



UNIVERSIDAD  
DE GRANADA



# 3DCurator

---

Sistema gráfico de ayuda al diagnóstico e intervención de esculturas mediante datos médicos volumétricos

Autor: Francisco Javier Bolívar Lupiáñez

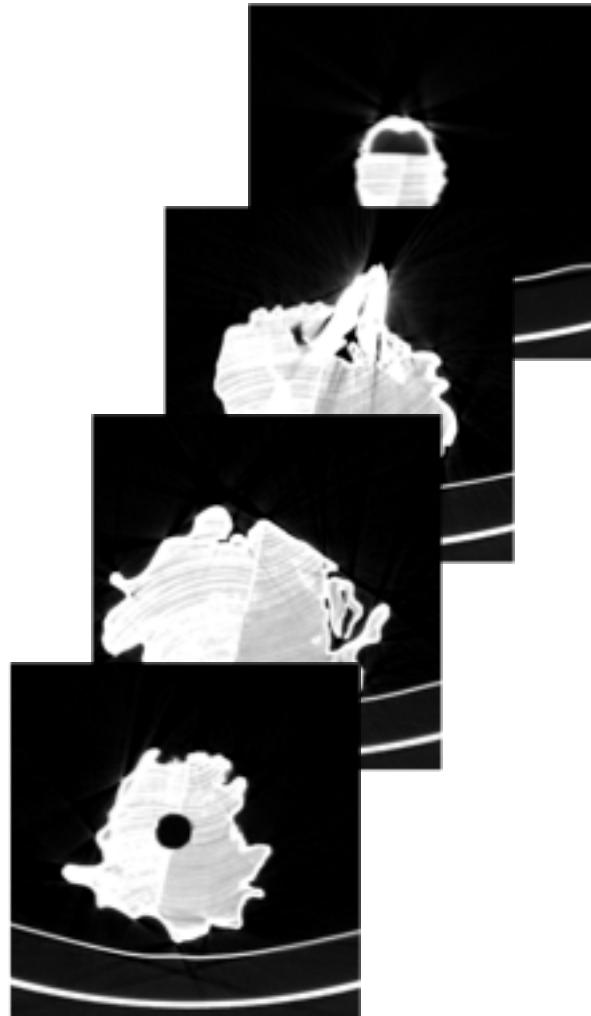
Director: Francisco Javier Melero Rus

# Índice

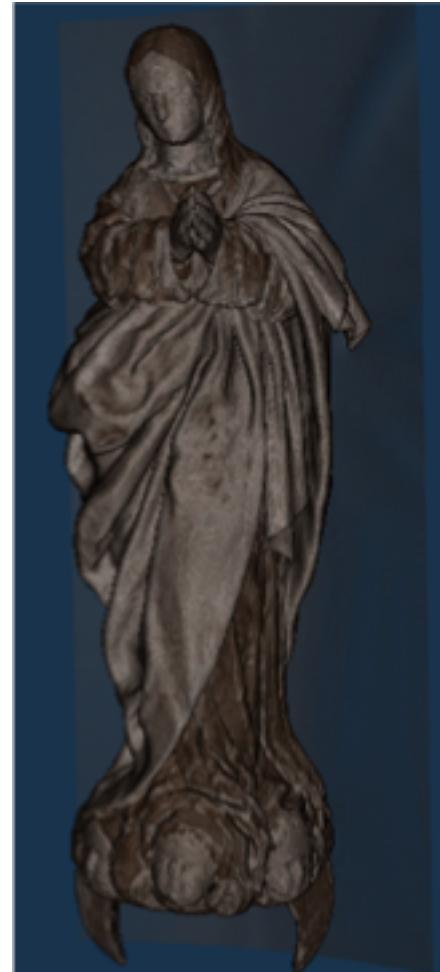
---

1. Introducción
2. Metodología de trabajo
3. Planificación
4. Análisis
5. Diseño
6. Desarrollo teórico
7. Implementación
8. Demostración
9. Conclusiones y trabajos futuros





Múltiples imágenes 2D



Reconstrucción volumétrica 3D

## Etapas del modelado de volúmenes

Obtención de imágenes



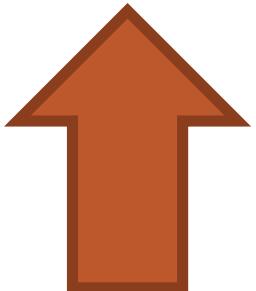
Filtrado



Segmentación

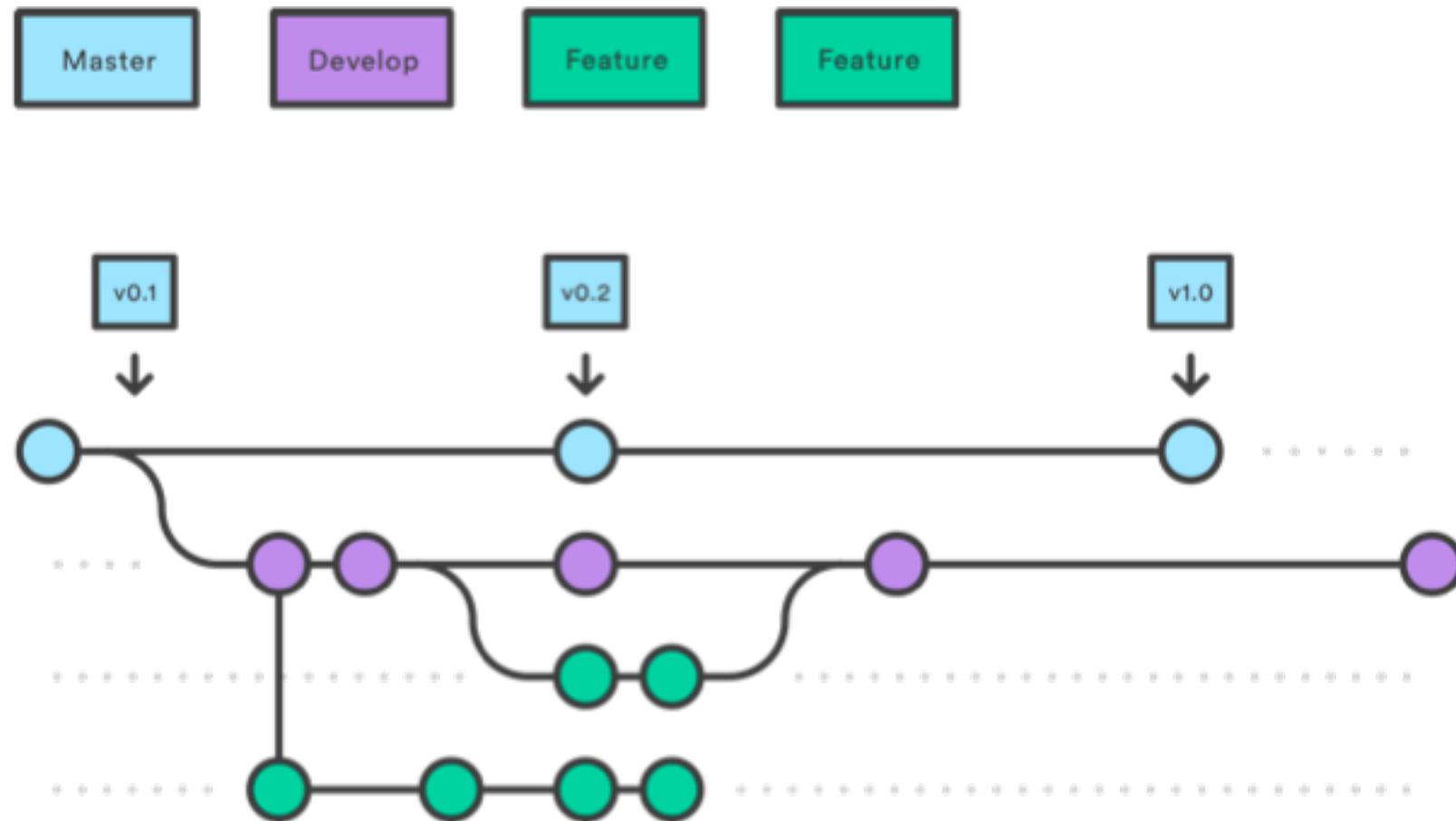


Renderizado



- Identificar cuál es la **técnica de obtención de datos** idónea para nuestro estudio
- Implementar **filtros** para reducir ruido
- **Dividir** la escultura en las distintas piezas de madera que componen su embón
- Desarrollar un conjunto de herramientas *in-app* para **documentar** los estudios realizados





# METODOLOGÍA DE TRABAJO

# GIT + GITHUB

fbilipi / 3DCurator

Code Issues Pull requests Projects Wiki Insights Settings

Milestones

6 Open 0 Closed

Sort ▾

GUI Improvements  
No due date Last updated about 22 hours ago

66% complete 3 open 6 closed

Edit Close Delete

Set-up environment  
No due date Last updated 27 days ago

72% complete 3 open 8 closed

Edit Close Delete

Automatic Segmentation  
No due date Last updated 7 months ago

0% complete 2 open 0 closed

Edit Close Delete

Semi-automatic Segmentation  
No due date Last updated 9 months ago

85% complete 3 open 17 closed

Edit Close Delete

Documentation tools  
No due date Last updated 10 months ago

100% complete 0 open 9 closed

Edit Close Delete

Noise reduction  
No due date Last updated 10 months ago

100% complete 0 open 4 closed

Edit Close Delete

fbilipi / 3DCurator

Code Issues Pull requests Projects Wiki Insights Settings

Filters Issue scope Labels Milestones New issue

11 Open 47 Closed

Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignees ▾ Sort ▾

Change from QVTKWidget to QVTKOpenGLWidget enhancement  
#71 opened 28 days ago by fbilipi Set-up environ...

Set-up environment for Mac OS X enhancement help wanted  
#65 opened 28 days ago by fbilipi Set-up environ...

Planes detection enhancement  
#64 opened on 7 Sep 2017 by fbilipi Automatic Segm...

Lines detection enhancement  
#63 opened on 7 Sep 2017 by fbilipi Automatic Segm...

Special characters in filenames bug  
#58 opened on 5 Aug 2017 by fbilipi GUI Improvements

3D Region Growing enhancement question  
#54 opened on 31 Jul 2017 by fbilipi Semi-automatic ...

Show degree symbol in protractor enhancement help wanted  
#50 opened on 24 Jul 2017 by fbilipi GUI Improvements

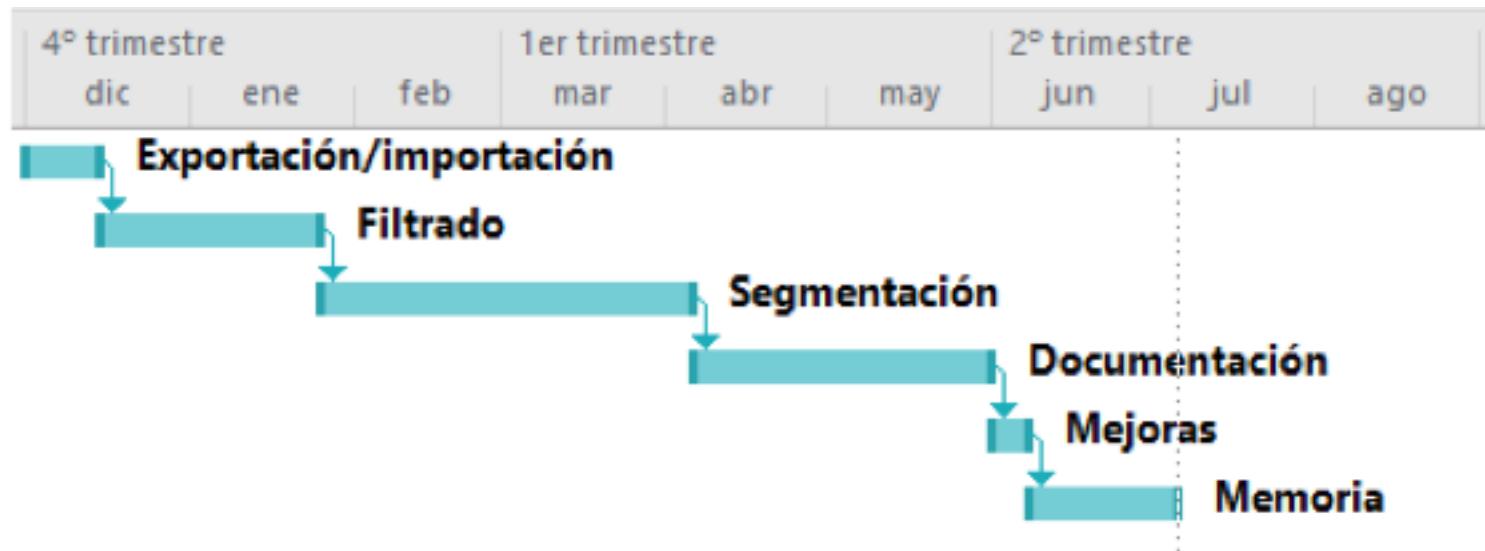
Show histogram in color and scalar opacity charts enhancement help wanted  
#39 opened on 13 Jul 2017 by fbilipi GUI Improvements

Segmentation using more than one line enhancement  
#15 opened on 2 May 2017 by fbilipi Semi-automatic ...

Parse plane coordinates into volume coordinates enhancement help wanted  
#4 opened on 28 Mar 2017 by fbilipi Semi-automatic ...

Set-up environment for Linux enhancement help wanted  
#2 opened on 1 Nov 2016 by fbilipi Set-up environ...





Inicio: 1/12/16  
Fin: 5/7/17  
Horas: 360\*

\*(redacción de memoria excluida)

- Se han registrado 312 horas en lugar de las 360 horas planificadas
- No se ha llegado a la fecha de entrega estimada
- Hubo que volver a planificar en dos ocasiones:
  - En junio por excesiva carga de trabajo durante el máster
  - En septiembre por empezar a trabajar a jornada completa



Inicio: 1/12/16  
Fin: 11/6/18  
Horas: 312

#	Descripción	Dep.	Est.	Prio.	#	Descripción	Dep.	Est.	Prio.
	<b>Auxiliar</b>				4.7	Eliminar regla	4.6	3	4
1.1	Exportar volumen	-	12	1	4.8	Editar regla	4.6	1	5
1.2	Importar volumen	1.1	8	5	4.9	Ocultar regla	4.6	2	6
	<b>Pre-procesamiento</b>				4.10	Mostrar regla	4.9	1	6
2.1	Filtro gaussiano	-	24	1	4.11	Crear transportador de ángulos	-	3	3
2.2	Filtro media	-	10	3	4.12	Eliminar transportador de ángulos	4.11	3	5
2.3	Filtro mediana	-	8	3	4.13	Editar transportador de ángulos	4.11	1	6
	<b>Segmentación</b>				4.14	Ocultar transportador de ángulos	4.11	2	7
3.1	Segmentar pieza de madera	1.1	64	1	4.15	Mostrar transportador de ángulos	4.14	1	7
	<b>Documentación</b>				4.16	Crear nota	-	4	2
4.1	Crear ROD	-	4	1	4.17	Eliminar nota	4.16	2	4
4.2	Eliminar ROD	4.1	4	3	4.18	Editar nota	4.16	3	5
4.3	Exportar ROD	4.1	2	2	4.19	Ocultar nota	4.16	2	6
4.4	Importar ROD	4.1	3	2	4.20	Mostrar nota	4.19	1	6
4.5	Cambiar ROD	4.1	3	1	<b>Mejoras de código</b>				
4.6	Crear regla	-	3	2	5.1.	Internacionalización	-	6	8

Total: 180 PH

Velocidad estimada: 2 h/PH

Velocidad real: 1,73 h/PH



Otros



Librerías de alto nivel



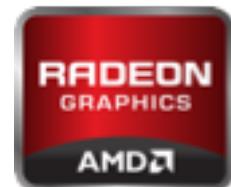
Librerías de bajo nivel

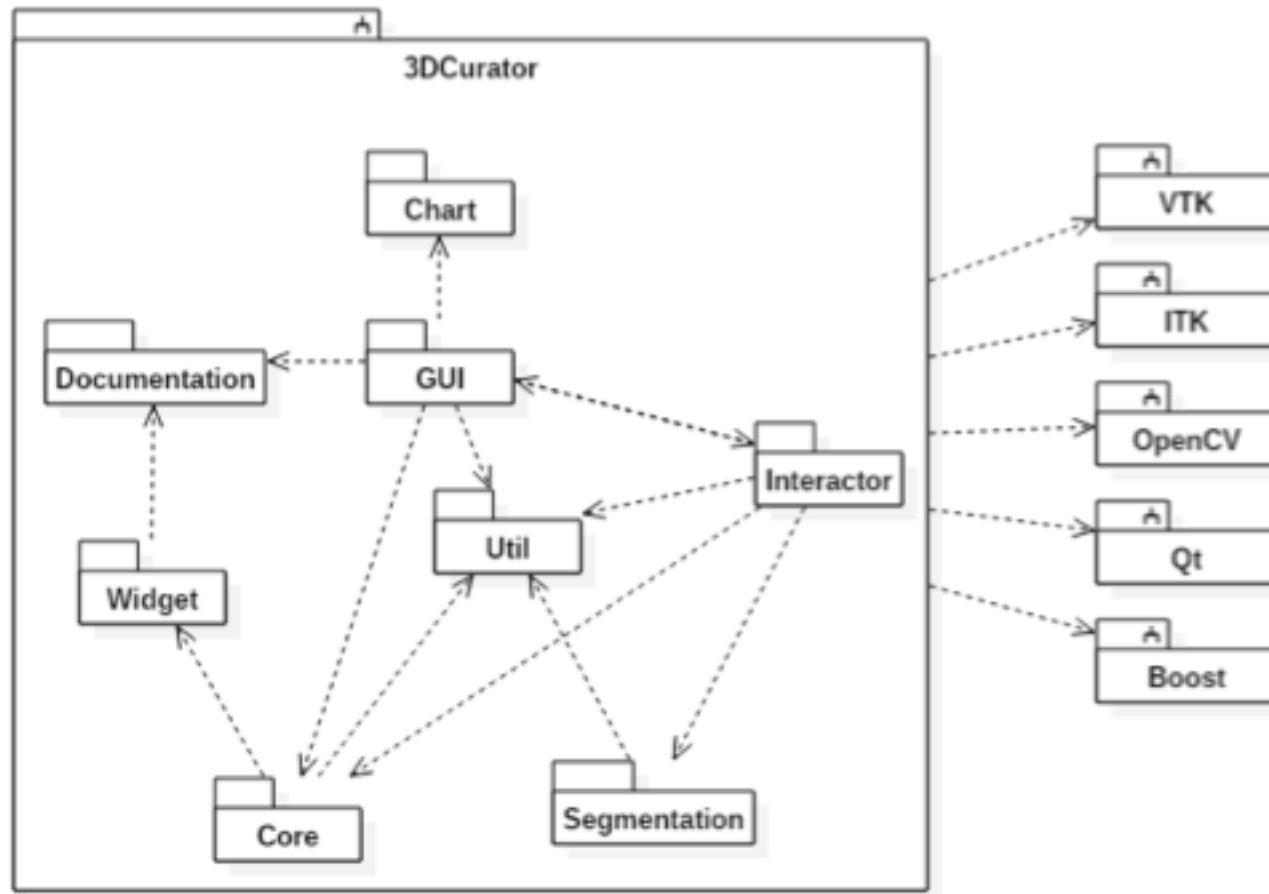


Lenguaje de programación

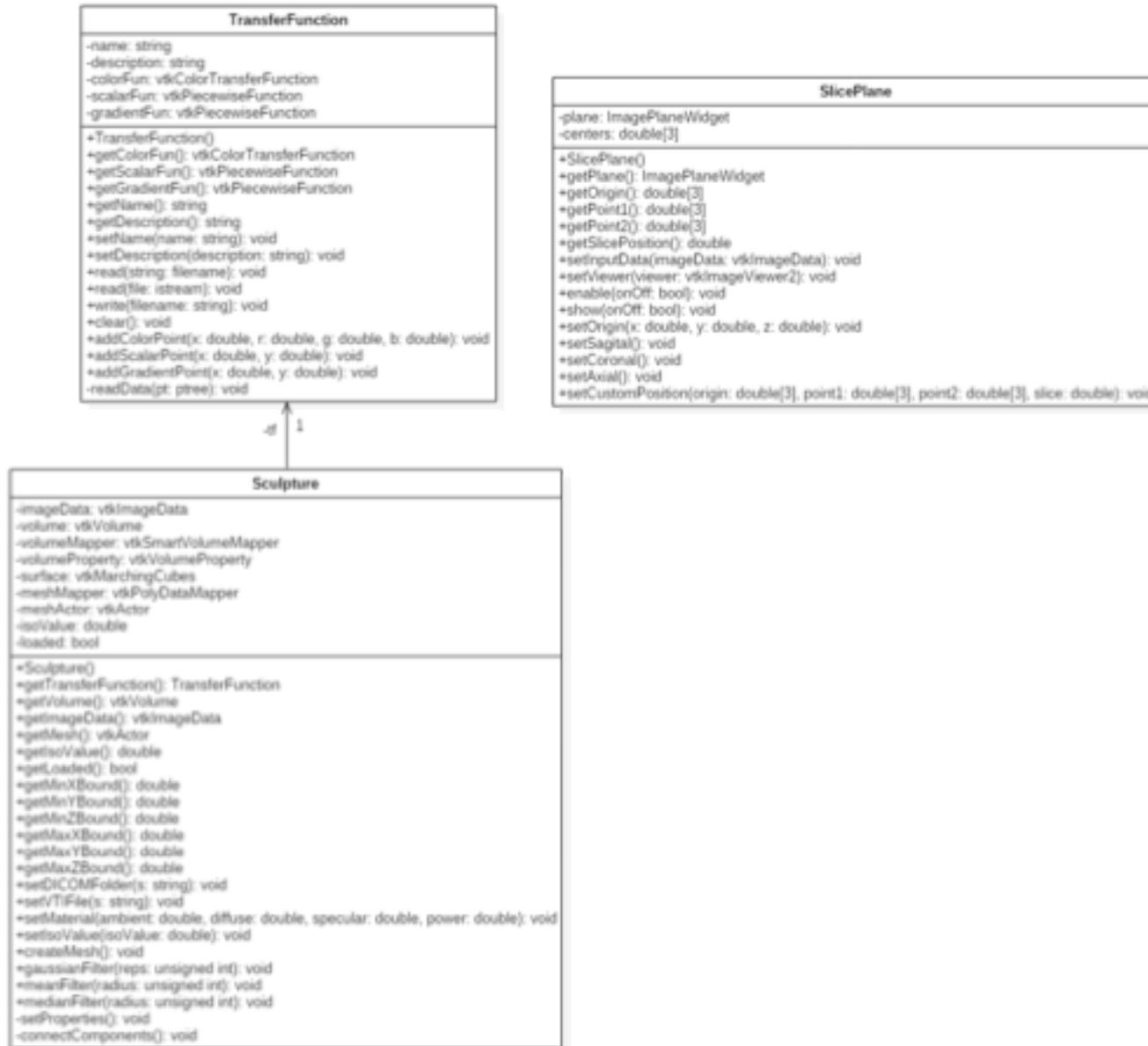


Hardware





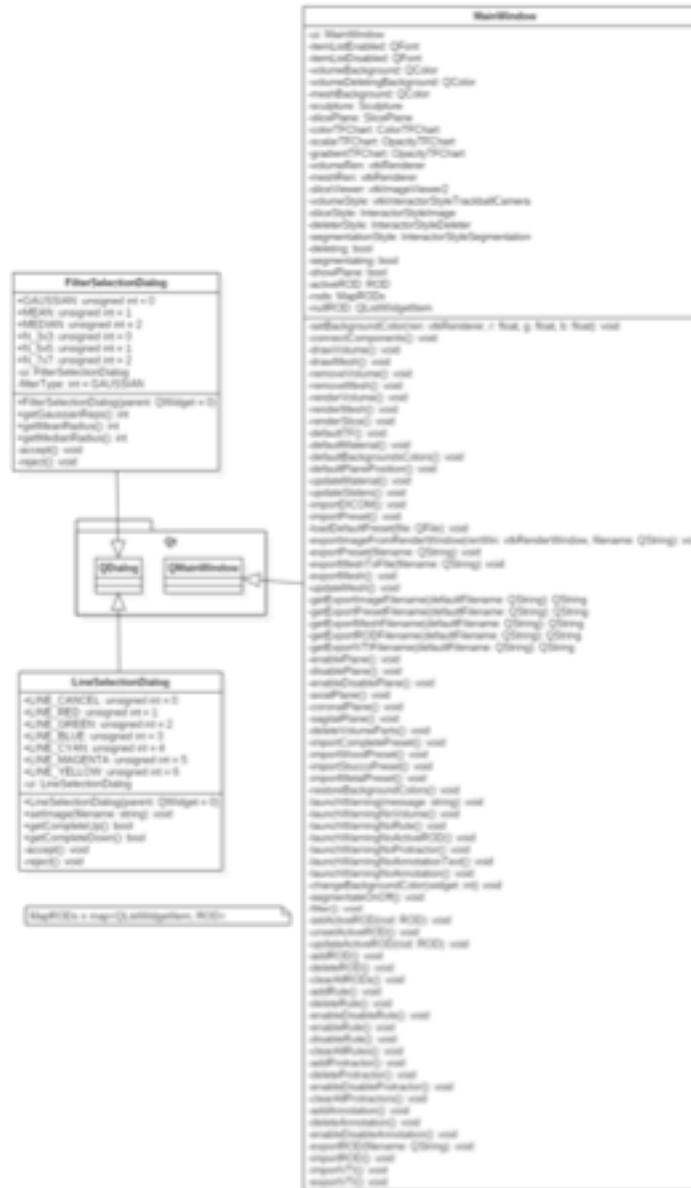


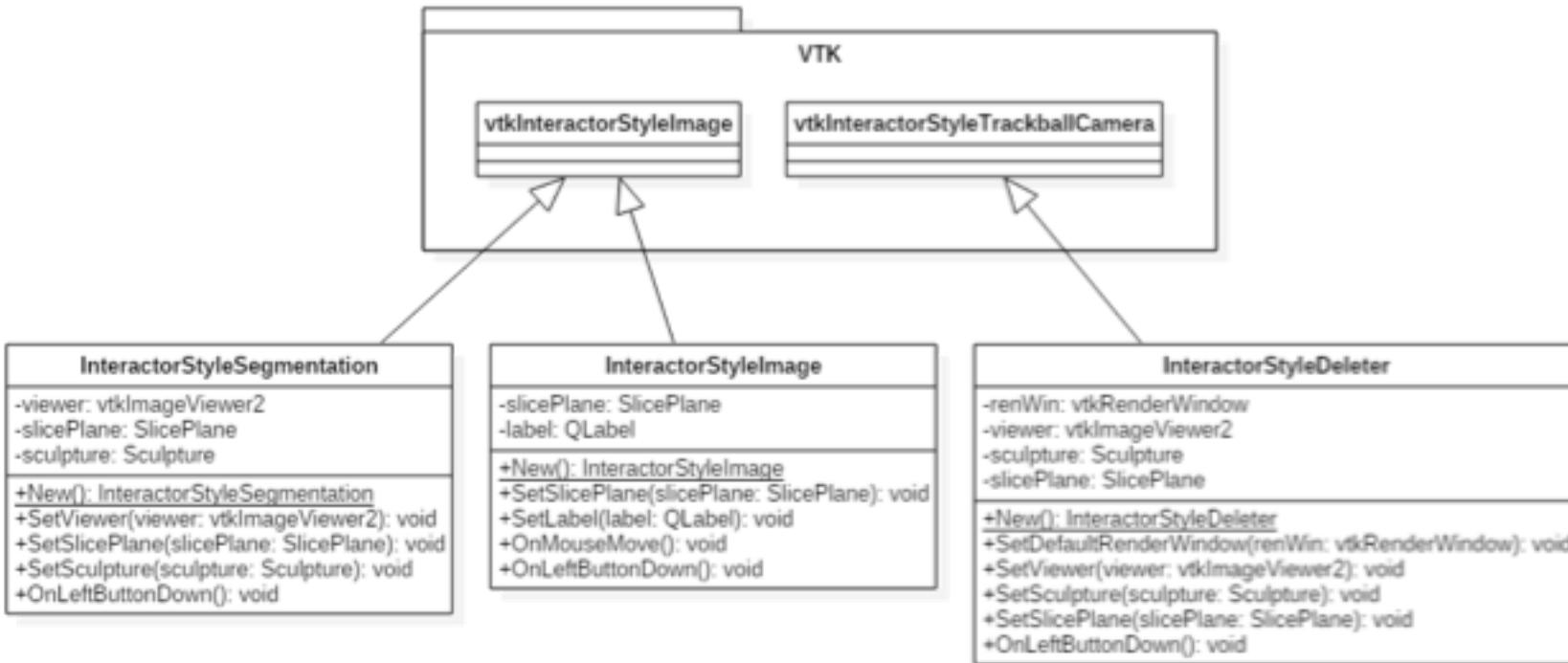


```
MapRules = map<QListWidgetItem, vtkDistanceWidget>
MapProtractors = map<QListWidgetItem, vtkAngleWidget>
MapAnnotations = map<QListWidgetItem, vtkCaptionWidget>
```

ROD
<pre>-name: string -origin: double[3] -point1: double[3] -point2: double[3] -slicePosition: double -rules: MapRules -protractors: MapProtractors -annotations: MapAnnotations -enabled: QFont -disabled: QFont -interactor: vtkRenderWindowInteractor  +ROD(name: string, origin: double[3], point1: double[3], point2: double[3], slice: double, enabled: QFont, disabled: QFont, interactor: vtkRenderWindowInteractor) +ROD(filename: string, enabled: QFont, disabled: QFont, interactor: vtkRenderWindowInteractor, ruleList: QListWidget, protractorList: QListWidget, annotationList: QListWidget) +getName(): string +getOrigin(): double[3] +getPoint1(): double[3] +getPoint2(): double[3] +getSlicePosition(): double +getRules(): MapRules +getProtractors(): MapProtractors +getAnnotations(): MapAnnotations +clearAllRules(): void +clearAllAnnotations(): void +addRule(item: QListWidgetItem): void +deleteRule(item: QListWidgetItem): void +enableDisableRule(item: QListWidgetItem): void +enableRule(item: QListWidgetItem): void +disableRule(item: QListWidgetItem): void +hideAllRules(): void +showAllRules(): void +addProtractor(item: QListWidgetItem): void +deleteProtractor(item: QListWidgetItem): void +enableDisableProtractor(item: QListWidgetItem): void +enableProtractor(item: QListWidgetItem): void +disableProtractor(item: QListWidgetItem): void +hideAllProtractors(): void +showAllProtractors(): void +clearAllProtractors(): void +addAnnotation(item: QListWidgetItem, text: string): void +deleteAnnotation(item: QListWidgetItem): void +enableDisableAnnotation(item: QListWidgetItem): void +enableAnnotation(item: QListWidgetItem): void +disableAnnotation(item: QListWidgetItem): void +hideAllAnnotations(): void +showAllAnnotations(): void +hideAll(): void +showAll(): void +samePlane(name: string, origin: double[3], point1: double[3], point2: double[3], slice: double): bool +write(filename: string): void +read(filename: string, ruleList: QListWidget, protractorList: QListWidget, annotationList: QListWidget): void</pre>







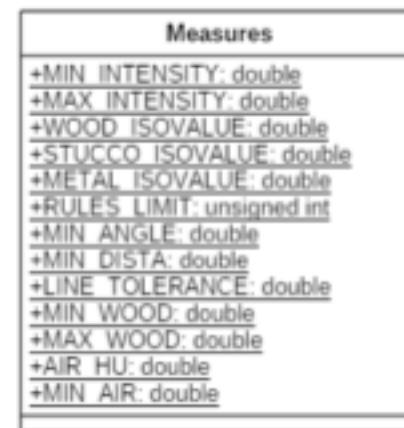
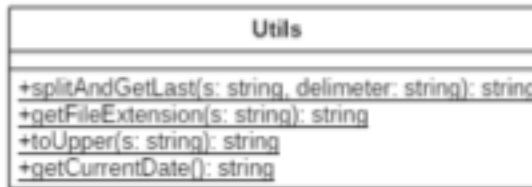
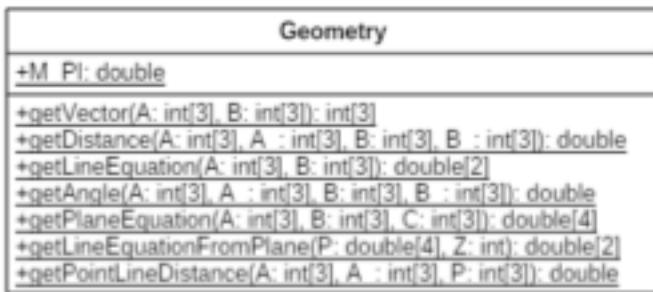
**RegionGrowingDeleter**

```
+searchInitialVoxel(imageData: vtkImageData, ijk: int[3], bounds: Bounds): Coord2D  
+deleteImage(imageData: vtkImageData, ijk: int[3], bounds: Bounds): void  
+deleteByImages(imageData: vtkImageData, ijk: int[3], bounds: Bounds): void
```

- Bounds = int[6]
- Line = pair<cv::Point, cv::Point>
- HoughLine = pair<Line, double>
- Coord2D = int[2]
- Coord3D = int[3]
- LineEq = double[2]
- PlaneEq = double[4]
- LineAngle = pair<Line, double>
- LineMatrix = Line[...][...]

**WoodSegmentation**

```
+longerLine(i: HoughLine, j: HoughLine): bool  
+getLinesFromImage(imageData: vtkImageData, colorFun: vtkColorTransferFunction, slice: int, bounds: Bounds): Line[...]  
+generateImage(imageData: vtkImageData, colorFun: vtkColorTransferFunction, slice: int, bounds: Bounds, lines: Line[...]): string  
+isInLine(coord: Coord2D, eq: LineEq, epsilon: int): bool  
+isAdjacent(imageData: vtkImageData, coord: Coord3D, MIN: double, MAX: double): bool  
+searchInitialVoxel(imageData: vtkImageData, ijk[3], bounds: Bounds, MIN: double, MAX: double, eq: LineEq): Coord2D  
+regionGrowingWithLineBoundImage(imageData: vtkImageData, ijk: int[3], bounds: Bounds, eq: LineEq): Coord2D  
+findNearestLine(lines: Line[...], goal: Line, originalZ: int, z: int): LineAngle  
+regionGrowingWithLineBoundVolume(imageData: vtkImageData, colorFun: vtkColorTransferFunction, ijk: int[3], bounds: Bounds, firstLine: Line, lines: LineMatrix, completeUp: bool, completeDown: bool): void
```





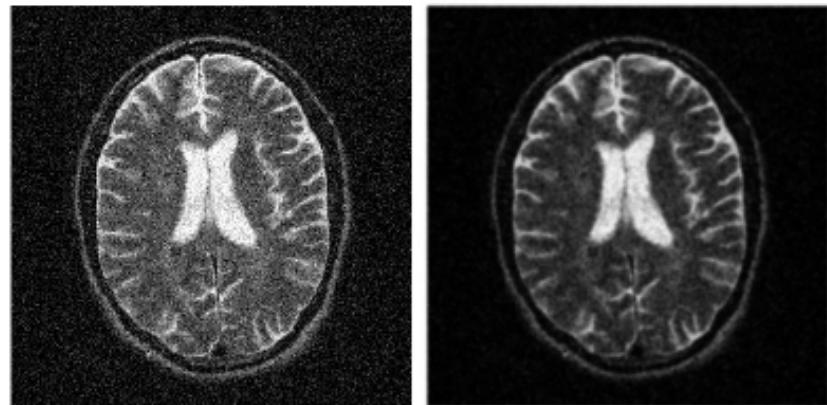
## Técnicas más utilizadas

- TC
  - IRM
  - PET
  - SPECT
- } Usan contrastes

	TC	IRM
Mayor resolución	✓	
Mejor contraste tejidos poco densos		✓
Menor tiempo de escaneo y costo	✓	
Unidades normalizadas	✓	

## Gaussiano

- Convolución utilizando la distribución de Gauss
- Suavizado



Suavizado usando filtro gaussiano

## Media

- Convolución utilizando la media
- Suavizado agresivo



Reducción de ruido tipo salt-and-pepper

## Mediana

- Convolución utilizando la mediana
- Reducción de ruido tipo *salt-and-pepper*

- **Vecindario:** dos vóxeles son vecinos si se encuentran a una distancia menor o igual a la definida para el vecindario
- **Adyacencia:** dos vóxeles son adyacentes si son vecinos y cumplen un criterio de similitud
- **Camino:** un camino entre dos vóxeles es una secuencia de vóxeles adyacentes entre uno y otro
- **Conectividad:** Existe conectividad entre dos vóxeles si se puede trazar un camino entre uno y otro
- **Componente conectado:** En un subconjunto de una imagen, para todos los vóxeles en ese subconjunto el conjunto de vóxeles conectados a éste se denomina componente conectado
- **Conjunto conectado:** Si en un subconjunto de una imagen solo hay un componente conectado este se denomina conjunto conectado
- **Región:** Un subconjunto de una imagen es una región si es un conjunto conectado
- **Borde:** El borde de una región es el conjunto de vóxeles que tienen algún vecino no perteneciente a la región

## Basada en umbrales

- Automática (2D y 3D)
- Parámetros: criterio de similitud
- Se basa en el criterio de similitud

## Crecimiento basado en umbrales

- Automática (2D y 3D)
- Parámetros: criterio de similitud y semilla
- Se basa en el concepto de adyacencia

## Watershed

- Automática (2D)
- Parámetros: factor de inundación
- Se basa en ver la imagen como un relieve topográfico

## Livewire

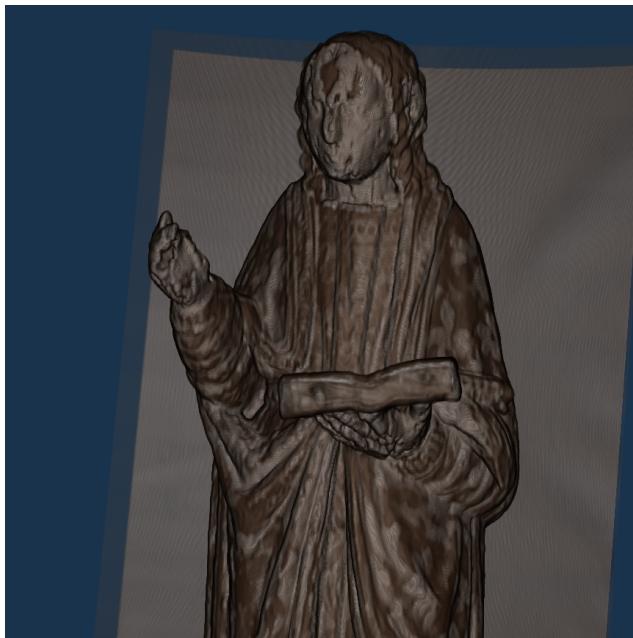
- Manual asistida (2D)
- Lazo magnético (Photoshop)
- Se basa en el concepto de borde y optimización (Dijkstra)



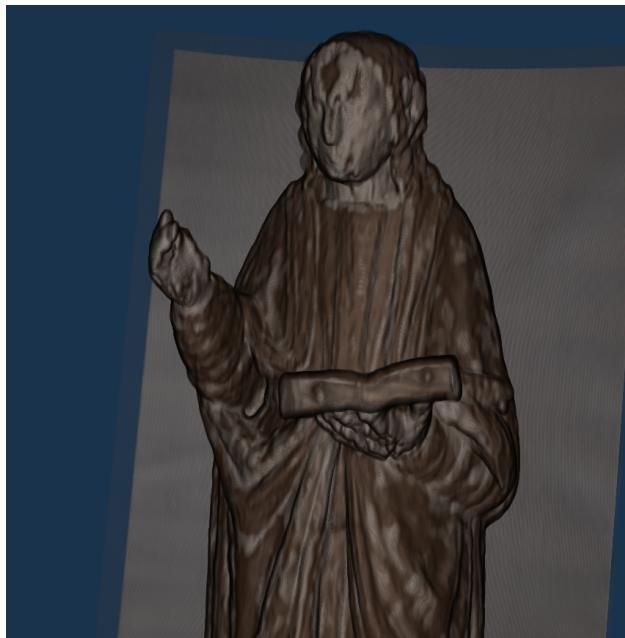
- Media: `itkMeanImageFilter`
- Mediana: `itkMedianImageFilter`
- Gaussiano: `itkBinomialBlurImageFilter`

VTK →  
ITK

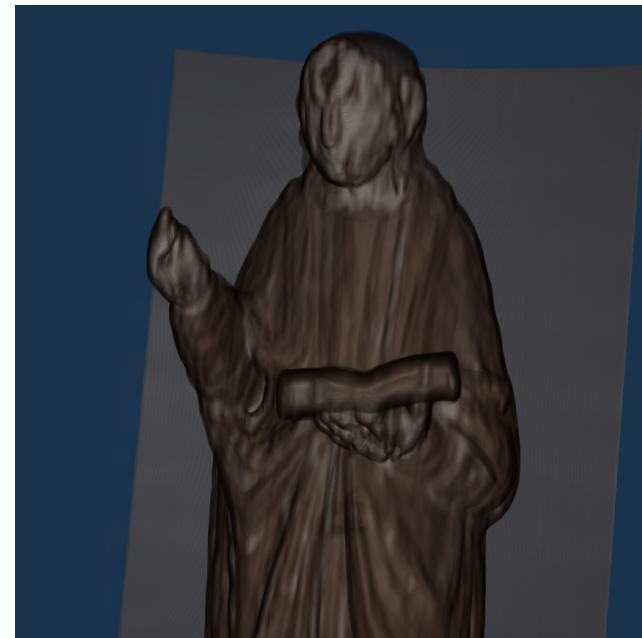
Filtro

ITK →  
VTK

Original



Gaussiano (2 rep.)

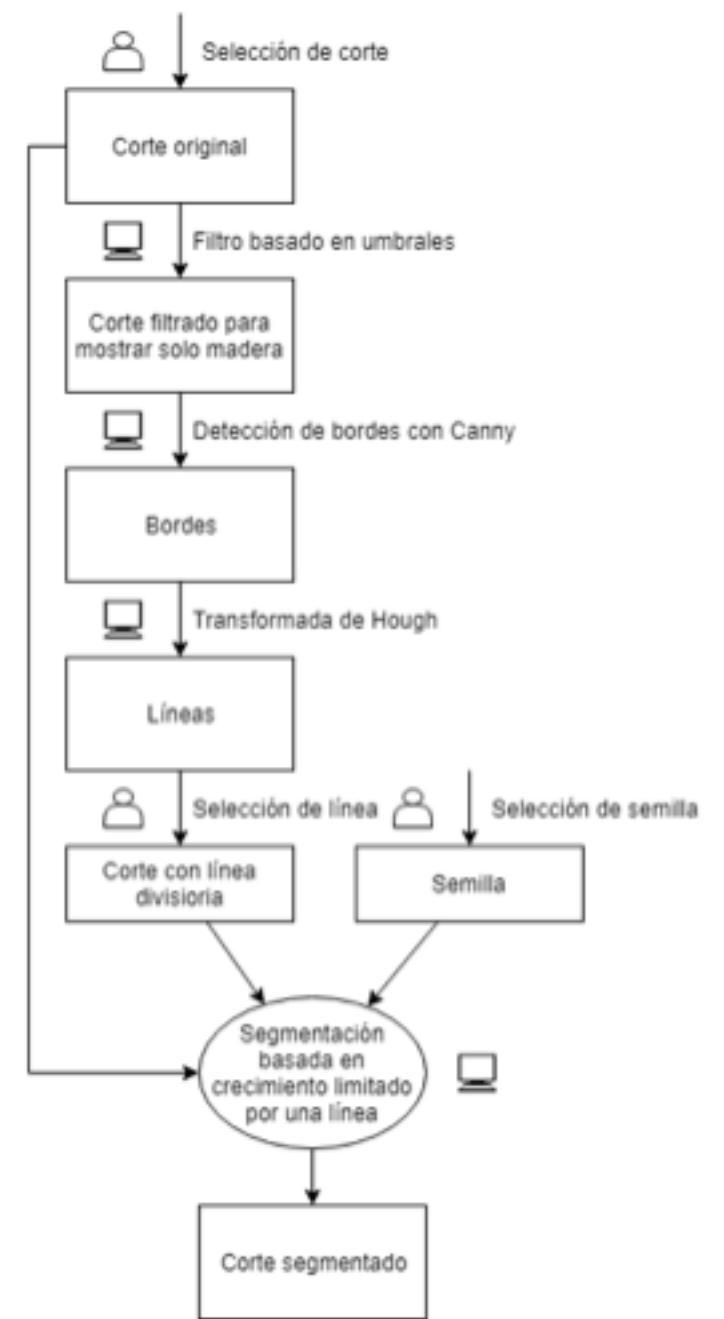
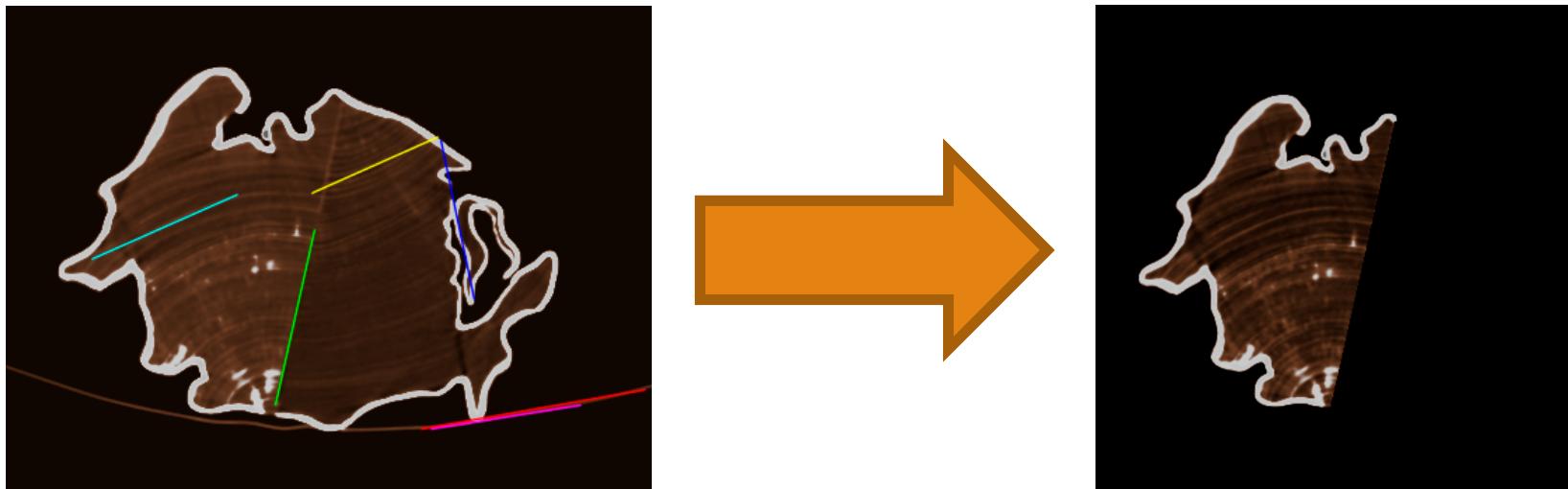


Media (vec. 5x5)

## Crecimiento basado en umbrales con línea divisoria



Proceso para 2D. Conversión a 3D con búsqueda automática de semillas y uso de plano auxiliar en cortes donde no se encuentre línea generada a partir de las encontradas justo antes y después





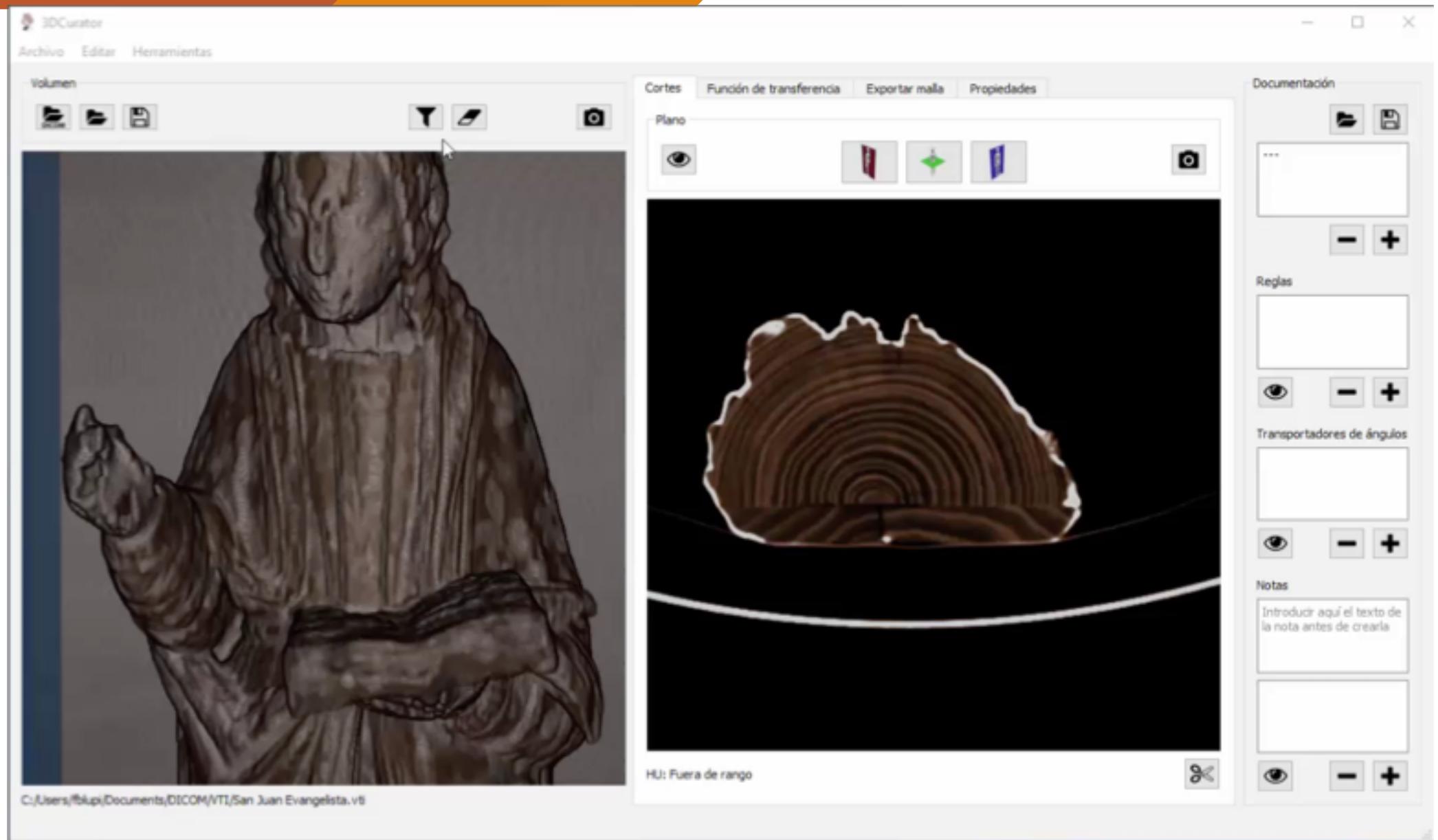
- Reglas: vtkDistanceWidget
- Transportador de ángulos: vtkAngleWidget
- Notas: vtkCaptionWidget

### ROD (*Region of Documentation*)

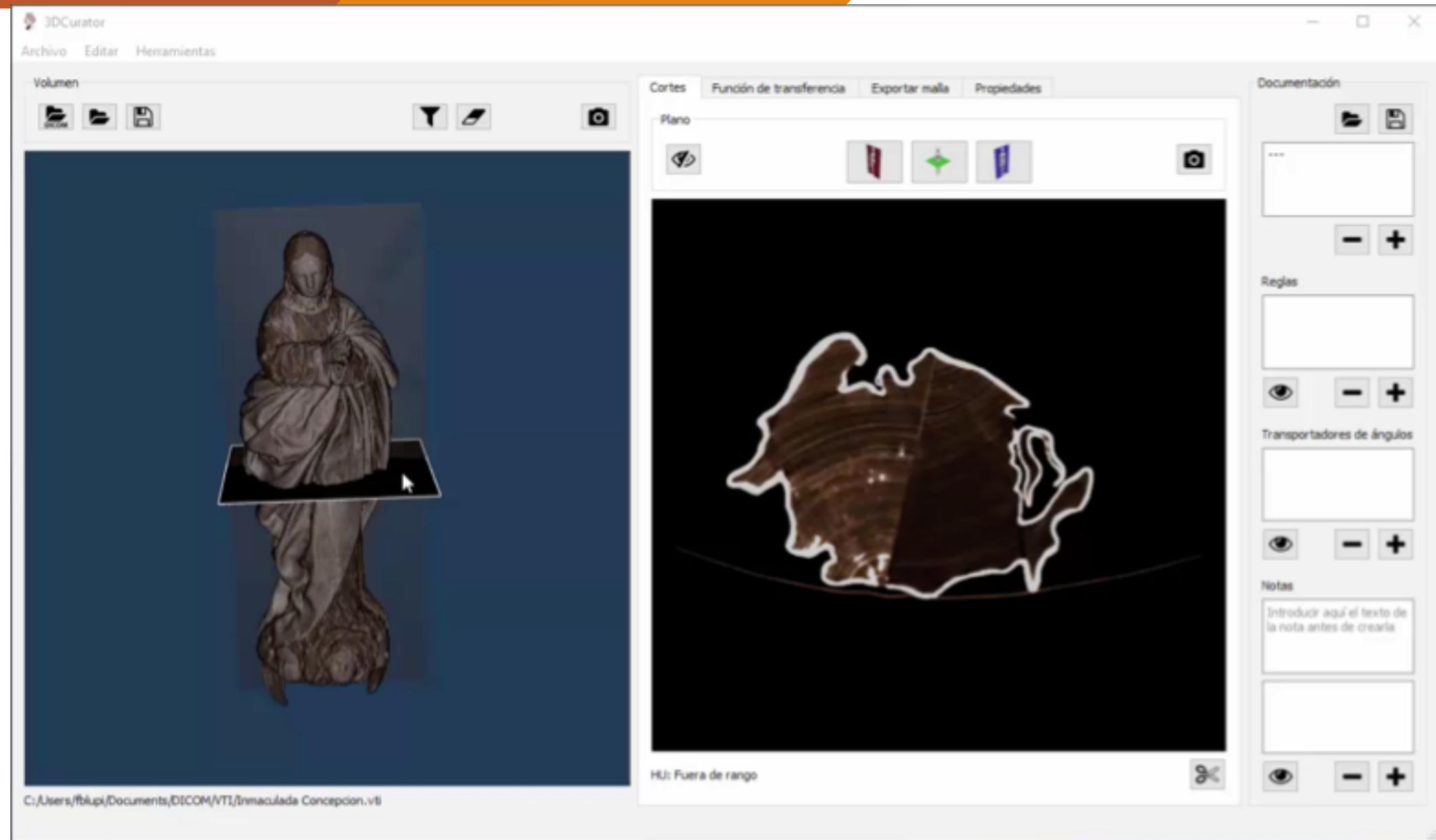
- Posición del plano
- Lista de reglas
- Lista de transportadores de ángulos
- Lista de notas



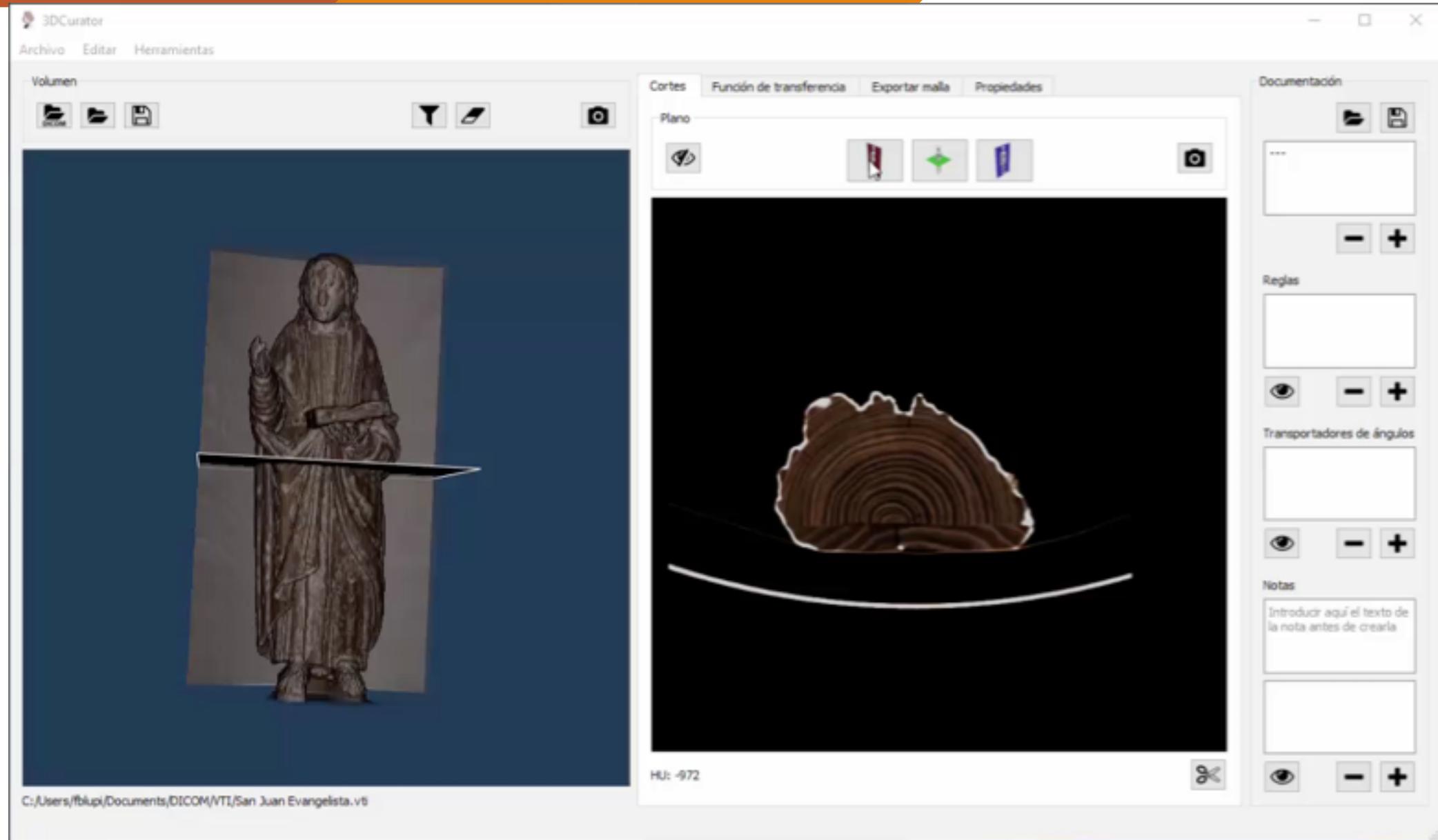
# DEMOSTRACIÓN ➤ FILTRADO



# DEMOSTRACIÓN ➤ SEGMENTACIÓN



# DEMOSTRACIÓN → DOCUMENTACIÓN

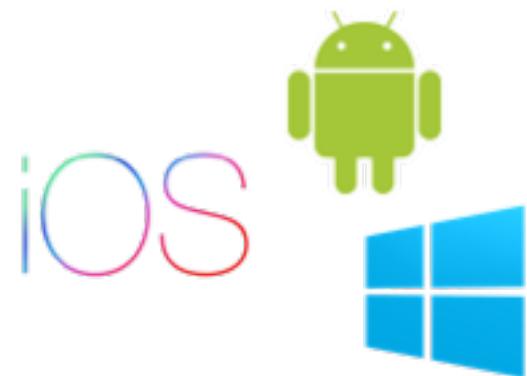
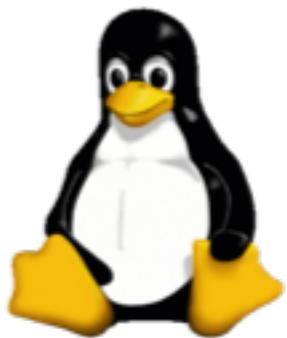


- Partíamos de un software que ya ayudaba a un restaurador o historiador del arte realizar un **estudio más exhaustivo del estado interno de una escultura**
- Se ha completado el proceso de **modelado y visualización de volúmenes** con el resto de pasos (filtrado y segmentación)
- Se ha implementado una serie de filtros que nos ayuda a **eliminar ruido** de los datos obtenidos con el escáner
- Se ha creado un método semiautomático bastante efectivo para **segmentar las piezas de madera** de una escultura
- Se ha implementado una serie de **herramientas para documentar** la escultura desde la propia aplicación
- Software validado por **restauradores profesionales**

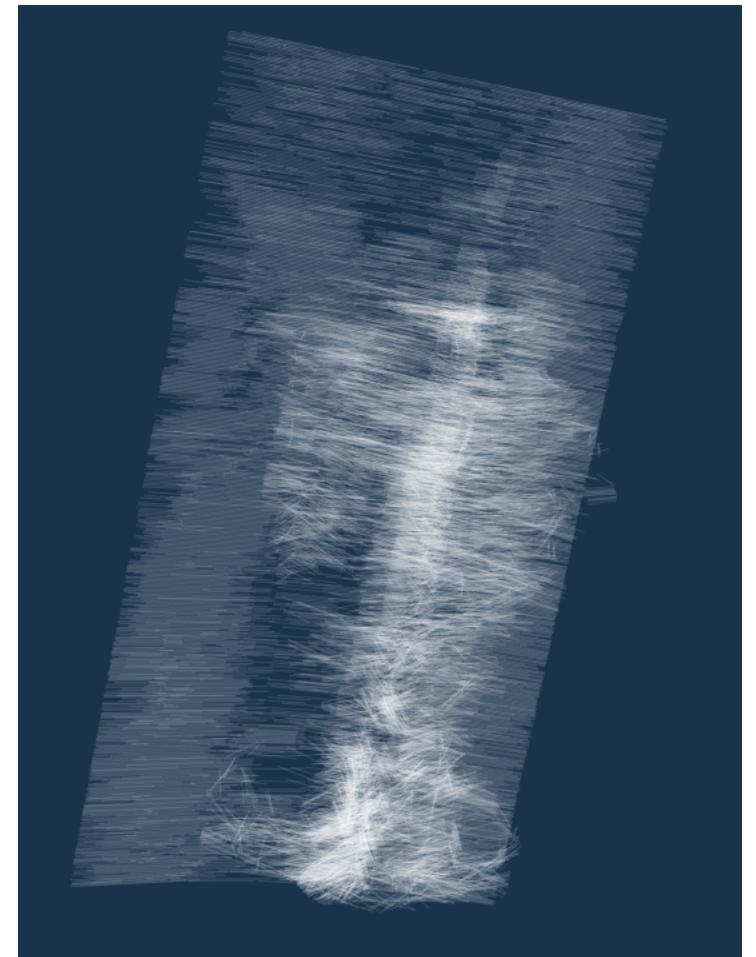


Compilar software existente en otras plataformas

Adaptación a dispositivos móviles



Segmentación 3D automática





**FIN**

---

GRACIAS POR SU ATENCIÓN