

# Auxiliar N°1

## Repaso del curso



---

Auxiliar: Pablo Pizarro R. [@ppizarror](#)  
Estudiante MSc. Ingeniería Estructural  
Universidad de Chile

# Contenidos del curso

- Introducción geométrica
  - Bases del curso de geometría computacional CC5502-1
- Shaders
- Cálculo paralelo en GPU
  - OpenCL
  - CUDA (Nvidia) -> compilador de cuda (mingw, gcc)
    - CUDA CORES
    - RT CORES (RTX)

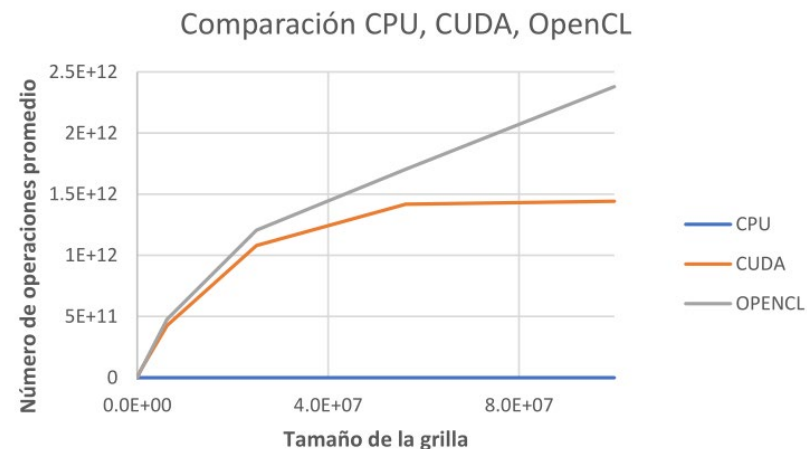
# Auxiliares

4/5 auxiliares en el semestre.

1. Introducción a C++, Clase (+=, \*=, “Friends”, herencia, .h, .cpp)
2. Shaders
  - Introducción
  - Dibujo de un cubo con shaders
  - Librerías (Three.js, Babylon, Etc)
3. OpenCL / CUDA
  - Descripción de los modelos y paradigmas de paralelización
  - Ejemplos

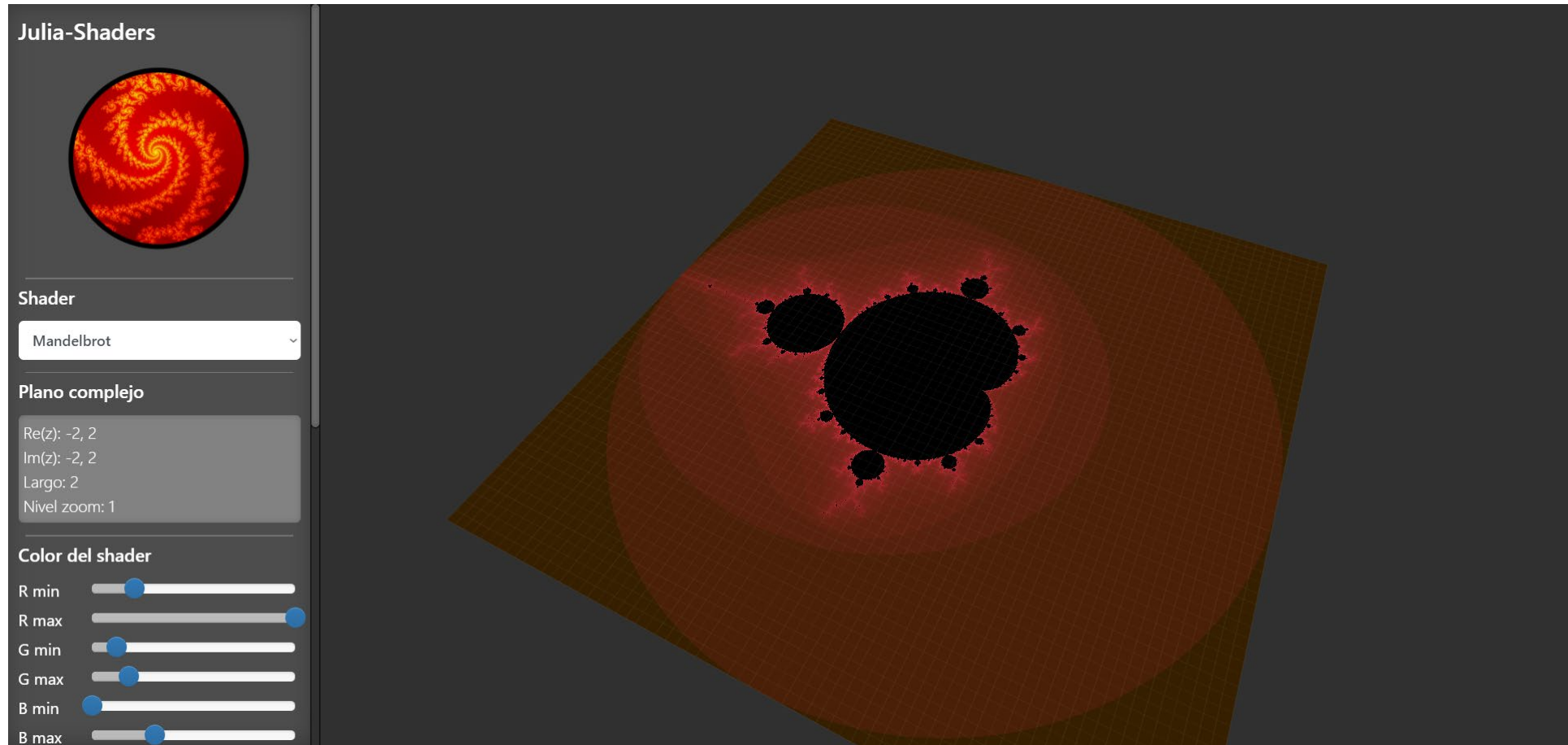
# Tareas

1. T1: C++, uso de clases
2. Shaders
  - Desarrollo de aplicación gráfica interactiva, uso de Python o herramientas web como Three.js
3. Cálculo y comparación C++ single threaded y multi-thread con OpenCL y CUDA



# Ejemplos – Tarea N°2 2018

## Visualizador Fractal



<https://github.ppizarror.com/julia-shaders-threejs/>

# Ejemplos – Tarea N°2 2019

## One Neuron Drawn with Maya Modeler as Object and loaded with vue-babylonjs

CC7515 GPU COMPUTING



<https://jthoth.github.io/neuronzoo/>

# Proyecto Semestral

- Trabajo sobre artículo, implementación y test de los resultados.
- Tendrán que aplicar distintas herramientas del curso, sobre todo implementación de los códigos para el calculo en paralelo con la GPU en CUDA o OpenCL.
- El tema puede ser un artículo, investigación personal, una rama de sus tesis, etc.
- Tendrán que hacer un video explicando sus trabajos. Éstos serán subidos a youtube.

# Proyecto Semestral

## Ejemplo – Plataforma de cálculo estructural en C++ con resolución matricial en CUDA (2018)



Computers and Structures

journal homepage: [www.elsevier.com/locate/compstruc](http://www.elsevier.com/locate/compstruc)



Technical Note

### A fast parallel Gauss Jordan algorithm for

Girish Sharma<sup>a</sup>, Abhishek Agarwala<sup>b</sup>, Baidurya Bhat

<sup>a</sup> Department of Civil Engineering, Indian Institute of Technology, Kharagpur 721302, India

<sup>b</sup> Archayne Labs, Gurgaon 122001, India

#### ARTICLE INFO

Article history:

Received 5 April 2012

#### ABSTRACT

The ability to invert large r  
tional tool. Current literatur

<https://github.com/ppizarror/FNELEM-GPU>

README.md

## FNELEM-GPU

Simple finite elements matrix structural analysis using GPU. Actually supports 2D planar membrane/shell.

### Requirements

C++ STANDARD 14 and CUDA. No other libraries are needed.

### Import

To fully use this library, you must include the following files:

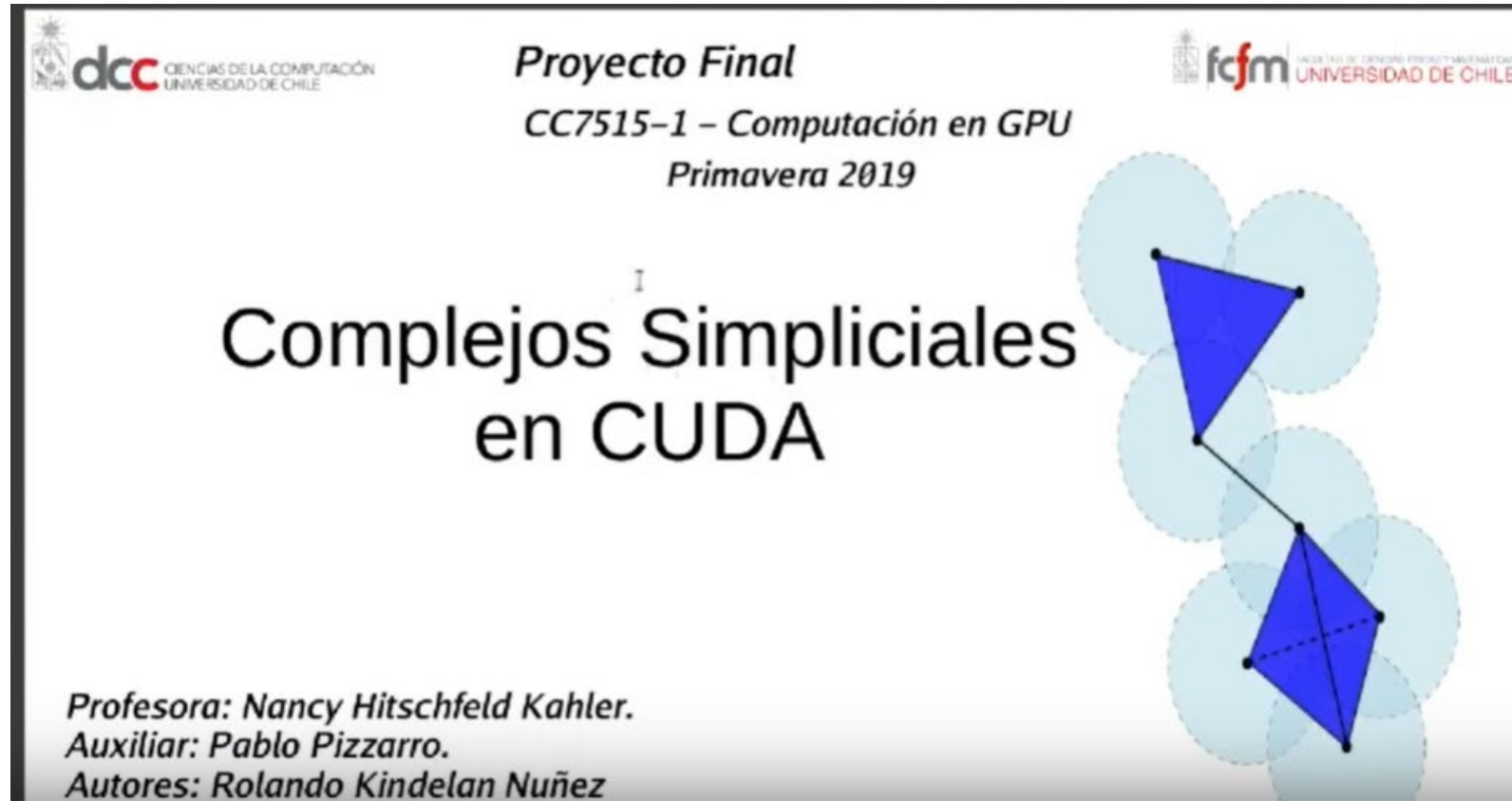
```
#include "fnelem/math/fematrix.cpp"
#include "fnelem/math/fematrix_utils.cpp"
#include "fnelem/math/matrix_inversion_cpu.cpp"
#include "fnelem/math/matrix_inversion_cuda.cu"

#include "fnelem/analysis/static_analysis.cpp"
```



# Proyecto Semestral

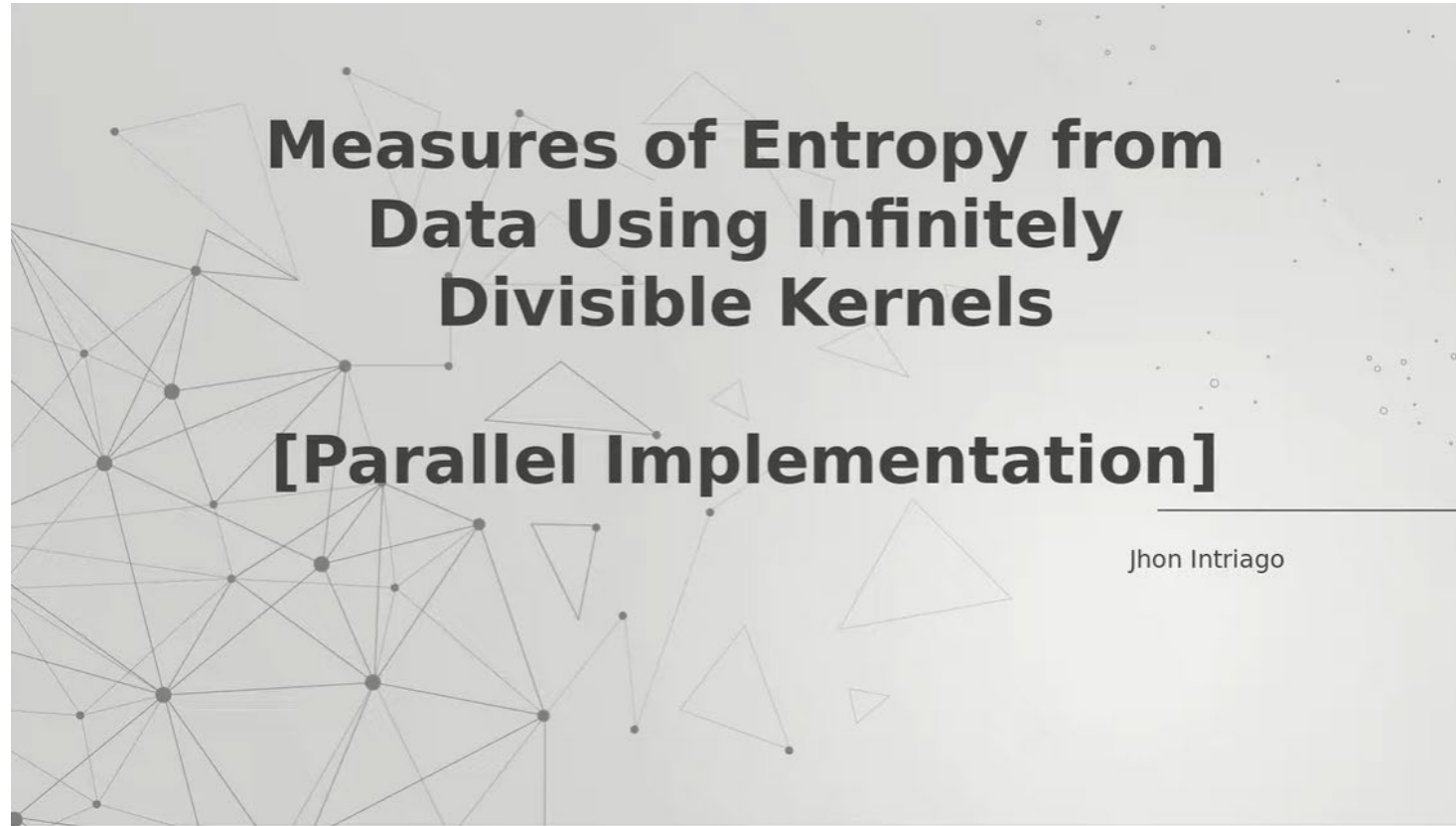
## Ejemplo – Complejos Simpliciales en CUDA (2019)



<https://www.youtube.com/watch?v=6DYsESb7kXE>

# Proyecto Semestral

## Ejemplo – Measures of Entropy from Data Using Infinitely Divisible Kernels(2019)



<https://www.youtube.com/watch?v=FMytRQy38nU>

# Herramientas a usar en el curso

- C++ Editor de código
  - CLion <https://www.jetbrains.com/es-es/clion/>
  - Visual Studio <https://visualstudio.microsoft.com/es/vs/>
- CUDA / OpenCL
  - OpenCL viene instalado en cada computador. Aunque requieren del compilador y los headers. (opencl.h)
  - Para CUDA, requieren de una tarjeta gráfica NVIDIA compatible con la librería.  
<https://developer.nvidia.com/cuda-downloads>