

Universidad Central “Marta Abreu” de las Villas  
Facultad de Matemática Física y Computación  
Licenciatura en Ciencia de la Computación



# TRABAJO DE DIPLOMA

*Herramienta computacional para la segmentación de tumores  
cerebrales en Imágenes de Resonancia Magnética*

**Autor:** Alvaro Javier Fuentes Suárez  
**Tutores:** Lic. Jorge Enrique Moreira Broche  
Lic. Marcelino Rodríguez Cancio

“Año 54 de la Revolución”  
Santa Clara  
**2012**



## Declaración Jurada

El que suscribe, \_\_\_\_\_, hago constar que el trabajo titulado \_\_\_\_\_ fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de los estudios de la especialidad de \_\_\_\_\_, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

\_\_\_\_\_  
Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

\_\_\_\_\_  
Firma del tutor

\_\_\_\_\_  
Firma del jefe del Laboratorio

\_\_\_\_\_  
Fecha

*A mis padres, por enseñarme el camino y recorrerlo conmigo...*

*A Beatriz, por ser mi inspiración...*

*A mi abuela y mi hermano...*

## ABSTRACT

This work aims to develop an overall method and application to the segmentation of brain tumors in magnetic resonance imaging that allows semi-automatically get the position, size, volume and shape of the tumor as well as reporting that details this information .

As a result this research presents an application for semi-automatic segmentation of brain tumors called **BrainApp**, whose main features are: segmenting images showing tumors, view segmentation, calculate the volume of the segmented tumor and its main dimensions, and display / save the calculated data. **BrainApp** ensures reproducibility, efficiency and robustness in image processing, also includes an interface that ensures a working method with the application simple and functional.

The semiautomatic method designed for the new tool consists of the following steps: smoothing the image using a filter that preserves edges between regions, image segmentation using a technique based on "region-growing" and ranges of intensities or statistical information, and finally, refinement of the segmentation.

The implementation of **BrainApp** consists of the following components: images, pipes, views and visualizations. They ensure the extensibility of the application in future versions and optional allow interconnection between the different elements involved in the process of obtaining the segmentation.

The tools used in the development of **BrainApp** were Qt, PyQt, ITK and VTK. The programming language used was Python.

Keywords: semiautomatic segmentation, brain tumor, volume and dimensions of a tumor, magnetic resonance imaging

---

## RESUMEN

El presente trabajo tiene como objetivo general desarrollar un método y una aplicación para la segmentación de tumores cerebrales en Imágenes de Resonancia Magnética que permita de manera semiautomática obtener la posición, tamaño, volumen y forma del tumor, así como presentar informes donde se detallen estos datos.

Como resultado de la presente investigación se presenta una aplicación para la segmentación semiautomática de tumores cerebrales denominada **BrainApp**, cuyas principales funcionalidades son: segmentar las imágenes donde se muestran los tumores, visualizar las segmentaciones, calcular el volumen del tumor segmentado y sus dimensiones principales, y mostrar/guardar los datos calculados. **BrainApp** garantiza reproducibilidad, eficiencia y robustez en el procesamiento de las imágenes, incluye además una interfaz que garantiza un método de trabajo con la aplicación de forma simple y funcional.

El método semiautomático diseñado para la nueva herramienta consta de los siguientes pasos: suavizado de la imagen usando un filtro que preserva los bordes entre las regiones, segmentación de la imagen usando una técnica basada en “region-growing” y rangos de intensidades o información estadística, y por último, refinamiento de la segmentación.

La implementación de **BrainApp** está conformada por los siguientes componentes: *imágenes, tuberías, vistas y visualizaciones*. Estos garantizan la extensibilidad de la aplicación en futuras versiones y permiten la interconexión opcional entre los diferentes elementos que intervienen en el proceso de obtención de la segmentación.

Las herramientas que se usaron en el desarrollo de **BrainApp** fueron Qt, PyQt, ITK y VTK. El lenguaje de programación usado fue Python.

Palabras clave: segmentación semiautomática, tumor cerebral, volumen y dimensiones de un tumor, imágenes de resonancia magnética

---

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO I. MARCO TEÓRICO.....	7
1.1. Tumores cerebrales .....	7
1.2. Imágenes de Resonancia Magnética .....	9
1.3. Diagnóstico de tumores usando IRM .....	10
1.4. Aspectos prácticos y apariencia en IRM de los tumores más comunes 12	
1.5. Segmentación de imágenes en el contexto de las IRM .....	15
1.5.1. Técnicas de segmentación y criterios de optimalidad .....	16
1.5.2. Segmentación de un tumor cerebral.....	18
1.5.3. Dificultades e ideas en el proceso de segmentación computacional de tumores cerebrales en IRM.....	21
1.6. Actualidad en la segmentación de tumores cerebrales.....	23
1.6.1. Segmentación manual.....	24
1.6.2. Segmentación semiautomática.....	26
1.6.3. Segmentación totalmente automática .....	29
1.6.4. Herramientas computacionales existentes para la segmentación de tumores cerebrales .....	29
1.6.5. Técnicas para la validación de segmentaciones obtenidas mediante un proceso computacional .....	30
1.7. Herramientas utilizadas para la implementación en la presente investigación.....	31
1.7.1. Qt.....	31
1.7.2. PyQt .....	32

---

## ÍNDICE

---

1.7.3. The Insight Segmentation and Registration Toolkit (ITK) .....	32
1.7.4. The Visualization Toolkit (VTK) .....	33
1.7.5. Python .....	33
1.8. Conclusiones parciales .....	33
<b>CAPÍTULO II. DESARROLLO DE BRAINAPP .....</b>	<b>35</b>
2.1. Introducción .....	35
2.2. Diseño del método de trabajo con BrainApp .....	36
2.3. Funcionalidades de BrainApp .....	37
2.4. Componentes de BrainApp .....	38
2.4.1. Imágenes .....	39
2.4.2. Tuberías .....	39
2.4.3. Vistas .....	45
2.4.4. Visualizaciones .....	48
2.5. Procesamiento de las imágenes en BrainApp .....	49
2.6. Manejo de los datos en BrainApp .....	52
2.7. Cálculo del volumen y dimensiones del tumor segmentado .....	53
2.8. Interfaz gráfica de BrainApp .....	54
2.9. Validación de resultados de la aplicación .....	57
2.10. Conclusiones parciales .....	61
<b>CAPÍTULO III. MANUAL DE USUARIO DE BRAINAPP .....</b>	<b>62</b>
3.1. Introducción .....	62
3.2. Instalación y requisitos del sistema .....	62
3.3. Ventana principal .....	62
3.3.1. Menús de BrainApp .....	63
3.3.2. Barra de herramientas de BrainApp .....	64

---

## ÍNDICE

---

3.4. Carga de imágenes.....	65
3.4.1. Lista de imágenes cargadas .....	67
3.4.2. Modificación de las propiedades de las imágenes.....	68
3.5. Visualizaciones de las imágenes .....	69
3.6. Procesamiento de imágenes y configuración de los filtros.....	70
3.6.1. Cálculo de los datos de la segmentación .....	74
3.7. Conclusiones parciales .....	75
CONCLUSIONES .....	76
RECOMENDACIONES .....	77
BIBLIOGRAFÍA .....	78
ANEXO 1. MÉTRICAS PARA COMPARAR SEGMENTACIONES .....	87
ANEXO 2. IMPLEMENTACIÓN DEL CÁLCULO DEL VOLUMEN Y DIMENSIONES PRINCIPALES .....	88
ANEXO 3. MUESTRA COMPARATIVA ENTRE SEGMENTACIONES SEMIAUTOMÁTICAS Y MANUALES .....	91

---



# INTRODUCCIÓN

El uso de Escáneres de Resonancia Magnética actualmente es una práctica médica en pleno desarrollo a nivel mundial. Esta técnica basada en el fenómeno de Resonancia Magnética Nuclear brinda imágenes del interior del cuerpo humano con una alta resolución y sin riesgos para la salud del paciente.

Las Imágenes de Resonancia Magnética (en la adelante IRM) son superiores a las obtenidas con otras técnicas en la visualización de los tejidos blandos del cuerpo humano porque presentan mejor contraste (Price, 2007). Esto, unido a que no necesitan radiación ionizante, como las Tomografías Computacionales, hacen que las IRM hayan devenido indispensable en el diagnóstico, tratamiento y monitoreo de numerosas afecciones cerebrales entre las que se encuentran los tumores (Gordillo, 2010).

A nivel internacional se publican numerosas investigaciones abordando temas relacionados con el análisis digital de las IRM y la extracción de datos representativos sobre las mismas. Uno de los campos donde se trabaja arduamente es en la segmentación de tumores cerebrales pues este es un tema de investigación abierto donde todavía no se han obtenido resultados concluyentes. Precisamente sobre este tema versa la presente investigación.

## Planteamiento del problema

La alta sensibilidad del cerebro humano y las características propias de los tumores cerebrales, muchas veces inaccesibles mediante cirugía, constatan la necesidad de contar con técnicas como las IRM para la planificación del tratamiento y el monitoreo de este padecimiento. En este contexto es muy importante contar con las segmentaciones del tumor mostrado en las IRM. En el caso de la planificación del tratamiento la correcta segmentación del tumor permite que se puedan obtener datos necesarios para la aplicación de radiaciones o la realización de una cirugía guiada por imágenes. Por otro lado,

si la segmentación se realiza en imágenes tomadas con fines de monitoreo, se pueden calcular parámetros que evalúan la evolución del tumor para verificar el estado del padecimiento y comparar con etapas anteriores e incluso con casos parecidos.

Los aspectos mencionados con anterioridad ilustran la situación problemática y motivan el planteamiento del problema de investigación que se define a continuación.

### **Problema de investigación**

¿Cómo segmentar de manera semiautomática un tumor cerebral mostrado en una Imagen de Resonancia Magnética?

Enmarcado en este problema se define el objetivo general de la presente investigación.

### **Objetivo general**

Desarrollar un método y una aplicación para la segmentación de tumores cerebrales en Imágenes de Resonancia Magnética que permita de manera semiautomática obtener la posición, tamaño, volumen y forma del tumor, así como presentar informes donde se detallen estos datos.

Para cumplir el objetivo general se definen algunos objetivos específicos, enunciados a continuación:

### **Objetivos específicos**

1. Desarrollar un método semiautomático para la segmentación de tumores cerebrales en IRM que minimice la interacción humana.
2. Desarrollar un algoritmo para el cálculo del volumen y dimensiones de un tumor a partir de la segmentación del mismo.
3. Implementar una interfaz gráfica que permita al personal médico obtener segmentaciones de los tumores cerebrales en IRM.

4. Validar las segmentaciones obtenidas y los valores de volumen y dimensión calculados.

Para lograr los objetivos planteados se asumen las siguientes tareas de investigación.

### **Tareas de investigación**

- Análisis de los métodos, algoritmos y aplicaciones más usados para la segmentación de tumores cerebrales en IRM.
- Fundamentación teórica, metodológica y práctica de un método semiautomático para la segmentación de tumores cerebrales en IRM que minimice la interacción humana.
- Diseño e implementación de una aplicación para la segmentación de tumores cerebrales en IRM.
- Validación de las segmentaciones obtenidas de manera semiautomática a través de la comparación con segmentaciones manuales usando diversas métricas.

Estas tareas permitirán validar la hipótesis de la presente investigación.

### **Hipótesis de investigación**

La segmentación de un tumor cerebral mostrado en una Imagen de Resonancia Magnética puede ser desarrollada a través de un método y una aplicación que permita obtener (con una combinación de técnicas basadas en rangos de intensidades y “region-growing”) de manera semiautomática, la posición, tamaño, volumen y forma del tumor, así como presentar informes donde se detallen estos datos.

### **Justificación de la investigación**

Actualmente la segmentación de los tumores cerebrales con fines prácticos se realiza de forma interactiva o semiautomática usando herramientas de software

que asisten al especialista (Gordillo, 2010). La segmentación manual sigue siendo el método aceptado como “más exacto” ya que no existe un “estándar de oro” en la segmentación de tumores cerebrales en IRM (Zou et al., 2004, O'Donnell, 2001, Prastawa, 2007). El proceso de segmentación manual trae consigo dos propiedades indeseadas en el resultado obtenido: las diferencias intra e inter-operadores, que provoca falta de reproducibilidad. Además es una labor que consume tiempo y resulta monótona (Cichosz et al., 2010, Jaya and Thanushkodi, 2011, Selvanayaki and Karnan 2010), lo que puede causar imprecisiones en la segmentación final debido a errores humanos. Otras cuestiones a tener en cuenta son los costos de la obtención de los resultados, que de por sí ya son altos por las características propias de la tecnología, y se ven aumentados por la necesidad de contar con el trabajo de varios técnicos especialistas en la segmentación manual. Todo esto hace deseable el desarrollo de métodos semiautomáticos o totalmente automáticos, para la segmentación de tumores cerebrales en IRM.

A pesar de los esfuerzos la segmentación de tumores cerebrales es un campo de investigación abierto, no existen métodos que se destaquen ampliamente en su utilización práctica (Gordillo, 2010) y la publicación de investigaciones científicas sobre este tema es muy rica debido a que el problema es altamente complejo y constituye un reto al intelecto humano, en el campo de la medicina y la computación.

La segmentación de un tumor cerebral en una IRM es particularmente difícil por las razones siguientes: (Moon et al., 2002, Prastawa et al., 2003)

- Los tumores pueden tener variadas formas y tamaños.
- Pueden estar presentes en cualquier parte del cerebro.
- Se visualizan de diferentes formas en las IRM, en dependencia del tipo de tumor.
- Pueden confundirse con tejido normal.

- Pueden ser tumores que ocupan espacio (desplazan el tejido normal) o tumores infiltrados (cambian las propiedades del tejido normal).
- Reaccionan de diferentes formas con los agentes contrastantes: resultan resaltados, parcialmente resaltados o sin ningún efecto.
- Pueden aparecer con presencia de edema, lo cual suaviza y distorsiona los bordes del tumor en las IRM.

Una herramienta capaz de realizar una segmentación semiautomática de un tumor cerebral en una IRM, minimizando la interacción humana, reduciría las características negativas de la segmentación manual y aumentaría la capacidad de procesamiento al reducir el costo y el tiempo para la obtención de la segmentación.

El impacto social que tiene cualquier avance en el campo de la segmentación automática de los tumores cerebrales es alto pues ayudaría a mejorar el tratamiento de este padecimiento, lo que influye positivamente en la calidad de vida de un gran número de personas: pacientes y familiares.

### **Viabilidad de la investigación**

La presente investigación se nutre de toda la producción científica existente en el campo de procesamiento de imágenes y en ella se utilizan herramientas de software libre al alcance de los investigadores cubanos. Desde el punto de vista de hardware no tiene necesidades adicionales, pues la segmentación se realiza con la imagen digital obtenida desde un Escáner de Resonancia Magnética.

### **Aportes de la investigación**

Desde el punto de vista teórico la presente investigación aporta la sistematización conceptual sobre la definición de segmentación de un tumor cerebral, sus clasificaciones, métodos de procesamiento y técnicas de validación, así como las características que debe tener una aplicación destinada a este tipo de segmentación.

Los aportes de significación práctica se centran en la posible introducción de los resultados de la presente investigación a la praxis médica y computacional, acercándose a una solución donde se trabaja arduamente en la actualidad en el mundo científico y en la cual todavía no hay resultados concluyentes. Se presenta una herramienta para el trabajo diario de los especialistas que facilite el proceso de segmentación de tumores cerebrales en IRM. Esta herramienta busca ser lo suficientemente fiable y autónoma para minimizar la interacción humana y así garantizar la veracidad y reproducibilidad de las segmentaciones obtenidas.

### **Estructura del documento**

El presente informe está dividido en tres capítulos, incluyendo conclusiones generales, recomendaciones y anexos. En el primer capítulo se realiza un análisis de los conceptos y métodos más importantes en la actualidad para el proceso de segmentación de tumores cerebrales, también se comenta sobre las herramientas utilizadas en el desarrollo de la nueva aplicación. En el segundo capítulo se explica el desarrollo de la aplicación incluyendo el diseño del método de trabajo con la misma, al final del capítulo se realiza una validación de los resultados obtenidos. El capítulo tres y final está dedicado a la presentación del manual de usuario. En los anexos se amplía la información relativa a las métricas de validación, la implementación de los algoritmos y la comparación visual de los resultados de la aplicación.

## CAPÍTULO I. MARCO TEÓRICO

La segmentación de un tumor cerebral en una imagen de resonancia magnética es un proceso computacional que opera con una imagen digital. Este es un proceso complejo que involucra varios adelantos científicos del siglo XX en el campo de la física, la biomedicina y la computación. Usualmente para realizar la segmentación se utilizan avanzados métodos y herramientas desarrollados especialmente para el tratamiento digital de imágenes médicas.

En este capítulo se abordarán los conceptos, métodos y herramientas involucrados en el desarrollo de una herramienta que responda al objetivo general de la presente investigación. De manera general se expondrá la siguiente información:

- El concepto de tumor cerebral.
- El mecanismo de obtención de imágenes usando resonancia magnética y los tipos de imágenes que se obtienen, así como sus características más relevantes.
- Los detalles concernientes a cómo los tumores cerebrales se visualizan en una imagen obtenida por resonancia magnética.
- El proceso de segmentación digital de imágenes y su aplicación a la medicina, esencialmente en el tratamiento de los tumores cerebrales.
- El estado actual de las herramientas y métodos que posibilitan la segmentación de un tumor cerebral en una imagen de resonancia magnética.
- Las técnicas de validación de las segmentaciones realizadas de forma computacional.

### 1.1. Tumores cerebrales

La palabra tumor se refiere a cualquier proceso patológico que produzca una masa o bulto en el cuerpo. Los tumores cerebrales son una de las mayores

complicaciones de un amplio grupo de enfermedades denominadas neoplasmas, comúnmente llamadas cáncer. Los neoplasmas se derivan de células normales del cuerpo que pierden la capacidad natural de responder al mecanismo fisiológico que controla el crecimiento y el crecimiento incontrolado de las células conlleva a la formación de un tumor (Kundel and Dean, 2002).

Los tumores que tienen un lento crecimiento y carecen de capacidad para difundirse a otras regiones del cuerpo son denominados benignos, mientras los que crecen rápidamente y son capaces de expandirse a distantes lugares del cuerpo (proceso de metástasis) son denominados malignos. Dentro de este contexto los tumores cerebrales tienen dos categorías: los primarios, que se originan en el cerebro; y los secundarios, que son parte de un proceso de expansión del cáncer desde otros lugares del cuerpo hacia el cerebro (Gordillo, 2010).

A las personas con tumores, o tumores potenciales, se les realiza un escaneo con el propósito de detectar, clasificar, y comparar. La detección puede ser dividida en diagnóstico, reconocimiento general, búsqueda causal y visualización; en dependencia del nivel de sospecha (Gordillo, 2010). La clasificación del tumor consiste, idealmente, en hacer un diagnóstico del tejido o al menos determinar si es una manifestación de una enfermedad benigna o maligna. El radiólogo es requerido para clasificar todas las regiones sospechosas en la imagen. También se hace un proceso de reconocimiento para determinar la extensión de la enfermedad de manera local y global (en todo el cuerpo), esta etapa es importante para la selección del tratamiento adecuado y para la estimación de la prognosis. La comparación de imágenes se realiza luego para determinar los efectos y para chequear la recurrencia del tumor. Frecuentemente es necesario discriminar si los cambios en la visualización del tumor están propiciados por el tratamiento o por la recurrencia del mismo (Gordillo, 2010).



Las características no invasivas de la Resonancia Magnética y la gran nitidez con que las IRM visualizan los tejidos blandos han hecho que esta técnica se use ampliamente en el diagnóstico de tumores cerebrales (Price, 2007).

## **1.2. Imágenes de Resonancia Magnética**

La obtención de Imágenes de Resonancia Magnética (IRM) es una técnica basada en el fenómeno físico de Resonancia Magnética Nuclear inducido por un campo magnético. Esta técnica es muy utilizada en la medicina porque posibilita obtener imágenes con mayor contraste entre los tejidos blandos del cuerpo que las técnicas basadas en Rayos-X como las Tomografías Computacionales. Además no representan ningún riesgo para los pacientes ya que no usa radiación ionizante (Kuperman, 2000, Prasad, 2006, Preim and Bartz, 2007).

El equipo utilizado para obtener las imágenes se denomina Escáner de Resonancia Magnética, la característica principal de estos radica en la potencia del campo magnético que generan (medido en Teslas, T). Con los escáneres se obtienen tres tipos básicos de imágenes (Westbrook and Kaut, 1998):

- T1: se basan en el proceso de dispersión de energía de los núcleos hacia el ambiente que los rodea.
- T2: se basan en el proceso de intercambio de energía de los núcleos con sus vecinos.
- PD (Densidad de Protones, según sus siglas en inglés): se basan en las diferencias en la cantidad de protones por unidad de volumen.

Conocer los diferentes tipos de imágenes que se pueden obtener como parte del proceso de IRM es indispensable porque los tejidos se visualizan con diferentes grados de intensidad en cada una de ellas (Figura 1).

Las imágenes obtenidas desde el Escáner de Resonancia Magnética están conformadas por un conjunto de medidas en un espacio tridimensional que codifican la amplitud de la radiofrecuencia emitida en cada punto (Pham et al.,

2000). Estas imágenes representan un volumen (imagen 3D) en forma de un conjunto de cortes (imágenes 2D) (Kuperman, 2000).

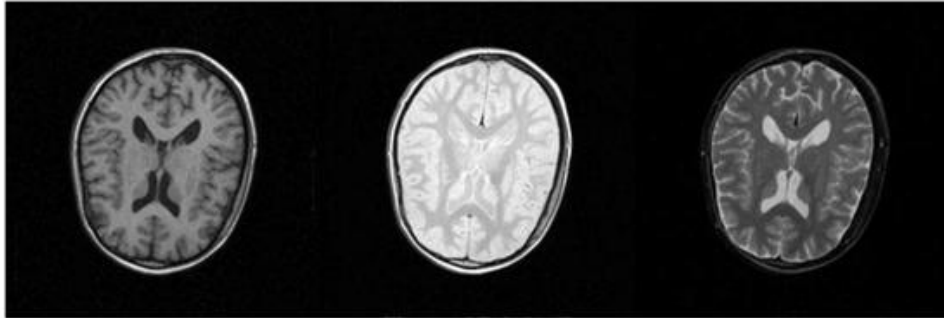


Figura 1. Tres tipos de IRM en una imagen del cerebro, de izquierda derecha: T1, PD, T2.

Las IRM contienen metadatos con información valiosa donde se especifican el nombre del paciente, edad, fecha del escaneo, etc. También se almacena información indispensable para reconstruir la imagen original: las coordenadas, el plano del corte, el número del corte y el espaciado. El espaciado de la imagen indica la distancia real que refleja un píxel de la imagen, tanto a lo largo del eje X como del eje Y, así como la distancia que hay entre un corte y otro (espaciado inter-cortes); un píxel junto con su espaciado en 3D se denomina vóxel (Westbrook and Kaut, 1998).

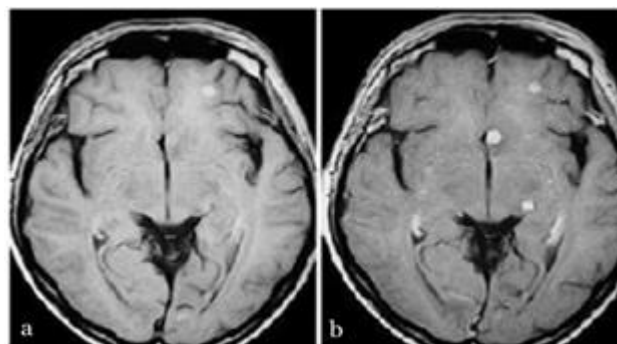
### 1.3. Diagnóstico de tumores usando IRM

La Resonancia Magnética es la técnica con mayor sensibilidad para la detección de tumores cerebrales así como para la determinación de la extensión de la enfermedad. Con ella se obtienen imágenes con una buena resolución y un alto grado de caracterización de las lesiones (Price, 2007). El uso de los diversos tipos de imágenes de resonancia magnética (T1,T2,PD) proveen al radiólogo de diferentes escalas de intensidades para las regiones o estructuras anatómicas de interés, y como los tumores cerebrales están conformados por tejidos biológicamente diferentes, resulta muy importante la combinación de la información de cada imagen para la determinación de la localización, extensión, diagnóstico y prognosis del tumor (Gordillo, 2010).

Para aumentar el contraste entre las señales capturadas como parte del proceso de IRM se utilizan agentes contrastantes externos que alteran la señal de los protones del agua permitiendo extraer información acerca de la distribución espacial, función del tejido, actividad metabólica, entre otras. Estos agentes pueden aumentar considerablemente el contraste entre el área perteneciente al tumor y la región circundante por lo que han llegado a ser un aspecto clave del diagnóstico con IRM (Prasad, 2006).

El análisis de los tumores cerebrales requieren un mayor contraste para las lesiones que el análisis del sistema nervioso central, lo cual depende de la intensidad en la señal de la lesión con relación a la intensidad del tejido normal que la rodea (Muroff and Runge, 1995).

Como la información detallada de la morfología de la lesión es esencial para el diagnóstico diferencial, determinación de la gravedad, y para la selección y planeamiento de la terapia; para la mayoría de las enfermedades, y para muchos de los métodos de Resonancia Magnética actualmente disponibles, el uso de contrastantes es obligatorio (Gordillo, 2010), un ejemplo de este hecho se muestra en la Figura 2 obtenida del libro “Magnetic Resonance Tomography” (Reiser et al., 2008).



**Figura 2. Efecto del agente contrastante: (a) Sólo se observa un tumor, (b) luego de una dosis triple de contrastante se observan tres tumores.**

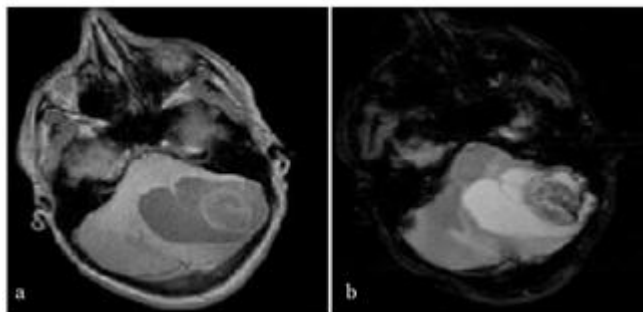
El mejoramiento del contraste en IRM también ayuda en la distinción de los tumores de otros procesos patológicos, y muestra signos básicos de la

respuesta del tumor al tratamiento como son: cambio en tamaño, morfología, y el grado de intensidad contrastante en la materia del tumor (Gordillo, 2010).

#### **1.4. Aspectos prácticos y apariencia en IRM de los tumores más comunes**

A continuación se mostrarán imágenes de los tumores cerebrales más comunes, todas fueron extraídas del libro “Magnetic Resonance Tomography” (Reiser et al., 2008). Los tumores se denotan según sus nombres médicos, de ellos solo se exponen algunos detalles que pueden ayudar a comprender la diversidad y características de las visualizaciones que presentan estas afecciones. También se muestran algunos porcentajes referidos al índice de aparición de cada tipo de tumor.

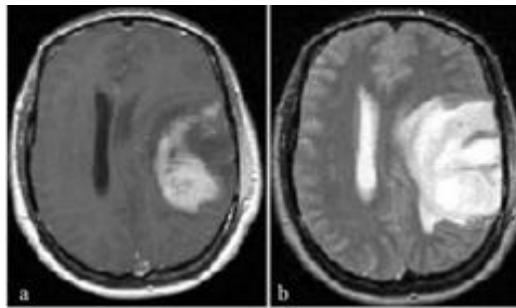
Los tumores astrocíticos agrupan el 80% (Gordillo, 2010) de los neoplasmas gliales, se presentan usualmente con una infiltración difusa del tumor. El borde del tumor (en T1 y T2) se observa mal definido, con una infiltración que destruye las estructuras anatómicas del cerebro (Ohgaki and Kleihues, 2005). Los dos tipos más comunes de tumores astrocíticos son el astrocitoma pilocítico y el glioblastomas multiforme (GBM).



**Figura 3. Apariencia típica de un Astrocitoma Pilocítico infratentorial. En (a) se muestra la imagen tipo T1 sin contrastante, y en (b) la tipo T2.**

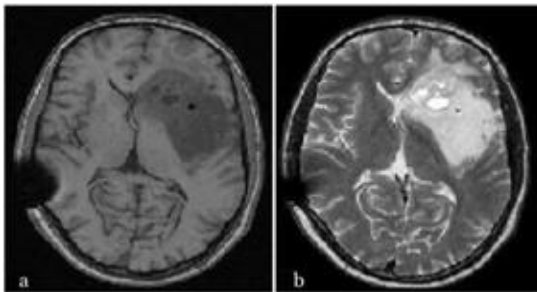
El Astrocitoma Pilocítico es una masa bien circunscrita que comúnmente tiene un quiste de tamaño considerable. La apariencia típica de un tumor de este tipo en los canales T1 y T2 se muestra en la Figura 3 donde se puede observar un tumor parcialmente sólido (Gordillo, 2010).

El astrocitoma maligno, también conocido como GBM, es el tumor cerebral más frecuente con aproximadamente el 12-15% de todos los tumores cerebrales y el 50-60% de todos los astrocíticos (Gordillo, 2010). El 95% de los GBM mejoran su visualización luego de la aplicación de un contrastante y usualmente exhiben intensidades heterogéneas en ambos tipos de imágenes (T1 y T2) causadas por la necrosis y hemorragia presente en muchos de los glioblastomas (Gordillo, 2010). Es normal que muestren altas intensidades en las imágenes tipo T2 como el ejemplo de la Figura 4.

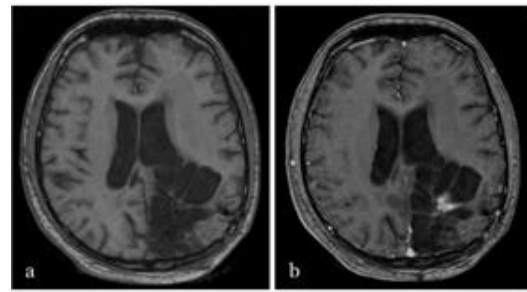


**Figura 4. GBM con presencia de una gran región sólida: (a) con un alto contraste, (b) patrón heterogéneo de la señal en T2.**

Los Oligodendroglioma, en las IRM, aparecen con muy bajas intensidades en las señales de las imágenes tipo T1 y con señales hiperintensas en T2, con un borde no muy bien delineado (Gordillo, 2010) (Figura 5, Figura 6).



**Figura 5. IRM de un Oligodendroglioma: (a) T1 (b) T2.**



**Figura 6. Oligodendroglioma anaplásico: (a) T1 sin contrastante, (b) con un fuerte contraste de las partes sólidas luego de aplicar el contrastante.**

Los tumores ependimarios, son comunes en niños, representan el 10% de los neoplasmas pediátricos del sistema nervioso central y el 5% de todos los tumores intra-axiales (Gordillo, 2010). Las IRM de ependimomas muestran la

heterogeneidad de los tumores y reflejan áreas de necrosis y sangramiento (Figura 7).

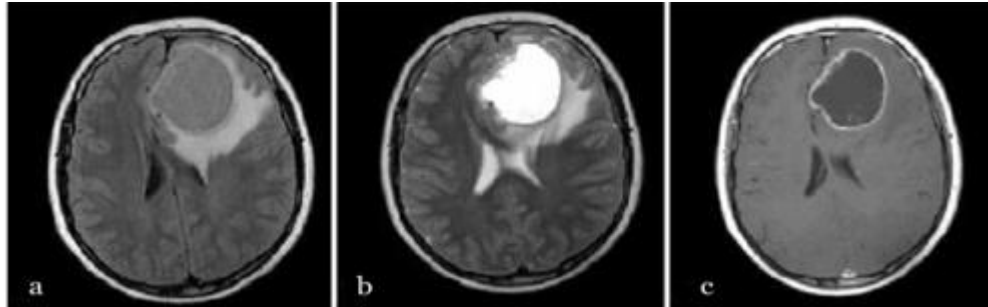


Figura 7. Ependimoma: (a) T1, (b) T2, (c) con resaltado en forma de anillo.

El meduloblastoma, es un tumor de tipo embrional maligno invasivo, que ocurre preponderantemente en niños. La apariencia característica en IRM es una lesión interventricular con una localización media en la imagen, este tumor muestra una señal isointensa en T2. El aumento de contraste es intenso de manera heterogénea y algunas veces en forma de anillo (Gordillo, 2010) (ver Figura 8).

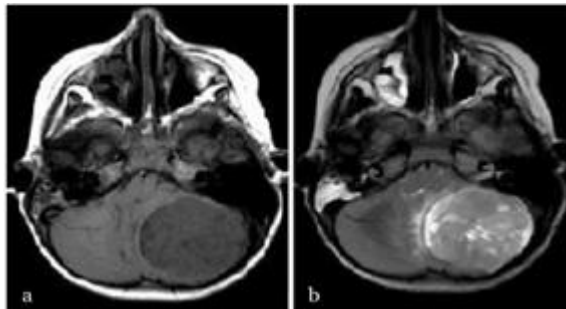


Figura 8. Meduloblastoma: se observa una masa heterogénea en (a) T1 y (b) T2.

Los meningiomas son los tumores intracraneales primarios no gliales más comunes. Estos son típicamente de lento crecimiento y benignos, normalmente se encuentran unidos a la dura madre. Los meningiomas se estima que comprendan entre el 13% y 26% de los tumores primarios del cerebro (Gordillo, 2010), la Figura 9 muestra un ejemplo típico.

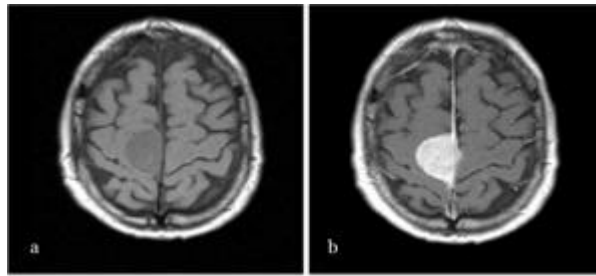


Figura 9. Meningioma en el canal T1, antes (a) y después (b) de la acción de un contrastante.

El hemangioblastoma capilar es un tumor de histogénesis incierta que ocurre esporádicamente y es asociado con el síndrome familiar von Hippel-Lindau (VHL). Se muestran de manera hiperintensa en T1 y generalmente no presentan edema alrededor (Gordillo, 2010), en la siguiente ilustración (Figura 10) se muestra un ejemplo.

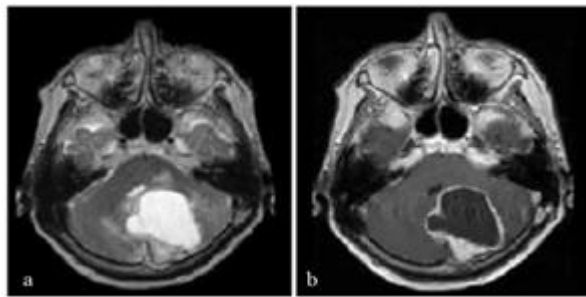


Figura 10. Hemangioblastoma capilar en un paciente con VHL: (a) normal, (b) luego de la administración de un contrastante.

Los ejemplos mostrados son casos típicos de tumores cerebrales donde se demuestra la amplia variedad de rasgos que estos presentan en las IRM, esta característica resulta de esencial importancia en la búsqueda de un método para la segmentación de los mismos, en el próximo epígrafe se comentará sobre este proceso.

### 1.5. Segmentación de imágenes en el contexto de las IRM

La segmentación es el proceso mediante el cual se particiona una imagen digital en regiones (o clases) que son homogéneas con respecto a una o más características bajo cierto criterio. Esto se logra identificando los píxeles o vóxeles (píxel en 3D) que pertenecen a una misma región/clase (Wong, 2005,

Ibáñez et al., 2005). También se denota como segmentación al resultado obtenido por el proceso homónimo (en la presente investigación se usará el término segmentación indistintamente salvo el caso en que exista ambigüedad).

El tratamiento digital de imágenes se divide en pequeñas unidades de procesamiento con un fin específico que se denominan filtros (Ibáñez et al., 2005). Los filtros aíslan operaciones tan diversas como el cálculo del histograma de la imagen o la reducción de ruido; y pueden ser tan simples como los que cambian el tipo de píxel de una imagen. Un proceso de segmentación puede estar compuesto por varios filtros.

Dentro de las aplicaciones de la segmentación se encuentran la delimitación de regiones de interés (ROI según sus siglas en inglés), usualmente para localizar estructuras anatómicas o regiones patológicas; la extracción de rasgos; medición de datos; la visualización de imágenes; e incluso se usa en la compresión de las mismas (Wong, 2005). En la presente investigación el proceso necesario para delimitar una ROI  $X$ , así como el resultado obtenido, se expresará de manera abreviada “*segmentación de  $X$* ”.

El proceso de segmentación puede verse afectado por numerosas dificultades, algunas de las más comunes son: pobre resolución espacial, variabilidad de la forma de los objetos de segmentación, ruido producido por artefactos dentro de la imagen (generados en el momento de la adquisición de las mismas) y las limitaciones físicas del proceso utilizado para la obtención de la imagen (Wong, 2005). Estas dificultades se evitan de una forma u otra con diferentes técnicas, a continuación se abordará con más profundidad este tema.

### **1.5.1. Técnicas de segmentación y criterios de optimalidad**

Un gran número de técnicas se han propuesto para el proceso de segmentación de una imagen digital, todas tienen sus debilidades y fortalezas, de manera general las técnicas se pueden dividir en cuatro grandes clases (Wong, 2005):



- **Las basadas en rangos de intensidades:** son las más simples que permiten una cierta independencia del especialista, su funcionamiento viene dado por la clasificación de los píxeles de una imagen según la intensidad de la señal que representan dentro de un rango de aceptación. Estos métodos pueden incluir un mecanismo de decisión extra para decidir si un píxel se incluye o no en la ROI, usualmente estos mecanismos involucran información espacial y/o estadística (Ahmed and Mohamad).
- **Las basadas en límites entre regiones:** tienen dos componentes fundamentales, detección de límites y unión de límites. De manera general un límite se define como un cambio abrupto de intensidades en una región de la imagen (transiciones claro-oscuro), para detectarlos se usa información matemática obtenida a través de gradientes de primer o segundo orden.
- **Las basadas en regiones:** esta técnica identifica regiones en la imagen y luego las fusiona o las divide. Los métodos para lograr estos dos pasos pueden ser muy variados y le dan gran riqueza a la técnica. Las regiones se identifican de manera que la imagen original pueda ser reconstruida de la totalidad de las mismas. La manera más simple de enfocar esta técnica es el método de “region-growing” que consiste en comenzar con una semilla e ir aumentando la región de acuerdo a algún criterio preestablecido que deben cumplir los píxeles vecinos a la región. Además se usa la idea contraria conocida como “region-splitting” que comienza con una región y la va dividiendo de acuerdo a un criterio de homogeneidad hasta obtener la región deseada (Ahmed and Mohamad).
- **Las basadas en la clasificación de píxeles:** usualmente son una generalización/automatización de los métodos basados en rangos de intensidades que permiten obtener los datos haciendo un análisis del histograma de la imagen.

Existen otras técnicas más avanzadas como son las basadas en modelos, las multimodales y las multivariantes (Wong, 2005), además resulta muy útil la aplicación de métodos puramente computacionales como la inteligencia artificial, predominantemente la lógica difusa. En la presente investigación se usará un enfoque semiautomático con una combinación de las técnicas basadas en rangos de intensidades y “region-growing”.

Producto de la dificultad inherente al proceso de segmentación de una imagen, en particular la segmentación de imágenes médicas con objetivos clínicos, se han propuesto algunos criterios que sirvan para medir la calidad de las mismas (Mazziotta et al. en Wong, 2005) algunos de los cuales son:

- Reproducibilidad.
- Precisión.
- Minimización de la subjetividad y dependencia del especialista.
- Robustez ante anatomías anormales.
- Capacidad de evolucionar la precisión a medida que la tecnología e información mejoren.
- Aplicabilidad en procesos clínicos y de investigación.
- Eficiencia tanto en la adquisición de los datos como en el procesamiento.

Después de haber aclarado algunos puntos sobre la segmentación de manera general se puede profundizar en su aplicación al tratamiento de tumores cerebrales.

### **1.5.2. Segmentación de un tumor cerebral**

Los tumores cerebrales son enfermedades complejas, no sólo porque son difíciles de tratar, sino también porque la obtención de información sobre ellas es un proceso difícil debido a las características del cerebro humano (órgano más sensible del cuerpo). Gracias a la aparición de las IRM se han hecho notables avances en la obtención del conocimiento necesario para diagnosticar,

tratar o monitorear las lesiones del sistema nervioso central, en particular los tumores.

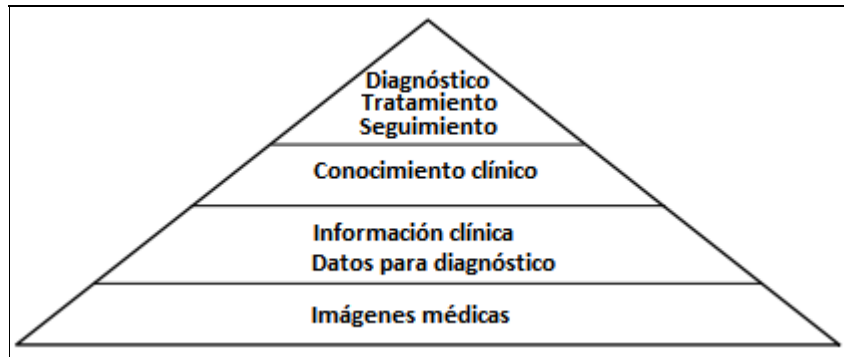


Figura 11. Pasos y objetivo final del análisis médico de imágenes en un ambiente clínico.

El análisis de imágenes médicas es un campo que ha tenido un gran auge en las últimas décadas del siglo veinte (Wong, 2005) y hoy es parte indispensable del trabajo diario de muchos médicos especialistas. La Figura 11 (adaptada de Wong, 2005) muestra el objetivo primordial de este tipo de análisis.

En la base de la pirámide están las imágenes médicas obtenidas por diversos equipos y en variados formatos, y en el tope, el objetivo principal del procesamiento de las mismas: hacer uso de la información extraída para formar un conjunto de conocimiento clínico que pueda devenir en el diagnóstico y tratamiento de una enfermedad específica. Esto define de manera inequívoca el objetivo de la segmentación en las imágenes médicas, pero:

1. ¿En qué consiste la segmentación de un tumor cerebral?
2. ¿Cuál es la información necesaria sobre un tumor cerebral que es posible obtener a partir de su segmentación en una IRM?

La respuesta a la primera interrogante así como una idea del nivel de complejidad del proceso de segmentación de un tumor cerebral la da Nelly Gordillo (Gordillo, 2010) en el siguiente fragmento:

*En el caso particular de un tumor cerebral, la segmentación consiste en separar los diferentes tejidos del tumor tales como tejido sólido o tumor activo, edema y necrosis; de los tejidos*

*normales tales como la materia gris, materia blanca y el líquido cefalorraquídeo. En los estudios sobre tumores cerebrales, la existencia de tejido anormal es fácilmente detectable la mayoría de las veces. No obstante, una segmentación precisa y reproducible así como la caracterización de las anormalidades, no es una tarea simple.*

Para la segunda interrogante primero hay que puntualizar cuál es la importancia de contar con una segmentación del tumor cerebral, esta va más allá de la importancia general mostrada en la Figura 11, ya que en muchos casos los tumores cerebrales son totalmente inaccesibles quirúrgicamente y el único tratamiento válido es el uso de radioterapia, donde la información con que se cuenta es la obtenida directamente de la segmentación (y la imagen de manera general). Además producto de la precisión necesaria en los tratamientos quirúrgicos también se han desarrollado técnicas de operación guiada por imágenes (mínimo acceso) donde se hace indispensable una descripción exacta del tejido patológico y su área circundante (Gering et al., 2002, Moon et al., 2002).

Aclarado este punto se enumeran algunas características que se pueden obtener directamente de la segmentación de un tumor cerebral que son indispensables para el diagnóstico y tratamiento:

1. Localización del tumor.
2. Dimensiones y extensión.
3. Volumen.

Además de la localización del tumor, también es importante la estimación del volumen del mismo y el cálculo de sus dimensiones principales. Estas son la base de metodologías que evalúan la evolución del tumor como la MacDonald, RECIST o la RANO (Grupo de Neurooncología, 2011). También con un análisis más profundo de la segmentación por parte de los especialistas se pueden determinar los tipos de tejidos que están presentes en la lesión.

### 1.5.3. Dificultades e ideas en el proceso de segmentación computacional de tumores cerebrales en IRM

Los tumores cerebrales poseen ciertas características que hacen que sea en extremo difícil su segmentación automática en IRM, principalmente se destacan las siguientes (Moon et al., 2002, Prastawa et al., 2003).

1. Los tumores pueden tener variadas formas y tamaños.
2. Pueden estar presentes en cualquier parte del cerebro.
3. Se visualizan de diferentes formas en las IRM, en dependencia del tipo de tumor.
4. Pueden confundirse con tejido normal.
5. Pueden ser tumores que ocupan espacio (desplazan el tejido normal) o tumores infiltrados (cambian las propiedades del tejido normal).
6. Reaccionan de diferentes formas con los agentes contrastantes: resultan resaltados, parcialmente resaltados o sin ningún efecto.
7. Pueden aparecer con presencia de edema (suaviza y distorsiona los bordes del tumor en las IRM).

Estas características hacen difícil la implementación de algunas ideas básicas iniciales que surgen en el proceso de búsqueda de un método para la segmentación de tumores cerebrales y entran en conflicto directo con algunas de ellas (ver resumen en la Tabla 1). A continuación se analizarán varias ideas y las dificultades para su implementación:

- **Utilizar la forma del tumor:** podría pensarse que los tumores tienen una forma predefinida (usualmente de forma globular) pero esto a pesar de ser parcialmente cierto no es muy útil pues, como anteriormente se mostró (en 1.4), las formas en que se visualizan los tumores en IRM son muy variadas y presentan muchos rasgos indefinidos que hacen muy difícil, si no imposible, clasificar una ROI analizando solamente su forma.
- **Basarse en la posición de las estructuras del cerebro:** esta idea tiene dos inconvenientes fundamentales, primero el hecho de que antes es

necesario determinar las estructuras en la imagen, lo cual se puede considerar un proceso de segmentación en sí que es muy propenso a fracasar pues los métodos de segmentación del cerebro sano normalmente fallan en presencia de un tumor (Gering et al., 2002). Como segundo inconveniente se tiene el hecho de que los tumores cerebrales pueden destruir, infiltrarse o desplazar las estructuras anatómicas del cerebro (Prastawa, 2007). También se puede agregar que los tumores pueden estar presentes en cualquier lugar del cerebro, es decir pueden trastornar cualquiera de sus estructuras, ninguna se puede presuponer totalmente sana.

- **Analizar las coincidencias entre los diferentes tipos de IRM:** cualquier análisis que se haga sobre un tumor en IRM que involucre varios de sus tipos de imágenes tendría que tener en cuenta las grandes diferencias en la visualización y sobre todo el hecho de que en algunos casos el tumor puede ser practicante indistinguible en un tipo de imagen, de todas formas esta es una idea válida que se usa mucho.
- **Analizar la simetría de la imagen:** el cerebro humano es relativamente simétrico y este conocimiento ha sido explotado por varios métodos de segmentación de tumores cerebrales que detectan asimetrías en la imagen, no obstante uno de los mayores inconvenientes de esta técnica es la búsqueda del plano de simetría que se ve seriamente dificultado por la presencia de ruido en las IRM y por las diferencias en las posiciones del paciente a la hora de adquirir la imagen (Khotanlou et al., 2009, Joshi et al.).
- **Basarse en las intensidades:** esta técnica sería ideal, por su fácil implementación, si los tumores tuvieran un rango de intensidades bien definido pero esto no es así. Como se graficó anteriormente (epígrafe 1.4) la intensidad de las visualizaciones de los tumores varían desde las más brillantes hasta las más oscuras. Además por si fuera poco los

efectos de los agentes contrastantes pueden no ser homogéneos y los tumores pueden parecer tejido normal.

- **Realizar una delineación de las regiones en la imagen:** esta técnica es muy útil para la segmentación de las estructuras anatómicas del cerebro sano, pero en muchos casos falla al aplicarla a los tumores pues estos normalmente tienen sus bordes difusos debido a la presencia de edema. También por el hecho de que algunos tumores se confunden con el tejido normal y por los diferentes niveles de reacción de algunos de ellos ante los agentes contrastantes.

**Tabla 1. Resumen de dificultades en el uso de ideas básicas para la segmentación de tumores cerebrales.**

<b>Idea base para la segmentación</b>	<b>Dificultades que inciden</b>
Forma de los tumores	1,2
Posición de las estructuras del cerebro	1,2,5
Coincidencias entre los tipos de IRM	3
Simetría del cerebro	4,5
Intensidad de la visualización	6,7
Delimitación de regiones	4,6,7

A pesar de las dificultades, en la bibliografía existente sobre el tema se encuentran métodos de segmentación que explotan una o más de las ideas comentadas, predominando los híbridos. Por estas razones el método sugerido en la presente investigación tiene un enfoque híbrido y se desarrolló para que funcione de manera semiautomática pues sólo un especialista puede determinar los parámetros adecuados para el proceso y la validación del resultado obtenido.

### **1.6. Actualidad en la segmentación de tumores cerebrales**

Para la segmentación de tumores cerebrales se han propuesto muchos métodos existiendo una gran fecundidad en el campo científico que los estudia (Selvanayaki and Karnan 2010). Todas las ideas expresadas en el epígrafe anterior han sido desarrolladas de una forma u otra, especialmente de manera

combinada, en técnicas que permitan obtener segmentaciones de los tumores cerebrales en IRM. Muchas de ellas han sido robustecidas usando métodos de inteligencia artificial.

A pesar del desarrollo alcanzado, en la comunidad científica hay un consenso general en el hecho de que no se ha desarrollado aún un método capaz de llevar a cabo una segmentación totalmente automática que brinde resultados indiscutiblemente superiores a los obtenidos mediante la segmentación manual realizada por los especialistas. Este hecho provoca que en la actualidad el “estándar de oro” para la segmentación de tumores cerebrales sea el realizado por un especialista manualmente (Zou et al., 2004, Warfield et al., 2002b, Prastawa, 2007).

La aceptación clínica de las técnicas ha dependido de la simplicidad del procesamiento y el grado de supervisión del usuario (Yao, 2006). Los métodos interactivos o semiautomáticos tienen altas probabilidades de mantenerse dominantes en la práctica por algunos años, especialmente en aplicaciones donde interpretaciones erróneas son inaceptables (McInerney and Terzopoulos, 2000) .

Las técnicas de segmentación de tumores cerebrales se pueden dividir en tres categorías de acuerdo al grado de interacción humana (Foo, 2006, Olabarriaga and Smeulders, 2001, Yao, 2006):

- segmentación manual,
- segmentación semiautomática, y
- segmentación totalmente automática.

### **1.6.1. Segmentación manual**

La segmentación manual de tumores cerebrales involucra trazar manualmente los límites del tumor y las estructuras de interés, o pintar las regiones de estructuras anatómicas con diferentes etiquetas. En la segmentación manual

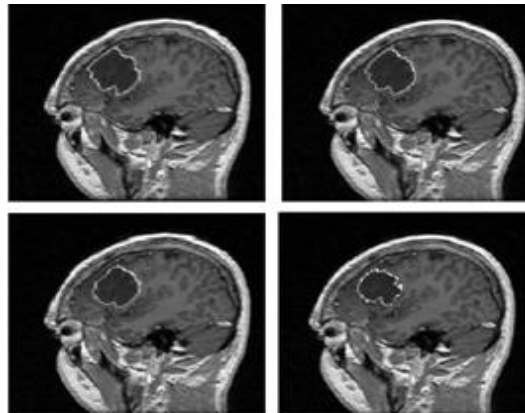


los expertos no sólo hacen uso de la información presentada en la imagen sino también de conocimiento adicional sobre anatomía (Gordillo, 2010).

La delineación manual requiere herramientas computacionales con sofisticadas interfaces de usuario para facilitar el trazado de la región de interés (ROI) y para mostrar las imágenes. En la práctica la selección de la región del tumor es una tarea tediosa y que consume tiempo ya que el especialista tiene que recorrer el conjunto de imágenes generadas por el escáner (ver epígrafe 1.1) para escoger las más representativas y en ellas cuidadosamente delinear la ROI (Wong, 2005).

La segmentación manual de imágenes tiene algunas características indeseadas que se describen a continuación (Cichosz et al., 2010, Jaya and Thanushkodi, 2011, Selvanayaki and Karnan 2010, Bokde et al., 2004):

- Las diferencias inter-operador: nunca dos operadores trabajando sobre la misma imagen obtienen la misma segmentación (Figura 12, (Luo et al., 2003)).



**Figura 12. Segmentación manual hecha por cuatro expertos diferentes, se pueden apreciar diferencias notables.**

- Las diferencias intra-operador: un mismo operador al repetir la segmentación no obtendrá una igual.
- La falta de reproducibilidad (por las dos dificultades anteriores).

- Presencia de irregularidades en los bordes: cuando el operador realiza la segmentación solo tiene en cuenta la información presente en el corte que está analizando, esto provoca que al interpolar la región segmentada (realizando un corte con un plano perpendicular al plano del corte original) esta se vea con un borde irregular, ver Figura 13 obtenida de (Prastawa et al., 2003).
- El costo en tiempo y agotamiento físico: un operador puede demorar mucho en el proceso de segmentación, monótono y repetitivo, además el tiempo que un radiólogo dedica a dicho proceso podría emplearse en la atención a otros pacientes.

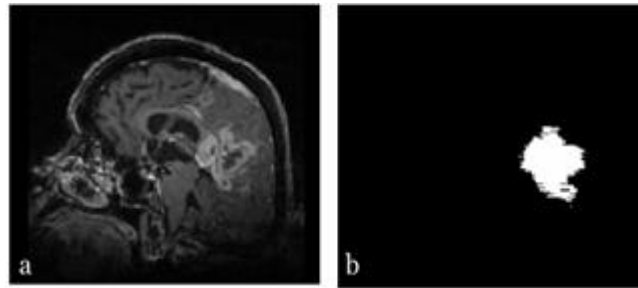


Figura 13. (a) Vista sagital de una imagen tipo T1 con contrastante, (b) segmentación manual con borde irregular, resultado de haberla realizado, corte por corte, en la dirección axial.

Estas dificultades son las que han propiciado que exista un gran interés en obtener métodos y herramientas computacionales que segmenten los tumores cerebrales automáticamente o al menos asistan al especialista en esta tarea. A pesar de los esfuerzos todavía no existe una herramienta totalmente eficaz en la realización de esta tarea.

### 1.6.2. Segmentación semiautomática

En este tipo de segmentación la intervención de un operador humano es necesaria para inicializar el método, chequear la precisión, o incluso corregir los resultados. Muchas de las investigaciones actuales están dirigidas hacia el desarrollo de técnicas de segmentación semiautomáticas, con la intención de minimizar la interacción humana (Gordillo, 2010).

En la Figura 14 (adaptada de Olabarriaga and Smeulders, 2001) se muestran los componentes principales de un método interactivo para la segmentación de un tumor cerebral. La parte computacional corresponde a una o más partes de software capaces de generar una delineación (segmentación) del tumor en cuestión, dados algunos parámetros. La parte interactiva es la encargada de mediar la información entre el usuario y la parte computacional: traduce la salida de la parte computacional en una visualización y los datos de entrada del usuario en parámetros para el método. La comunicación se logra usando los equipos de entrada/salida controlados por la interfaz gráfica de usuario del programa.

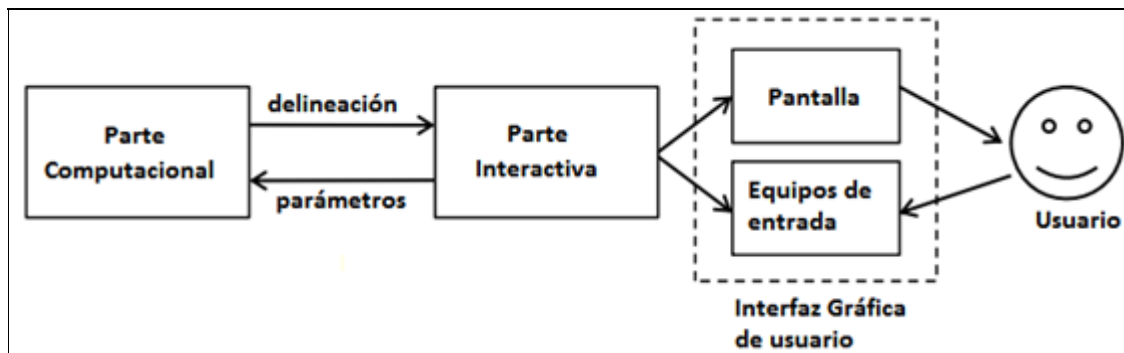


Figura 14. Componentes principales de un método interactivo para la segmentación de tumores cerebrales.

La interacción del usuario en el proceso de segmentación del tumor ha sido agrupada por Foo (Foo, 2006) en tres categorías:

1. **Inicialización:** puede involucrar la entrada de algunos parámetros, el pre-procesamiento de la imagen para eliminar el ruido o mejorar la visualización, un análisis inicial de la imagen para refinar la elección de parámetros, y la elección por parte del usuario de una posición donde comenzar la segmentación.
2. **Intervención:** debe ofrecer una dirección continua o intermitentemente del proceso para lograr un resultado deseado, responder a las entradas de datos realizadas por el usuario y brindar la posibilidad de detener el proceso en caso de ser necesario.

3. **Evaluación:** permitir la evaluación del resultado final del proceso, en caso de no ser satisfactorio reiniciar el procesamiento, desechando parcial o totalmente el resultado anterior.

La mayoría de los métodos pueden incluir uno o más de estos tipos de interacción (Gordillo, 2010).

Otro aspecto muy importante para los métodos semiautomáticos es la correcta combinación de las ventajas de la computación con la experiencia del operador, para lograr esto se deben tener en cuenta algunas de las siguientes estrategias (Gordillo, 2010, Olabarriaga and Smeulders, 2001):

- Diseñar un proceso que integre la interacción y la computación.
- Usar una entrada gráfica.
- Minimizar la interacción necesaria brindando opciones de selección al usuario.
- Involucrar al usuario en el proceso inicial de la segmentación para que provea información que pueda acelerar el cómputo de la misma.
- Visualizar correctamente el funcionamiento del proceso para permitir una apropiada respuesta del usuario.
- Mantener al usuario con el control durante todo el proceso para generar resultados precisos.
- Enfatizar en el procesamiento luego de cada interacción para generar resultados repetibles.
- Agregar comportamiento para elevar la abstracción en la interacción.
- Adicionar inteligencia para aprender de la interacción y reducir la necesidad de futuras intervenciones.

Aunque los métodos semiautomáticos pueden llegar a ser muy eficientes con el uso de las anteriores estrategias, de todas formas están sujetos a la interacción del usuario lo cual puede generar diferencias en las segmentaciones obtenidas por diferentes expertos e inclusive por el mismo usuario (Gordillo, 2010), esto

provoca que la búsqueda de un proceso de segmentación totalmente automático sea un objetivo con plena validez.

### **1.6.3. Segmentación totalmente automática**

En este tipo de segmentación no es necesaria ninguna interacción humana, toda la tarea la realiza un ordenador. Estos métodos generalmente incorporan en los algoritmos inteligencia humana en forma de conocimiento previo, y en muchos casos son desarrollados haciendo uso de la computación blanda (soft computing) y métodos basados en modelos, como los modelos deformables (Gordillo, 2010).

La segmentación totalmente automática, debido a su nivel de complejidad, queda fuera del alcance de la presente investigación que pretende brindar una herramienta semiautomática que sirva de ayuda a los expertos, para apoyar esta afirmación se reproducen las siguientes palabras de la doctora Nelly Gordillo (Gordillo, 2010):

*El estudio de la segmentación automática de tumores cerebrales representa un tema interesante de investigación en Aprendizaje Automático y Reconocimiento de Patrones, ya que muestra un problema que los humanos pueden aprender a hacer de manera efectiva. No obstante, el desarrollo de métodos automáticos con un alto grado de precisión representa un gran reto. Esto es fácilmente expresable en el hecho de que los humanos deben usar un complejo procesamiento visual e incorporar conocimiento especializado del dominio para realizar esta tarea.*

### **1.6.4. Herramientas computacionales existentes para la segmentación de tumores cerebrales**

Dentro de las aplicaciones que se usan para la segmentación de tumores cerebrales más citadas en la bibliografía se encuentran BrainVisa, FSL y Brainsuite. Estas aplicaciones (distribuidas gratuitamente) brindan muchas facilidades para la segmentación de las estructuras anatómicas cerebrales pero fallan ante la segmentación de los tumores producto de las dificultades ya enumeradas en 1.5.3, por esto se hace necesario el desarrollo de una

herramienta especializada capaz de facilitarle al especialista un ambiente de trabajo cómodo y útil.

Las aplicaciones con licencias privativas no pudieron ser accedidas para el desarrollo de la presente investigación.

#### **1.6.5. Técnicas para la validación de segmentaciones obtenidas mediante un proceso computacional**

Las segmentaciones de tumores cerebrales obtenidas de manera computacional se validan comparándolas con las segmentaciones realizadas por especialistas de forma manual (Cichosz et al., 2010, Zou et al., 2004).

Para cuantificar la calidad de una segmentación de un tumor cerebral se han propuesto varias medidas. Entre las más usadas se encuentran el Coeficiente de Similitud de Jaccard (JSC, según sus siglas en inglés) (Jaccard, 1912) y el Coeficiente de Similitud de Dice (DSC, según sus siglas en inglés) (Dice, 1945). También se utilizan las que miden la distancia entre dos superficies usando la medida de Hausdorff. Además en la literatura se pueden encontrar varias investigaciones que proponen nuevos métodos de comparación que se basan en ideas más complejas (Zou et al., 2004, Warfield et al., 2002c), no obstante JSC-DSC y las medidas basadas en la distancia de Hausdorff son las más usadas (Prastawa et al., 2004, Zou et al., 2004, Warfield et al., 2002a, Schmidt et al., 2005).

El Coeficiente de Similitud de Dice cuantifica el solapamiento entre dos conjuntos. Toman valores entre 0 y 1. Se acepta 0.8 como un *buen valor* para las comparaciones entre las segmentaciones realizadas computacionalmente y de forma manual (Prastawa, 2007).

Las medidas que involucran distancia entre superficies son varias: las que usan distancia clásica de Hausdorff y las que calculan promedios de distancias (inside/negative, outside/positive, and absolute), para estas técnicas se asumen como buenos resultados valores menores a 1mm (Prastawa, 2007).

La formulación matemática de las métricas se presenta en el Anexo 1.

Para calcular de manera automática todas estas medidas se puede utilizar un software de validación denominado VALMET (Gerig et al., 2001). Este software es una herramienta ampliamente utilizada para la validación por muchos investigadores en el campo de la segmentación de tumores cerebrales (Prastawa et al., 2004, Ho et al., 2002).

### **1.7. Herramientas utilizadas para la implementación en la presente investigación**

Para la manipulación y visualización de las imágenes médicas se necesitan herramientas computacionales especializadas las cuales son capaces de trabajar con diversos tipos de formatos, incluidos los formatos médicos estándares como el DICOM. Para el proceso de segmentación se necesitan dos etapas: la primera que aglutina varios procesos de iteración sobre la imagen, alteración de la misma, búsqueda de rasgos, análisis de intensidades, etc.; y una segunda etapa que se encarga de la visualización de la imagen que muestra la segmentación. Todo esto debe estar dentro de una interfaz capaz de interactuar con el usuario. Para cada una de estas necesidades se utilizaron herramientas computacionales de las cuales se precisaban dos características esenciales: que fueran software libre y multiplataforma. A continuación se comentará sobre cada una de ellas.

#### **1.7.1. Qt**

Es un framework multiplataforma para el desarrollo de aplicaciones, ampliamente utilizado en programas con una interfaz gráfica de usuario. Algunos de los proyectos que han usado Qt para su implementación son Adobe Photoshop Album, Google Earth, KDE, Opera, Skype, y VirtualBox.

Qt usa C++ estándar, pero hace extensivo uso del preprocesador de C para enriquecer el lenguaje. Qt también puede ser usado desde otros lenguajes a través de envolturas, soporta las plataformas más extendidas y es distribuido

bajo la licencia LGPL (GNU Lesser General Public License) lo que hace a Qt un software open-source.

En el desarrollo de la herramienta computacional que se propone en la presente investigación se escogió este framework como plataforma para la interfaz gráfica de usuario por el gran número de prestaciones que brinda y la calidad de las mismas, que vienen avaladas por los numerosos usuarios (privados e institucionales) que tiene. Además el soporte multiplataforma y el rendimiento de Qt es excelente.

### **1.7.2. PyQt**

Es una envoltura de Qt para Python. Una de las alternativas para la programación visual en Python, PyQt es software libre y está implementado como un módulo de extensión para este lenguaje. PyQt soporta Linux y otras variantes de Unix, así como Mac OS X y Windows. Se escogió este envoltorio pues brinda un excelente soporte para la mayoría de las funcionalidades de Qt.

### **1.7.3. The Insight Segmentation and Registration Toolkit (ITK)**

ITK es un framework open-source con licencia BSD para el desarrollo de aplicaciones destinadas a la manipulación/segmentación de imágenes. Está especialmente diseñado para el trabajo con imágenes biomédicas en sus formatos más usados (DICOM, RAW, ANALYZE, etc.). Implementado en C++ con un uso intensivo de la programación genérica y plantillas, puede ser usado desde las plataformas más extendidas: Unix, Windows, MacOS X. Posee un sistema de envolturas que sirven como interfaz para varios lenguajes: Tcl, Python y Java (Ibáñez et al., 2005). Está soportado por una amplia comunidad de colaboradores que han propiciado la liberación en el año 2012 de la versión 4.0.

Se decidió utilizar ITK porque brinda una amplia gama de filtros para el procesamiento de imágenes que abarcan las ramas de la segmentación, registro, suavización, análisis estadístico, análisis morfológico, etc. Además



posee un poderoso mecanismo para la iteración sobre imágenes de N dimensiones ( $N > 1$ ). La capacidad multiplataforma de ITK y su licencia como software libre cumplen con las características buscadas. La versión utilizada para el desarrollo de la presente investigación es la 3.16.

### **1.7.4. The Visualization Toolkit (VTK)**

VTK es un framework open-source para la visualización, con licencia BSD. Posee un motor gráfico multiplataforma (Windows, Unix, MacOS) con soporte para paralelismo. Tiene un mecanismo de envoltura para usar el motor gráfico desde Tcl, Perl, Python y Java. Soporta múltiples formatos de imágenes y además viene preparado para lograr una perfecta comunicación con ITK y todos sus componentes, así como para funcionar con Qt como interfaz gráfica. VTK ha sido usado en múltiples ocasiones para el desarrollo de aplicaciones destinadas a la visualización de imágenes médicas. Estas razones hacen de VTK una herramienta ideal para la presente investigación.

### **1.7.5. Python**

El lenguaje de programación de propósito general Python fue escogido para la implementación del código que controla los componentes de la aplicación porque tiene uno de los mayores grados de productividad (en cuanto a velocidad de desarrollo) de los lenguajes modernos, es multiplataforma, soporta la programación orientada a objetos y además está capacitado con todas las herramientas necesarias para la manipulación, visualización e interacción con las imágenes médicas.

## **1.8. Conclusiones parciales**

En el campo de la segmentación de imágenes existen varias técnicas que se pueden aplicar a la tarea de segmentación de un tumor cerebral en IRM, estas se clasifican en cuatro grupos: las basadas en rangos de intensidades, las basadas en regiones, las basadas en límites entre regiones y las basadas en clasificación de píxeles.

La literatura especializada recomienda los métodos semiautomáticos para la segmentación de tumores cerebrales en IRM y para la validación de los mismos se han propuesto diferentes alternativas donde el estándar de comparación es la segmentación manual realizada por un especialista. También existen herramientas científicas para realizar la validación de manera automática.

Existen herramientas computacionales altamente especializadas en la visualización y manipulación de imágenes médicas, entre ellas destacan ITK y VTK. Para la creación de interfaces de usuario, resalta por su alto grado de interoperabilidad y rendimiento, Qt con su envoltorio para Python, PyQt.

## CAPÍTULO II. DESARROLLO DE BRAINAPP

En este capítulo se comenta sobre el diseño e implementación de una nueva herramienta denominada **BrainApp**, la cual es fruto de la presente investigación. También se detallan los componentes fundamentales de la aplicación, así como los filtros de ITK y VTK usados en el procesamiento de las imágenes y en la visualización de las mismas.

El capítulo expone la siguiente información:

- Diseño e implementación del método de trabajo con **BrainApp**.
- Funcionalidades de la aplicación.
- Descripción de los componentes fundamentales de la aplicación, el procesamiento que realiza y la modelación de los datos.
- Desarrollo del algoritmo para el cálculo del volumen y dimensiones principales.
- Diseño de la interfaz gráfica de **BrainApp**.
- Validación de resultados obtenidos con la nueva herramienta.

### 2.1. Introducción

**BrainApp** puede ser definida como una herramienta para la segmentación semiautomática de tumores cerebrales en IRM, destinada a ser utilizada por los especialistas, con la cual se podrá obtener una segmentación y extraer las características fundamentales de la misma.

En el epígrafe 1.6.2 del capítulo anterior se especificaron algunas características imprescindibles que debe tener cualquier aplicación que pretenda servir como herramienta al especialista en el proceso de segmentación de tumores cerebrales en IRM. Estas características son las que han regido el desarrollo de **BrainApp**.

La aplicación que se propone consta de varios componentes visuales y de procesamiento. En su desarrollo se utilizó el patrón Modelo-Vista-Controlador (Gamma et al., 2003) que ha demostrado ser una solución útil y eficiente para este problema de diseño, así las clases en la implementación de **BrainApp** quedan divididas en tres paquetes: **model**, **view**, **controller**. En el diseño de **BrainApp** también se tuvo en cuenta su posible desarrollo futuro, por lo que se implementó de manera que pueda ser extendida y mejorada, cambiando o agregando nuevos componentes.

## 2.2. Diseño del método de trabajo con BrainApp

Teniendo en cuenta las necesidades de los usuarios para los cuales está destinada **BrainApp**, se implementó la herramienta de manera que facilite un método de trabajo que cumpla los requisitos de reproducibilidad e interacción analizados en el capítulo anterior (epígrafe 1.6.2).

El método se diseñó de manera que permita mantener el control sobre el procesamiento y sus resultados. Para esto se desarrolló un sistema de visualización que permite mostrar los datos que se van calculando y se creó un mecanismo de interacción con los componentes computacionales del proceso de segmentación que permite el ajuste de los parámetros a medida que se van ejecutando cada uno de los pasos, de manera eficiente y robusta.

Los pasos del método de trabajo son:

1. Suavizado de la imagen con un método que preserve los bordes entre las regiones.
2. Segmentación de la imagen con un método basado en “region-growing” y rangos de intensidades para los píxeles.
3. Segmentación de la imagen con un método basado en “region-growing” e información estadística (varianza y media de las intensidades de los píxeles en la región).

4. Refinamiento de la segmentación usando un método que re-clasifica los píxeles con un mecanismo de votación basado en la información de los vecinos.

Al inicio de cada paso el usuario deberá configurar los parámetros necesarios para su correcta ejecución y, luego de esta, se ofrecerá una visualización del resultado obtenido.

La decisión de incluir dos métodos de segmentación en el procedimiento general se debe a las características de la visualización de algunos tumores que poseen regiones con intensidades homogéneas y similares a las del tejido normal circundante (ver epígrafe 1.5.3), esto hace difícil una discriminación de los píxeles basados sólo en su intensidad, de aquí que se haga necesario contar con información extra, la cual se obtiene de los estadísticos descriptivos propios de la región que se está segmentando. La implementación del proceso de segmentación estadístico y cada uno de los componentes computacionales del procesamiento general se detallará más en el epígrafe 2.4.2, y en el 2.5 se explicará cómo se logró un mecanismo de trabajo en **BrainApp** eficiente y robusto.

### 2.3. Funcionalidades de BrainApp

Con base en el análisis realizado en el capítulo anterior y atendiendo a las necesidades de los usuarios potenciales de la nueva herramienta que se propone, se identificaron varias funcionalidades que debe brindar **BrainApp**:

1. Cargar imágenes en varios formatos: JPG, TIF, PNG, BMP, DICOM, RAW.
2. Cargar imágenes volumétricas (3D).
3. Cargar series de imágenes 2D.
4. Visualizar las imágenes de manera útil al especialista.
5. Mejorar las imágenes con un proceso de suavización.
6. Segmentar las imágenes usando una técnica basada en rangos de intensidades.

7. Segmentar las imágenes usando una técnica estadística.
8. Refinar las segmentaciones obtenidas.
9. Visualizar las segmentaciones.
10. Calcular el volumen del tumor segmentado y sus dimensiones. Mostrar y guardar los datos calculados.
11. Guardar la imagen procesada (con o sin segmentación).
12. Guardar/Cargar la configuración del procesamiento.

Todas estas funcionalidades se implementaron en **BrainApp**. Sobre la implementación de las mismas se comentará a continuación, comenzando por los componentes conceptuales de la aplicación que rigen todo el proceso de implementación y garantizan generalidad e interoperabilidad entre las partes de **BrainApp**.

## 2.4. Componentes de BrainApp

De manera conceptual la implementación de **BrainApp** está conformada por *imágenes*, *tuberías*, *vistas* y *visualizaciones*, que se denotarán como *componentes*. Las *imágenes* son los contenedores de los datos que se procesarán, las *tuberías* son los objetos que usando los filtros de ITK se encargan de la manipulación y/o segmentación de las *imágenes*, las *vistas* son las que usan los filtros de VTK para preparar las *imágenes* antes de ser visualizadas y por último las *visualizaciones* son las encargadas de mostrar los datos y/o *imágenes*. La Figura 15 (elaborada por el autor) grafica la interacción entre estos componentes.

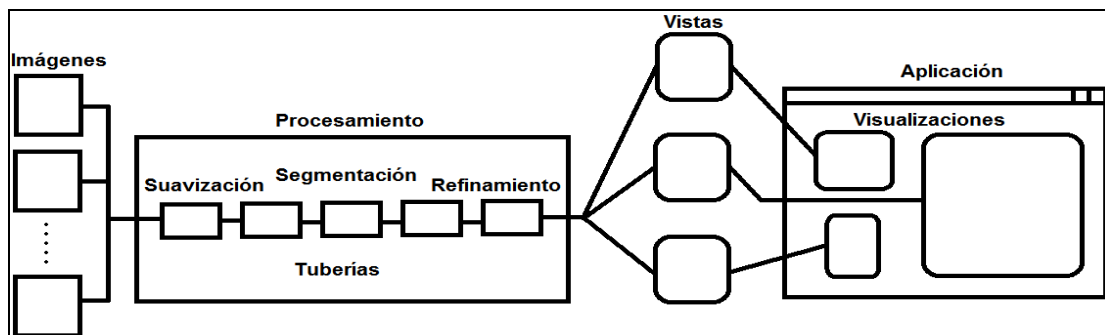


Figura 15. Interacción entre los componentes de la aplicación.

Los componentes anteriormente descritos se encuentran distribuidos en los paquetes de **BrainApp** de la siguiente forma: las *imágenes*, *tuberías* y *vistas* forman parte del modelo de datos y están en el paquete **model**, mientras que las *visualizaciones* como trabajan tanto con las *vistas* como con la interfaz gráfica de usuario se encuentran en el paquete de control (**controller**).

#### 2.4.1. Imágenes

Las imágenes médicas, y en particular las IRM, tienen algunos parámetros que son imprescindibles para su análisis y manipulación como ya se comentó en el epígrafe 1.1 del capítulo anterior. En **BrainApp** las imágenes están implementadas usando la clase **itk.Image**. Esta clase es capaz de almacenar los datos de imágenes en múltiples dimensiones y soporta varios tipos de valores para los píxeles, además maneja internamente las propiedades deseadas: espaciado, origen, etc. (Ibáñez et al., 2005)

ITK tiene un variado soporte para los tipos de píxel, entre los que incluye todos los tipos de datos básicos de C++ (char, int, float, unsigned char, etc.) así como algunos más complejos que se usan para la representación de píxeles RGB o píxeles vectoriales. En el caso de **BrainApp** sólo se usan los tipos de píxeles escalares pues en las IRM en cada píxel se muestra el valor de la intensidad de la señal que representan.

La clase que se implementó para modelar las imágenes en **BrainApp** se denomina **Image**, ella extiende la funcionalidad de **itk.Image** para lograr una interacción eficiente y armónica con el resto de los componentes.

#### 2.4.2. Tuberías

Las *tuberías* constituyen el núcleo de **BrainApp** pues son las encargadas de realizar el procesamiento de las imágenes y de obtener la segmentación del tumor. Son objetos que envuelven a un filtro de ITK, el cual realiza el procesamiento *real* de los datos. Todas las *tuberías* tienen como clase base a

**Pipe**, en ella se define el comportamiento general de las mismas, que se podría resumir de la siguiente forma:

- Las *tuberías* son objetos que tienen al menos una entrada y al menos una salida por donde reciben y devuelven un objeto de tipo **Image**.
- Todas las *tuberías* tienen una referencia a un filtro de ITK, encargado de realizar el procesamiento.
- Las *tuberías* realizan un procesamiento con los datos que le entran y devuelven los datos que resultan de este procesamiento.
- Las *tuberías* soportan cualquier tipo válido de imagen de ITK como entrada/salida para el filtro de ITK que contienen.
- Las *tuberías* son capaces de conectarse con otras *tuberías* de manera que si la *tubería* A se conecta con la B entonces la salida de A será la entrada de B.
- Las *tuberías* tienen un sistema de actualización inteligente que ante una petición de actualización decide si se debe volver a realizar el procesamiento o si los datos de la salida ya se encuentran correctamente calculados, esto permite optimizar el procesamiento general cuando se actualiza una *tubería* que está interconectada con otras.

**BrainApp** consta de seis *tuberías* que envuelven filtros de ITK, estas se encargan de convertir las imágenes, suavizar las imágenes, segmentar las imágenes y refinar las segmentaciones. La Tabla 2 muestra las *tuberías* de la aplicación y los respectivos filtros de ITK que estas envuelven.

Tabla 2. *Tuberías* de BrainApp y los filtros de ITK que estas envuelven.

Clase de BrainApp	Filtro de ITK
model.CastPipe	itk.CastImageFilter
model.SmoothPipe	itk.CurvatureAnisotropicDiffusionImageFilter
model.SegmentPipe	itk.ConnectedThresholdImageFilter
model.ConfidenceConnectedPipe	itk.ConfidenceConnectedImageFilter
model.VotingPipe	itk.VotingBinaryImageFilter



A continuación se describirá el funcionamiento de cada *tubería* y sus parámetros de configuración, comenzando por la clase base.

**Pipe:** es la clase base para todas las *tuberías*, en ella se definen los métodos de entrada y salida así como el mecanismo de actualización. Además en esta clase queda bien definido que para instanciar una *tubería* se debe especificar una dimensión y el tipo de píxel tanto de la entrada como de la salida (que no tienen que coincidir). Esta clase es abstracta y contiene una referencia al filtro de ITK que realiza el procesamiento.

**CastPipe:** es la *tubería* encargada de convertir imágenes, internamente usa el filtro **CastImageFilter** de ITK. Este filtro se encarga de cambiar el tipo de píxel que sirve para representar los datos de la imagen (Ibáñez et al., 2005). Es decir, si se desea cambiar una imagen que almacena sus píxeles con tipo *unsigned char* por otra que lo haga con tipo *float* solamente hay que instanciar la clase **CastPipe** con tipo de píxel *unsigned char* en la entrada y *float* en la salida. Esta *tubería* no tiene ningún parámetro de configuración.

**SmoothPipe:** se encarga de suavizar la imagen usando el filtro de ITK **CurvatureAnisotropicDiffusionImageFilter**. Este filtro realiza una difusión anisotrópica en la imagen que preserva los límites entre las regiones (transiciones claro-oscuro), con una buena sensibilidad al contraste (Ibáñez et al., 2005, Whitaker and Xue, 2001). Es un filtro iterativo que necesita para su correcto funcionamiento el ajuste de varios parámetros que definen la *ecuación modificada de difusión de curvatura* basada en la definición de la ecuación de difusión anisotrópica de Perona-Malik (Perona and Malik, 1990). Los siguientes parámetros se deben ajustar para obtener el resultado adecuado (Ibáñez et al., 2005).

- Paso de tiempo: esta constante influye en el mecanismo de resolución de la ecuación por el método de diferencias finitas, se recomiendan valores menores a 0.0625 para imágenes 3D y menores de 0.125 para 2D.

- **Parámetro de conductancia:** controla la sensibilidad de la difusión de los bordes de las regiones, mientras más pequeño el valor, mejor se conservan las características de la imagen. Los valores típicos varían entre 0.5 y 3.0 aunque el valor correcto depende de la imagen y las necesidades del usuario así como del número de iteraciones que se realicen.
- **Número de iteraciones:** afecta el rendimiento general del filtro, mientras más iteraciones más lento es y más exactitud se alcanza en el resultado final.

**SegmentPipe:** realiza la segmentación de la imagen siguiendo un método de crecimiento de regiones (“region-growing”) basado en rangos de intensidades. El filtro de ITK que usa se denomina **ConnectedThresholdImageFilter** el cual actúa de la siguiente forma: en cada iteración el filtro analiza los píxeles vecinos de la región actual (los que comparten al menos un vértice con algún píxel ya incluido en la misma) y, si sus valores están dentro del rango de intensidades válidas, estos píxeles pasan a formar parte de la región para la nueva iteración. Cuando haya una iteración que no agregue píxeles a la región se da por concluido el procesamiento. El rango de intensidades válidas se define por un umbral máximo y mínimo. La región inicial del procesamiento viene dada por un píxel de la imagen que sirve como semilla. El resultado del procesamiento es una región que se muestra en forma de máscara donde los píxeles que forman parte de la segmentación tienen el valor de relleno especificado y los demás píxeles tienen valor cero (Ibáñez et al., 2005).

Para el funcionamiento de esta *tubería* se deben configurar los siguientes parámetros:

- El umbral mínimo.
- El umbral máximo.
- El píxel que sirve como semilla.
- El valor de relleno para la máscara.

**ConfidenceConnectedPipe:** esta *tubería* también realiza una segmentación con un método de crecimiento de regiones. Usa el filtro **ConfidenceConnectedImageFilter** de ITK que trabaja de la siguiente forma: para cada iteración calcula la varianza y la media de los valores de los píxeles ya incluidos en la región, luego todos los píxeles vecinos a la región que tengan una intensidad comprendida en el intervalo (definido por la media más/menos el valor de la varianza por un multiplicador) se incluyen en la región para la próxima iteración. El proceso finaliza luego de un número predefinido de iteraciones. Para comenzar, este filtro necesita un píxel inicial que sirva como semilla y un radio que se usa para saber cuántos vecinos de la semilla se deben utilizar al calcular la varianza y media de la primera iteración. El resultado se obtiene en forma de máscara donde los píxeles pertenecientes a la segmentación se muestran con el valor de relleno especificado y los demás tendrán valor cero (Ibáñez et al., 2005).

Los siguientes parámetros son los que definen el comportamiento de esta tubería:

- Semilla: es el píxel inicial que sirve como semilla para la región de la segmentación.
- Radio inicial: define cuántos píxeles vecinos a la semilla se deben incluir para calcular la media y la varianza al inicio de la primera iteración. Es importante señalar que un radio con valor  $r$  define en 3D una vecindad (con las longitudes medidas en píxeles) en forma de cubo de lado  $2r+1$  y centro en el píxel semilla, y en el caso 2D define un cuadrado de lado  $2r+1$  con características análogas.
- Multiplicador: se usa para controlar la longitud de la vecindad de la media que sirve como intervalo de admisión para los nuevos píxeles. El intervalo se define como  $media \pm varianza * multiplicador$ .
- Valor de relleno: es el valor de los píxeles que finalmente formen parte de la segmentación.

- Número de iteraciones: define la cantidad de iteraciones del filtro, este parámetro tiene una incidencia directa en el crecimiento que pueda alcanzar la región.

**VotingPipe:** el objetivo de esta *tubería* es refinar una segmentación, para ello usa el filtro **VotingBinaryImageFilter** de ITK. Este filtro se utiliza para modificar imágenes binarias, es decir aquellas imágenes en las cuales los píxeles sólo poseen uno de dos posibles valores: *fondo* y *primer plano*. En el caso de **BrainApp** el *primer plano* es el valor que define la región de la segmentación previamente obtenida y el *fondo* es precisamente el fondo de la máscara que representa dicha segmentación (Ibáñez et al., 2005).

El filtro **VotingBinaryImageFilter** realiza una *votación* para cada píxel de la imagen de entrada con vistas a decidir si este pasa a formar parte del *primer plano* o si se considerará un píxel de *fondo*. Para realizar la *votación* el filtro opera de la siguiente forma:

1. Para cada píxel de la imagen se cuentan cuántos vecinos de este forman parte del *primer plano*.
2. Luego se verifica si el píxel que se está analizando es de *fondo* o de *primer plano*.
  - 2.1. Para un píxel de *primer plano*, este pasará a formar parte de los píxeles de *fondo*, si la cantidad de sus vecinos que forman parte del *primer plano* es menor que el *umbral de supervivencia*.
  - 2.2. Para un píxel de *fondo*, este pasará a formar parte de los píxeles de *primer plano*, si la cantidad de sus vecinos que forman parte del *primer plano* es mayor que el *umbral de nacimiento*.

Cuando se realiza la *votación* se considera *vecino* a todo píxel que se encuentre dentro de un ortoedro definido por un radio N-dimensional ( $N=2,3$ ), centrado en el píxel que se está analizando y cuya longitud se mide en píxeles. Las componentes del radio definen las longitudes de los lados del ortoedro, por

ejemplo un radio  $r=(r_1, r_2, r_3)$ , en una imagen 3D, define un ortoedro de lados  $(2r_1+1) \times (2r_2+1) \times (2r_3+1)$ . Luego de realizada la *votación* se le ajusta el valor al píxel en dependencia del resultado de la misma (Ibáñez et al., 2005).

Para el correcto funcionamiento de esta *tubería* se deben ajustar los siguientes parámetros:

- Radio: define cuáles píxeles se consideran vecinos del píxel en cuestión (como ya se explicó anteriormente).
- Valor de fondo: especifica el valor que se considerará como *fondo* en la imagen binaria que sirve de entrada a la tubería.
- Valor de primer plano: define el valor que se considerará como *primer plano*.
- Umbral de nacimiento: es la cantidad que se considerará a la hora de decidir si un píxel de *fondo* pasa a ser de *primer plano*.
- Umbral de supervivencia: es la cantidad que se considerará a la hora de decidir si un píxel de *primer plano* pasa a ser de *fondo*.

Por último, es necesario destacar que combinando de manera adecuada el radio y los umbrales se pueden lograr múltiples efectos en la segmentación, algunos de los cuales son muy utilizados como es el caso de la *erosión* y la *dilatación* (Ibáñez et al., 2005).

### 2.4.3. Vistas

Las *vistas* en **BrainApp** sirven de intermediarias entre las imágenes que están siendo usadas por la aplicación y las visualizaciones que el usuario puede percibir de las mismas. **BrainApp** se diseñó e implementó de manera que permita múltiples vistas, actualmente consta con una sola, definida en la clase **CanonicalSliceView**.

Las *vistas* están diseñadas para que hereden de la clase **View** que define el comportamiento general y para que se puedan agregar tantas como se necesite, algo muy importante para el desarrollo futuro de **BrainApp**.

Como VTK permite visualizar múltiples tipos de objetos (imágenes, objetos 3D, funciones o superficies) está diseñado con un modelo de visualización muy amplio, que pueda visualizar todos estos tipos de objetos de manera adecuada y lograr que se integren correctamente. En el modelo de VTK se usan varios conceptos que sirven como abstracción en el proceso de visualización, de estos los más relevantes para la implementación de **BrainApp** son (Kitware Inc., 2006, Schroeder et al., 2002):

- **Objetos de datos:** son los objetos que sirven como orígenes de los datos que se desean visualizar, estos pueden ser desde imágenes, funciones, hasta cuerpos.
- **Objetos de procesamiento:** son los objetos encargados de procesar una imagen para lograr un efecto en su visualización. Se denotarán como *filtros* de VTK.
- **Actores:** un actor de VTK es un objeto que contiene los datos que se deben visualizar, además posee una posición, orientación, etc. que definen la forma en que estos se visualizan. En **BrainApp** la mayoría de los actores son instancias de **vtk.vtkImageActor**.
- **Renderizadores:** son los encargados de calcular la forma en que se van a visualizar los actores, teniendo en cuenta la posición, luces y la presencia de otros componentes. Todas las clases de VTK que representan renderizadores son subclases de **vtk.vtkRenderer**.
- **Ventanas de renderización:** son las que controlan las porciones de la pantalla donde se visualizan realmente los datos, estas pueden funcionar con un *mediador* que se encarga de manejar la interacción entre el usuario y la ventana. Las clases que representan a estas ventanas son subclases de **vtk.vtkRenderWindow** y los objetos encargados de mediar en la interacción son instancias de **vtk.vtkRenderWindowInteractor**.

El funcionamiento general de las *vistas* en **BrainApp** se implementó de la siguiente forma: las *vistas* reciben como entrada una imagen (**Image**), realizan

una manipulación de los datos que estas contienen para lograr un efecto en su visualización y luego devuelven un actor de VTK que representa los datos listos para ser visualizados. En este contexto se puede describir cada una de las clases que implementan las *vistas* de la aplicación.

**View:** es la clase base para todas las *vistas*. Define el comportamiento general. En ella se especifica que cada *vista* debe tener una referencia a uno o varios filtros de VTK que son los que realizan el procesamiento de los datos, además aporta el mecanismo inicial para convertir los datos de **itk.Image** a un formato que VTK pueda usar. **View** también predefine que todas las *vistas* brindan como salida un *actor* de VTK.

**CanonicalSliceView:** permite obtener una imagen 2D realizando un corte en una imagen 3D. Esta es la *vista* más importante de **BrainApp** y su verdadera potencia radica en las múltiples formas en que se puede realizar el corte, en el sentido de cualquiera de los ejes canónicos y con cualquier orientación. En total permite 24 combinaciones para realizar un corte. Estos cortes se definen teniendo en cuenta las direcciones y el sentido de los ejes que quedan como ejes horizontal y vertical respectivamente en la imagen final, por ejemplo un corte denotado por *YNegX* significa que la imagen es cortada por un plano paralelo al plano XY pero donde el eje horizontal de la imagen obtenida tiene la dirección del eje Y con sentido positivo mientras que el eje vertical tiene la dirección del eje X con sentido negativo. La funcionalidad de esta *vista* se implementó usando el filtro **vtk.vtkImageReslice**.

Esta *vista* se usa para mostrar tres cortes perpendiculares dos a dos que muestran una imagen cortada por los planos canónicos y constituye el componente fundamental de la *visualización* principal de **BrainApp**.

#### 2.4.4. Visualizaciones

Las *visualizaciones* son las encargadas de mostrar la información, previamente procesada por una *vista*, en la interfaz gráfica. Las características principales de su diseño e implementación son las siguientes:

- Contienen las vistas que procesan los datos antes de la visualización.
- Controlan los componentes visuales que permiten mostrar los datos en pantalla: las *ventanas de renderización* y los *renderizadores* (ver el epígrafe anterior), así como los componentes de la interfaz gráfica de usuario necesarios para la visualización.
- Proveen la interacción entre el usuario y la visualización de los datos usando los *interactores* de VTK.
- Pueden ser compuestas, es decir, una *visualización* puede constar con otras a las cuales controla.

La implementación de **BrainApp** cuenta con tres clases que representan las visualizaciones, de las cuales una define el comportamiento general.

**Visualization:** es la clase base para todas las *visualizaciones*. Define que todas tienen al menos una *vista*, un *renderizador*, y un *interactor*. Además controlan una porción de la interfaz que contiene una *ventana de visualización* y los componentes visuales necesarios para la interacción.

**SliceVisualization:** esta *visualización* se encarga de mostrar un corte de una imagen 3D usando la vista **CannonicalSliceView**. Brinda los objetos de interacción necesarios para controlar la *vista*. Permite cambiar los planos de cortes en la dirección de cualquiera de los planos canónicos, cambiar el punto donde se realiza el corte y usar cualquier sentido en los ejes de la imagen 2D mostrada.

**ThreePlanesVisualization:** muestra tres instancias de **SliceVisualization** que representan cortes de la imagen 3D que se recibe como entrada. Los cortes están sincronizados para que al hacer clic en un punto de la imagen en alguno



de ellos los demás muestren también ese punto, independientemente del plano de corte que estos estén usando. Además, esta *visualización* muestra las coordenadas donde se hizo clic y el valor de la intensidad en el punto seleccionado. Por defecto **ThreePlanesVisualization** muestra tres cortes perpendiculares dos a dos aunque este comportamiento no es fijo y el usuario lo puede configurar según sus necesidades. Esta *visualización* es el componente visual principal de **BrainApp** porque brinda la información necesaria para configurar los parámetros de los filtros en el proceso de segmentación.

## 2.5. Procesamiento de las imágenes en BrainApp

Ya se ha especificado que las *tuberías* son las encargadas del procesamiento en **BrainApp** pero ¿cómo se logra obtener una segmentación del tumor?, ¿de qué forma se deben conectar las *tuberías* para realizar el procesamiento deseado?

Para responder a estas interrogantes se debe explicar mejor la implementación del procesamiento con las *tuberías* en **BrainApp**.

El nombre *tubería* surge porque los objetos encargados de procesar las imágenes deben comportarse de una forma peculiar, que asemeja el funcionamiento de una tubería; es decir, deben ser capaces de interconectarse de manera que cuando dos de ellos se conecten la salida del primero sirva de entrada al segundo. Estos objetos interconectados entre sí, uno detrás del otro, funcionan como una tubería por donde entra la imagen original y va pasando desde un objeto a otro a medida que cada uno de ellos la modifica según el objetivo para el que fueron programados. Así al terminar el procesamiento (a la salida de la tubería) se tiene una imagen que muestra el resultado deseado que puede ser una segmentación u otro efecto.

**BrainApp** se implementó de manera que el procesamiento de la imagen siempre empieza con la *tubería* **CastPipe** para cambiar el tipo de píxel de la

imagen hacia tipo *float* (ya que con este tipo de dato funcionan las demás *tuberías*), igualmente al final se conecta otro **CastPipe** para devolver una imagen con tipo de píxel *unsigned char*. Entre estas dos *tuberías* se encuentra la *tubería* de suavizado, las dos *tuberías* de segmentación y la de refinamiento, en el mismo orden de la Tabla 2.

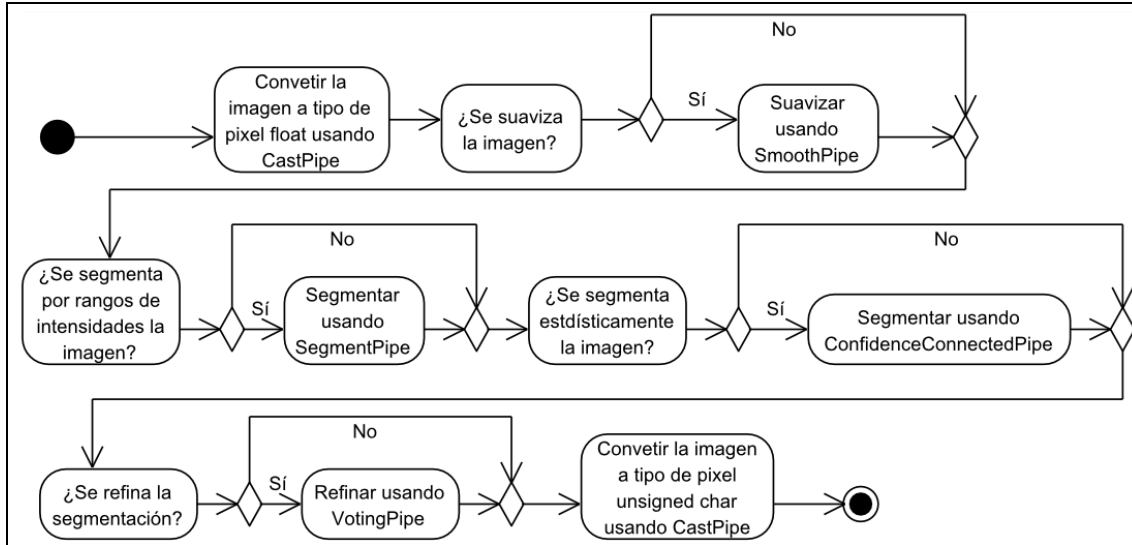


Figura 16. Diagrama de actividad para el procesamiento de la imagen.

En la implementación se tuvo en cuenta brindarle al usuario la posibilidad de configurar el procesamiento especificando cuáles *tuberías* desea usar entre la *tubería* inicial y la final, ajustando los parámetros de cada una de ellas. En dependencia de las *tuberías* que el usuario configure, se instancian los objetos necesarios y se realizan las conexiones oportunas para así formar una *tubería* que conforme el procesamiento general. Luego, cuando el usuario decida, puede ejecutar el procesamiento y observar los resultados obtenidos.

El diseño de **BrainApp**, con el método de división del proceso en unidades más pequeñas, permite ir conectando los filtros que se deseen al ir agregando poco a poco *tuberías* al procesamiento (Figura 16, elaborada por el autor). Por ejemplo en un caso típico el usuario comienza cargando una imagen que se debe suavizar para mejorar su visualización, aquí se usa el filtro de suavizado (a través de una instancia interna de **SmoothPipe**), es probable que tenga que

ejecutarlo varias veces variando la configuración de los parámetros hasta que obtenga el resultado deseado. Después agrega una de las tuberías capaces de realizar la segmentación (**SegmentPipe** o **ConfidenceConnectedPipe**) las cuales debe configurar y probablemente ejecutar varias veces. Al final, es posible que necesite refinar la segmentación usando **VotingPipe**. Todo esto lo puede realizar el usuario de **BrainApp** porque internamente en cada uno de estos pasos se implementó la funcionalidad que permite que la aplicación se encargue de crear y manejar los objetos necesarios para el procesamiento y la interconexión de los mismos con la configuración definida por el usuario.

No obstante, donde realmente se evidencia la potencia del método planteado es en el nivel de adaptabilidad del procesamiento a las necesidades del proceso. Es decir, cuando el usuario decide que el resultado del filtro de suavizado es el correcto, al agregar la *tubería* encargada de la segmentación, esta se conecta con la de suavizado y como ya estaba calculado el resultado del filtro anterior el procesamiento simplemente continúa desde donde se quedó (sin necesidad de recalcular los datos). Esto mismo ocurre luego si conecta la *tubería* de refinamiento. También es posible configurar todas las *tuberías* y ejecutar el procesamiento completamente o se puede *desconectar* una tubería (luego de haber calculado un primer resultado) porque no es necesario ese paso para lograr un resultado óptimo. Todo esto se puede hacer en **BrainApp** de una manera eficiente gracias a la forma en que se implementó el procesamiento con las *tuberías* que permite conservar los datos anteriores al punto del procesamiento que se cambió (*desconexión* de una *tubería*) y sólo recalcula los datos que dependen del punto cambiado. Este mecanismo brinda un alto grado de rendimiento que el usuario percibe en la rapidez con que ejecuta la aplicación.

Para garantizar la completa reproducibilidad del procesamiento, se implementó un mecanismo para guardar el estado de la configuración de los filtros y para cargar uno previamente guardado. Así, si se tiene el fichero que contiene la

configuración de las *tuberías*, cualquier usuario puede cargar los parámetros y, usando la imagen original, ejecutar el procesamiento, de esta forma se garantiza que el resultado será siempre el mismo. Esto añade reproducibilidad al trabajo con **BrainApp**.

## 2.6. Manejo de los datos en BrainApp

La forma en que **BrainApp** maneja los datos y configuraciones se diseñó pensando que pueda ser extendido o refactorizado. Se implementó un diseño que permite extender la aplicación en futuras versiones para que trabaje con múltiples documentos, es decir en un ambiente MDI (siglas en inglés para Interfaz de Múltiples Documentos), donde el usuario pueda tener cargado diferentes estudios que no estén relacionados entre sí (*documentos*), cada uno con varias imágenes y sus propias configuraciones de las *tuberías*, sin la necesidad de abrir dos instancias de **BrainApp**. En la versión actual solo se soporta un *documento* por aplicación.

La manera en que se concibieron e implementaron los distintos niveles de agrupamiento de datos en **BrainApp** fue la siguiente:

- La clase **Application**: agrupa los mecanismos de acceso a todos los componentes necesarios para el correcto funcionamiento de la aplicación. Sólo contiene métodos de clase (un tipo de método estático de Python) y está implementada para que sólo exista una instancia de la misma (patrón Singleton Gamma et al., 2003). Con ella se pueden acceder a todos los *documentos* de la aplicación, así como a las imágenes cargadas y a las tuberías con sus respectivas configuraciones. Es la única clase que TODOS los componentes pueden usar para acceder a los datos, incluyendo al propio *documento*. Brinda un método que devuelve el documento actual que está usando la aplicación.
- La clase **Document**: encargada de modelar el *documento* que aísla los datos del estudio que el usuario está realizando de otras posibles instancias. Contiene una lista con todas las *imágenes* cargadas por la

aplicación (para un estudio específico) y mantiene las instancias de las *tuberías* que se usan en el procesamiento con sus respectivas configuraciones. Aquí se brindan métodos de acceso y manipulación a las *imágenes* y *tuberías*. Esta es la clase que realmente contiene todos los datos necesarios para un procesamiento específico de las *imágenes*.

## 2.7. Cálculo del volumen y dimensiones del tumor segmentado

En **BrainApp** se implementaron algoritmos que permiten calcular el volumen de la segmentación realizada y además obtener las dimensiones principales de la misma en cada uno de los ejes (ver Anexo 2).

El algoritmo para calcular el volumen funciona de la siguiente forma:

1. Inicializar el **contador** de los píxeles de la segmentación en cero.
2. Para cada uno de los píxeles de la imagen:
  - 2.1. Si el píxel es de *primer plano* incrementar el **contador** en 1
  - 2.2. Si el píxel es de *fondo* continuar.
3. Al finalizar, multiplicar el valor del **contador** por los tres valores de espaciado en los vóxeles de la imagen y retornar el resultado obtenido como valor para el volumen de la segmentación.

Para obtener las dimensiones principales en cada uno de los ejes se elaboró e implementó el siguiente algoritmo:

1. Escoger uno de los ejes X, Y o Z, al cual se le calcularán los datos.
2. Inicializar la **longitud máxima** del eje seleccionado en cero.
3. Inicializar los índices de **inicio** y **fin** del segmento de longitud máxima en cero.
4. Seleccionar un eje diferente (del ya seleccionado) para realizar los cortes de la imagen de forma que sean perpendiculares a este nuevo eje.
5. Para cada uno de estos cortes recorrer las rectas definidas por la dirección del eje seleccionado en el paso 1, y para cada una de estas rectas:

- 5.1. Inicializar el **contador** de píxeles consecutivos en cero.
- 5.2. Inicializar el índice de **inicio del contador** en cero.
- 5.3. Para cada píxel de la recta actual:
  - 5.3.1. Si el píxel es de *primer plano*:
    - 5.3.1.1. En caso de que el **contador** sea cero hacer **inicio del contador** igual al índice del píxel actual.
    - 5.3.1.2. Incrementar el **contador**.
  - 5.3.2. Si el píxel es de *fondo*:
    - 5.3.2.1. En caso de que el **contador** tenga un valor mayor que el de la **longitud máxima**:
      - 5.3.2.1.1. Actualizar la **longitud máxima** al valor del **contador**.
      - 5.3.2.1.2. Actualizar el **inicio** de la longitud máxima con el índice almacenado en **inicio del contador**.
      - 5.3.2.1.3. Actualizar el **fin** de la longitud máxima con el índice del píxel actual.
    - 5.3.2.2. Hacer cero el **contador**.
6. Multiplicar el valor de la **longitud máxima** calculada por el valor del espaciado de los vóxeles de la imagen en la dirección del eje seleccionado en el paso 1.
7. Retornar el resultado anterior como longitud máxima de la segmentación en la dirección del eje seleccionado y retornar los índices de inicio y fin del segmento que presenta la longitud máxima.

Al terminar de ejecutar este algoritmo se tendrá la longitud máxima de la segmentación en la dirección del eje seleccionado inicialmente. También se obtiene el índice de inicio y fin del segmento que representa esta longitud.

## 2.8. Interfaz gráfica de BrainApp

Para lograr que la herramienta propuesta fuera útil y cómoda de usar se diseñó una interfaz gráfica que tuviera en cuenta los criterios señalados en el epígrafe 1.6.2. De forma abstracta se propone una interfaz como la mostrada en la

Figura 17 para facilitarle al usuario la configuración y uso de las *tuberías* que procesan la imagen en función de obtener la segmentación del tumor.

En el diseño mostrado se observan varios recuadros denotados por letras, sus funciones se explican a continuación.

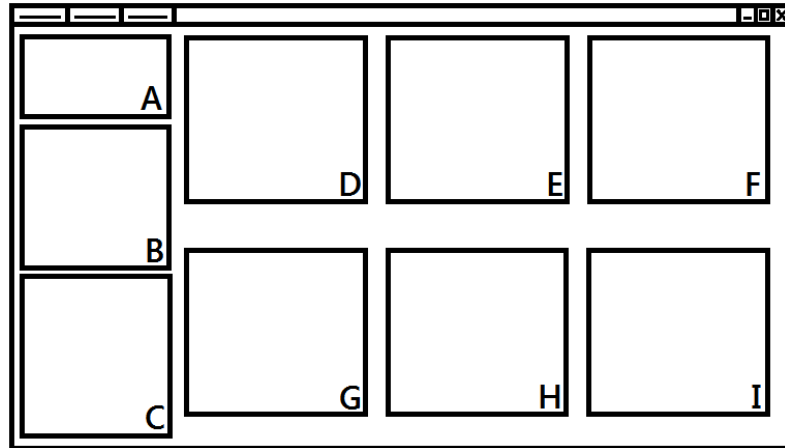


Figura 17. Diseño de la interfaz gráfica de BrainApp.

- **A:** tiene como propósito mostrar una lista que especifique todas las imágenes que se han cargado en la aplicación.
- **B:** muestra las propiedades de la imagen que se decidió procesar, además permite editar algunas de ellas.
- **C:** muestra los valores de los parámetros para los filtros del procesamiento y permite la configuración de los mismos.
- **D, E, F:** muestran los tres cortes de la imagen original que se está procesando (sin cambios). Estos recuadros funcionan en conjunto como parte de la visualización principal (que es instancia de **ThreePlanesVisualization**) para la imagen seleccionada en **A**.
- **G, H, I:** estos recuadros son análogos a los tres anteriores, sólo que muestran los cortes de la imagen que resulta del procesamiento realizado sobre la imagen seleccionada. Aquí se visualizará la segmentación final u otro resultado que se obtenga dada la configuración de los filtros especificada en **C**.

Con este diseño se pretende lograr que el usuario tenga el control total sobre el procesamiento y cómo se realiza, permitiendo la visualización de cada etapa en conjunto con la imagen original. Así el usuario puede realizar una comparación y corregir los errores encontrados.

La interfaz gráfica también posee diálogos para calcular/mostrar el volumen y las dimensiones principales del tumor, así como para guardar/cargar la configuración de las *tuberías* y las *imágenes*.

En la implementación de la interfaz se concibieron las siguientes clases controladoras (paquete **controller**):

- **ImageLoader, ImageSaver:** se encargan de la carga/salva de imágenes 3D.
- **SeriesLoader:** se encarga de la carga de series de imágenes 2D.
- **ConfigurationManager:** carga/salva la configuración de las *tuberías*.
- **InterfaceRunner:** inicializa la configuración de la aplicación de Qt que maneja la interfaz gráfica.
- **MainWindow:** es la clase encargada de la interconexión de todos los componentes y del control sobre el procesamiento.
- **SegmentationData:** muestra los datos de la segmentación y brinda al usuario la posibilidad de ejecutar los algoritmos de cálculo del volumen y dimensiones, así como guardar los resultados en un fichero.
- **FiltersUiSync:** permite al usuario configurar las *tuberías* y controla los componentes que visualizan la configuración.
- **ImageUiSync:** maneja los componentes encargados de mostrar las propiedades de las imágenes, permite al usuario editar algunas.
- **WaitProcess:** se encarga de mostrar un diálogo que informa al usuario el avance del procesamiento que está realizando la aplicación.



Para la implementación de la apariencia visual y el manejo de sus componentes (botones, menús, etiquetas, etc.) se implementaron las clases del paquete **view** que se detallan a continuación

- **ImageLoaderDialog:** representa el diálogo encargado de cargar imágenes 3D.
- **ImageSaverDialog:** representa el diálogo para guardar imágenes.
- **SeriesLoaderDialog:** representa el diálogo de carga de series de imágenes 2D.
- **WaitDialog:** representa el diálogo de espera que muestra información del avance del procesamiento.
- **MainWindowWidget:** representa la ventana principal con todos sus componentes (incluyendo las ventanas flotantes).
- **SliceVisualizationWidget:** representa la parte visual de la clase **CanonicalSliceVisualization**.
- **ThreePlanesWidget:** representa la parte visual de **ThreePlanesVisualization**.

## 2.9. Validación de resultados de la aplicación

Para comprobar la calidad de los resultados obtenidos con la herramienta que se propone en la presente investigación, se realizaron varias pruebas divididas en dos grupos. En el primer grupo de pruebas se utilizaron nueve imágenes IRM de tumores a las cuales se les realizó una segmentación semiautomática con la nueva aplicación, los resultados se compararon con segmentaciones manuales aplicando las métricas definidas en 1.6.5 y utilizando el software VALMET (Gerig et al., 2001) para el cálculo de las mismas.

Los dos métodos de segmentación semiautomática realizados, cuyos resultados se comparan en la Tabla 3, son:

1. Segmentación semiautomática sobre la imagen original.
2. Segmentación semiautomática sobre la imagen suavizada.

En ambos casos se utilizó el filtro segmentador (no estadístico) para el proceso de segmentación. Para suavizar la imagen se utilizó el filtro de suavizado que brinda **BrainApp**. Todo el proceso de los dos métodos se realizó completamente con **BrainApp**.

En la Tabla 3 se muestran los valores que resultan al aplicar las métricas de comparación entre los resultados obtenidos con los dos métodos semiautomáticos y las segmentaciones manuales. Se observa que no existe una supremacía de uno de los métodos según el “Coeficiente de Similitud de Dice”, con ambos se alcanzan resultados catalogados como *buenos* según la literatura consultada (ver epígrafe 1.6.5) ya que sólo en dos casos están por debajo de 0.8 y como promedio general se obtiene el valor 0.81805.

Tabla 3. Comparación de similitud entre las segmentaciones manuales y las obtenidas usando BrainApp.

	Segmentación semiautomática	DSC	Hausdorff inside	Hausdorff outside
Caso1	Normal	0.855233	0.724925	19.244682
	Suavizada	0.813498	1.027511	19.553579
Caso2	Normal	0.914506	0.850309	4.104105
	Suavizada	0.91335	0.833362	4.117365
Caso3	Normal	0.871574	1.093121	16.951918
	Suavizada	0.8809	0.714552	16.971914
Caso4	Normal	0.728943	0.999906	0.9375
	Suavizada	0.722094	1.0141	0.92892
Caso5	Normal	0.874289	0.450451	0.9375
	Suavizada	0.874289	0.450451	0.9375
Caso6	Normal	0.677119	1.071931	2.093842
	Suavizada	0.678239	1.011307	2.125935
Caso7	Normal	0.809821	0.298491	6.029385
	Suavizada	0.810981	0.285768	6.193849
Caso8	Normal	0.82231	1.343587	2.43984
	Suavizada	0.812387	1.989521	2.503953
Caso9	Normal	0.83923	0.283802	10.29485
	Suavizada	0.826099	0.214574	10.293857
Promedios	Normal	0.82145	0.79072	7.00374
	Suavizada	0.81465	0.83791	7.06965

La distancia de Hausdorff también demuestra que hay igualdad y en el caso particular de Hausdorff-inside se obtienen muy buenos resultados con valores como promedio menores que 1 mm, lo cual según la literatura es un resultado bueno (Prastawa, 2007).

La paridad en los resultados viene dada principalmente porque las características del procesamiento de las imágenes de tumores es muy variable en dependencia de la imagen, es decir no se puede afirmar *a priori* que siempre es mejor suavizar la imagen antes de segmentarla, aunque hay casos donde este paso resulta imprescindible para el correcto crecimiento de la región (Figura 18).

En el segundo grupo de pruebas se compararon las medidas planas (ancho y alto) de cinco tumores.

Los resultados mostrados en la Tabla 4 (columna: Medidas semiautomáticas) se obtuvieron a partir de segmentaciones realizadas con **BrainApp**. Los datos se calcularon usando el algoritmo implementado en la aplicación que posibilita obtener las medidas principales del tumor.

Las medidas manuales fueron obtenidas con Imagis, un software que posibilita realizar mediciones de longitudes de segmentos dentro de una imagen. Para obtener el resultado, el especialista traza dos segmentos que representen los diámetros principales del tumor y el software calcula la longitud de acuerdo al espaciado de los píxeles en la imagen.

En la Tabla 4 se observan diferencias entre las medidas manuales y semiautomáticas, una de las razones de que esto ocurra es el hecho de que las segmentaciones realizadas con la aplicación no son muy precisas en los bordes del tumor, debido a las dificultades inherentes al proceso como los bordes difusos (epígrafe 1.5.3), esto provoca los milímetros de diferencia que se observan en la comparación de los resultados.

Tabla 4. Comparación de las medidas planas.

	Dimensión	Medidas manuales	Medidas semiautomáticas
Caso1	Ancho	26.07 mm	29.65 mm
	Alto	26.45 mm	26.05 mm
Caso2	Ancho	25.09 mm	26.37 mm
	Alto	21.45 mm	21.97 mm
Caso3	Ancho	25.49 mm	23.93 mm
	Alto	14.57 mm	16.07 mm
Caso4	Ancho	30.02 mm	31.92 mm
	Alto	16.11 mm	15.21 mm
Caso5	Ancho	35.12 mm	36.25 mm
	Alto	24.58 mm	22.98 mm

A pesar de que la diferencia en milímetros parece significativa se debe puntualizar que el espaciado promedio de los píxeles en los cortes principales de las imágenes de prueba era de 1.0 mm, es decir que por cada milímetro de diferencia en las mediciones, existe una diferencia real entre las segmentaciones de solo un píxel.

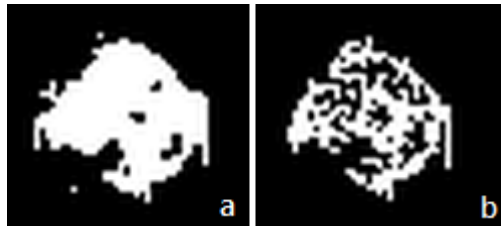


Figura 18. Efecto del uso del filtro suavizador: (a) segmentación luego de suavizar, (b) segmentación sin suavizar.

Como consecuencia del uso de la aplicación al realizar las pruebas anteriormente descritas se constató que para los tumores cuya visualización se obtuvo sin la utilización de contrastante resulta importante el uso de la técnica de suavizado en el método de segmentación, pues de lo contrario se observan efectos como el de la Figura 18. Esta característica se debe principalmente a que el rango de intensidades de los píxeles pertenecientes al tumor en las imágenes sin contrastante es relativamente estrecho y con pequeñas variaciones en el interior del cuerpo del tumor.

En el Anexo 3 se incluyen imágenes que grafican dos segmentaciones semiautomáticas comparadas con sendas segmentaciones manuales.

### 2.10. Conclusiones parciales

En el diseño de **BrainApp** se tuvo en cuenta su posible desarrollo futuro, por lo que se implementó de manera que pueda ser extendida y mejorada, cambiando o agregando nuevos componentes.

El procesamiento en **BrainApp** se implementó siguiendo una estrategia que permite dividirlo en pequeñas unidades que se pueden interconectar o desconectar para ajustarlas al procesamiento deseado de manera que se puedan reutilizar resultados ya calculados y que solo se recalculen los resultados que dependen de datos que hayan sido modificados. Todo este mecanismo aumenta la eficiencia y el rendimiento general de **BrainApp**.

La interfaz gráfica de **BrainApp** se diseñó e implementó teniendo en cuenta los requerimientos del proceso de segmentación y de manera que brinde la mayor cantidad de facilidades posibles al usuario, para lograr una conjugación armónica entre el proceso computacional y la experiencia del especialista.

Las validaciones realizadas a las segmentaciones obtenidas con la aplicación, comparadas con las realizadas manualmente, arrojaron un grado de similitud catalogado como bueno, con valores promedios por encima de 0.8. Las validaciones de los cálculos de las dimensiones principales del tumor también mostraron resultados alentadores.

## CAPÍTULO III. MANUAL DE USUARIO DE BRAINAPP

En el presente capítulo se describirá cómo trabajar con **BrainApp**, la información que se expone es la siguiente:

- Requisitos e instalación de la aplicación.
- Descripción de la ventana principal, menús, ventanas flotantes y barra de herramientas de la aplicación.
- Descripción de las visualizaciones de las imágenes.
- Mecanismos de configuración del procesamiento y obtención de los datos asociados a las segmentaciones.

### 3.1. Introducción

**BrainApp** es una herramienta para obtener la segmentación de un tumor cerebral de manera semiautomática usando Imágenes de Resonancia Magnética. Está especialmente diseñada para servir de apoyo al especialista en la labor de segmentación y cuenta con una interfaz gráfica de cómodo uso con un conjunto de herramientas muy útiles. La versión actual del software es la 0.1.

### 3.2. Instalación y requisitos del sistema

**BrainApp** no necesita ser instalado, basta con copiar los archivos del código fuente de la carpeta BrainApp hacia donde se desee ubicar la aplicación. Esta herramienta puede ser utilizada desde cualquiera de los siguientes sistemas operativos: Windows9x/NT/XP/Vista/7, Linux, MacOS X; y sólo necesita que se halla instalado la versión 2.6 o posterior del intérprete de Python, junto con la distribución de PyQt versión 4.6 o posterior, la biblioteca para Python de VTK versión 5.6 y la de ITK versión 3.16.

### 3.3. Ventana principal

Al abrir **BrainApp** se muestra la ventana principal de la aplicación (Figura 19).

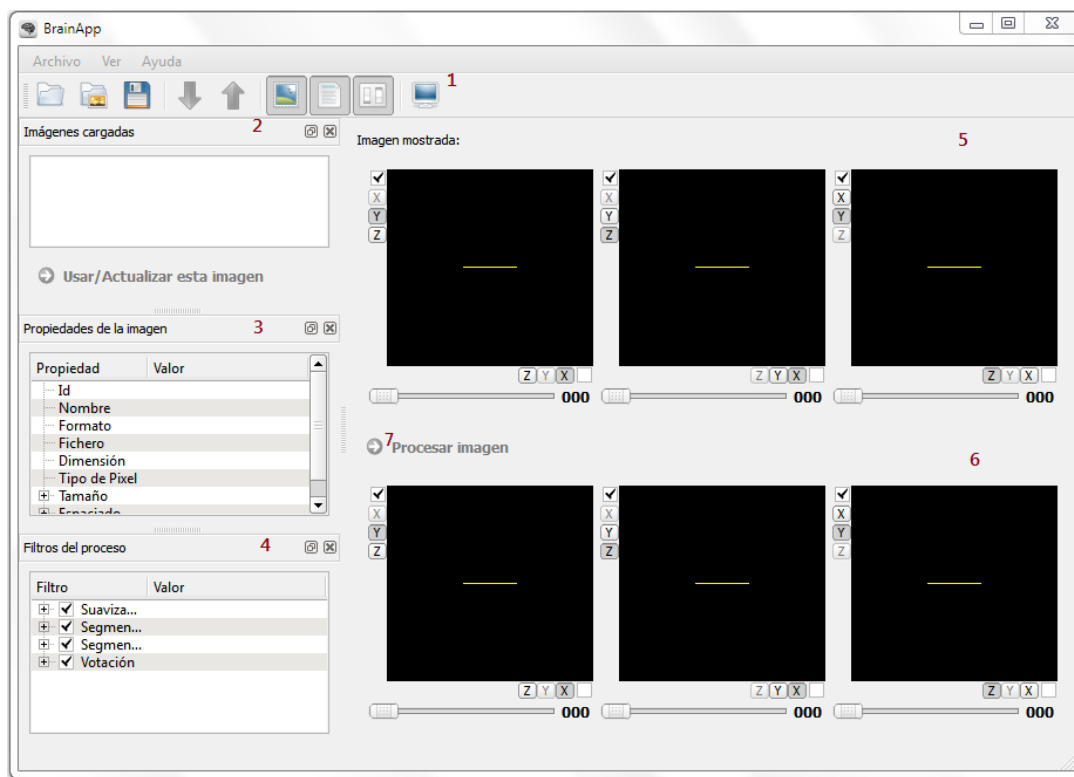


Figura 19. Ventana principal de BrainApp.

Componentes de la ventana principal de **BrainApp**:

1. Barra de herramientas.
2. Lista de imágenes cargadas por la aplicación.
3. Propiedades de la imagen seleccionada en la lista de imágenes.
4. Configuración de los filtros del procesamiento
5. Visualización principal de la imagen a procesar.
6. Visualización del resultado del procesamiento.
7. Botón que ejecuta el procesamiento (atajo de teclado: Ctrl+F5).

### 3.3.1. Menús de BrainApp

**BrainApp** tiene tres menús: Archivo, Ver y Ayuda (Figura 20).

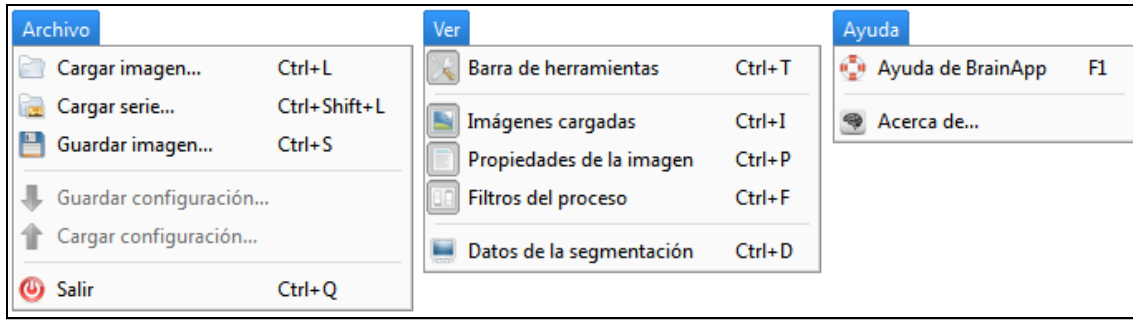


Figura 20. Menús de BrainApp.

Tabla 5. Descripción del menú Archivo

Acción	Descripción
Cargar imagen...	cargar imágenes volumétricas
Cargar serie...	cargar series de imágenes
Guardar imagen...	guardar la imagen resultado del procesamiento
Guardar configuración	guardar la configuración de los filtros del procesamiento
Cargar configuración	cargar la configuración de los filtros del procesamiento
Salir	cerrar <b>BrainApp</b>

Tabla 6. Descripción del menú Ver.

Acción	Descripción
Barra de herramientas	mostrar la barra de herramientas
Imágenes cargadas	cargar imágenes volumétricas
Propiedades de la imagen	cargar series de imágenes
Filtros del proceso	guardar la imagen resultado del procesamiento
Datos de la segmentación	guardar la configuración de los filtros del procesamiento

Tabla 7. Descripción del menú Ayuda

Acción	Descripción
Ayuda de BrainApp	mostrar la ayuda de la aplicación
Acerca de...	mostrar información sobre <b>BrainApp</b>

### 3.3.2. Barra de herramientas de BrainApp

Para facilitar el uso de **BrainApp**, esta aplicación tiene una barra de herramientas (Figura 21) que permite acceder a las opciones más importantes



con sólo un clic. La barra se puede mostrar u ocultar a través de Ver → Barra de herramientas.



Figura 21. Barra de herramientas.

Esta barra muestra los siguientes botones:

1. Cargar una imagen 3D.
2. Cargar una serie de imágenes.
3. Guardar la imagen que resulta del procesamiento.
4. Cargar una configuración del procesamiento desde un fichero.
5. Guardar la configuración actual del procesamiento hacia un fichero.
6. Mostrar/Ocultar la lista de imágenes cargadas.
7. Mostrar/Ocultar las propiedades de las imágenes.
8. Mostrar/Ocultar la configuración del procesamiento.
9. Mostrar/Ocultar los datos de la segmentación.

### 3.4. Carga de imágenes

**BrainApp** soporta el trabajo con imágenes en escala de grises para los siguientes formatos: BMP, JPG, TIF, PNG, DICOM y RAW. No soporta imágenes a color.

Para cargar imágenes en **BrainApp** existen dos opciones:

1. Cargar una imagen 3D.
2. Cargar una serie de imágenes 2D que conformen un volumen.

La primera opción se utiliza mediante el siguiente diálogo al cual se accede por el menú Archivo → Cargar Imagen... (Figura 22).

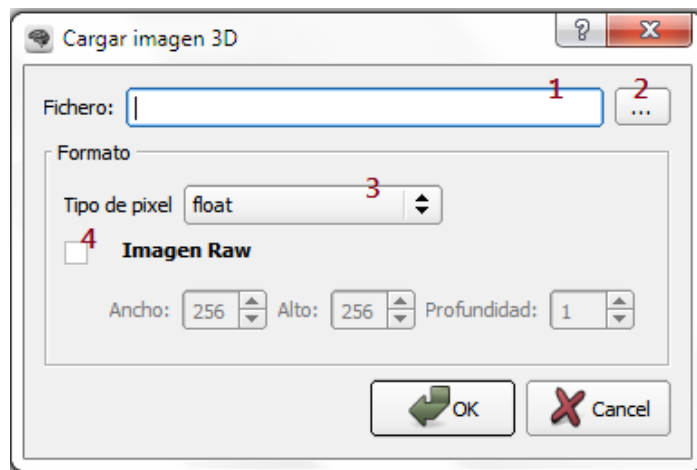


Figura 22. Diálogo para cargar imágenes volumétricas.

Los elementos de este diálogo son:

1. Nombre del fichero que contiene la imagen.
2. Botón para seleccionar el fichero manualmente.
3. Selección del tipo de píxel de la imagen. Esta opción debería obviarse para el caso de imágenes con formatos JPG, TIF, BMP, PNG o DICOM; donde sí es indispensable esta opción es en el caso de imágenes RAW.
4. Esta casilla se debe marcar en el caso de que se desee cargar una imagen RAW para especificar el ancho, alto y profundidad de la misma.



Figura 23. Diálogo para cargar series de imágenes.

En el caso de la carga de series de imágenes se utiliza el diálogo mostrado en la Figura 23, accesible desde Archivo → Cargar serie.... Este diálogo contiene elementos análogos al anterior, con las siguientes diferencias:

1. El nombre del fichero debe contener el comodín %d en el lugar donde se deben sustituir los números de la serie de imágenes. Por ejemplo para cargar las imágenes “file1.bmp”, “file2.bmp” y “file3.bmp” se debe usar el nombre “file%d.bmp”.
2. Número inicial en la numeración de los ficheros que conforman la serie de imágenes.
3. Número final en la numeración de los ficheros que conforman la serie de imágenes.

### 3.4.1. Lista de imágenes cargadas

Las imágenes que están cargadas en **BrainApp** se muestran en la ventana flotante “Imágenes cargadas” (Figura 24), que tiene los siguientes componentes:

1. Lista de las imágenes que se encuentran cargadas en **BrainApp**.
2. Botón que permite visualizar la imagen seleccionada o actualizarla en caso de que se hayan cambiado algunas de sus propiedades (atajo de teclado: F5).

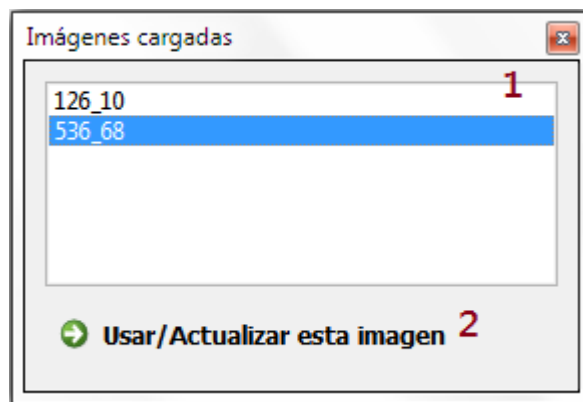


Figura 24. Ventana flotante "Imágenes cargadas".

### 3.4.2. Modificación de las propiedades de las imágenes

La ventana flotante “Propiedades de la imagen” (accesible desde Ver → Propiedades de la imagen) sirve para modificar algunas propiedades de las imágenes cargadas en **BrainApp**. Esta ventana, al seleccionar una imagen en la ventana flotante “Imágenes cargadas”, muestra las siguientes propiedades de dicha imagen (Figura 25).

1. Id: identificador interno para la imagen.
2. Nombre: nombre de la imagen (este es el que se muestra en la lista de imágenes cargadas).
3. Formato: formato de la imagen original.
4. Fichero: fichero desde el cual se cargó la imagen.
5. Dimensión: dimensión de la imagen (2D o 3D).
6. Tipo de píxel: píxel interno usado por **BrainApp**.
7. Tamaño: dimensiones de la imagen (ancho, alto, profundidad).
8. Espaciado: espaciado de los vóxeles de la imagen en cada uno de los ejes (X, Y, Z).

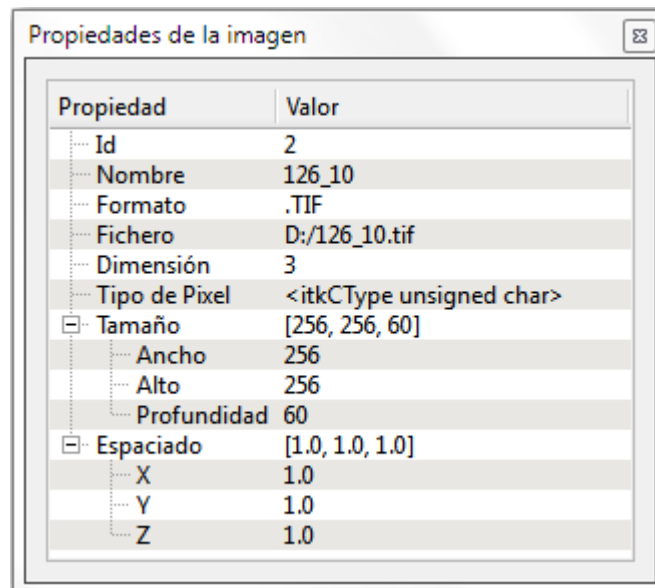


Figura 25. Ventana flotante "Propiedades de la imagen".

De todas estas propiedades sólo se pueden editar la que representa el nombre de la imagen y las que muestran el espaciado.

### 3.5. Visualizaciones de las imágenes

Las visualizaciones en la ventana principal de **BrainApp**, tanto la de la imagen original como la del resultado del procesamiento, muestran tres cortes de la imagen que grafican. Cada uno de estos cortes se puede configurar y por defecto están inicializados de manera que sean perpendiculares dos a dos. Además pueden tener un punto seleccionado que se mantiene sincronizado entre los tres.

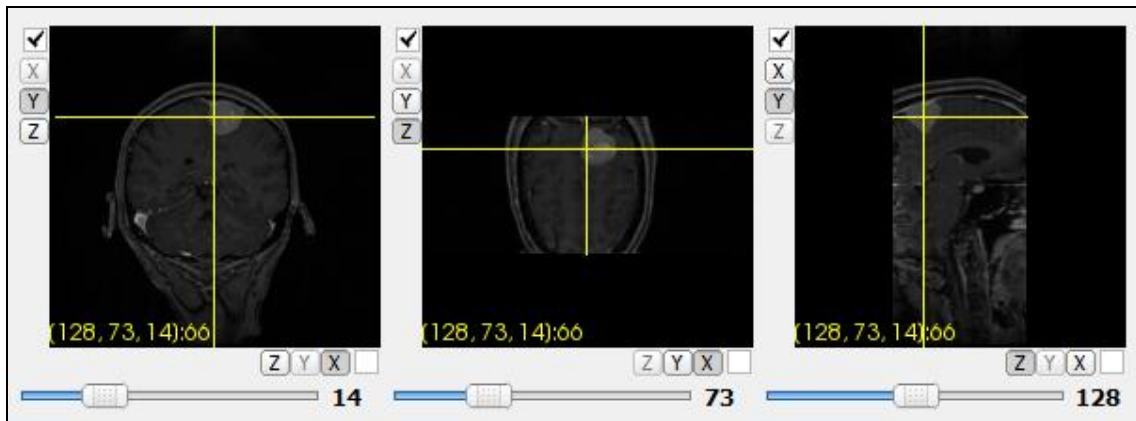


Figura 26. Visualización de una imagen.

Los elementos que permiten configurar los cortes se describen a continuación (Figura 27).

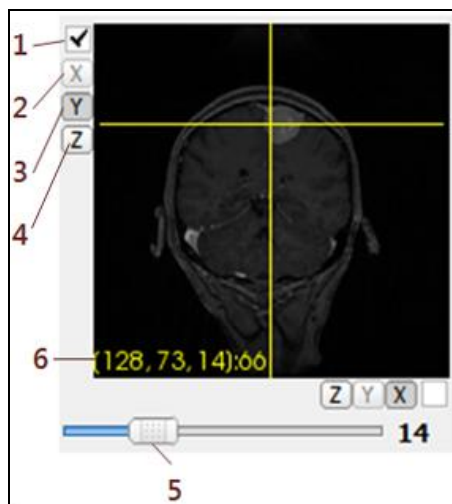


Figura 27. Visualización de un corte.

1. Casilla que invierte el eje utilizado (marcada: invertido, sin marcar: normal).
2. Eje que no se puede utilizar (pues se está usando en la otra dirección, en el caso de la imagen de ejemplo se usa en la dirección horizontal).
3. Eje en uso en esta dirección (vertical en la imagen de ejemplo).
4. Eje que se puede usar en esta dirección.
5. Selector del corte para el plano definido por los ejes (en el ejemplo: el corte 14).
6. Coordenadas y valor del punto seleccionado (marcado por las líneas amarillas).

Además de estos elementos también es posible interactuar directamente con la imagen mostrada en la visualización de la siguiente forma:

- Ctrl + Clic Izq. sostenido: rota la imagen.
- Shift + Clic Izq. sostenido: mueve la imagen.
- Rueda del ratón arriba: aumenta zoom.
- Rueda del ratón abajo: disminuye zoom.

### **3.6. Procesamiento de imágenes y configuración de los filtros**

Para la configuración del procesamiento con **BrainApp** se usa la ventana flotante “Filtros del proceso” que permite especificar cuáles filtros se desean utilizar y editar los parámetros para cada uno de ellos. Esta ventana es accesible a través de Ver → Filtros del proceso.

En la Figura 28 se muestran los elementos de esta ventana flotante:

1. Selector de uso del filtro: especifica si se usará el filtro en el procesamiento (marcado, se usa en el procesamiento; desmarcado, no)
2. Parámetros de los filtros: muestra todos los parámetros configurables de los filtros.

3. Valores de los parámetros: muestran los valores de los parámetros de los filtros y además permiten la edición de los mismos.

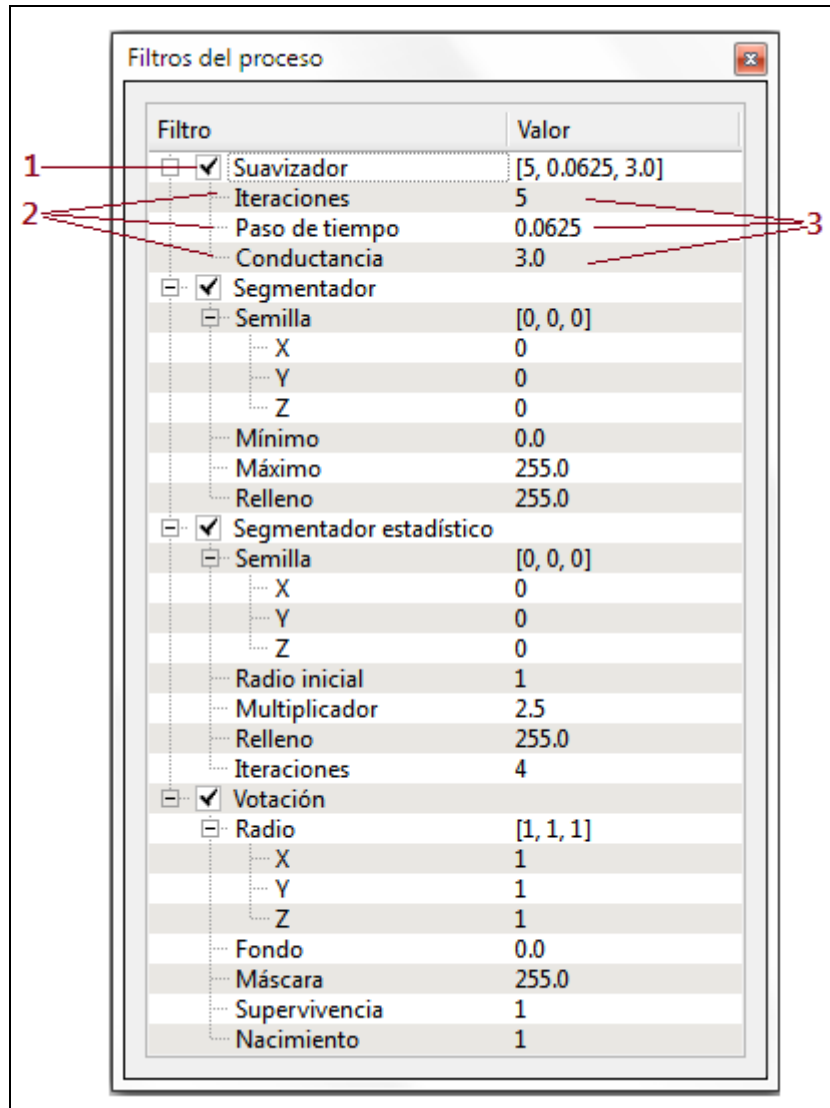


Figura 28. Ventana flotante "Filtros del proceso".

Los filtros de **BrainApp** funcionan de la siguiente forma.

**Suavizador:** realiza una difusión anisotrópica en la imagen que preserva los límites entre las regiones (transiciones claro-oscuro), con una buena sensibilidad al contraste. Es un filtro iterativo que necesita para su correcto funcionamiento el ajuste de varios parámetros que definen la *ecuación*

*modificada de difusión de curvatura* basada en la definición de la ecuación de difusión anisotrópica de Perona-Malik (Perona and Malik, 1990).

**Segmentador:** en cada iteración el filtro analiza los píxeles vecinos de la región actual (los que comparten al menos un vértice con algún píxel ya incluido en la misma) y, si sus valores están dentro del rango de intensidades válidas, estos píxeles pasan a formar parte de la región para la nueva iteración. Cuando haya una iteración que no agregue píxeles a la región se da por concluido el procesamiento. El rango de intensidades válidas se define por un umbral máximo y mínimo. La región inicial del procesamiento viene dada por un píxel de la imagen que sirve como semilla. El resultado del procesamiento es una región que se muestra en forma de máscara donde los píxeles que forman parte de la segmentación tienen el valor de relleno especificado y los demás píxeles tienen valor cero.

**Segmentador estadístico:** para cada iteración calcula la varianza y la media de los valores de los píxeles ya incluidos en la región, luego todos los píxeles vecinos a la región que tengan una intensidad comprendida en el intervalo (definido por la una vecindad de la media con radio igual al valor de la varianza por un multiplicador) se incluyen en la región para la próxima iteración. El proceso finaliza luego de un número predefinido de iteraciones. Para comenzar este filtro necesita un píxel inicial que sirva como semilla y un radio que se usa para saber cuántos vecinos de la semilla se deben utilizar al calcular la varianza y media de la primera iteración. El resultado se obtiene en forma de máscara donde los píxeles pertenecientes a la segmentación se muestran con el valor de relleno especificado y los demás tendrán valor cero.

**Filtro de Votación:** lo que hace es realizar una *votación* para cada píxel de la imagen de entrada con vistas a decidir si este pasa a formar parte del *primer plano* o si se considerará un píxel de *fondo*. Para el proceso de *votación* el filtro cuenta los píxeles vecinos a la región que son de *primer plano*. Cuando se realiza la *votación* se considera *vecino* a todo píxel que se encuentre dentro de



un ortoedro definido por un radio N-dimensional ( $N=2,3$ ), centrado en el píxel que se está analizando. Las componentes del radio definen las longitudes de los lados del ortoedro (medidos en píxeles), por ejemplo un radio  $r = (r_1, r_2, r_3)$ , en una imagen 3D, define un ortoedro de lados  $(2r_1+1) \times (2r_2+1) \times (2r_3+1)$ . Luego de realizada la *votación* se le ajusta el valor al píxel en cuestión en dependencia del resultado de la misma.

Para cada filtro de **BrainApp** (Figura 28) en la ventana de configuración del procesamiento se muestran los parámetros de configuración, de los cuales se presenta un resumen en las siguientes tablas.

Tabla 8. Parámetros de configuración del filtro de suavizado.

Parámetro	Significado
<b>Iteraciones</b>	Cantidad de iteraciones del filtro.
<b>Paso de tiempo</b>	Influye en el refinamiento de la imagen y en el rendimiento del filtro, se recomiendan valores menores a 0.0625.
<b>Conductancia</b>	Influye en la precisión de los límites entre las regiones presentes en la imagen, entre más pequeño más preciso. Se recomiendan valores entre 0.5 y 3.0.

Tabla 9. Parámetros de configuración del filtro segmentador.

Parámetro	Significado
<b>Semilla</b>	Es el píxel inicial usado como semilla, en las sub-propiedades X, Y, Z se deben especificar las coordenadas del mismo.
<b>Máximo</b>	Umbral máximo a tener en cuenta a la hora de obtener la región de la segmentación.
<b>Mínimo</b>	Umbral mínimo tener en cuenta a la hora de obtener la región de la segmentación.
<b>Relleno</b>	Valor de sustitución para los píxeles de la región segmentada.

Tabla 10. Parámetros de configuración del filtro segmentador estadístico.

Parámetro	Significado
<b>Semilla</b>	Es el píxel inicial usado como semilla, en las sub-propiedades X, Y, Z se deben especificar las coordenadas del mismo.
<b>Radio inicial</b>	Radio que se tiene en cuenta para incluir los píxeles vecinos a la semilla en el cálculo inicial de la varianza y media de la región.
<b>Multiplicador</b>	Valor que modifica la varianza de la región
<b>Relleno</b>	Valor de sustitución para los píxeles de la región segmentada.
<b>Iteraciones</b>	Cantidad de iteraciones del filtro.

Tabla 11. Parámetros de configuración del filtro de votación.

Parámetro	Significado
<b>Radio</b>	Radio a tener en cuenta para seleccionar los píxeles vecinos al píxel para el cual se realiza la votación
<b>Fondo</b>	Valor que representa el fondo en la imagen
<b>Máscara</b>	Valor que representa la máscara o primer plano de la imagen.
<b>Supervivencia</b>	Umbral de supervivencia para la votación.
<b>Nacimiento</b>	Umbral de nacimiento para la votación.

Como parte de la estrategia de **BrainApp** para garantizar la reproducibilidad también se brinda la opción de guardar/cargar la configuración actual de los filtros en la aplicación hacia/desde un fichero. Estas opciones son accesibles desde Archivo → Guardar configuración y Archivo → Cargar configuración.

La ejecución del procesamiento se inicia desde el botón “Procesar imagen” de la ventana principal (3.3).

### 3.6.1. Cálculo de los datos de la segmentación

Para calcular los datos relativos a una segmentación se usa la ventana flotante “Datos de la segmentación” que se muestra a través del menú Ver → Datos de la segmentación... (Figura 29).

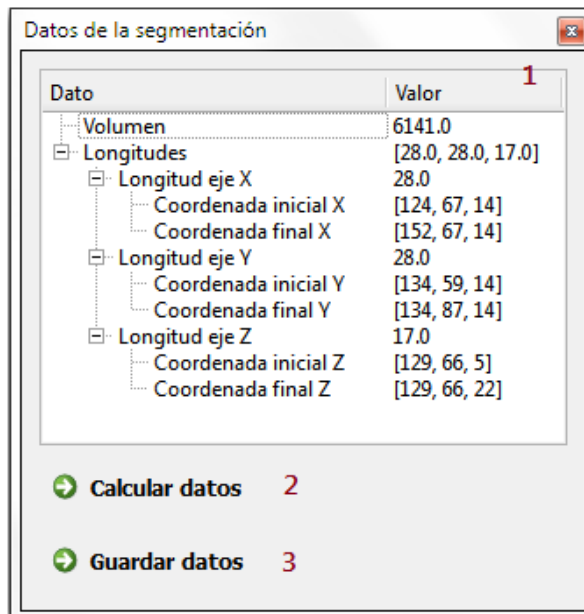


Figura 29. Ventana flotante "Datos de la segmentación".

Elementos de la ventana:

1. Listado de datos de la segmentación.
2. Botón para calcular los datos de la segmentación.
3. Botón para guardar los datos calculados.

Datos de una segmentación que se pueden calcular usando **BrainApp**:

- El volumen, dado en unidades cúbicas, y calculado teniendo en cuenta el espaciado de los vóxeles.
- Las longitudes máximas de la segmentación en cada una de las direcciones canónicas, calculadas en unidades planas teniendo en cuenta el espaciado de los vóxeles.
- Las coordenadas de inicio y fin, en la imagen original, de los segmentos que representan las longitudes máximas.

### 3.7. Conclusiones parciales

En este capítulo se profundizó en los componentes de la interfaz gráfica de **BrainApp**, para aclarar el método de trabajo con la aplicación. Se documentó cada uno de los elementos de los menús, la barra de herramientas y las ventanas flotantes que permiten la visualización de datos y configuración del procesamiento. También se describieron las visualizaciones y sus métodos de interacción.

Como resultado principal de este capítulo queda el manual de usuario oficial que se distribuirá junto con la versión actual de **BrainApp**.

## CONCLUSIONES

En la presente investigación se desarrolló un método y una aplicación para la segmentación de tumores cerebrales en IRM. El método semiautomático desarrollado se basa en técnicas de “region-growing” con un pre-procesamiento de suavizado de la imagen y un post-procesamiento de refinamiento. Con la implementación realizada se logró:

- Segmentar semiautomáticamente tumores mostrados en IRM.
- Visualizar, mediante una segmentación, la posición y forma del tumor.
- Calcular el volumen y dimensiones principales del tumor.
- Mostrar los datos calculados, con facilidades para guardar los resultados.

Los algoritmos desarrollados en la presente investigación para el cálculo del volumen y las dimensiones principales del tumor son eficientes y presentan un alto grado de exactitud.

La interfaz gráfica implementada para la aplicación conjuga elementos de uso intuitivo con útiles áreas de datos, garantizando el control del usuario sobre el proceso, brindando información en todo momento sobre el resultado parcial del procesamiento.

Con el método y la aplicación implementados se obtuvieron segmentaciones semiautomáticas que presentan buenos resultados comparados con el estándar de segmentación manual.

La aplicación que surge como producto de la presente investigación garantiza la reproducibilidad de las segmentaciones.

## RECOMENDACIONES

1. Introducir el uso de la aplicación en el trabajo clínico en los hospitales de nuestro país.
2. Continuar el desarrollo de nuevas versiones de la aplicación y elaborar más algoritmos para extraer información de las segmentaciones, como información de las estructuras cerebrales afectadas.
3. Profundizar en la investigación de nuevos métodos para la segmentación de tumores cerebrales.
4. Investigar nuevas vías para aumentar el grado de automatización del proceso y disminuir la interacción humana.
5. Explorar la utilidad de una visualización volumétrica para su posible implementación, que permita visualizar el tumor de manera resaltada, junto con las demás estructuras cerebrales, en un espacio 3D.

## BIBLIOGRAFÍA

- ACR–ASNR 2008. ACR–ASNR practice guideline for the performance and interpretation of magnetic resonance imaging (MRI) of the brain. The American College of Radiology.
- AHMED, M. M. & MOHAMAD, D. B. Segmentation of Brain MR Images for Tumor Extraction by Combining Kmeans clustering and Perona-Malik Anisotropic Diffusion Model. *International Journal of Image Processing*, 2.
- ALLOUI, K. & NACEUR, M. S. 2009. 3D Brain Tumor Segmentation Using Level-Sets Method and Meshes Simplification from Volumetric MR Images. *World Academy of Science, Engineering and Technology*, 57.
- ALLISON, W. 2006. *Fundamental Physics for Probing and Imaging*, Oxford University Press.
- ATKINS, M. S., SIU, K., LAW, B., ORCHARD, J. & ROSENBAUM, W. Difficulties of T1 brain MRI segmentation techniques. *In*: SONKA, M. & FITZPATRICK, J. M., eds. *SPIE Medical Imaging 2002: Image Processing*, 2002.
- BOKDE, A. L. W., TEIPEL, S. J., SCHWARZ, R., LEINSINGER, G., BUERGER, K., MOELLER, T., MÖLLER, H.-J., HAMPEL, H. & ALLISON, W. 2004. Reliable manual segmentation of the frontal, parietal, temporal, and occipital lobes on magnetic resonance images of healthy subjects. *Brain Research Protocols*, 14.
- CATES, J. E., WHITAKER, R. T. & JONES, G. M. 2004. Case Study: An Evaluation of User-Assisted Hierarchical Watershed Segmentation.
- CICHOSZ, S. L., VANGSGAARD, S., JØRGENSEN, A. S., KANNIK, K. E., STEFFENSEN, E. & ESKILDSEN, S. F. 2010. Brain tumor segmentation from MRI: a comparative study. *IADIS International Conferences Computer Graphics, Visualization, Computer Vision and Image Processing 2010*

- CLARK, M. C., HALL, L. O., GOLDFOF, D. B., VELTHUIZEN, R., MURTAUGH, F. R. & SILBINGER, M. S. 1998. Unsupervised Brain Tumor Segmentation Using Knowledge-Based and Fuzzy Techniques.
- CUI, M., WONKA, P., RAZDAN, A. & HU, J. 2007. A new image registration scheme based on curvature scale space curve matching.
- DAN, E. B. 2003. *Committing Medical Image Analysis*.
- DICE, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26, 297–302.
- DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE. 2009. Strategic document.
- DROZDOWICZ, B., BERNASCONI, G., REYES, M., SABA, F. & SIMÓN, G. 2005. Segmentación semiautomática de imágenes de resonancia magnética, basada en redes neuronales artificiales. *Ciencia, Docencia y Tecnología*, 30, 117-155.
- DUBEY, R. B., M.HANMANDLU, GUPTA, S. K. & GUPTA, S. K. 2010. An Advanced Technique for Volumetric Analysis *International Journal of Computer Applications*, 1.
- FLETCHER-HEATH, L. M., HALL, L. O., GOLDFOF, D. B. & MURTAGH, F. R. 2001. Automatic Segmentation of Non-enhancing Brain Tumors in Magnetic Resonance Images.
- FOO, J. L. 2006. A Survey of User Interaction and Automation in Medical Image Segmentation Methods. Iowa State University.
- GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J. 2003. *Design Patterns*, Addison-Wesley.
- GERIG, G., JOMIER, M. & CHAKOS, M. 2001. VALMET: a new validation tool for assessing and improving 3D object segmentation. *In: NIESSEN, W. & VIERGEVER, M. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2001*. New York: Springer.
- GERING, D. T., GRIMSON, W. E. L. & KIKINIS, R. 2002. Recognizing Deviations from Normalcy for Brain Tumor Segmentation.

- GOKCAY, E. A new clustering algorithm for segmentation of Magnetic Resonance Images. 2000 University of Florida.
- GORDILLO, N. 2010. *Contributions to Automatic and Unsupervised MRI Brain Tumor Segmentation: A New Fuzzy Approach*. Programa de Doctorat, Universitat Politècnica de Catalunya.
- GRUPO DE NEUROONCOLOGÍA 2011. Criterios de respuesta de los tumores cerebrales *Sociedad Española de Neurorradiología*.
- HANEY, S. M., THOMPSON, P. M., CLOUGHESY, T. F., ALGER, J. R. & TOGA, A. W. 2001. Tracking Tumor Growth Rates in Patients with Malignant Gliomas: A Test of Two Algorithms. *AJNR Am J Neuroradiol*, 22, 73-82.
- HO, S., BULLITT, E. & GERIG, G. 2002. Level Set Evolution with Region Competition: Automatic 3-D Segmentation of Brain Tumors.
- HO, S., KIM, Y., BULLITT, E. & GERIG, G. Medical image segmentation by 3-D level set evolution.
- IBÁÑEZ, L., SCHROEDER, W., NG, L. & CATES, J. 2005. The ITK Software Guide. Insight Software Consortium.
- JACCARD, P. 1912. The distribution of flora in the alpine zone. *New Phytologist*, 11, 37–50.
- JAYA, J. & THANUSHKODI, K. 2011. Certain Investigation on MRI Segmentation for the Implementation of CAD System *Wseas Transactions on Computers*, 10.
- JOE, B. N., FUKUI, M. B., MEITZER, C. C., HUANG, Q.-S., DAY, R. S., GREER, P. J. & BOZIK, M. E. 1999. Brain Tumor Volume Measurement: Comparison of Manual and Semiautomated Methods. *Radiology*, 212, 811-816.
- JOSHI, S., LORENZEN, P., GERIG, G. & BULLIT, E. Structural and Radiometric Asymmetry in Brain Images.



- KARAYIANNIS, N. B. & PAI, P.-I. Segmentation of Magnetic Resonance Images Using Fuzzy Algorithms for Learning Vector Quantization. *In: VANNIER, M., ed., 1996.*
- KAUS, M., WARFIELD, S. K., JOLESZ, F. A. & KIKINIS, R. 1999. Adaptive Template Moderated Brain Tumor Segmentation in MRI.
- KAUS, M. R., WARFIELD, S. K., NABAVI, A., BLACK, P. M., JOLESZ, F. A. & KIKINIS, R. 2001. Automated Segmentation of MR Images of Brain Tumors. *Radiology*, 218, 858-591.
- KAZEROONI, A. F., AHMADIAN, A., SEREJ, N. D., RAD, H. S., SABERI, H., YOUSEFI, H. & FARNIA, P. Segmentation of Brain Tumors in MRI Images Using Multi-scale Gradient Vector Flow. 33rd Annual International IEEE EMBS Conference, 2011.
- KHOTANLOU, H., COLLIOT, O., ATIF, J. & BLOCH, I. 2009. 3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models. *Fuzzy Sets and Systems*, 160, 1457–1473.
- KITWARE INC. 2006. *VTK User's Guide*, Colombia, Kitware Inc.
- KUNDEL, H. & DEAN, P. 2002. Tumor Imaging. *Image-Processing Techniques for Tumor Detection*.
- KUPERMAN, V. 2000. *Magnetic Resonance Imaging. Physical Principles and Applications.*, Academic Press.
- LOGESWARI, T. & KARNAN, M. 2009. An improved implementation of brain tumor detection using segmentation based on soft computing. *Journal of Cancer Research and Experimental Oncology*, 2.
- LUO, S., LI, R. & OURSELIN, S. 2003. A New Deformable Model Using Dynamic Gradient Vector Flow and Adaptive Balloon Forces. *APRS Workshop on Digital Image Computing*, 9-14.
- MANOHAR, V. & GU, Y. Digital Image Processing - CAP 5400 MRI Segmentation Using Fuzzy C-Means and Finite Gaussian Mixture Model

- MAYER, A. & GREENSPAN, H. 2009. An Adaptive Mean-Shift Framework for MRI Brain Segmentation. *IEEE Transactions on Medical Imaging*, 28.
- MCINERNEY, T. & TERZOPOULOS, D. 2000. Deformable Models. *Handbook of Medical Imaging Processing and Analysis*.
- MOON, N., BULLITT, E., VAN LEEMPUT, K. & GERIG, G. 2002. Model-Based Brain and Tumor Segmentation. *International Conference on Pattern Recognition*.
- MOONIS, G., LIU, J., UDUPA, J. K. & HACKNEY, D. B. 2002. Estimation of Tumor Volume with Fuzzy-Connectedness Segmentation of MR Images. *AJNR Am J Neuroradiol*, 23, 356-363.
- MORAIN-NICOLIER, F., LEBONVALLET, S., BAUDRIER, É. & RUAN, S. Hausdorff Distance based 3D Quantification of Brain Tumor Evolution from MRI Images. 29th Annual International Conference of the IEEE EMBS, 2007 Lyon, France. 5597-5600.
- MUROFF, L. R. & RUNGE, V. M. 1995. The Use of MR Contrast in Neoplastic Disease of the Brain. *Top Magnetic Resonance Imaging*, 7, 137-157.
- NATIONAL CANCER INSTITUTE. 2011. *Tumores cerebrales en adultos: Tratamiento (PDQ®)* [Online]. Available: <http://www.cancer.gov/espanol/pdq/tratamiento/cerebralesadultos/HealthProfessional/page1> [Accessed 13/11/2011].
- O'DONNELL, L. 2001. *Semi-Automatic Medical Image Segmentation*. Thesis of Master of Science in Computer Science and Engineering, Massachusetts Institute of Technology.
- OHGAKI, H. & KLEIHUES, P. 2005. Population-Based Studies on Incidence, Survival Rates, and Genetic Alterations in Astrocytic and Oligodendroglial Gliomas. *Journal of Neuropathology and Experimental Neurology*, 64, 479-489.
- OLABARRIAGA, S. D. & SMEULDERS, A. W. M. 2001. Interaction in the Segmentation of Medical Images: A Survey. *Medical Image Analysis*, 5, 127-142.

- PADOLE, V. P. & CHAUDHARI, D. S. 2011. A Review of Segmentation Methods for Detection of Brain Tumor in MRI. *International Journal of Electronics, Communication & Soft Computing Science and Engineering*, 1.
- PERONA, P. & MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2, 629-639.
- PHAM, D. L., XU, C. & PRICE, J. L. 2000. Current Methods in Medical Image Segmentation. *Annu. Rev. Biomed. Eng.*, 2, 315-37.
- PHAM, D. L., XU, C. & PRINCE, J. L. 1998. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*.
- PRADHAN, N. & SINHA, A. K. 2010. Development of a composite feature vector for the detection of pathological and healthy tissues in FLAIR MR images of brain. *ICGST-BIME*, 10.
- PRASAD, P. V. (ed.) 2006. *Magnetic Resonance Imaging. Methods and Biologic Applications.*, Totowa, New Jersey: Humana Press.
- PRASTAWA, M. 2007. An MRI Segmentation Framework for Brains with Anatomical Deviations.
- PRASTAWA, M., BULLIT, E., MOON, N., VAN LEEMPUT, K. & GERIG, G. 2003. Automatic Brain Tumor Segmentation by Subject Specific Modification of Atlas Priors. *Academic Radiology*, 10, 1341-1348.
- PRASTAWA, M., BULLITT, E., HO, S. & GERIG, G. 2004. A Brain Tumor Segmentation Framework Based On Outlier Detection. *Medical Image Analysis*.
- PRECIOSO, F., BARLAUD, M., BLU, T. & UNSER, M. Robust real-time segmentation of images and videos using a smoothing-spline snake-based algorithm. *IEEE Transactions on Image Processing*.
- PREIM, B. & BARTZ, D. 2007. *Visualization in medicine*, Elsevier, Morgan Kaufman.

- PRICE, S. J. 2007. The Role of Advanced MR Imaging in Understanding Brain Tumour Pathology. *Br J Neurosurg*, 21, 562-675.
- REDDY, K. K., SOLMAZ, B., YAN, P., AVGEROPOULOS, N. G., RIPPE, D. J. & SHAH, M. 2012. Confidence guided enhancing brain tumor segmentation in multi-parametric mri. *ISBI*.
- REISER, M. F., SEMMLER, W. & HRICAK, H. 2008. Magnetic Resonance Tomography.
- ROUSSEEUW, P. J. & VAN DRIESSEN, K. 1998. A Fast Algorithm for the Minimum Covariance Determinant Estimator
- SALMAN, Y. M. 2009. Modified technique for volumetric brain tumor measurements. *Biomedical Science and Engineering*, 2, 16-19
- SCHMIDT, M., LEVNER, I., GREINER, R., MURTHA, A. & BISTRITZ, A. Segmenting Brain Tumors using Alignment-Based Features. Fourth International Conference on Machine Learning and Applications (ICMLA'05) 2005.
- SCHROEDER, W., MARTIN, K. & LORENSEN, B. 2002. *The Visualization Toolkit*, Kitware Inc.
- SELVANAYAKI, K. & KARNAN, M. 2010. CAD System for Automatic Detection of Brain Tumor through Magnetic Resonance Image-A Review. *International Journal of Engineering Science and Technology* 2.
- STYNER, M., BRECHBÜHLER, C., SZÉKELY, G. & GERIG, G. 2000. Parametric Estimate of Intensity Inhomogeneities Applied to MRI. *IEEE Transactions on Medical Imaging*, 19.
- TEI, A., TALOS, F., BHARATHA, A., FERRANT, M., BLACK, P. M., JOLESZ, F. A., KIKINIS, R. & WARFIELD, S. K. Tracking Volumetric Brain Deformation During Image Guided Neurosurgery.
- THACKER, N. A., WILLIAMSON, D. C. & POKRIC, M. 2004. Voxel based analysis of tissue volume from MRI data. *The British Journal of Radiology*, 77.

- WANG, J. & JIANG, T. 2006. Nonrigid registration of brain MRI using NURBS. *Pattern Recognition Letters*, 28.
- WARFIELD, S. K., TALOS, F., TEI, A., BHARATHA, A., NABAVI, A., FERRANT, M., BLACK, P. M., JOLESZ, F. A. & KIKINIS, R. 2002a. Real-time registration of volumetric brain MRI by biomechanical simulation of deformation during image guided neurosurgery. *Computing and Visualization in Science*, 5.
- WARFIELD, S. K., ZOU, K. H., KAUS, M. R. & WELLS, W. M. 2002b. Simultaneous Validation of Image Segmentation and Assessment of Expert Quality. In: FESSLER, J. & VANNIER, M. (eds.) *International Symposium on Biomedical Imaging: Macro to Nano*.
- WARFIELD, S. K., ZOU, K. H. & WELLS, W. M. 2002c. Validation of Image Segmentation and Expert Quality with an Expectation-Maximization Algorithm. *MICCAI 2002: Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention*. Tokyo, Japan.
- WEINMAN, J., BISSIAS, G., RISEMAN, E., HANSON, A. & HOROWITZ, J. 2003. Confidence-Based Segmentation of MR Imagery Using Region and Boundary Information with Nonlinear Scale-Space and Fast Marching Level Sets.
- WESTBROOK, C. & KAUT, C. 1998. *MRI in Practice*, Blackwell Science.
- WHITAKER, R. & XUE, X. 2001. Variable-Conductance, Level-Set Curvature for Image Denoising. *International Conference on Image Processing*.
- WONG, K.-P. 2005. Medical Image Segmentation: Methods and Applications in Functional Imaging. In: SURI, J. S., WILSON, D. L. & LAXMINARAYAN, S. (eds.) *Handbook of Biomedical Image Analysis: Segmentation Methods*. Kluwer Academic / Plenum Publishers.
- XIE, Z. & FARIN, G. E. 2004. Image Registration Using Hierarchical B-Splines. *IEEE Transactions on Visualization and Computer Graphics*, 10.
- YAO, J. 2006. Image Processing in Tumor Imaging. *New Techniques in Oncologic Imaging*, 79-102.

ZOU, K. H., WELLS, W. M., KIKINIS, R. & WARFIELD, S. K. 2004. Three validation metrics for automated probabilistic image segmentation of brain tumours. *Statistics in Medicine*, 23, 1259-1282.

## ANEXO 1. MÉTRICAS PARA COMPARAR SEGMENTACIONES

En este anexo se definen las métricas más utilizadas en la literatura para la validación de segmentaciones computacionales (Prastawa, 2007). A la hora de aplicarlas se deben sustituir los conjuntos A y B de las definiciones por los que se desean comparar, en este caso, los conjuntos de píxeles que representan al tumor en la segmentación computacional y la manual respectivamente.

Para dos conjuntos cualesquiera A y B. Se definen:

- El Coeficiente de Similitud de Dice:

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

- El Coeficiente de Similitud de Jaccard:

$$JSC(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- La fórmula de reducción entre los coeficientes de similitud:

$$JSC(A, B) = \frac{DSC(A, B)}{2 - DSC(A, B)}$$

- La distancia simétrica de Hausdorff:

$$H(\delta A, \delta B) = \max\{dist_H(\delta A, \delta B), dist_H(\delta B, \delta A)\} \text{ donde,}$$

$$dist_H(P, Q) = \max_{p \in P} \min_{q \in Q} \{distancia(p, q)\}$$

Para P y Q conjuntos de puntos cualesquiera y  $\delta A, \delta B$  los puntos de superficie de A y B.

Las distancias pueden ser restringidas para puntos hacia dentro o hacia fuera de las superficies determinadas por las segmentaciones.

## ANEXO 2. IMPLEMENTACIÓN DEL CÁLCULO DEL VOLUMEN Y DIMENSIONES PRINCIPALES

A continuación se muestra la implementación del algoritmo encargado de calcular el volumen y las dimensiones principales de un tumor a partir de una segmentación del mismo. La implementación se hizo en C++ para lograr eficiencia y utilizar todas las potencialidades de ITK.

```
ImageType::Pointer inputImage = reader->GetOutput();

typedef itk::ImageLinearConstIteratorWithIndex< ImageType > ConstIteratorType;

ConstIteratorType inputIt( inputImage, inputImage->GetRequestedRegion() );
inputIt.SetDirection(0);

ImageType::IndexType start_idx0, end_idx0, max_start_idx0, max_end_idx0;
int len0, maxLen0, vol = 0;
maxLen0 = 0;

char a[1000];
for(inputIt.GoToBegin(); ! inputIt.IsAtEnd(); inputIt.NextLine()) {
    inputIt.GoToBeginOfLine();
    len0 = 0;
    while ( ! inputIt.IsAtEndOfLine() ) {
        if(inputIt.Get() == 255) {
            vol++;
            if(len0==0)
                start_idx0 = inputIt.GetIndex();
            len0++;
        } else {
            if(maxLen0 < len0) {
                maxLen0 = len0;
                len0 = 0;
                max_start_idx0 = start_idx0;
                max_end_idx0 = inputIt.GetIndex();
            }
        }
        ++inputIt;
    }
    if(maxLen0 < len0 && inputIt.IsAtEndOfLine()) {
        maxLen0 = len0;
        len0 = 0;
        max_start_idx0 = start_idx0;
        max_end_idx0 = inputIt.GetIndex();
    }
}

inputIt.SetDirection(1);
```



```

ImageType::IndexType start_idx1, end_idx1, max_start_idx1, max_end_idx1;
int len1, maxLen1;
maxLen1 = 0;

for(inputIt.GoToBegin(); ! inputIt.IsAtEnd(); inputIt.NextLine()) {
    inputIt.GoToBeginOfLine();
    len1 = 0;
    while ( ! inputIt.IsAtEndOfLine() ) {
        if(inputIt.Get() == 255) {
            if(len1==0)
                start_idx1 = inputIt.GetIndex();
            len1++;
        } else {
            if(maxLen1 < len1) {
                maxLen1 = len1;
                len1 = 0;
                max_start_idx1 = start_idx1;
                max_end_idx1 = inputIt.GetIndex();
            }
        }
        ++inputIt;
    }
    if(maxLen1 < len1 && inputIt.IsAtEndOfLine()) {
        maxLen1 = len1;
        len1 = 0;
        max_start_idx1 = start_idx1;
        max_end_idx1 = inputIt.GetIndex();
    }
}

inputIt.SetDirection(2);
ImageType::IndexType start_idx2, end_idx2, max_start_idx2, max_end_idx2;
int len2, maxLen2;
maxLen2 = 0;

for(inputIt.GoToBegin(); ! inputIt.IsAtEnd(); inputIt.NextLine()) {
    inputIt.GoToBeginOfLine();
    len2 = 0;
    while ( ! inputIt.IsAtEndOfLine() ) {
        if(inputIt.Get() == 255) {
            if(len2==0)
                start_idx2 = inputIt.GetIndex();
            len2++;
        } else {
            if(maxLen2 < len2) {
                maxLen2 = len2;
                len2 = 0;
                max_start_idx2 = start_idx2;
                max_end_idx2 = inputIt.GetIndex();
            }
        }
        ++inputIt;
    }
    if(maxLen2 < len2 && inputIt.IsAtEndOfLine()) {
        maxLen2 = len2;
    }
}

```

```
        len2 = 0;
        max_start_idx2 = start_idx2;
        max_end_idx2 = inputIt.GetIndex();
    }
}
```

## ANEXO 3. MUESTRA COMPARATIVA ENTRE SEGMENTACIONES SEMIAUTOMÁTICAS Y MANUALES

En las figuras que se muestran a continuación se observan varios cortes de dos tumores segmentados con **BrainApp** comparados con sus respectivas segmentaciones manuales. La segmentación semiautomática obtenida se muestra con un relleno rojo y la segmentación manual está delineada en amarillo.

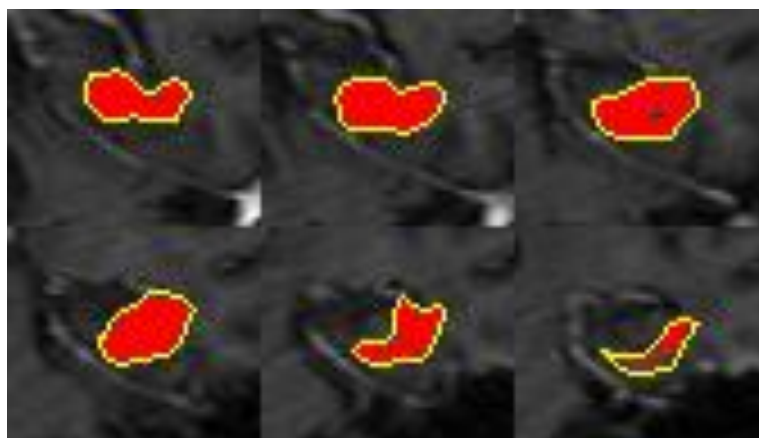


Figura 30. Muestra de 6 cortes de un tumor segmentado con BrainApp.

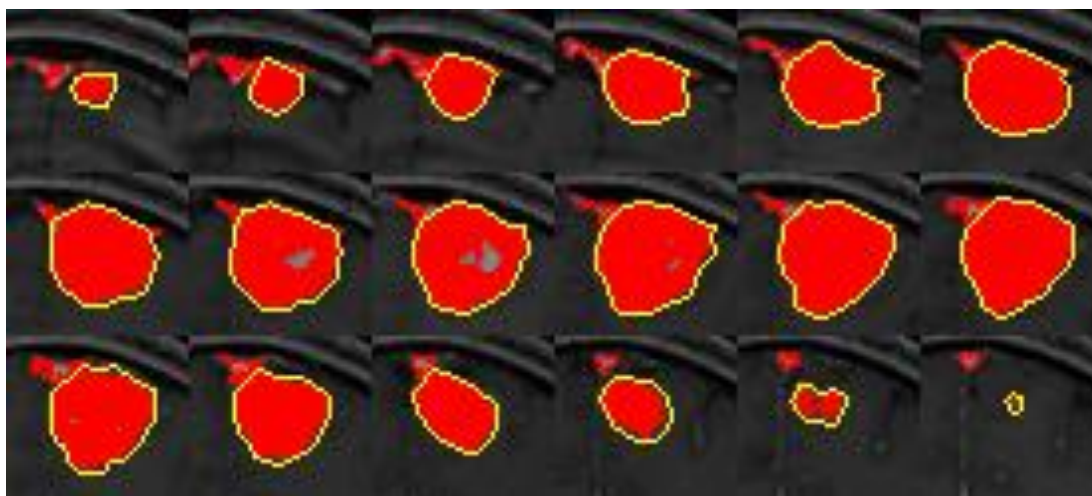


Figura 31. Muestra de 18 cortes de un tumor segmentado con BrainApp.