuso.
- [35] “THE C++ PROGRAMMING LANGUAGE”, THIRD EDITION, DE BJARNE STROUSTRUP  
Edición 1997, Addison-Wesley, 911 páginas  
Libro de referencia para la implementación. Este fue el mejor, casi siempre fué el que tuvo las respuestas para nuestras dudas en la implementación

### 14.3 HERRAMIENTAS UTILIZADAS

- [36] IMAGEMAGICK  
<http://imagemagick.com>  
Librería utilizada para tratamiento y segmentación de imágenes.
- [37] XERCES  
<http://www.apache.org/xerces-c/index.html>  
Librería utilizada para el manejo de XML desde Visual C++.
- [38] DOXYGEN  
<http://www.doxygen.com>  
Sistema para generación de documentación técnica a partir de código fuente en C++.

# Capítulo 15

## 15 AGRADECIMIENTOS

---

**En este capítulo nombraremos a tan solo algunas de las personas que recordamos como apoyo importante en distintas etapas del proyecto.**

Ing. Alvaro Pardo  
Ing. Anselmo Martínez  
Dr. Gustavo Stellardi  
Dra. Cristina Balbiani  
Eduardo Blanco  
Gustavo Beiró  
Leonardo García  
Mauro Flores  
Pablo Costa  
Paola Ortiz  
Rafael Grompone  
Salvador Tercia