

# Verificación de Programas Java para el Procesamiento de Imágenes Digitales

Grado en Ingeniería Informática



UNIVERSIDAD DE LA RIOJA

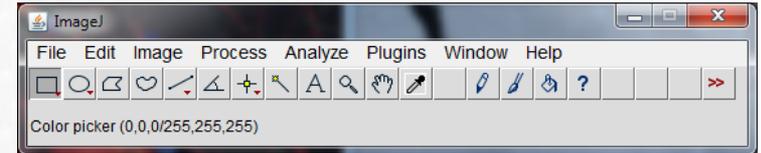
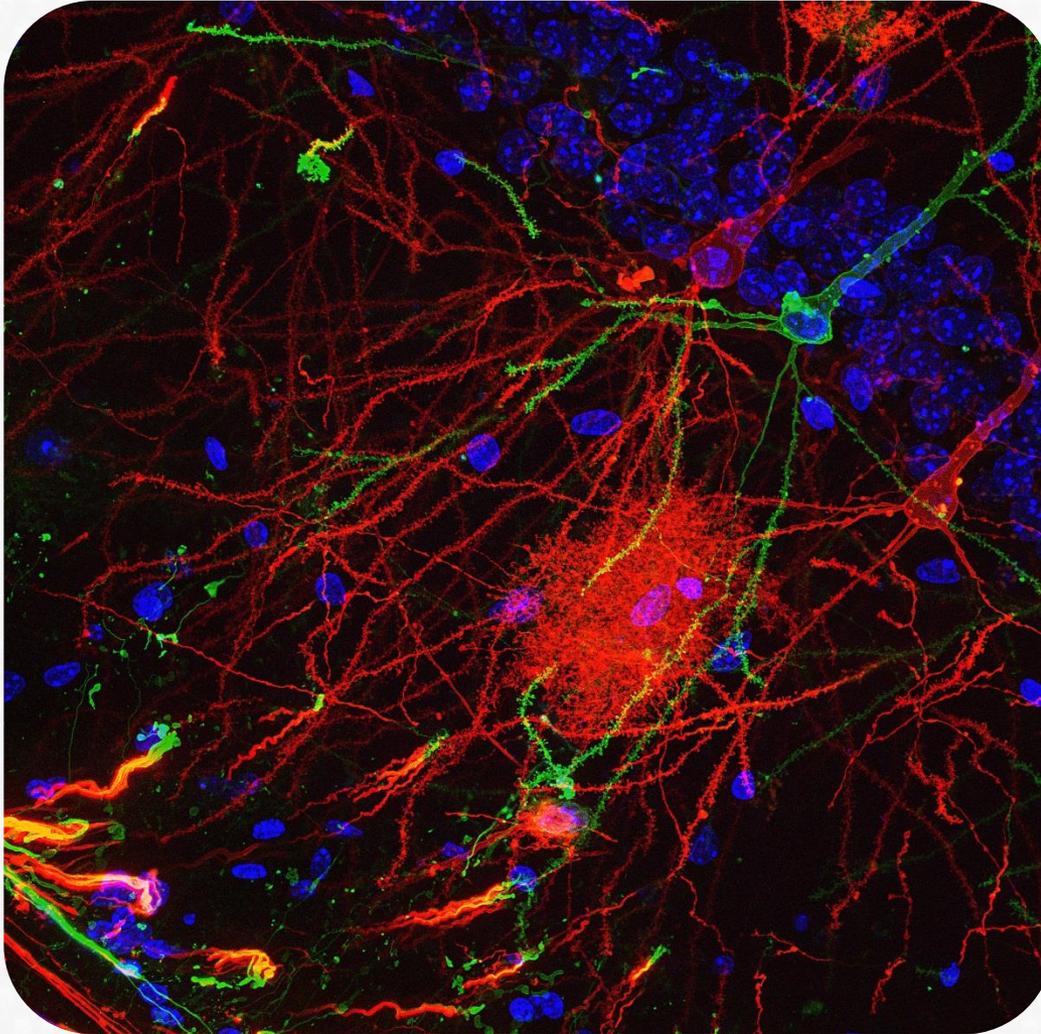
**Rubén Sáenz Francia**

**Tutor: Julio Rubio García**

# Índice

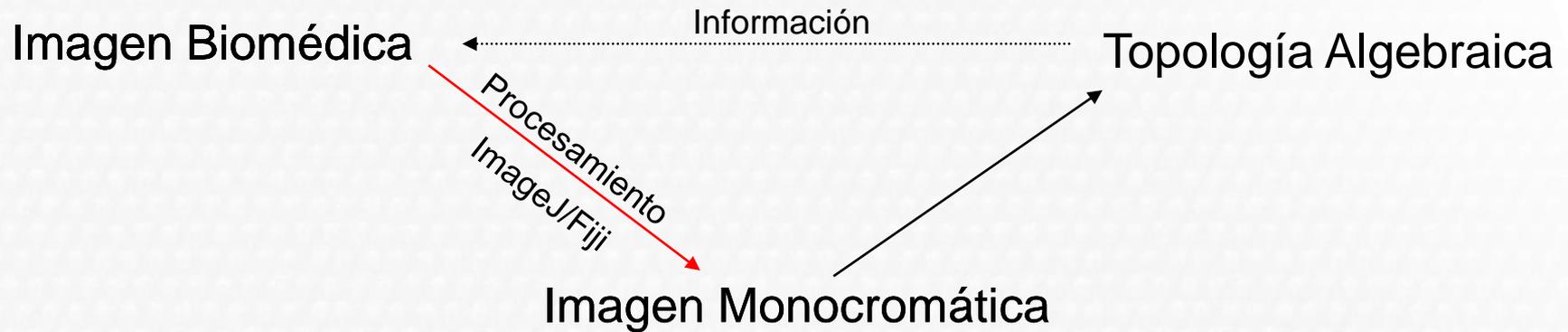
- 1. Imágenes Biomédicas**
- 2. Proyecto Europeo ForMath**
- 3. Fechas Importantes**
- 4. Krakatoa**
- 5. NeuronPersistentJ**
- 6. Automatización – Class Unifier**
- 7. Gestión**

# Imágenes Biomédicas



- ImageJ/Fiji
- Imágenes de neuronas

# Proyecto Europeo ForMath



**Formalización === Confiabilidad**

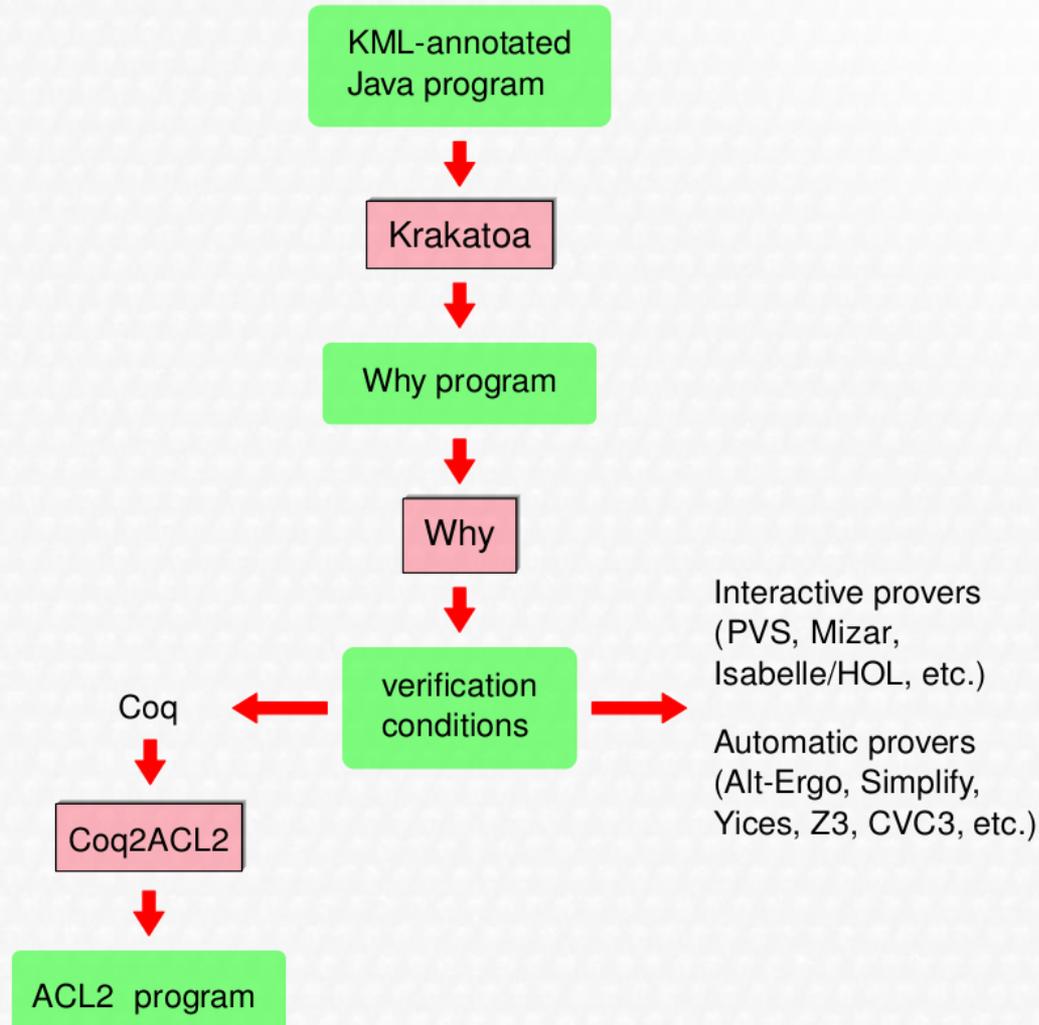
# Fechas Importantes



1. Formalización de parte  
de ImageJ/Fiji

2. Depurar NeuronPersistentJ  
3. Automatizar el proceso

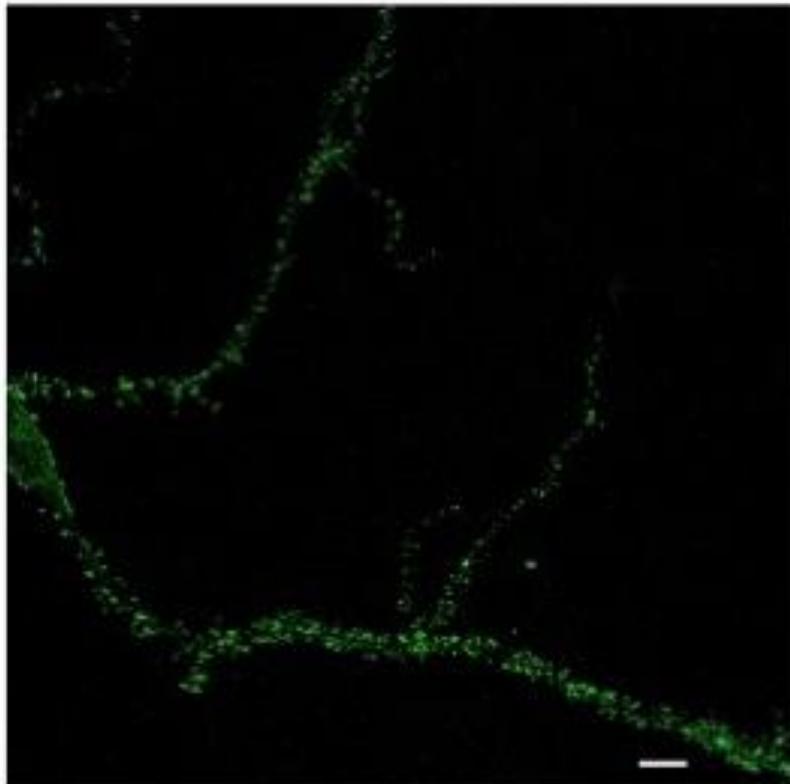
# Formalización del código Java - Krakatoa



# Formalización del código Java - Metodo

1. Código de ImageJ/Fiji → Código compilable por Krakatoa.
2. Especificar programas Java.
3. Aplicar la herramienta Why.
4. ¿Todas las pruebas están verificadas ?
  - Si, la verificación ha concluído.
  - No, Ir al paso 5.
5. ¿Los casos que no se han probado pueden ser mejor especificados?
  - Si, Ir a 2.
  - No, Ir a 6.
6. Exportar código Coq.
7. Código Coq → Código ACL2.
8. ¿Las pruebas han sido verificadas automáticamente en ACL2?
  - Si, La verificación ha concluído.
  - No, Son necesarias más especificaciones? Si, Ir a 2. No . . .

# NeuronPersistentJ



# ClassUnifier – Automatizar del proceso



**Eclipse Plugin**

↔ **ClassUnifier** ↔



**Java Parser**

## 7. Gestión. Tabla de horas estimadas/reales

	Horas estimadas	Horas reales	Desfase
Dirección	17 h	15 h	-2 h
Iniciación	63 h	71 h	+8 h
Formación	178 h	213 h	+35 h
Análisis	124 h	104 h	-20 h
Diseño	205 h	175 h	-30 h
Construcción	198 h	226 h	+28 h
Memoria	99 h	135 h	+36 h
Manuales	18 h	4 h	-14 h
Defensa	8 h	10 h	+2 h
<b>Total</b>	<b>910 h</b>	<b>953 h</b>	<b>+43 h</b>

## 7. Conclusiones

- La formalización es un proceso muy laborioso.
- La formalización requiere mucho trasfondo matemático.
- Es igual de difícil depurar que diseñar un programa.
- En general ImageJ/Fiji está bien programado.
- Aún conociendo perfectamente el comportamiento de los programas los invariantes son difíciles de descubrir y probar.
- Un programa no es correcto si no asegura su finitud y que las operaciones que se realizan dentro son seguras.

**Gracias por  
su  
asistencia**



**UNIVERSIDAD DE LA RIOJA**

Julio Rubio

Ana Romero

Jonathan Heras

Gadea Mata