

# GRAPHIC CARDS AND SCIENTIFIC COMPUTATION

Desarrollo con Tecnologías Emergentes – TG3

2018-2019

- Adrián López Godoy
- Daniel Stanus
- Ignacio Burgos Lucha
- Borja Ordóñez
- Javier Luque

# Contenido

1. Autores del trabajo, planificación y entrega.....	3
1.1 Autores.....	3
1.2 Planificación .....	3
1.3 Entrega .....	3
2. Requisitos del prototipo a implementar .....	4
2.1 Requisitos funcionales.....	5
2.2 Otros requisitos .....	5
3. Criterios de comparación en la implementación.....	6
3.1 Criterio 1: Requisitos de Hardware .....	6
3.2 Criterio 2: Facilidad de implementación.....	6
3.3 Criterio 3: Rendimiento Test Luxmark .....	6
3.4 Criterio 4: Rendimiento Test Compubench .....	6
3.5 Criterio 5: Prueba de estabilidad con el test MSI Kombustor .....	6
3.6 Criterio 6: Software científico .....	7
3.7 Criterio 7: Puntuación Software Científico .....	7
3.8 Criterio 8: Software específico de testeo. ....	7
3.9 Criterio 9: Multisistema.....	7
3.10 Criterio 10: Escalabilidad.....	7
4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología CUDA .....	8
4.1 Documentación de diseño .....	8
4.2 Documentación de construcción.....	8
4.3 Documentación de pruebas.....	9
4.4 Documentación de instalación.....	19
La utilidad GPUCaps nos confirma que estamos utilizando CUDA:.....	24
4.5 Manual de usuario.....	25
5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología OpenCL .....	45
5.1 Documentación de diseño .....	45
5.2 Documentación de construcción.....	45
5.3 Documentación de pruebas.....	46
5.4 Documentación de instalación.....	56
5.5 Manual de usuario.....	66
6. Comparación de las dos implementaciones .....	87
6.1 Evaluación de los criterios en la implementación usando la tecnología CUDA ..	94
6.2 Evaluación de los criterios en la implementación usando la tecnología OpenCL	95

7. Comparación de la implementación de las tecnologías.....	96
8. Conclusiones .....	98

# **1. Autores del trabajo, planificación y entrega**

## **1.1 Autores**

El presente documento ha sido elaborado por:

- Daniel Stanus
- Borja Ordóñez
- Javier Luque
- Ignacio Burgos
- Adrián López

## **1.2 Planificación**

Nuestra planificación ha sido desarrollada íntegra de forma online, y viene detallada en el siguiente enlace:

<https://app.teamweek.com/#pp/gUFolydeHretNiSvYDpQmVW-q9IDbtZn>

## **1.3 Entrega**

El enlace de Github donde se encuentra el presente documento es el siguiente:

<https://github.com/algoy/DTE/blob/master/TG3T3/>

## 2. Requisitos del prototipo a implementar

En este trabajo, vamos a comparar el diferente funcionamiento de las tecnologías de computación gráfica OpenCL y CUDA, operando cada una de ella en diferentes tarjetas gráficas de rango equivalente.

CUDA es una arquitectura de cálculo en paralelo de NVIDIA que aprovecha la potencia de la GPU para proporcionar un incremento en el rendimiento del sistema. Esta potencia se utiliza para la computación en paralelo. CUDA también incluye un compilador y herramientas de desarrollo creadas por NVIDIA que ayudan a los programadores a usar diferentes lenguajes de programación como por ejemplo C, C++, Python y Java.

Esta tecnología funciona con todas las tarjetas de NVIDIA a partir de la serie G8X incluyendo las familias GeForce, Quadro, ION y Tesla.

CUDA intenta explotar las ventajas de las GPU frente a las CPU de propósito general utilizando el paralelismo que ofrecen sus múltiples núcleos, que permiten el lanzamiento de un altísimo número de hilos simultáneos. Por ello, si una aplicación está diseñada utilizando numerosos hilos que realizan tareas independientes (que es lo que hacen las GPU al procesar gráficos).

Junto a una interfaz de programación y de un lenguaje (C) permite crear aplicaciones con paralelismo a nivel de datos y tareas que pueden ejecutarse tanto en unidades de procesamiento (CPU) como en unidades de procesamiento gráfico (GPU).

OpenCL provee del API que permite a los programas acceder a múltiples procesadores de forma simultánea para lograr el procesamiento en paralelo. Al distribuir el procesamiento entre múltiples procesadores mejora el rendimiento del programa.

Para que OpenCL funcione, el hardware y el software deben de soportar la API. Actualmente las tarjetas gráficas de AMD, NVIDIA e INTEL soportan este estándar.

Para poder llevar a cabo este proyecto, nos hemos dividido en dos equipos; un equipo A, que trabajará con un portátil Asus con una gráfica **Nvidia GT630m de 2GB**, con el fin de implementar el prototipo en un ordenador compatible con CUDA, y un Packard Bell Easynote TJ76 que será compatible con OpenCL al emplear una gráfica **AMD HD5470 de 512MB**.

## 2.1 Requisitos funcionales

Los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

REQ.	DESCRIPCIÓN
RF01	El sistema deberá permanecer estable bajo condiciones de carga
RF02	El sistema deberá ser capaz de ejecutar Luxmark
RF03	El sistema deberá pasar las pruebas obteniendo una puntuación
RF04	El sistema deberá ser capaz de ejecutar CompuBench
RF05	El sistema deberá pasar las pruebas obteniendo una puntuación
RF06	El sistema deberá ejecutar MSI Kombustor durante al menos 5 minutos, con temperaturas adecuadas al TDP de cada gráfica.
RF07	El sistema deberá ser capaz de unirse al menos a un proyecto científico (SETI@home)
RF08	El sistema deberá ser capaz de ejecutar las tareas encomendadas en al menos un proyecto científico (SETI@home)
RF09	El sistema OpenCL deberá ser capaz de ejecutar una herramienta de benchmark exclusiva (Julia4D OpenCL)
RF10	El sistema CUDA deberá ser capaz de ejecutar una herramienta de benchmark exclusiva (herramienta)
RF11	El sistema debe ser funcional y operacionalmente escalable

## 2.2 Otros requisitos

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

REQ.	DESCRIPCIÓN
R01	El sistema deberá tener bien instalado el hardware
R02	El sistema deberá cumplir los requisitos mínimos de gráfica
R03	El sistema deberá cumplir los requisitos mínimos de RAM
R04	El sistema deberá tener instalado sistema operativo adecuado y SDK/Herramientas adecuadas para la tecnología
R05	Los drivers gráficos deben estar correctamente instalados
R06	Los drivers de CUDA/OpenCL deberán estar operativos

### **3. Criterios de comparación en la implementación**

En el presente trabajo, hay que definir criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

#### **3.1 Criterio 1: Requisitos de Hardware**

Nombre del criterio: Requisitos hardware

Descripción: Condiciones que debe tener el sistema para operar con el hardware. Cuanto menos requiera, mejor.

Tipo de valor: Texto libre.

#### **3.2 Criterio 2: Facilidad de implementación**

Nombre del criterio: Implantación.

Descripción: Complejidad para instalar el sistema.

Tipo de valor: Texto libre (Alta, Media, Baja).

#### **3.3 Criterio 3: Rendimiento Test Luxmark**

Nombre del criterio: Rendimiento Test Luxmark.

Descripción: Capacidad del sistema para responder ante el test Luxmark y dar un resultado superior al otro.

Tipo de valor: Numérico.

#### **3.4 Criterio 4: Rendimiento Test Compubench**

Nombre del criterio: Rendimiento Test Compubench.

Descripción: Capacidad del sistema para responder ante la suite de test Compubench y dar un resultado superior al otro, o pasar mayor cantidad de pruebas.

Tipo de valor: Numérico.

#### **3.5 Criterio 5: Prueba de estabilidad con el test MSI Kombustor**

Nombre del criterio: Prueba de estabilidad con el test MSI Kombustor.

Descripción: Prueba con MSI Kombustor. Software que somete al equipo a una prueba donde fuerza CPU y GPU al 100%, viendo si se bloquea o no.

Tipo de valor: Booleano (Si/No).

### **3.6 Criterio 6: Software científico**

Nombre del criterio: Software científico.

Descripción: Indica si la gráfica CUDA y OpenCL son capaces de ejecutar el proyecto científico (de BOINC) correctamente.

Tipo de valor: Booleano (Si/No).

### **3.7 Criterio 7: Puntuación Software Científico**

Nombre del criterio: Puntuación Software Científico.

Descripción: Indica cuál ha sido el resultado de ambas gráficas, y se compara.

Tipo de valor: Numérico.

### **3.8 Criterio 8: Software específico de testeo.**

Nombre del criterio: Software específico de testeo.

Descripción: Indica si el prototipo es capaz de enfrentarse a pruebas específicas para una gráfica de su tipo (software para CUDA u OpenCL exclusivo).

Tipo de valor: Booleano (Si/No).

### **3.9 Criterio 9: Multisistema.**

Nombre del criterio: Multisistema Windows/Linux/Mac.

Descripción: Indica si la tecnología es capaz de ejecutarse en los diferentes sistemas operativos comerciales más usados (Windows/Linux/Mac/Unix).

Tipo de valor: Booleano (Si/No).

### **3.10 Criterio 10: Escalabilidad.**

Nombre del criterio: Soporte de sistema escalable.

Descripción: Indica si la tecnología es capaz de adaptarse siendo escalable, o no.

Tipo de valor: Booleano (Si/No).

## **4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología CUDA**

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

### **4.1 Documentación de diseño**

El prototipo consistirá en dos ordenadores portátiles, equipados con tarjetas gráficas y procesadores capaces de ejecutar CUDA y OpenCL para cálculos científicos.

Ambos contarán además con el software necesario para poder realizar las pruebas pertinentes para poder comprobar cuál de las dos tecnologías podría considerarse más eficiente o competitiva, según varios criterios, en dos equipos con un nivel de hardware gráfico similar (de igual gama).

El prototipo no podrá ser mostrado en máquinas en caso de carecer de las instrucciones necesarias.

### **4.2 Documentación de construcción**

Para construir el prototipo, tenemos que diseñar, por un lado, hardware totalmente compatible con lo que queremos construir, y diseñar un ecosistema software que también lo sea.

Intentando conseguir un resultado lo más preciso, ambos equipos deberán estar lo más equilibrado en condiciones de sistema, HDD, Ram y procesador.

CAPA SOFTWARE	<ul style="list-style-type: none"><li>• Drivers</li><li>• SDK</li><li>• CUDA driver</li><li>• IDE</li><li>• Testing Tools</li><li>• Scientific Software</li></ul>
CAPA SISTEMA OPERATIVO	<ul style="list-style-type: none"><li>• Sistema (Windows/Mac/Linux)</li><li>• Actualizaciones</li></ul>
CAPA HARDWARE	<ul style="list-style-type: none"><li>• Procesador &gt; SSE2</li><li>• Ram &gt; 4Gb</li><li>• Gráfica compatible Nvidia</li></ul>

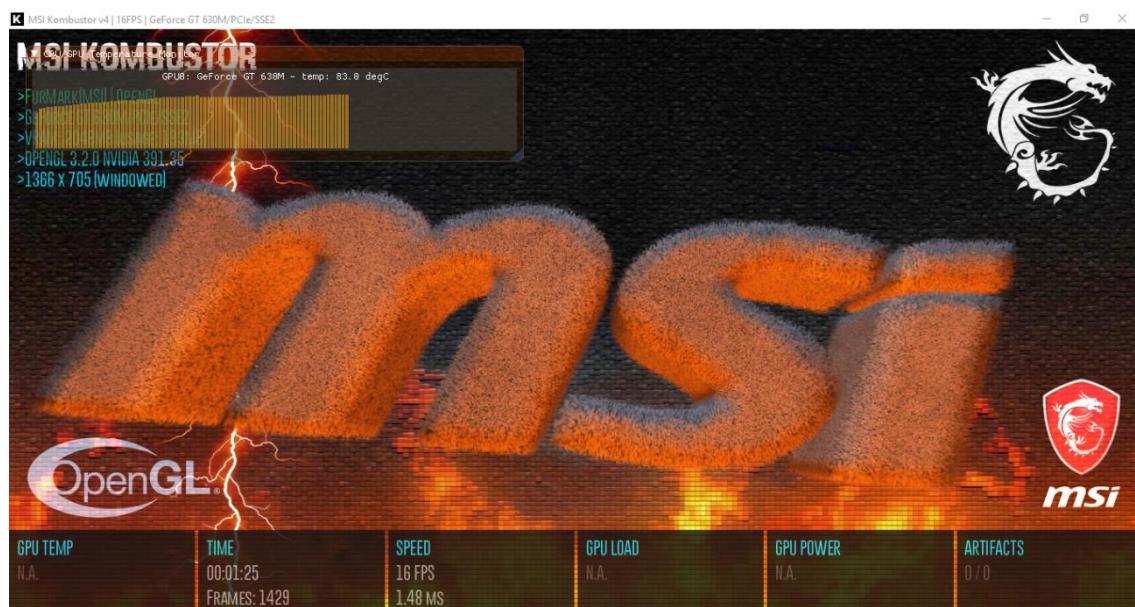
#### 4.3 Documentación de pruebas

Vamos a comprobar el correcto funcionamiento de las tarjetas gráficas Nvidia y la tecnología CUDA, usando dos tipos de aplicaciones.

Por una parte, usaremos software de testeo para ver si se está realmente usando la potencia de la gráfica, y por otra, emplearemos software científico real, vinculado a el proyecto BOINC, que busca emplear la potencia de cómputo sobrante de los equipos para proyectos científicos. Es un modelo distribuido, en el que tanto supercomputadores como equipos corrientes ponen a disposición de los centros investigadores sus recursos.

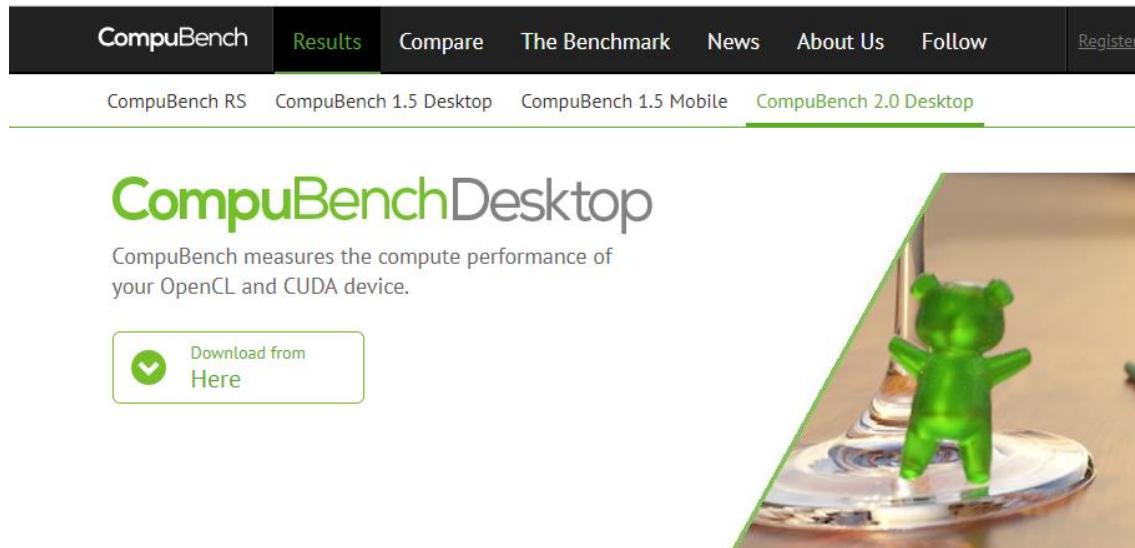
##### ----- Pruebas Testing -----

-Vamos a comenzar empleando el software MSI Kombustor. Nos dará una cifra aproximada en FPS de rendimiento en juegos, pero más importante, nos permitirá poner al máximo los dos equipos.



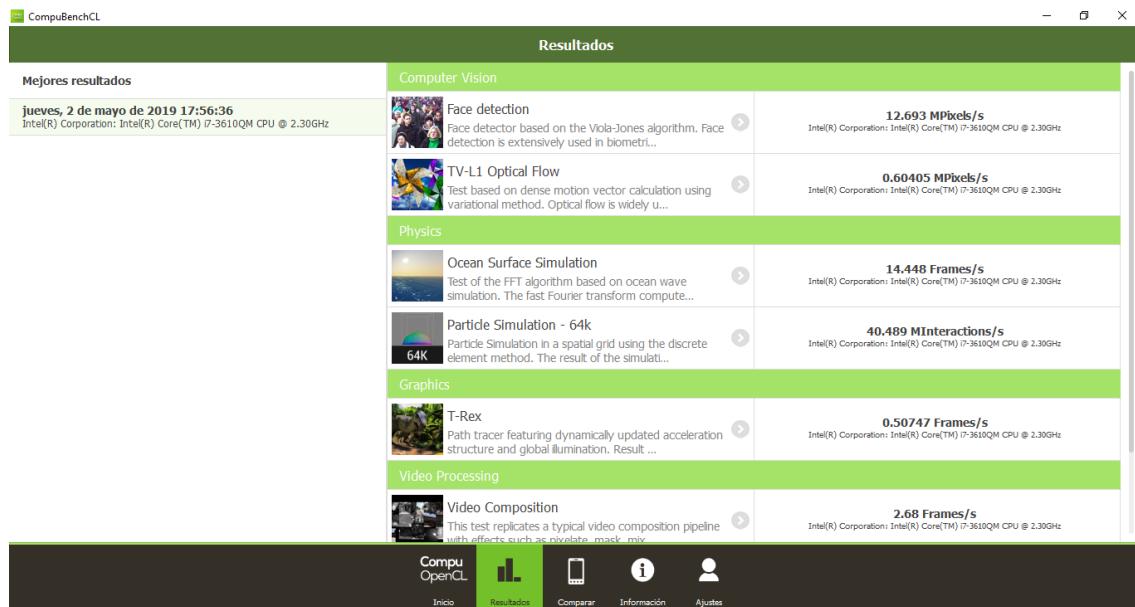
Vemos que, no se producen bloqueos en el sistema y la temperatura no es excesivamente alta. El sistema pasa la prueba de estabilidad.

El segundo programa que vamos a usar es CompuBench Desktop, que mide el rendimiento de dispositivos OpenCL y CUDA, realizando las mismas pruebas en ambos equipos. Observamos los resultados que nos ha dado empleando CUDA:



The screenshot shows the CompuBench Desktop website. The top navigation bar includes links for CompuBench, Results (highlighted in green), Compare, The Benchmark, News, About Us, Follow, and Register. Below the navigation are links for CompuBench RS, CompuBench 1.5 Desktop, CompuBench 1.5 Mobile, and CompuBench 2.0 Desktop (also highlighted in green). The main content area features a large green title 'CompuBenchDesktop'. Below it, a subtext reads 'CompuBench measures the compute performance of your OpenCL and CUDA device.' A green button with a download icon and the text 'Download from Here' is visible. To the right, there's a photograph of a small green teddy bear figurine standing on a reflective surface.

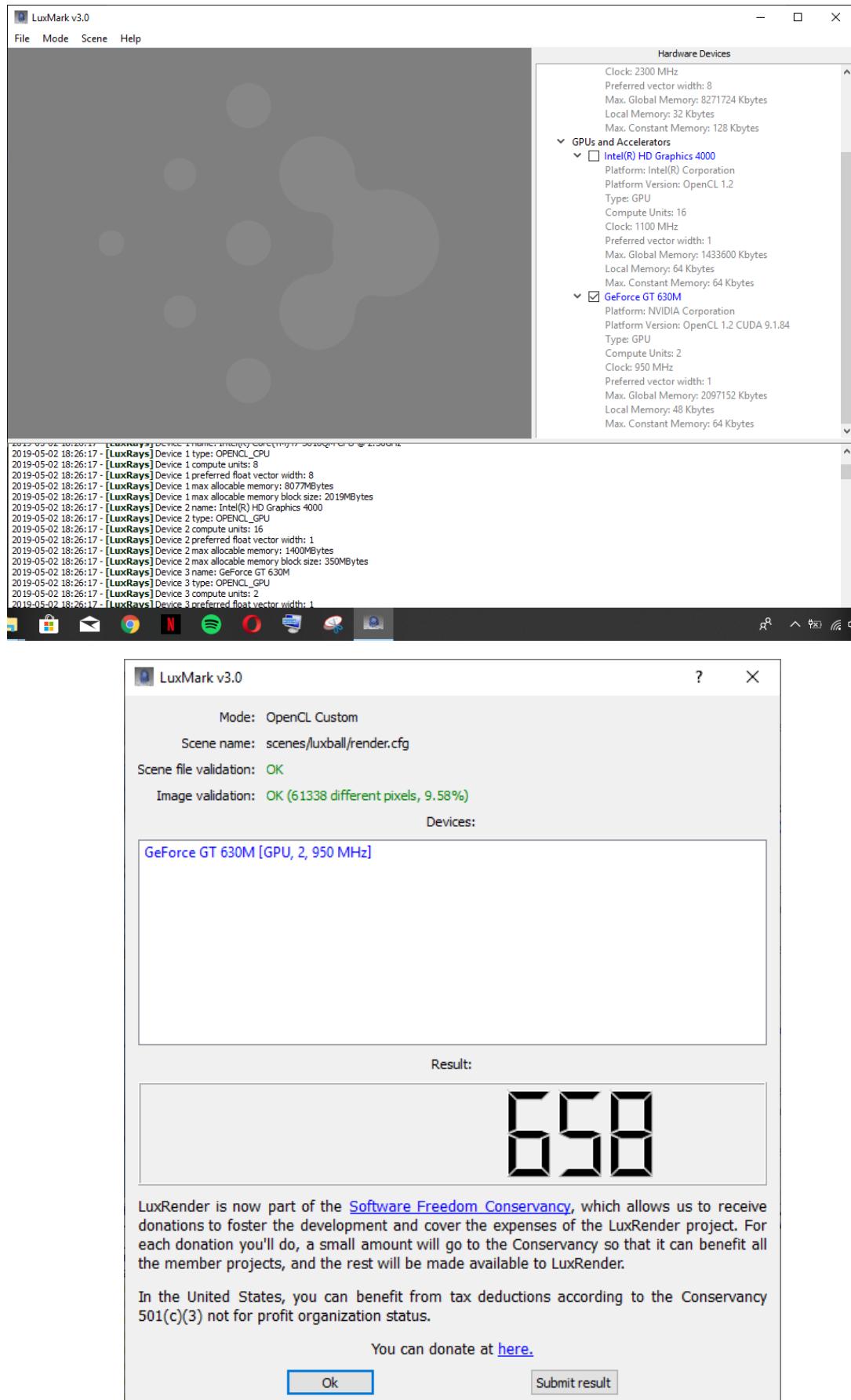
Tras instalarlo y ejecutarlo, vemos el siguiente resultado: todas las pruebas se han ejecutado satisfactoriamente, y hemos obtenido un resultado que podremos usar como base para comparar posteriormente con nuestra gráfica AMD.



The screenshot shows the CompuBenchCL application window titled 'Resultados'. It displays the 'Mejores resultados' for a run on 'jueves, 2 de mayo de 2019 17:56:36' using an Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz. The results are categorized into Computer Vision, Physics, Graphics, and Video Processing. Each category lists a test, its description, and its performance metric. At the bottom, there are links for 'Compu OpenCL', 'Inicio', 'Resultados' (highlighted in green), 'Comparar', 'Información', and 'Ajustes'.

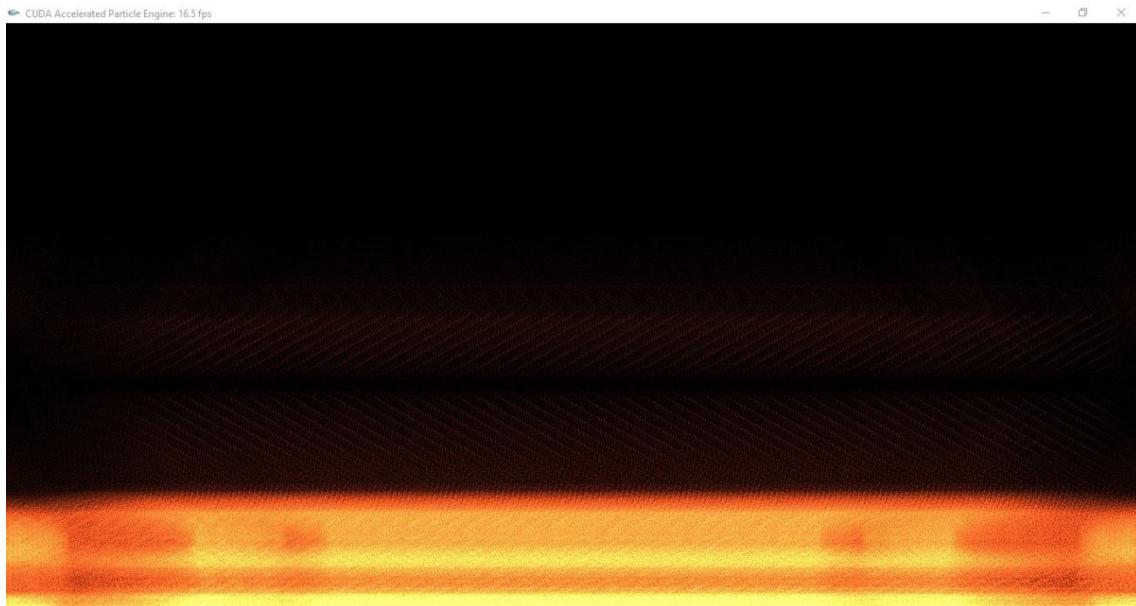
Categoría	Prueba	Métrica	Configuración
Computer Vision	Face detection	12.693 MPixels/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
	TV-L1 Optical Flow	0.60405 MPixels/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
Physics	Ocean Surface Simulation	14.448 Frames/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
	Particle Simulation - 64k	40.489 MInteractions/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
Graphics	T-Rex	0.50747 Frames/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
	Video Composition	2.68 Frames/s	Intel(R) Corporation: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz

Vamos a proceder a ejecutar el software de benchmarking **Luxmark**:



Y obtenemos un resultado numérico, que especifica la puntuación obtenida por el sistema gráfico usando CUDA, como vimos en la captura anterior. **Se ejecuta sin anomalías, prueba pasada** dando un resultado numérico y usando la tecnología CUDA.

Vamos a proceder a realizar la prueba específica para gráficas CUDA (Nvidia), en nuestro caso usando el software :

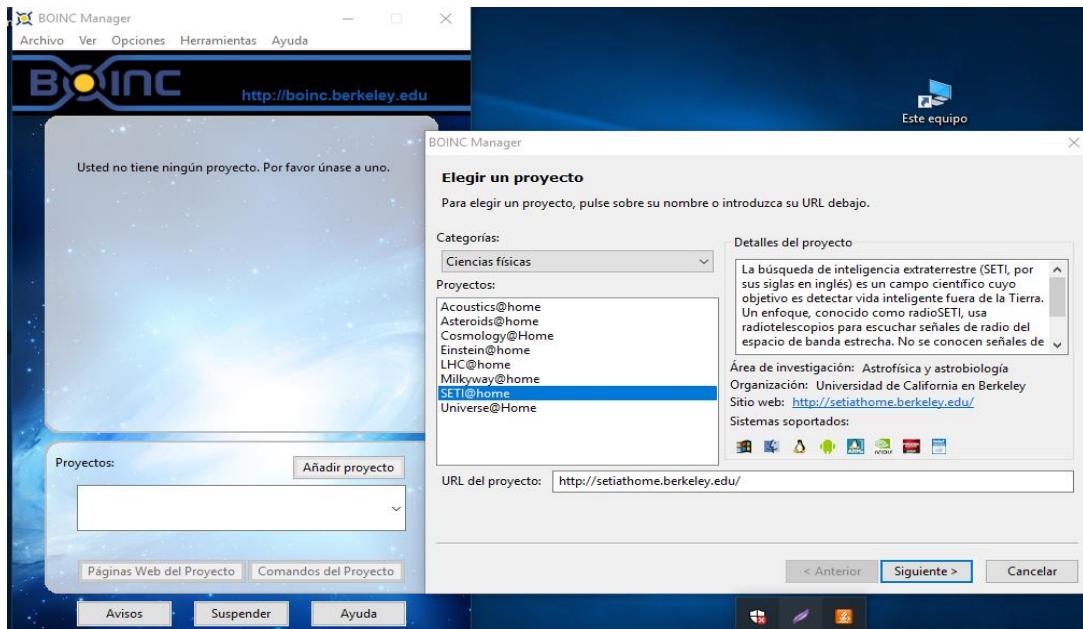


Como vemos, es capaz de ejecutarlo con unos 16 FPS sostenidos. Se ejecuta correctamente, superando la prueba.

El software usado es CUDA Accelerated Particle Engine, que busca simular la creación de 20 millones de partículas.

## ----- Pruebas científicas -----

Ahora, vamos a hacer las pruebas en el proyecto seti@home. Vamos a adjuntar pantallazos también de pruebas que hicimos intentando unirnos a los proyectos GPUGRID y Denis.

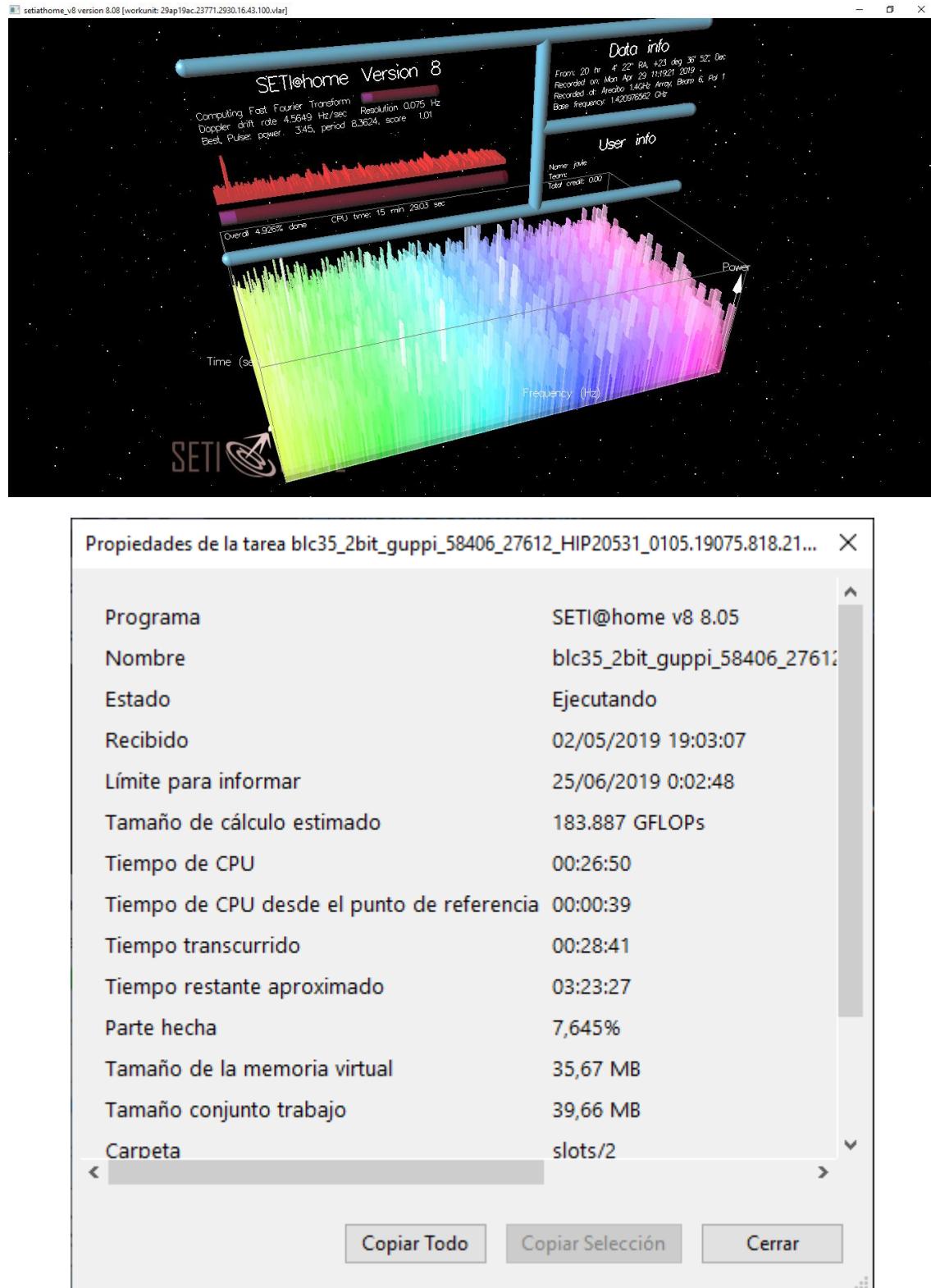




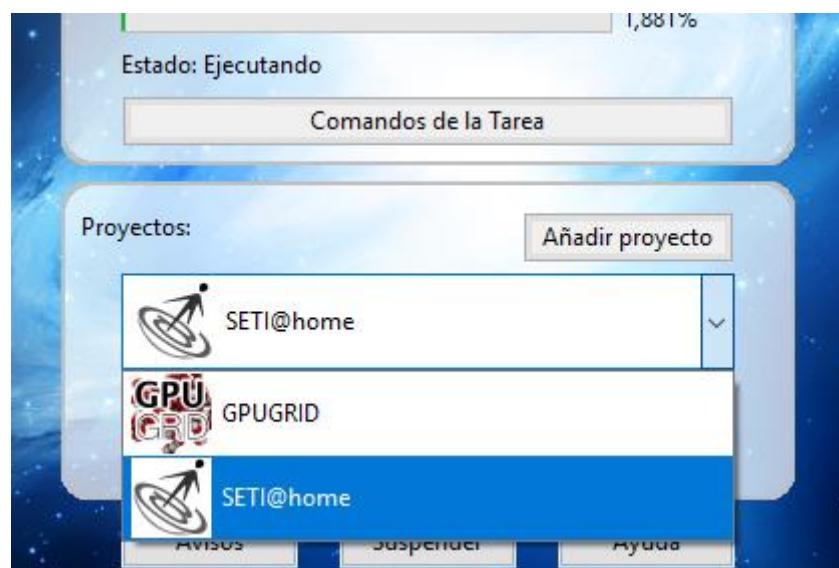
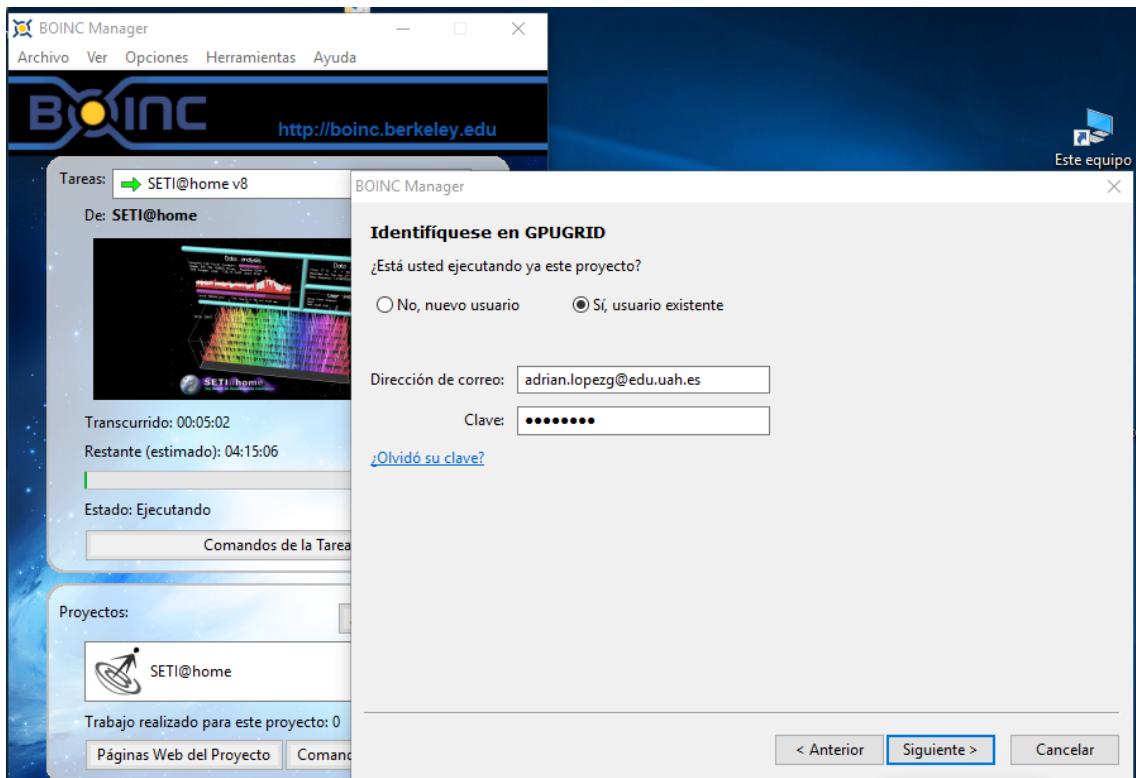
Seleccionamos el proyecto a unirnos. Nos deja sin problema. Intentamos ejecutar carga de trabajo en nuestro ordenador: Ejecuta un test de rendimiento. Vemos que lo pasa, e inicia el test, prueba superada:



Además, observamos que está usando la tecnología OpenCL usando la gráfica también:



**El resultado es satisfactorio** usando CUDA. Sin embargo, no con todos los programas fue así. Con gpugrid@home y con Denis@home no pudimos conectarnos, esto es debido a que eran proyectos obsoletos.



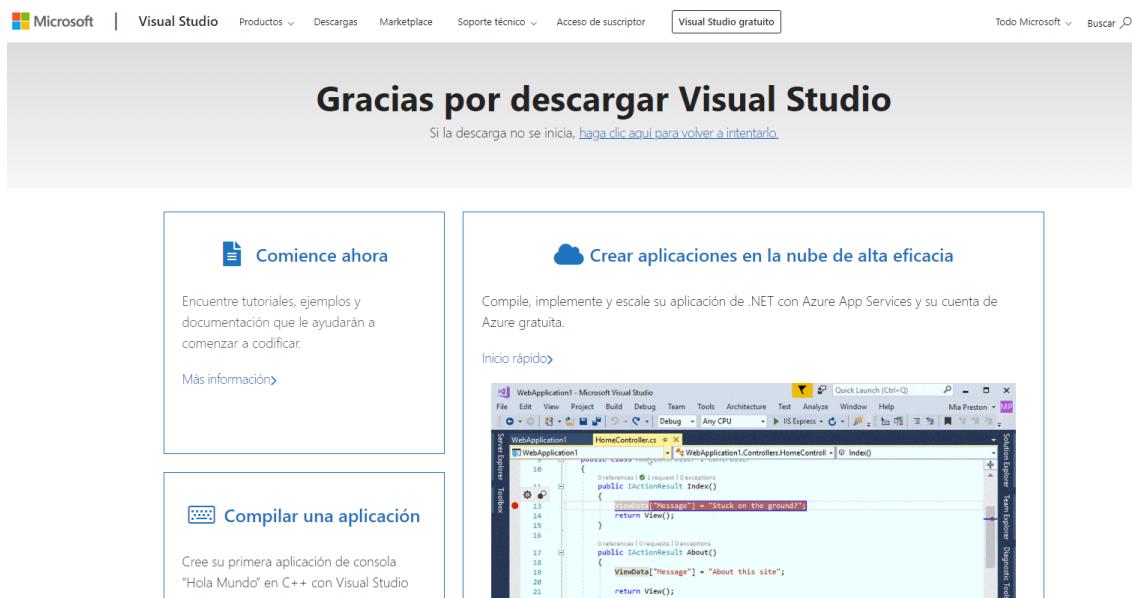
El error consistía en que nunca llegaban lotes de trabajo de los dos anteriores proyectos, siempre llegaban de Seti@home. Ello es porque ya no hay forma de colaborar, son proyectos que datan de principios del 2000 y dependen de universidades públicas. Sin embargo, el proyecto SETI tiene detrás a la Universidad de Berkeley, en USA, y de momento se prevé que sigan aportando apoyo.

**Nota: en ambas pruebas solo cambian capturas con drivers y tecnología usada, así como conclusiones. En el apartado 5.4, pues, se repetirán bastantes cosas.**

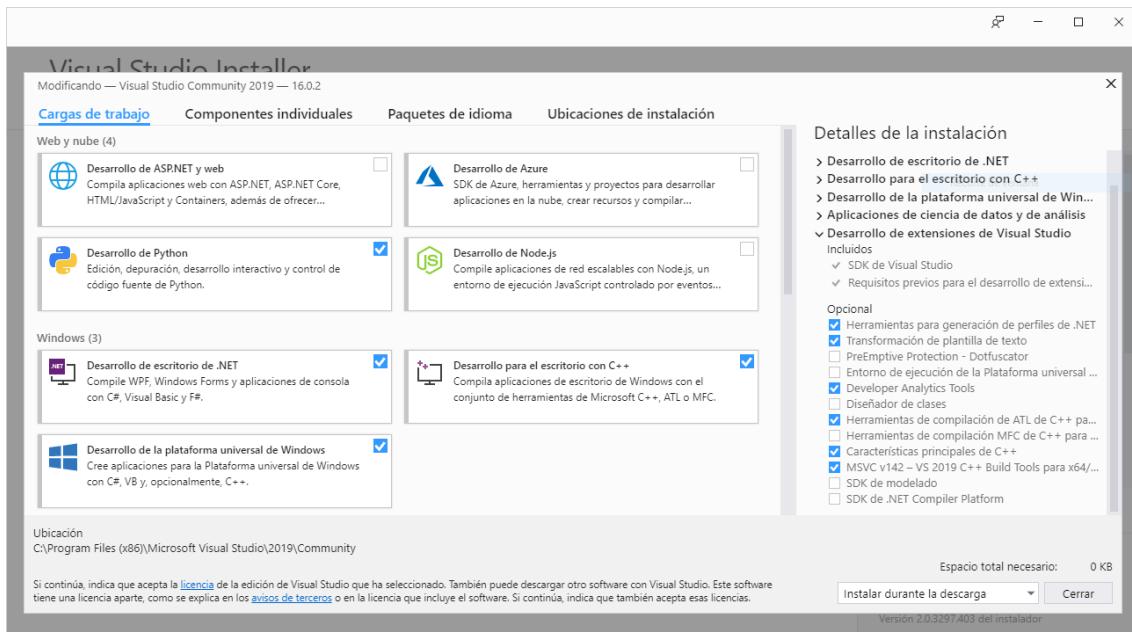
## 4.4 Documentación de instalación

Para poder usar CUDA será necesario instalarse Visual Studio para poder operar y programar en esta arquitectura. Mediante el siguiente enlace se podrá acceder a la descarga del instalador de Visual Studio:

<https://visualstudio.microsoft.com/es/thank-you-downloading-visual-studio/?sku=Community&rel=16>



Cuando se abra y se instale el instalador nos aparecerá la siguiente pantalla:



Se deberá escoger los entornos de trabajo que se le quieren asignar para poder programar en Visual Studio dependiendo de lo que se quiera hacer y qué herramientas vamos a necesitar. En este caso con la instalación de CUDA ya vienen las librerías y el entorno de trabajo de esta para poder programar.

Ahora, tras haber instalado Visual Studio se realizará la instalación de CUDA. Para la instalación de la arquitectura CUDA se puede instalar mediante el siguiente enlace:

<https://developer.nvidia.com/cuda-zone>

Tras este enlace nos llevará a la siguiente página:



CUDA® is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). With CUDA, developers are able to dramatically speed up computing applications by harnessing the power of GPUs.

In GPU-accelerated applications, the sequential part of the workload runs on the CPU – which is optimized for single-threaded performance – while the compute intensive portion of the application runs on thousands of GPU cores in parallel. When using CUDA, developers program in popular languages such as C, C++, Fortran, Python and MATLAB and express parallelism through extensions in the form of a few basic keywords.

The CUDA Toolkit from NVIDIA provides everything you need to develop GPU-accelerated applications. The CUDA Toolkit includes GPU-accelerated libraries, a compiler, development tools and the CUDA runtime.

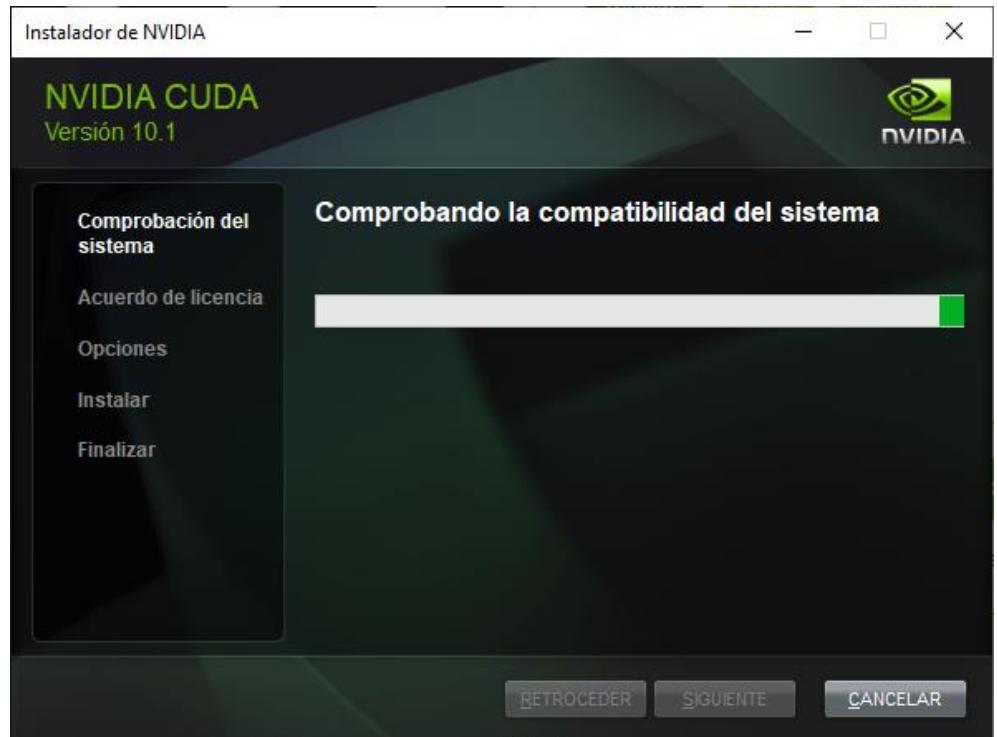
[Download Now >](#)

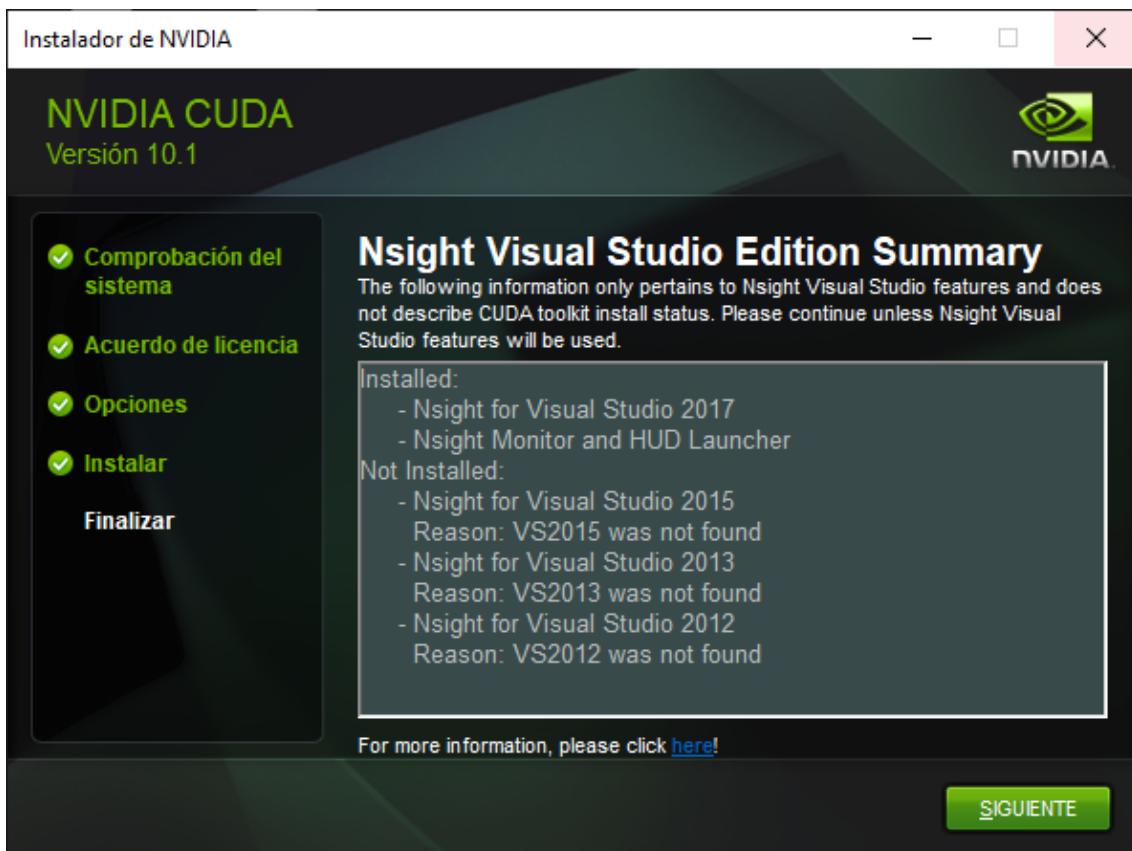
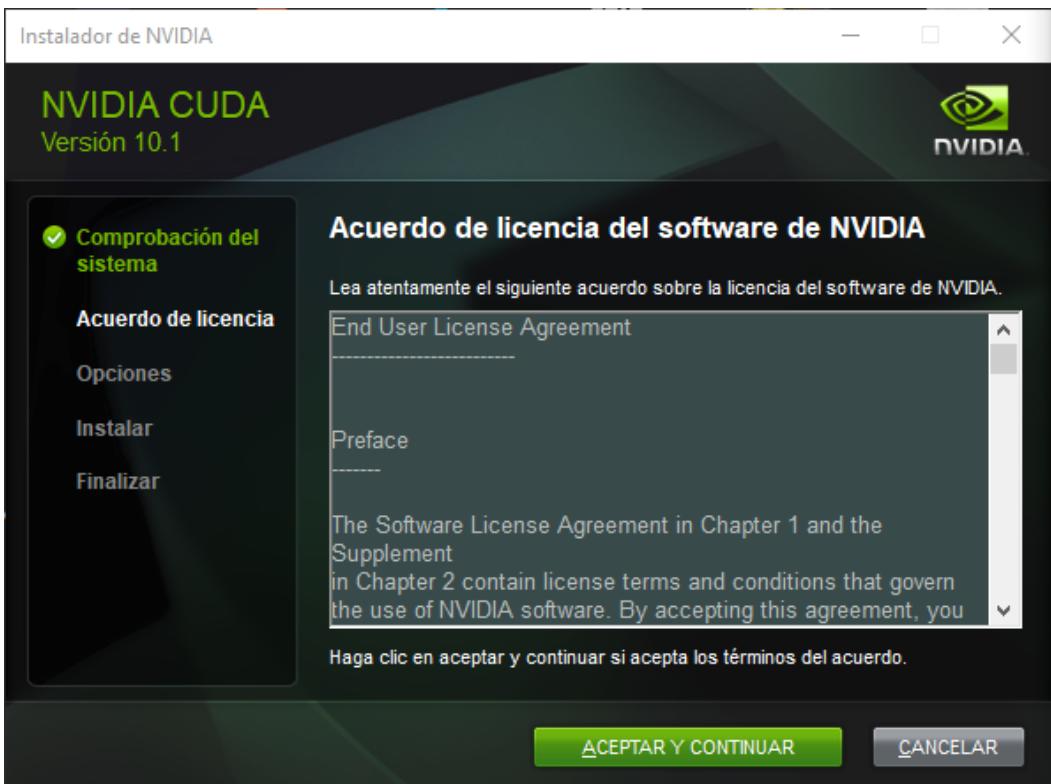
Siguiendo los pasos podremos descargarnos el instalador dependiendo de las características que cuenta nuestro dispositivo. En este caso nos instalaremos los controladores correspondientes a Windows 10 x86\_64 de forma local.

A screenshot of the CUDA Installer selection page. It shows filter options for "Operating System" (Windows, Linux, Mac OSX), "Architecture" (x86\_64), "Version" (10, 8.1, 7, Server 2019, Server 2016, Server 2012 R2), and "Installer Type" (exe [network], exe [local]).

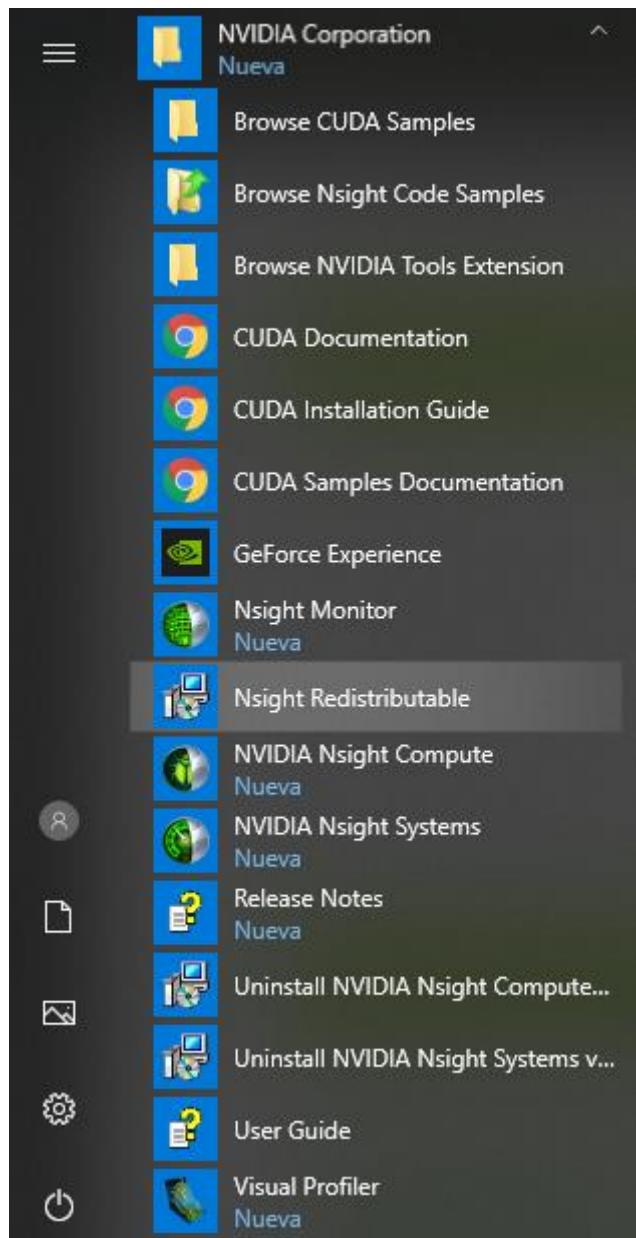
A screenshot of the CUDA Base Installer download page. It shows the title "Download Installer for Windows 10 x86\_64" and a message stating "The base installer is available for download below." A "Base Installer" button with a "Download (2.4 GB)" link is shown. Below it, "Installation Instructions" list steps 1 and 2. At the bottom, there is information about checksums and further resources.

Tras esto abrimos el instalador y seguimos todos los pasos que se nos indica:

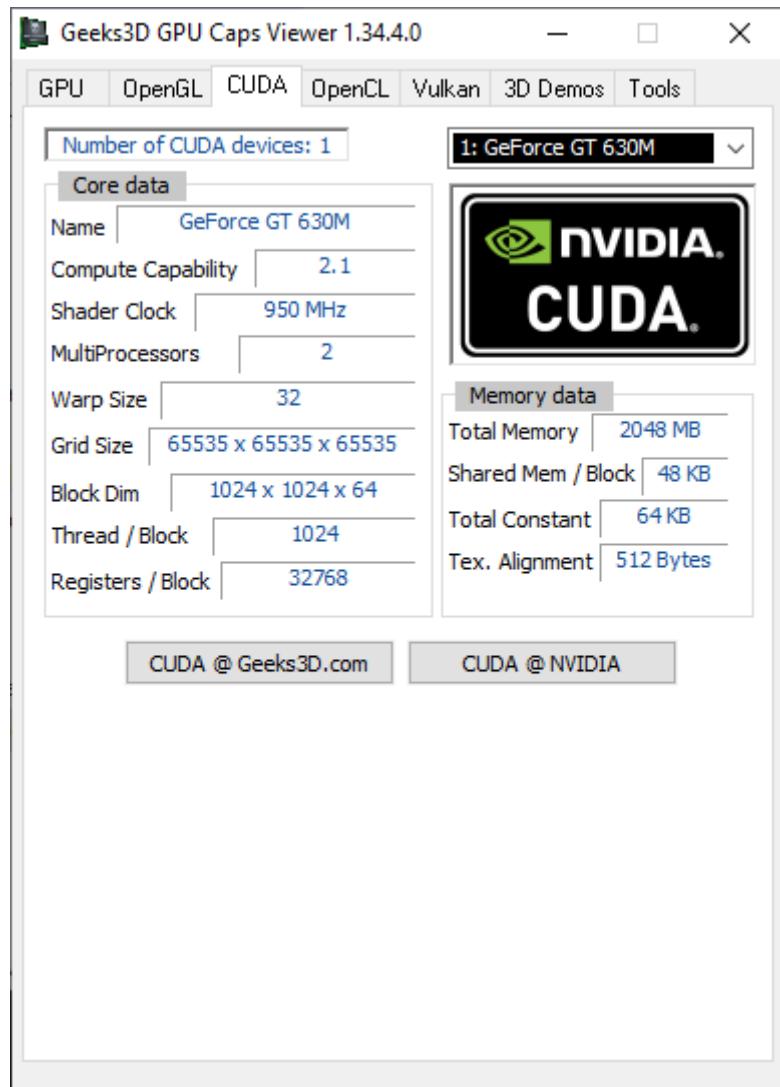




Tras esto se obtendrá un repertorio de aplicaciones en las cuales nos ayudarán a la hora de querer implementar el código por Visual Studio correctamente:



La utilidad GPUCaps nos confirma que estamos utilizando CUDA:



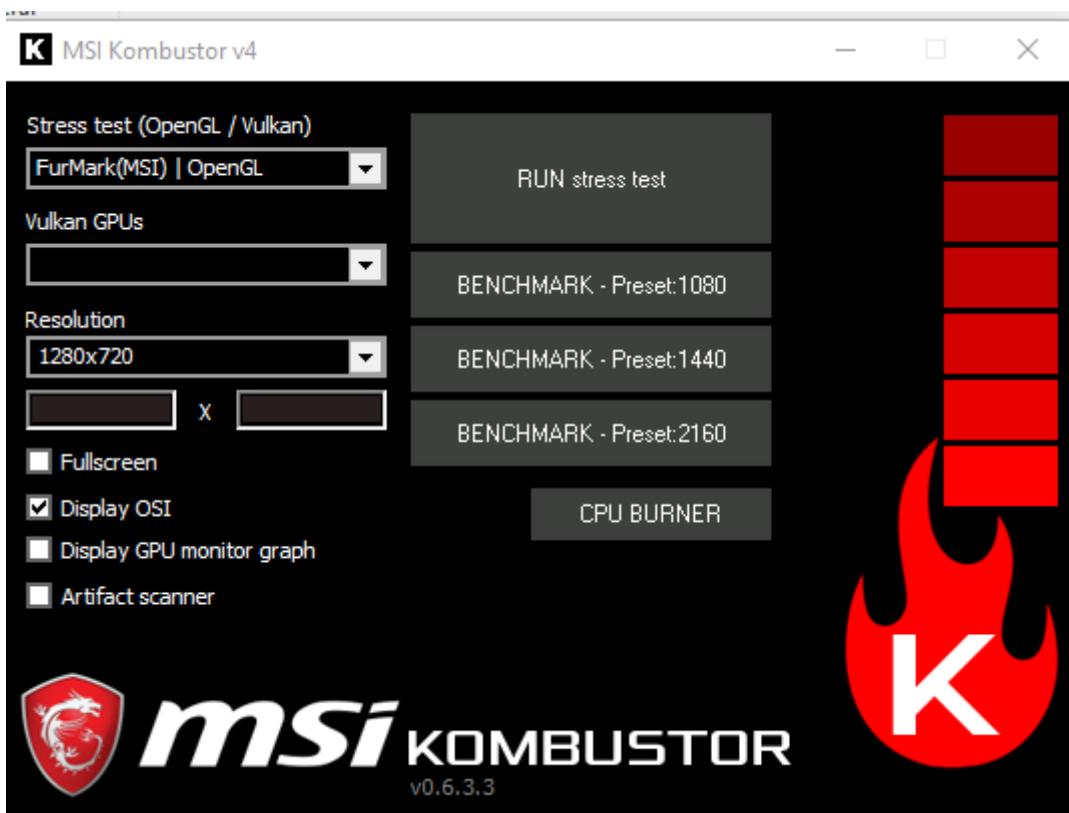
## 4.5 Manual de usuario

Una vez supuesto la instalación del hardware, vamos a ver cómo instalar y qué son los diferentes programas que hemos usado en el apartado anterior.

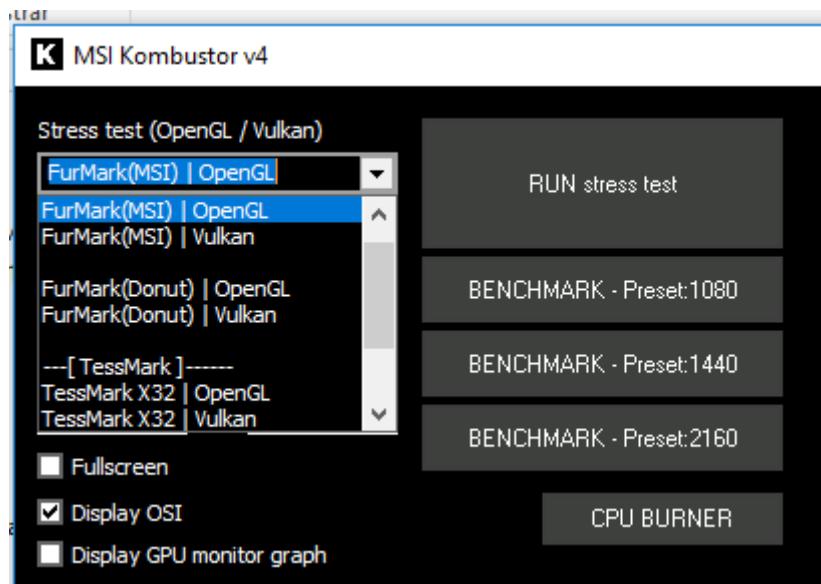
Vamos a comenzar instalando la utilidad MSI Kombustor. Es una suite software que incluye varios programas para realizar test de estrés. Podemos bajarlo gratuitamente de la siguiente página:

- <https://geeks3d.com/furmark/kombustor/>

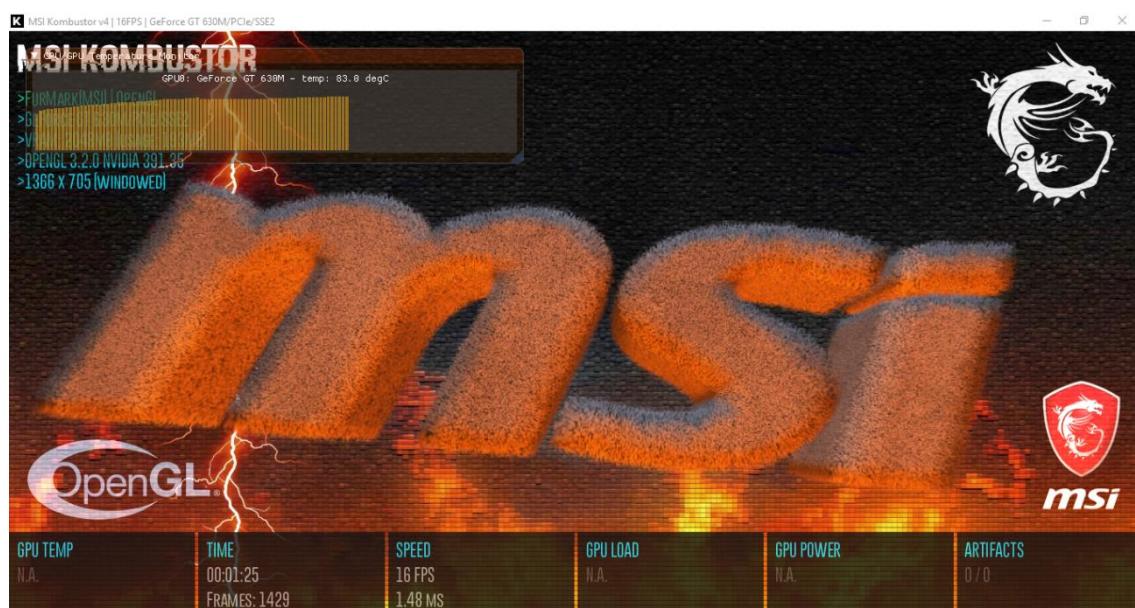
Lo abrimos, tras instalarlo, y seleccionamos el test que queremos. La pantalla principal es la siguiente:



Tenemos que seleccionar la resolución que queremos, y el test a ejecutar. Para estresar GPU + CPU, lo mejor es usar el test Furmark, poner la resolución más alta a la que vaya a trabajar el equipo, y pulsamos “RUN stress test”:

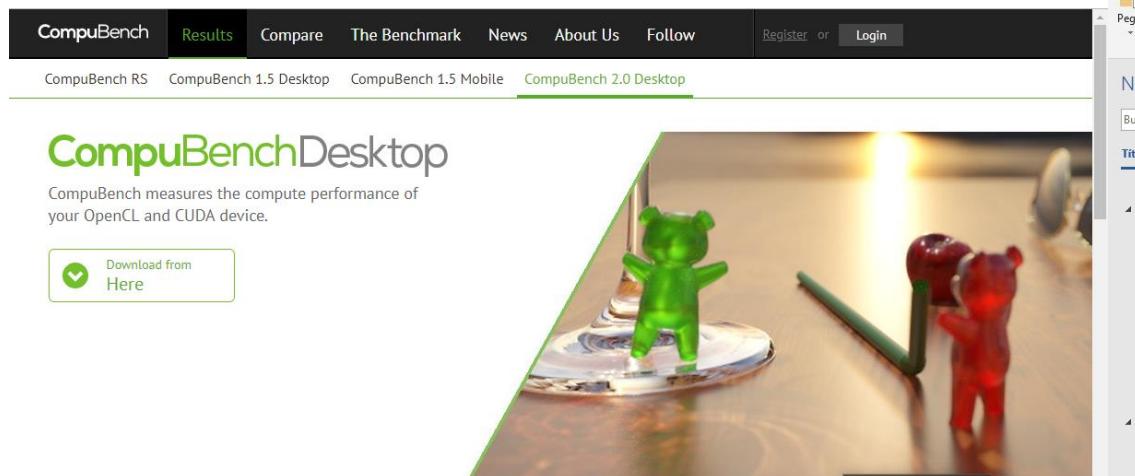


Arrojando como resultado el que vimos en los casos de prueba:



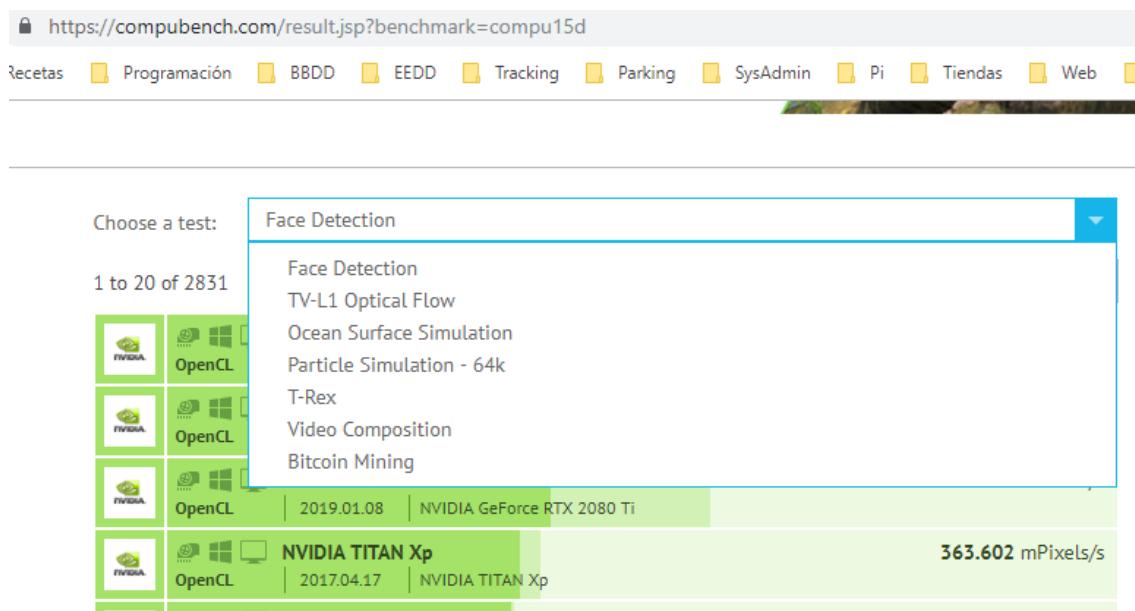
Esto es igual para ambas plataformas. Tenemos que ver qué temperatura alcanzan ambos, si se bloquea, y cuál es capaz de alcanzar mayor número de FPS o Frames Por Segundo. **Pasa la prueba**, de forma bastante notable. Alcanza unos 85º, el máximo de la gráfica es 100º.

Vamos ahora a observar cómo funciona la suite de pruebas o benchmarking CompuBench. Emplea el desempeño de la gráfica usando OpenCL o CUDA si están los drivers correctamente instalados. Nos da una idea de cómo se comportará nuestro equipo y en qué tareas puede ser empleado.



The screenshot shows the homepage of CompuBench Desktop. At the top, there's a navigation bar with links for 'CompuBench', 'Results', 'Compare', 'The Benchmark', 'News', 'About Us', 'Follow', 'Register or Login'. Below the navigation, there are links for 'CompuBench RS', 'CompuBench 1.5 Desktop', 'CompuBench 1.5 Mobile', and 'CompuBench 2.0 Desktop' (which is highlighted). The main title 'CompuBenchDesktop' is displayed in large green letters. A subtext below it says 'CompuBench measures the compute performance of your OpenCL and CUDA device.' There's a green button with a checkmark icon and the text 'Download from Here'. To the right of the text, there's a photograph of two small, translucent gummy bear figurines, one green and one red, standing on a reflective surface.

Es multiplataforma, multidispositivos, y cuenta con las siguientes pruebas:

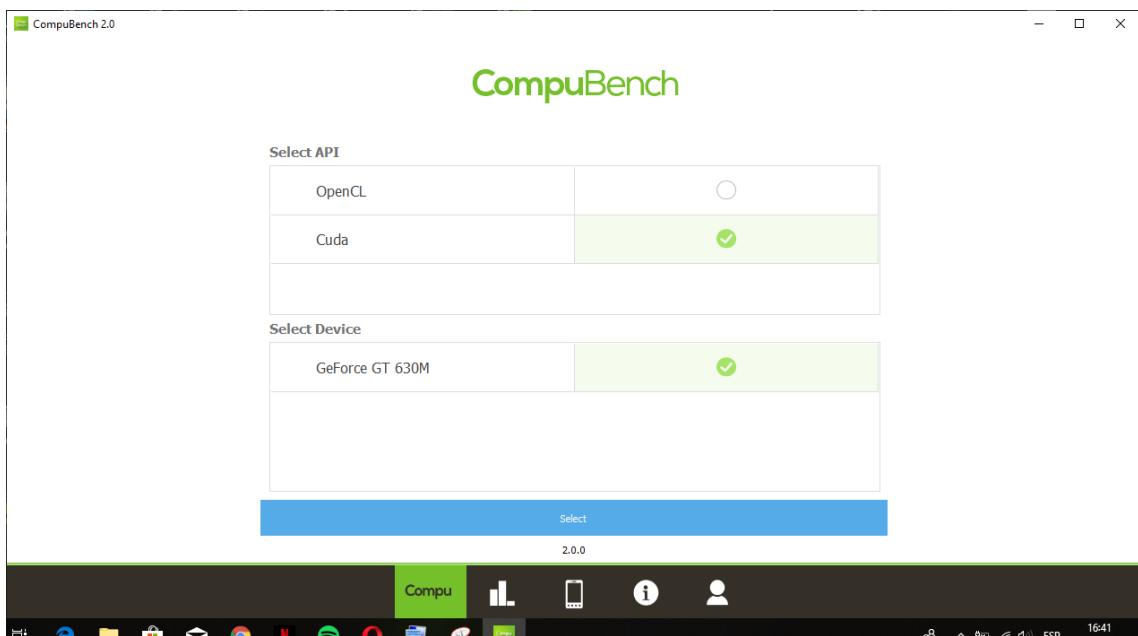


The screenshot shows the 'result.jsp?benchmark=compu15d' page. At the top, there's a header with a lock icon and the URL. Below it, a navigation bar includes 'Recetas', 'Programación', 'BBDD', 'EEDD', 'Tracking', 'Parking', 'SysAdmin', 'Pi', 'Tiendas', 'Web', and 'Tienda'. On the left, there's a sidebar with the text 'Choose a test:' and '1 to 20 of 2831'. It lists four items under 'NVIDIA' and 'OpenCL'. On the right, a dropdown menu is open, showing a list of tests: 'Face Detection', 'TV-L1 Optical Flow', 'Ocean Surface Simulation', 'Particle Simulation - 64k', 'T-Rex', 'Video Composition', and 'Bitcoin Mining'. Below the dropdown, there's a table with two rows. The first row shows a green bar for 'NVIDIA GeForce RTX 2080 Ti' with the date '2019.01.08'. The second row shows a green bar for 'NVIDIA TITAN Xp' with the date '2017.04.17'. Both rows show the result '363.602 mPixels/s'.

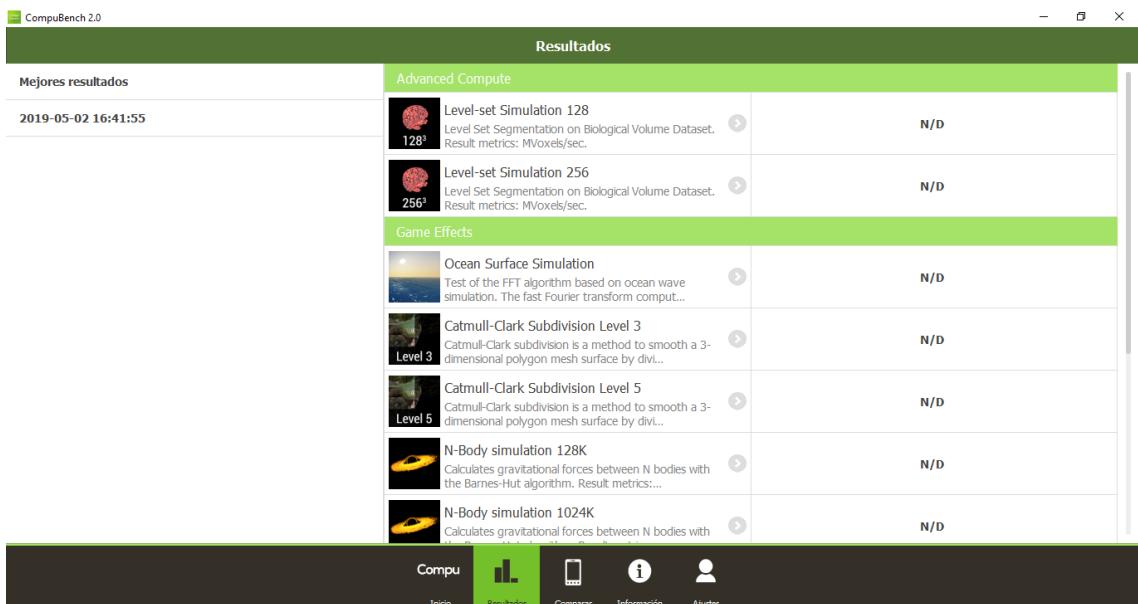
Nos lo bajamos de su sitio oficial, en nuestro caso usaremos la versión gratuita “CompuBench Desktop 1.5”:

- <https://compubench.com/result.jsp?benchmark=compu15d>

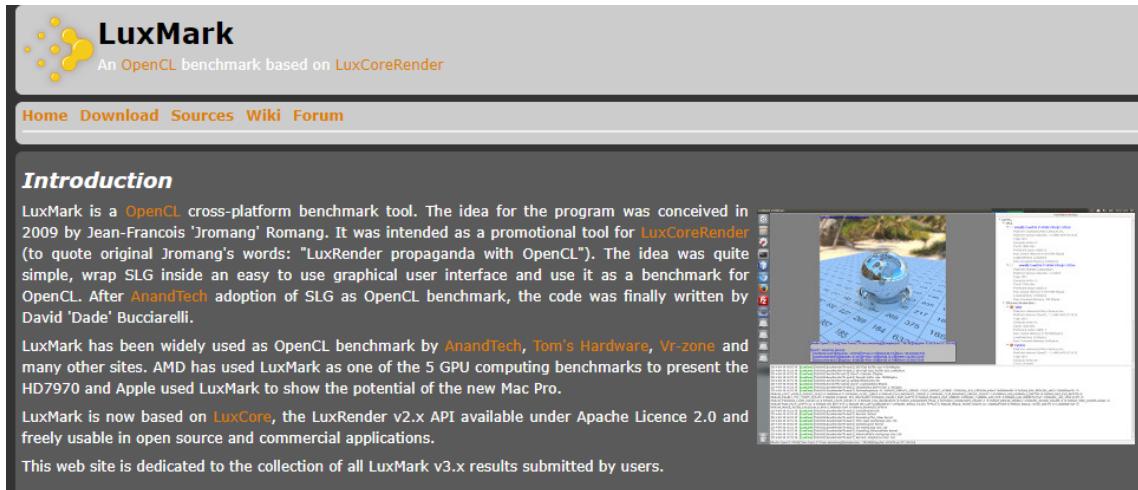
Lo ejecutamos tras instalarlo, elegimos usar la tecnología CUDA:



Y vamos a la pantalla principal, donde podremos elegir test a realizar. Damos a iniciar, y comenzará la prueba, con el resultado visto en el documento de pruebas:



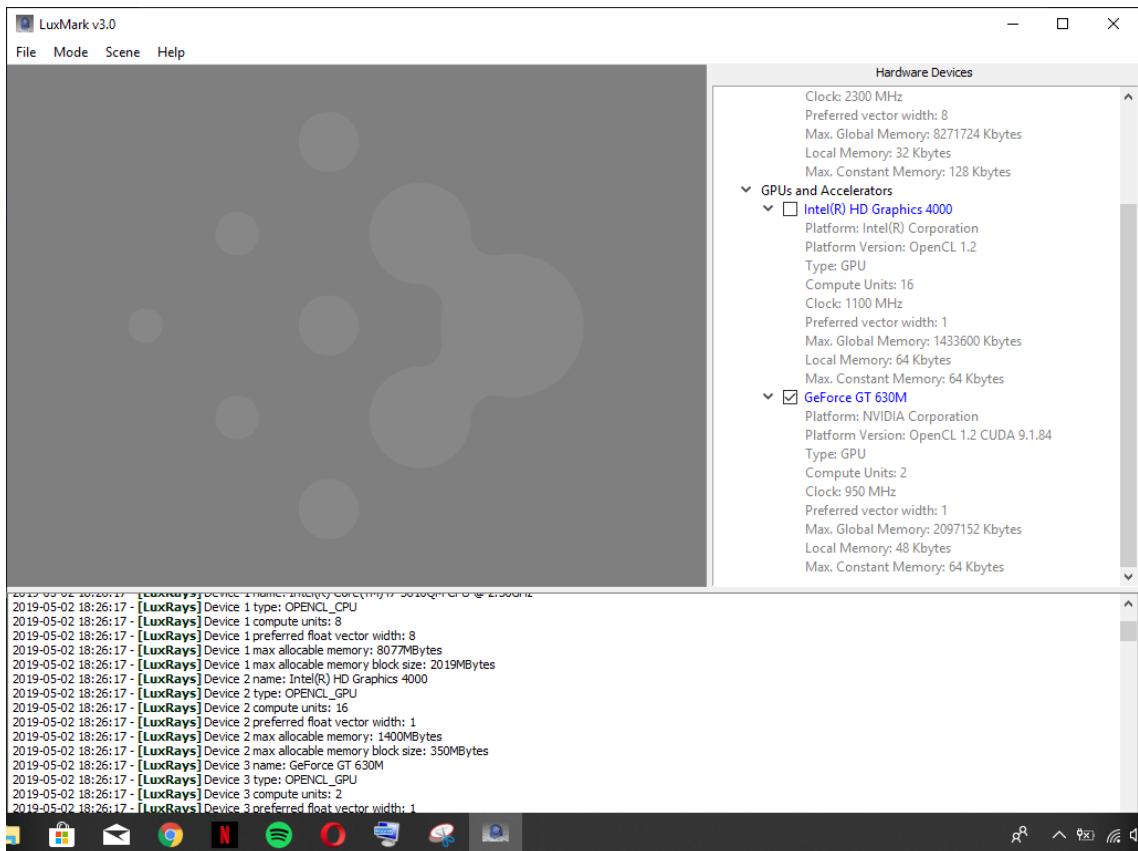
El software siguiente es la suite Luxmark, que mide de una forma más sencilla, ejecutando algoritmos matemáticos, qué equipo es más potente. Emplea también la potencia OpenCL y CUDA de las gráficas. En origen servía para comprobar la potencia OpenCL, pero se fue abriendo a más tecnologías:



Es multiplataforma. Lo podemos obtener del siguiente servidor oficial:

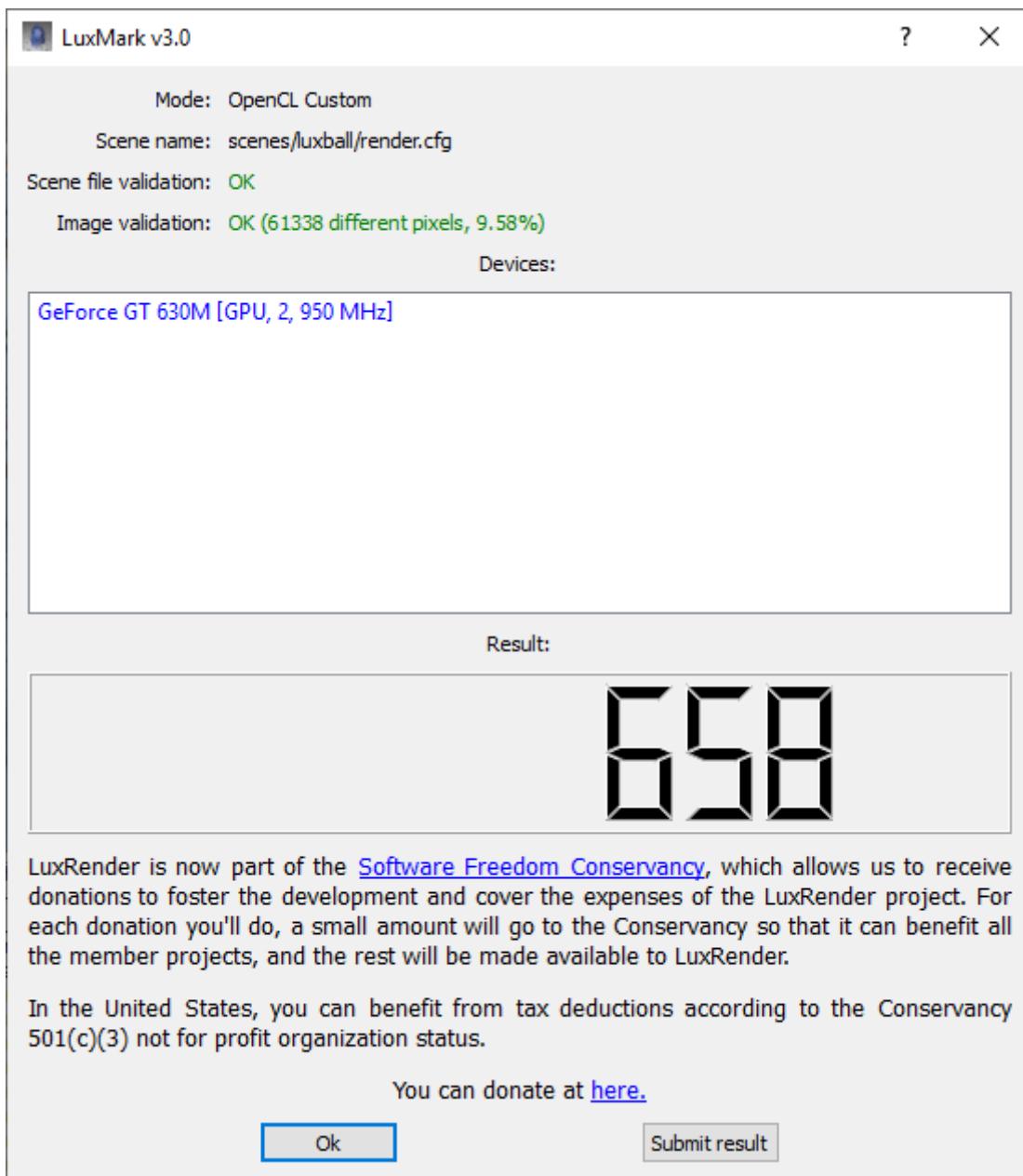
- <http://wiki.luxcorerender.org/LuxMark#Binaries>

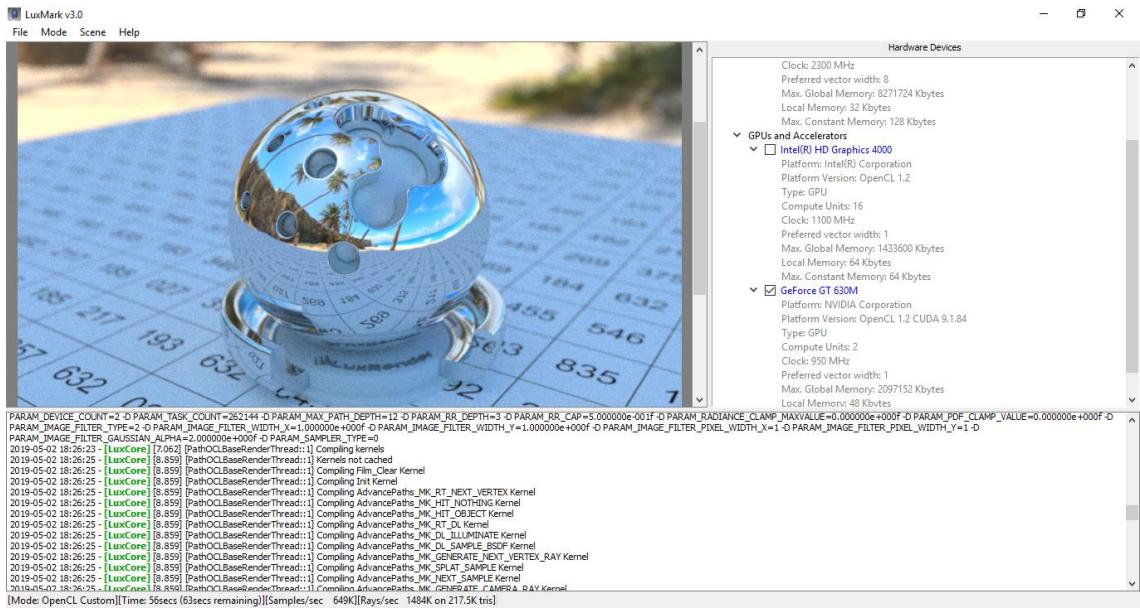
Al ejecutarlo por primera vez, veremos un diseño muy sobrio:



Observamos en el panel izquierdo, que es capaz de detectarnos la gráfica y los drivers OpenCL -CUDA, habilitándose esta última opción..

Ahora, solo tenemos que pulsar en Mode -> run y ejecutará las pruebas, dándonos un resultado final .En nuestro caso, hemos testeado solo la gráfica, pues en nuestro caso compiten un i3 y un i7:





Vamos ahora con nuestras pruebas científicas. En nuestro caso, nos vamos a unir a la red de ordenadores distribuidos del BOINC o “Compute for Science Project” iniciado por la universidad de Berkeley, y que nos permite donar poder computacional de nuestro equipo a diversas investigaciones científicas, desde búsqueda extraterrestre a cura del cáncer:

**Compute for Science**

BOINC lets you help cutting-edge science research using your computer (Windows, Mac, Linux) or Android device. BOINC downloads scientific computing jobs to your computer and runs them invisibly in the background. It's easy and safe.

About 30 science projects use BOINC; examples include Einstein@Home, IBM World Community Grid, and SETI@home. These projects investigate diseases, study global warming, discover pulsars, and do many other types of scientific research.

You can participate in either of two ways:

- Choose science areas**  
To contribute to science areas (biomedicine, physics, astronomy, and so on) use [Science United](#). Your computer will do work for current and future projects in those areas.  
[Join Science United](#)
- Choose projects**  
To contribute to specific projects, download BOINC and follow the directions.  
[Descargar BOINC](#)

## La búsqueda de aliens de SETI en crisis: el Bitcoin les está dejando sin GPUs para sus radiotelescopios



En nuestro caso, como acabamos de adelantar, hemos decidido emplear ambos equipos en Seti. Vamos a mostrar cómo hacerlo. Primero, ¿qué es Seti?:

SETI@home

Proyecto ▾ Science ▾ Procesando ▾ Comunidad ▾ Sitio ▾

¿Qué es SETI@home?

SETI@home is a scientific experiment, based at UC Berkeley, that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI). You can participate by running a free program that downloads and analyzes radio telescope data.

Unirse a SETI@home

Already joined? [Log in.](#)

Noticias

Ciencia

An up  
30 Apr

Long  
We ha  
Oscar  
will be

## GPU computing

Most computers are equipped with a **Graphics Processing Unit (GPU)** that handles their graphical output, including the 3-D animated graphics used in computer games. The computing power of GPUs has increased rapidly, and they are now often much faster than the computer's main processor, or CPU.

Some BOINC-based projects have applications that run on GPUs. **These applications run from 10X to 200X faster than the CPU-only version depending on the application, CPU and GPU in question.** We urge BOINC participants to use them if possible. Just follow these instructions:

### Check whether your computer has a capable GPU

- Identify the model name of your GPU.
  - On Windows, use GPU-Z found [here](#). (note: The listed capabilities of the card may be inaccurate on multi GPU systems.)
  - On Linux, in a console use: `lspci | grep VGA`
  - On Macintosh, Select *About this Mac* from the Apple menu, then click *More Info*. Under *Hardware* select *Graphics/Displays*.
- Note the Adapter Type and Memory Size.
- To find out if your NVIDIA GPU is compatible: check [NVIDIA's list of CUDA-enabled products](#). If your GPU is listed here and has at least 256MB of RAM, it's compatible.
- ATI GPUs: you need a platform based on the AMD R600 or AMD R700 GPU or later. R600 GPUs are found on [ATI Radeon HD2400, HD2600, HD2900 and HD3800](#) graphics board. R700 GPUs are found on [HD4350 to HD4890](#) graphics boards. Check [ATI's list of OpenCL-enabled products](#) to see which GPUs use OpenCL.
- Intel GPUs: Currently [Ivy Bridge](#) and [Haswell](#) are the only Intel CPUs with an OpenCL capable Intel GPU, however future embedded GPUs may also support OpenCL. You will need to install [Intel graphics drivers](#) to enable OpenCL support. It's also needed to add a monitor or VGA dummy-plug before the Intel GPU is recognized.
- AMD APUs: The AMD Kaveri, Brazos, Kabini and Lynx platforms -Zacate and Kabini for tablet and embedded products and Llano, Trinity and Richland APUs for desktops and laptops- are capable of doing GPU work on their own.

*Note: Some projects may have additional requirements. Check with the website of your project of interest for more details.*

### Contents [hide]

- 1 Check whether your computer has a capable GPU
- 2 Get the latest BOINC software
- 3 Get the latest driver
- 4 Attach to projects with GPU applications
- 4.1 Things to be aware of

Procedemos a instalarlo. Para ello, obtendremos el software oficial de BOINC de su propia página. Los requisitos previos, tener un ordenador que lo soporte con sistema Windows/Linux/Mac, y tener los drivers de la gráfica y CUDA/OpenCL instalados:

- <https://boinc.berkeley.edu/download.php>

## Install BOINC

BOINC es un programa que le permite donar el tiempo en que su computador está inactivo a proyectos científicos como SETI@home, Climateprediction.net, Rosetta@home, World Community Grid y muchos otros. Tras instalar BOINC en su computador, puede conectarlo con tantos proyectos como quiera.

We recommend that you also install [VirtualBox](#), so your computer can work on science projects that require it. Saber más de [VirtualBox](#).

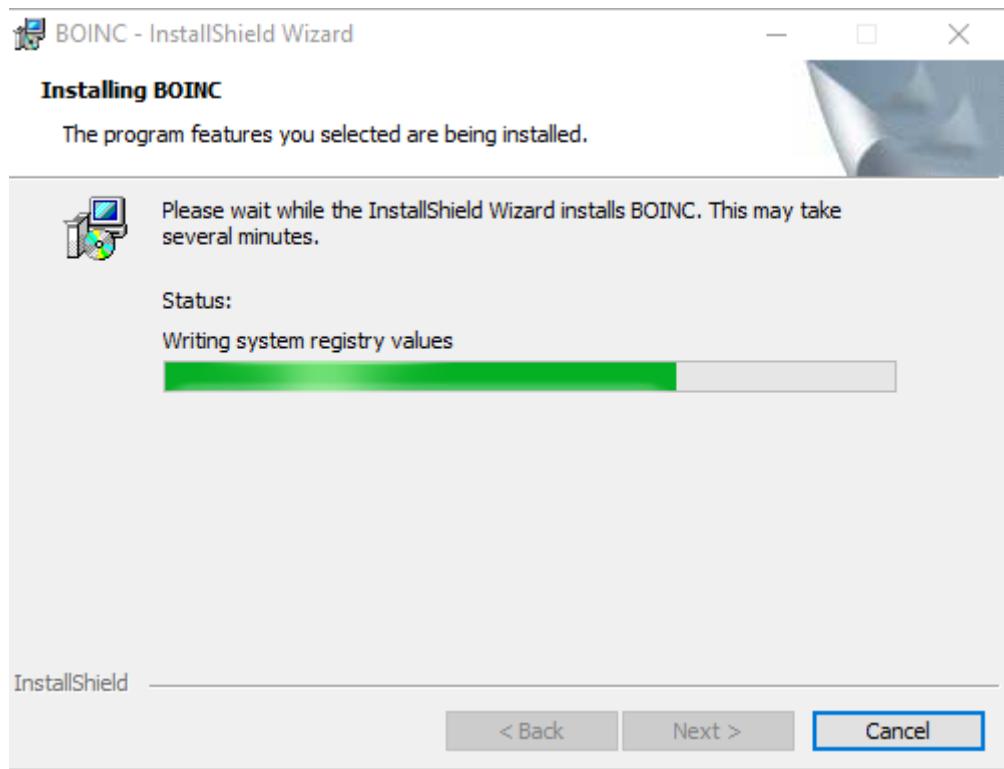
[Bajar BOINC y VirtualBox](#)  
for Windows 64-bit (92.93 MB)  
(BOINC 7.14.2, VirtualBox 5.2.8)

[Descargar BOINC](#)  
for Windows 64-bit (9.66 MB)  
(BOINC 7.14.2)

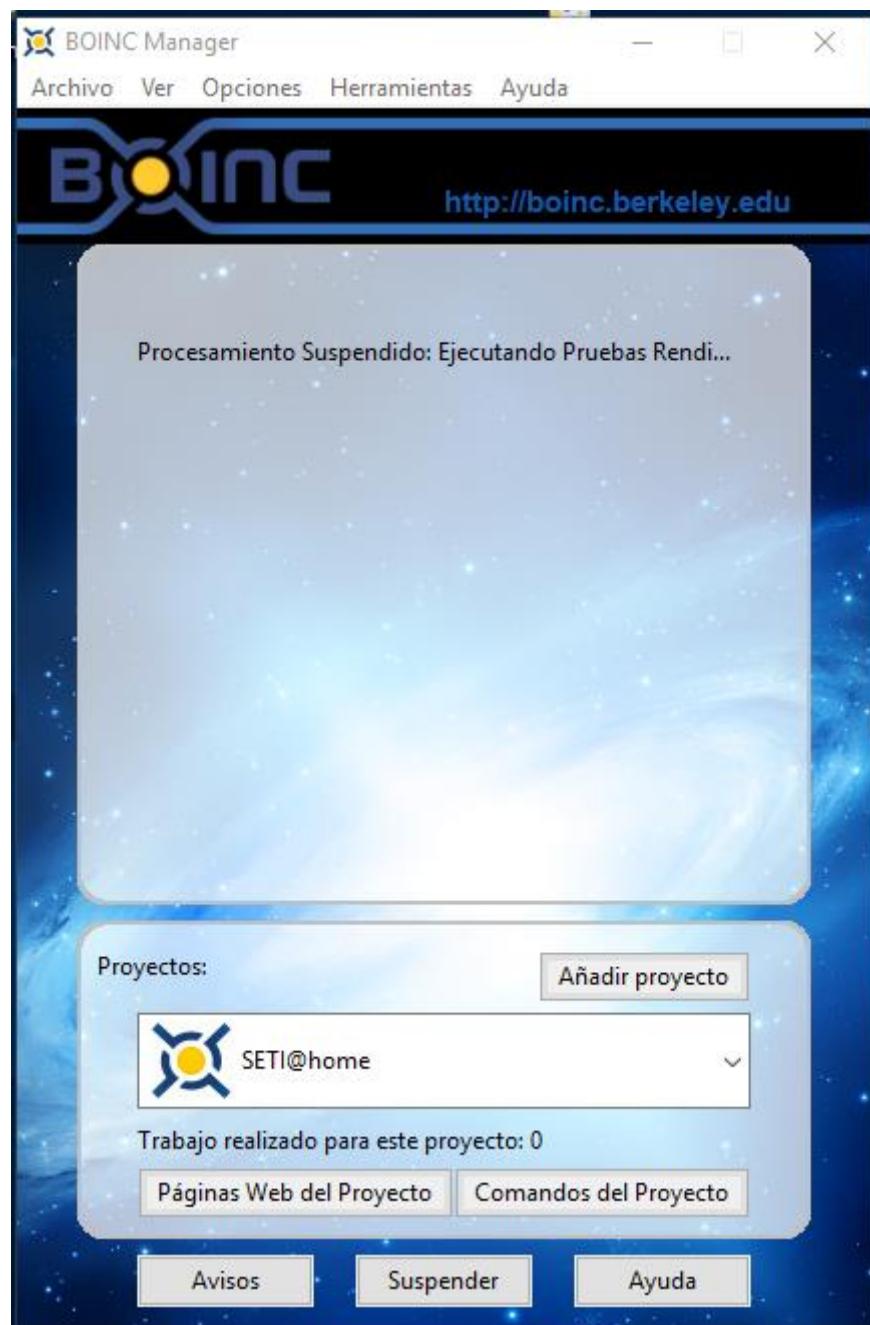
After downloading BOINC you must **install** it: typically this means double-clicking on the file icon when the download is finished.

When you first run BOINC, you will be asked to choose a project. For instructions, see the [BOINC User Manual](#).

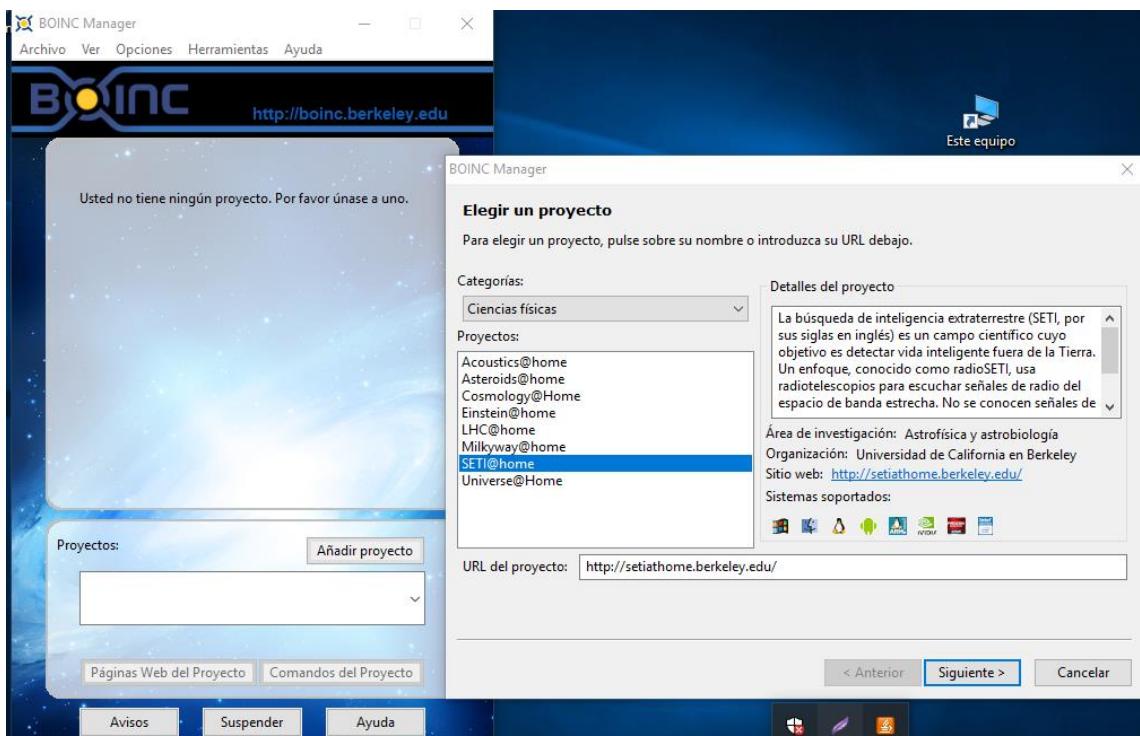
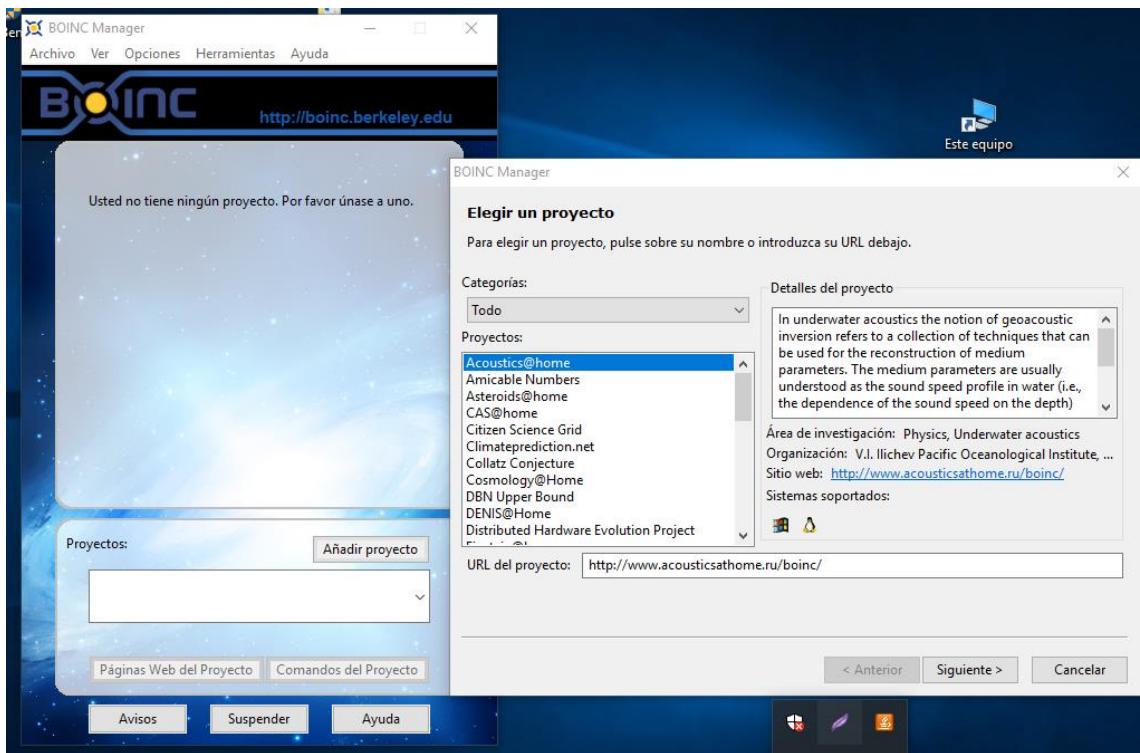
Debemos seleccionar la segunda opción y bajarnos el ejecutable. Ahora, configuraremos por primera vez el programa tras ejecutarlo:



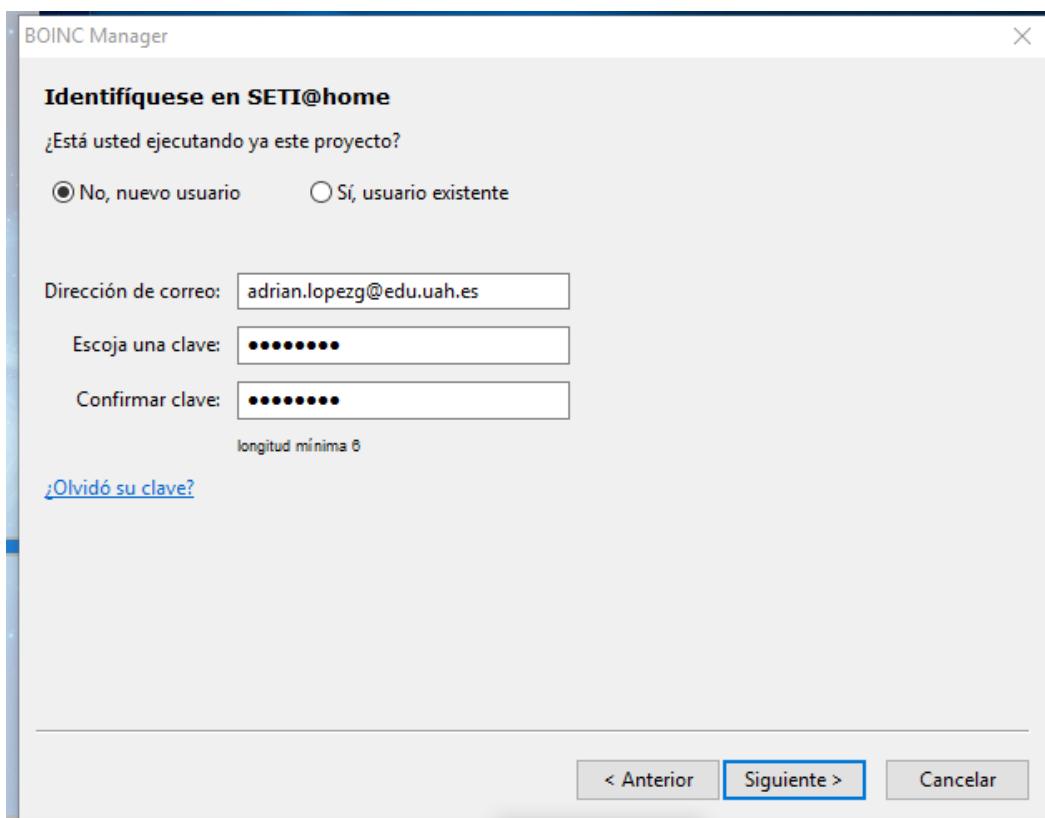
En el menú principal, seleccionamos “añadir proyecto”, y se abre la siguiente pantalla:



Seleccionamos el proyecto a añadir:

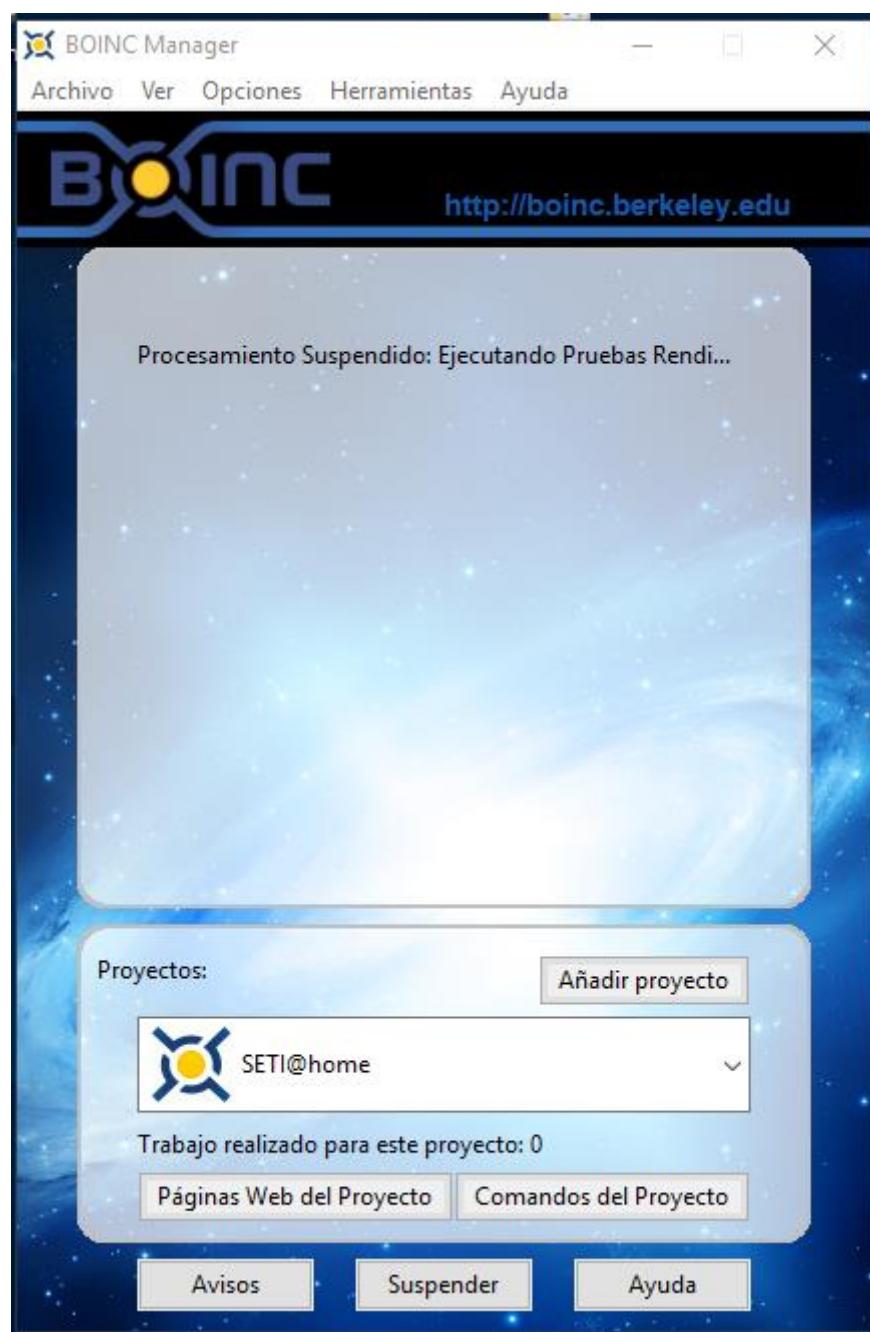


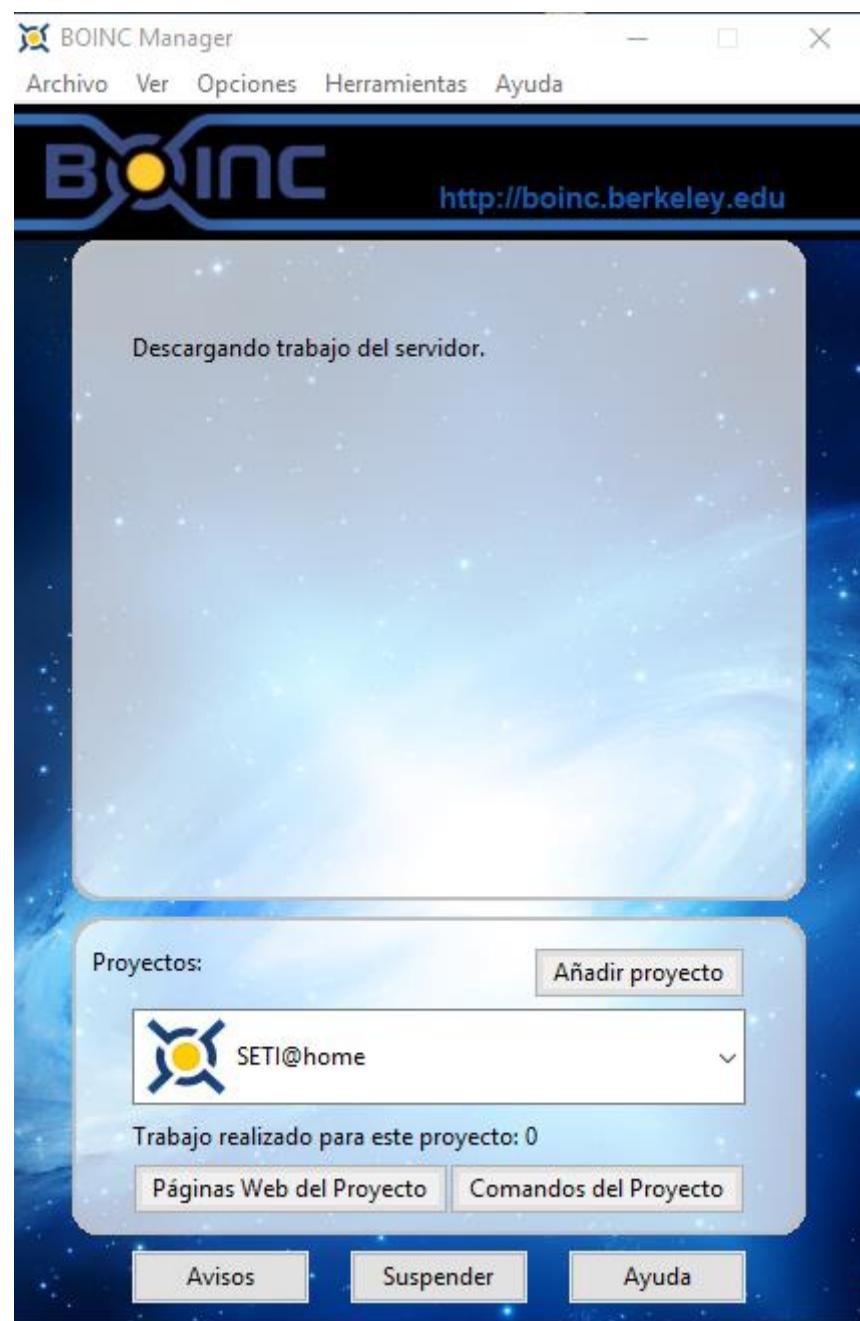
Creamos una cuenta para participar;



Proyecto	Hora	Mensaje
	05/05/2019 12:54:19	Starting BOINC client version 7.14.2 for windows_x86_64
	05/05/2019 12:54:19	log flags: file_xfer, sched_ops, task
	05/05/2019 12:54:19	Libraries: libcurl/7.47.1 OpenSSL/1.0.2g zlib/1.2.8
	05/05/2019 12:54:19	Data directory: C:\ProgramData\BOINC
	05/05/2019 12:54:19	Running under account javie
	05/05/2019 12:54:20	CUDA: NVIDIA GPU 0: GeForce GT 630M (driver version 391.35, CUD..
	05/05/2019 12:54:20	OpenCL: NVIDIA GPU 0: GeForce GT 630M (driver version 391.35, de..
	05/05/2019 12:54:20	OpenCL: Intel GPU 0: Intel(R) HD Graphics 4000 (driver version 10.18..
	05/05/2019 12:54:20	OpenCL CPU: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz (Open...
	05/05/2019 12:54:20	Host name: DESKTOP-ITN53ND
	05/05/2019 12:54:20	Processor: 8 GenuinelIntel Intel(R) Core(TM) i7-3610QM CPU @ 2.30.
	05/05/2019 12:54:20	Processor features: fpu vme de pse tsc msr pae mce cx8 apic sep mt.
	05/05/2019 12:54:20	OS: Microsoft Windows 10: Core x64 Edition, (10.00.17763.00)
	05/05/2019 12:54:20	Memory: 7.89 GB physical, 9.14 GB virtual
	05/05/2019 12:54:20	Disk: 930.11 GB total, 748.34 GB free
	05/05/2019 12:54:20	Local time is UTC +2 hours
	05/05/2019 12:54:20	No WSL found.

Y esperamos a que el servidor mida el rendimiento de nuestro equipo, y nos comience a mandar carga de trabajo:

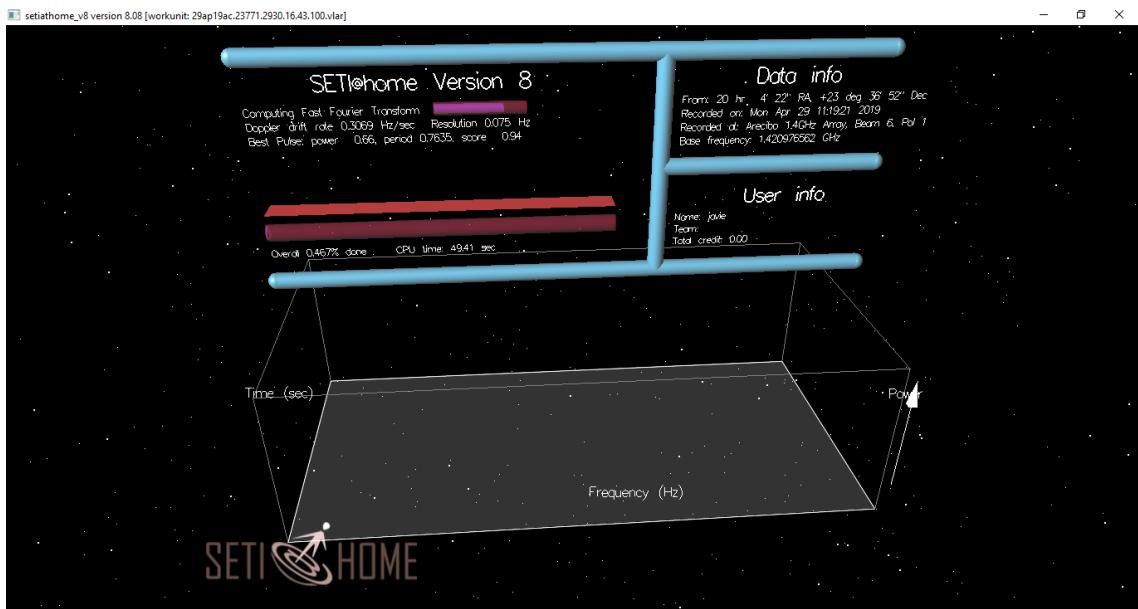




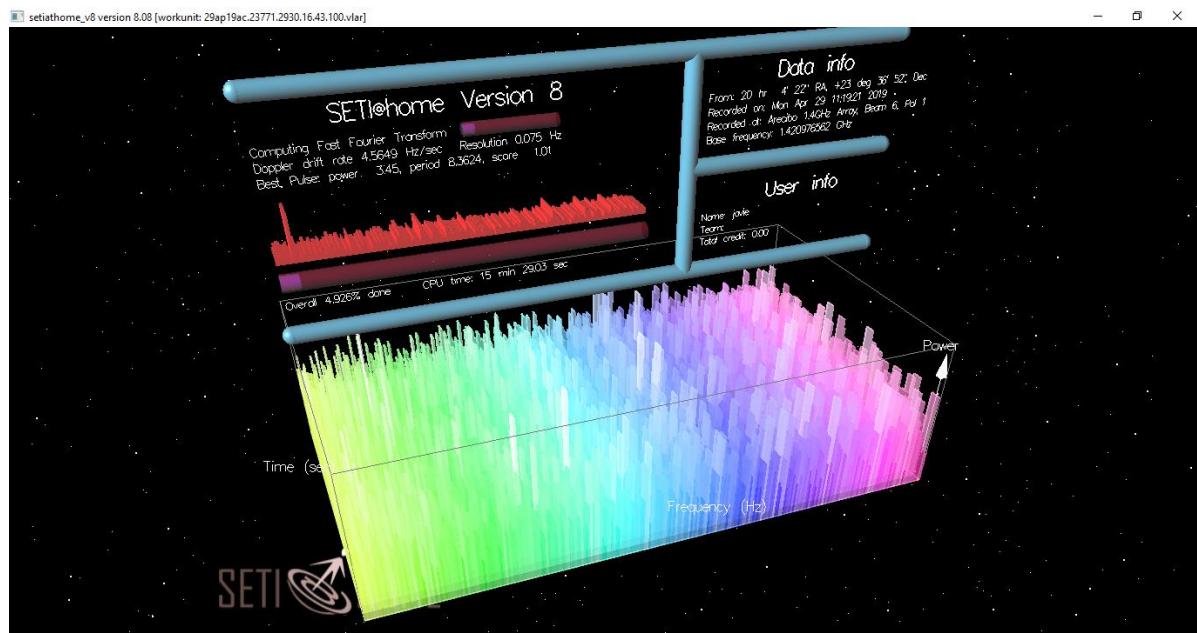
Si seguimos estos simples pasos, ya debemos ser capaces de ver el programa ejecutándose correctamente:



En “Comandos de la Tarea”, seremos capaces de ver cómo ejecuta nuestro equipo el trabajo mandado:



En la pantalla superior, nos irá indicando cómo es capaz de ejecutar las transformadas de Fourier, y las fórmulas gaussianas en la siguiente captura:



Propiedades de la tarea blc35_2bit_guppi_58406_27612_HIP20531_0105.19075.818.21...	
Programa	SETI@home v8 8.05
Nombre	blc35_2bit_guppi_58406_27612
Estado	Ejecutando
Recibido	02/05/2019 19:03:07
Límite para informar	25/06/2019 0:02:48
Tamaño de cálculo estimado	183.887 GFLOPs
Tiempo de CPU	00:26:50
Tiempo de CPU desde el punto de referencia	00:00:39
Tiempo transcurrido	00:28:41
Tiempo restante aproximado	03:23:27
Parte hecha	7,645%
Tamaño de la memoria virtual	35,67 MB
Tamaño conjunto trabajo	39,66 MB
Carpeta	slots/2

[Copiar Todo](#)

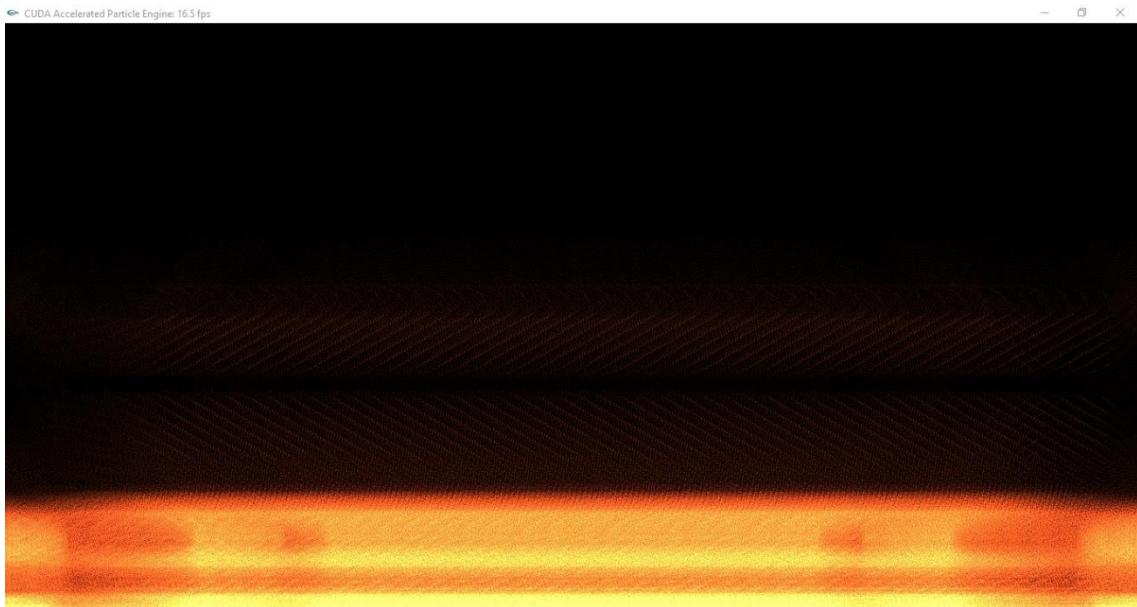
[Copiar Selección](#)

[Cerrar](#)

Ahora, ejecutamos el software específico para gráficas compatible con CUDA. El programa elegido es, que bajaremos del enlace oficial:

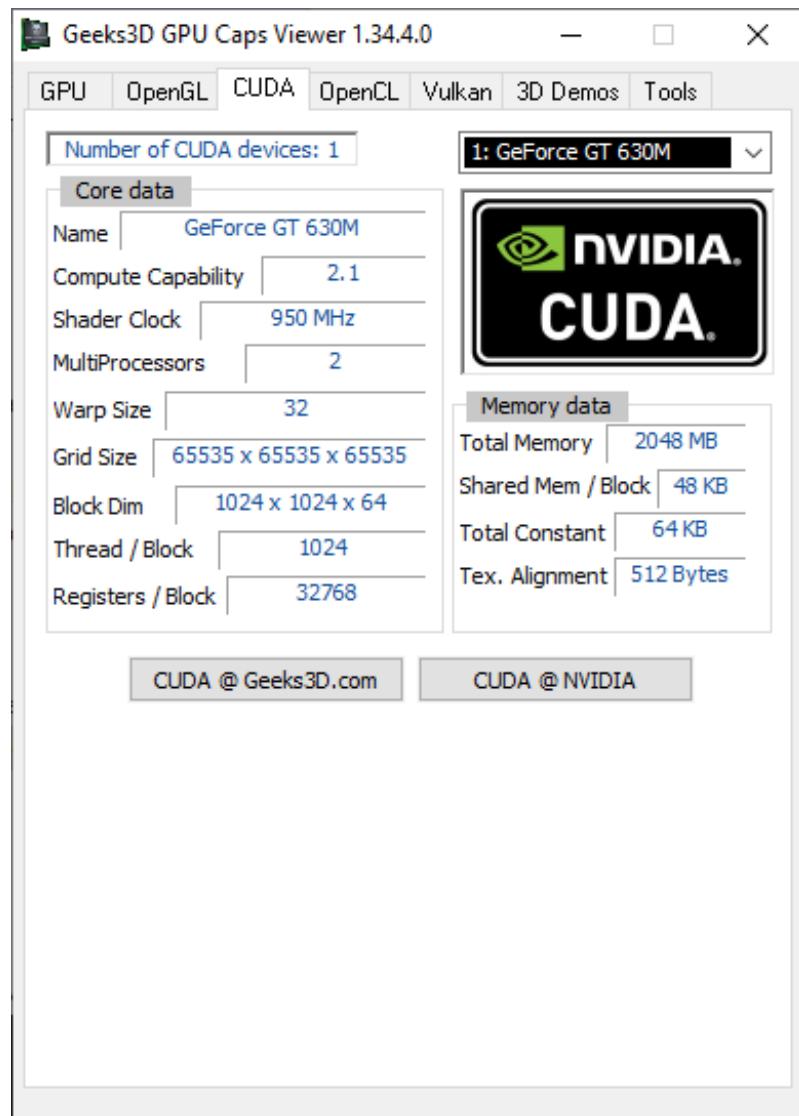
- <http://www.craigmouser.com/random/cudaparticles.zip>
- <https://www.geeks3d.com/20100830/gpu-computing-particle-systems-with-opencl-and-cuda/>

Lo ejecutamos, y esta será la primera pantalla que veamos si todo está instalado correctamente, el propio programa ejecutándose, y arriba, los FPS que alcanza:



En este caso, supera la prueba, aunque de media alcanza unos 15FPS, un resultado bastante bajo.

Comprobamos que CUDA está reconocido y activo, de nuevo:



## **5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología OpenCL**

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B (OpenCL), del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

### **5.1 Documentación de diseño**

El prototipo consistirá en dos ordenadores portátiles, equipados con tarjetas gráficas y procesadores capaces de ejecutar CUDA y OpenCL para cálculos científicos.

Ambos contarán además con el software necesario para poder realizar las pruebas pertinentes para poder comprobar cuál de las dos tecnologías podría considerarse más eficiente o competitiva, según varios criterios, en dos equipos con un nivel de hardware gráfico similar (de igual gama).

El prototipo no podrá ser mostrado en máquinas en caso de carecer de las instrucciones necesarias.

### **5.2 Documentación de construcción**

Para construir el prototipo, tenemos que diseñar, por un lado, hardware totalmente compatible con lo que queremos construir, y diseñar un ecosistema software que también lo sea.

Intentando conseguir un resultado lo más preciso, ambos equipos deberán estar lo más equilibrado en condiciones de sistema, HDD, Ram y procesador.

CAPA SOFTWARE	<ul style="list-style-type: none"><li>• Drivers</li><li>• SDK</li><li>• CUDA/OpenCL driver</li><li>• IDE</li><li>• Testing Tools</li><li>• Scientific Software</li></ul>
CAPA SISTEMA OPERATIVO	<ul style="list-style-type: none"><li>• Sistema (Windows/Mac/Linux)</li><li>• Actualizaciones</li></ul>
CAPA HARDWARE	<ul style="list-style-type: none"><li>• Procesador &gt; SSE2</li><li>• Ram &gt; 4Gb</li><li>• Gráfica compatible</li></ul>

### 5.3 Documentación de pruebas

Vamos a comprobar el correcto funcionamiento de las tarjetas gráficas AMD y la tecnología OpenCL, usando dos tipos de aplicaciones.

Por una parte, usaremos software de testeo para ver si se está realmente usando la potencia de la gráfica, y por otra, emplearemos software científico real, vinculado a el proyecto BOINC, que busca emplear la potencia de cómputo sobrante de los equipos para proyectos científicos. Es un modelo distribuido, en el que tanto supercomputadores como equipos corrientes ponen a disposición de los centros investigadores sus recursos.

#### ----- Pruebas Testing -----

Vamos a comenzar empleando el software MSI Kombustor, que nos dará una cifra aproximada en FPS de rendimiento en juegos, pero más importante, nos permitirá poner al máximo los dos equipos, en tareas gráficas. Consiste en una serie de test integrados, dirigidos a estresar, como hemos comentado, la tarjeta gráfica. Soporta OpenGL, Vulkan, OpenCL y Nvidia CUDA. Veremos si es capaz de detectar los drivers OpenCL:



Iniciamos el programa, ya instalado en el sujeto de pruebas 2:



Vemos que, aunque la temperatura es alta, se mantiene estable. El sistema pasa la prueba de estabilidad.

El segundo programa que vamos a usar es CompuBench Desktop, que mide el rendimiento de dispositivos OpenCL y CUDA, realizando las mismas pruebas en ambos equipos. Observamos los resultados que nos ha dado en OpenCL:

CompuBench

Results

Compare

The Benchmark

News

About Us

Follow

Register

CompuBench RS

CompuBench 1.5 Desktop

CompuBench 1.5 Mobile

CompuBench 2.0 Desktop

# CompuBenchDesktop

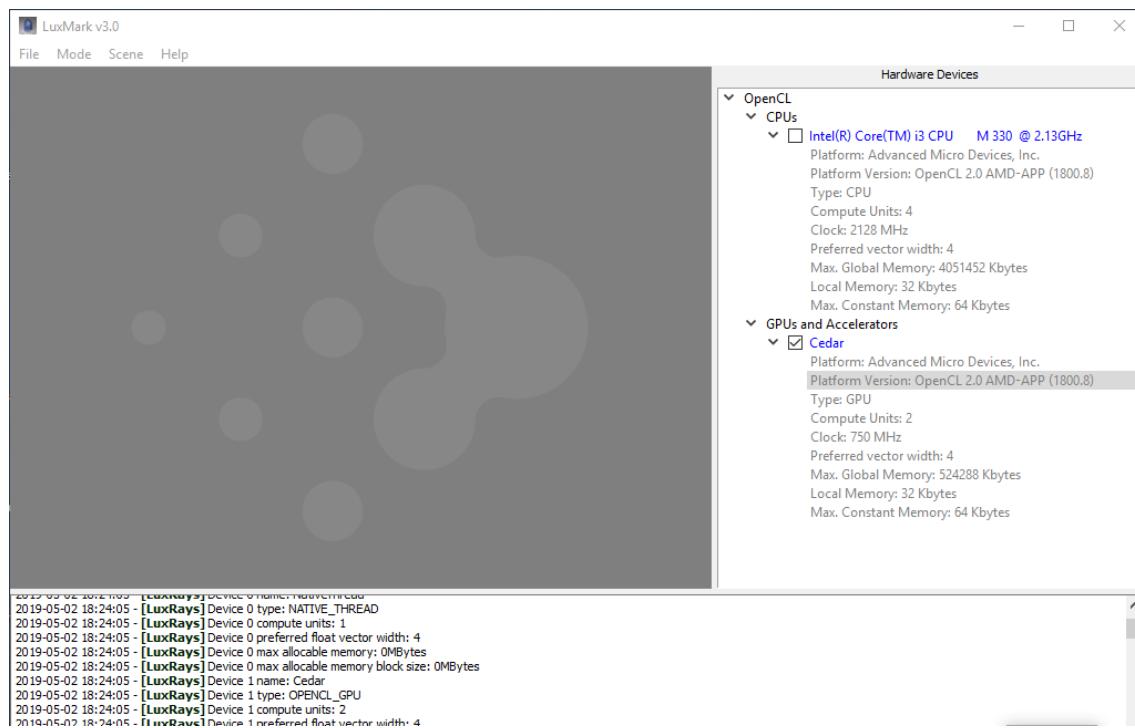
CompuBench measures the compute performance of your OpenCL and CUDA device.

Download from Here

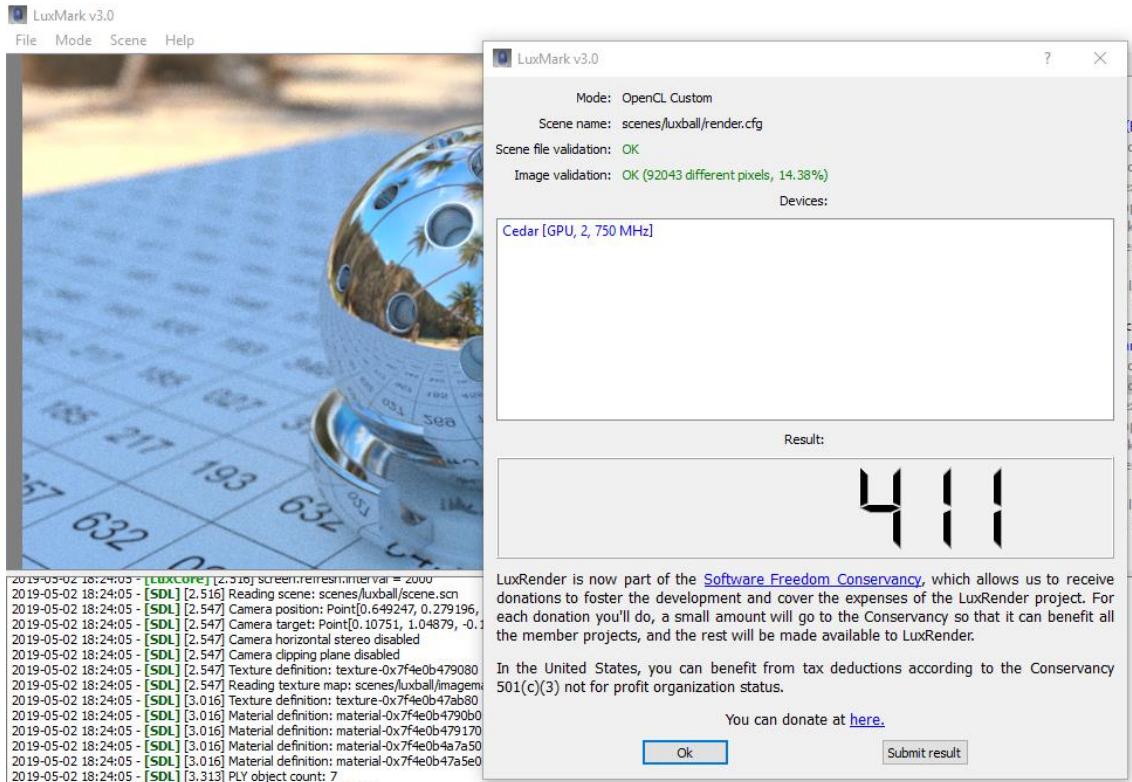
Tras instalarlo y ejecutarlo, vemos el siguiente resultado. Nuestro equipo OpenCL no es capaz de ejecutar todas las pruebas. Ello es debido a carencia de instrucciones y tecnologías de la gráfica AMD:

Resultados			
Mejores resultados			
jueves, 2 de mayo de 2019 17:53:37 Advanced Micro Devices, Inc.: Cedar	 Face detection	Face detector based on the Viola-Jones algorithm. Face detection is extensively used in biometri...	 <b>Failed</b> CL_BUILD_PROGRAM_FAILURE
jueves, 2 de mayo de 2019 17:50:15 Advanced Micro Devices, Inc.: Cedar	 TV-L1 Optical Flow	A test based on dense motion vector calculation using variational method. Optical flow is widely u...	 <b>0.7775 MPixels/s</b> Advanced Micro Devices, Inc.: Cedar
Physics			
	 Ocean Surface Simulation	Test of the FFT algorithm based on ocean wave simulation. The fast Fourier transform compute...	 <b>Failed</b> CL_BUILD_PROGRAM_FAILURE
	 Particle Simulation - 64k	Particle Simulation in a spatial grid using the discrete element method. The result of the simulati...	 <b>34.86 MInteractions/s</b> Advanced Micro Devices, Inc.: Cedar
Graphics			
	 T-Rex	Path tracer featuring dynamically updated acceleration structure and global illumination. Result ...	 <b>0.19187 Frames/s</b> Advanced Micro Devices, Inc.: Cedar
Video Processing			
	 Video Composition	This test replicates a typical video composition pipeline with effects such as pixelate, mask, mix...	 <b>Failed</b> CL_BUILD_PROGRAM_FAILURE

Vamos a proceder a ejecutar el software de benchmarking **Luxmark**:

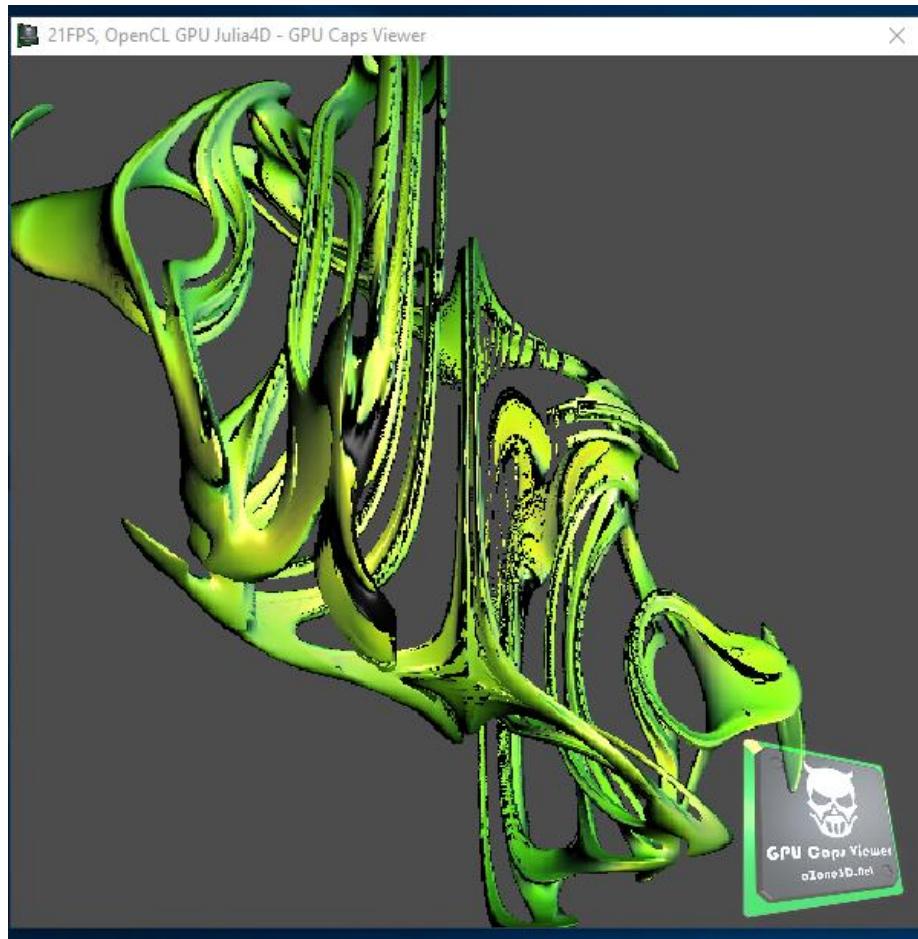


Y obtenemos un resultado numérico, que especifica la puntuación obtenida por el sistema gráfico usando OpenCL, como vimos en la captura anterior:



Se ejecuta sin anomalías, detectando los drivers OpenCL.

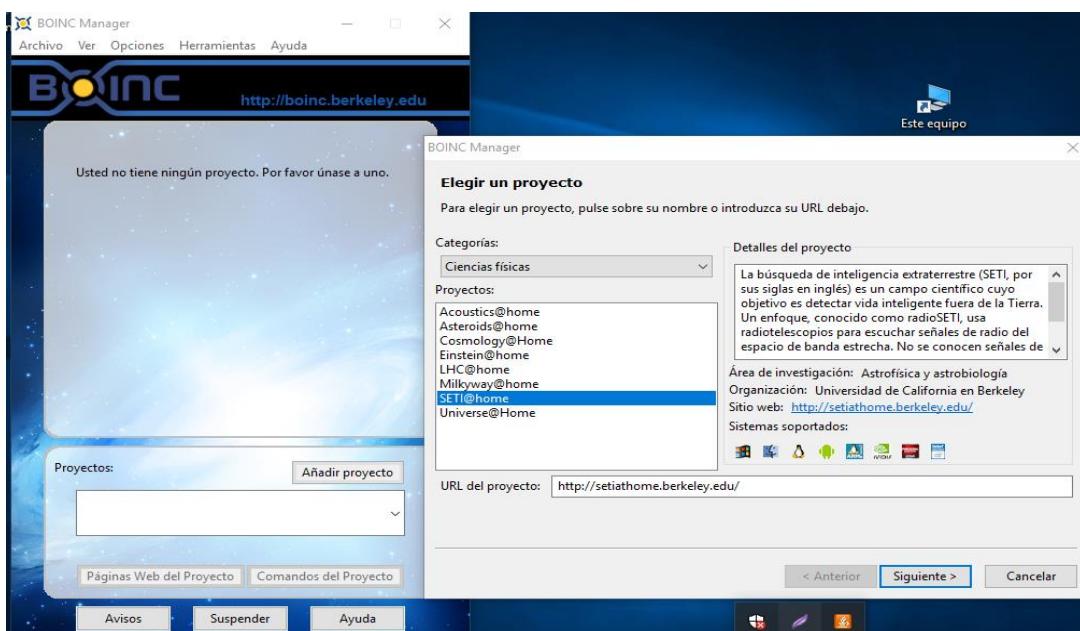
Vamos a proceder a realizar la prueba específica para gráficas OpenCL (AMD/Intel/Nvidia), en nuestro caso usando el software **Julia4D**:

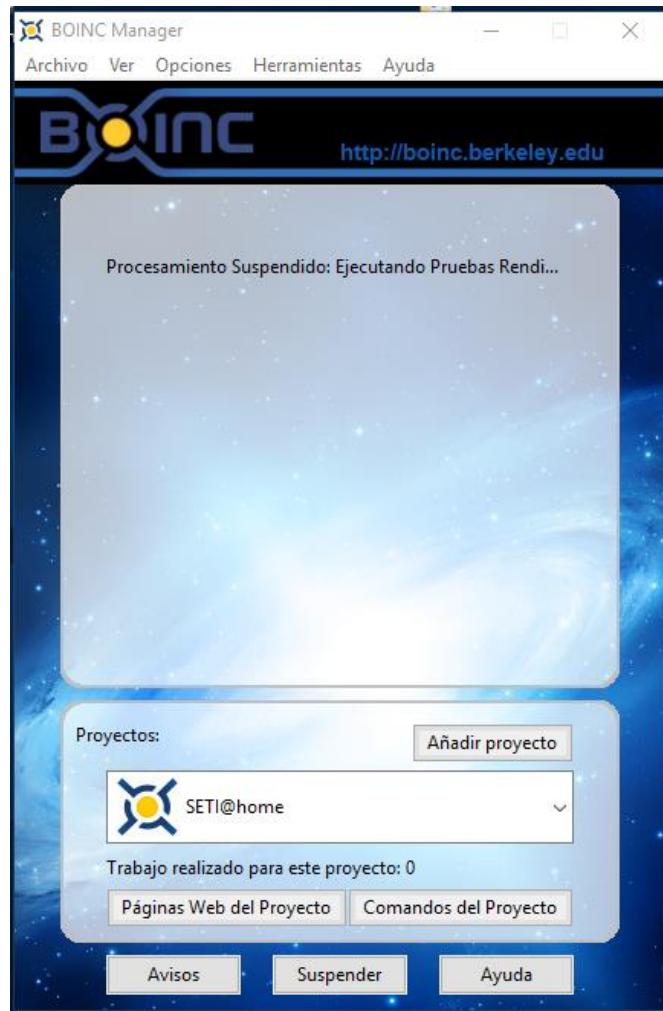


Como vemos, es capaz de ejecutarlo con unos 21FPS sostenidos. Se ejecuta correctamente, superando la prueba.

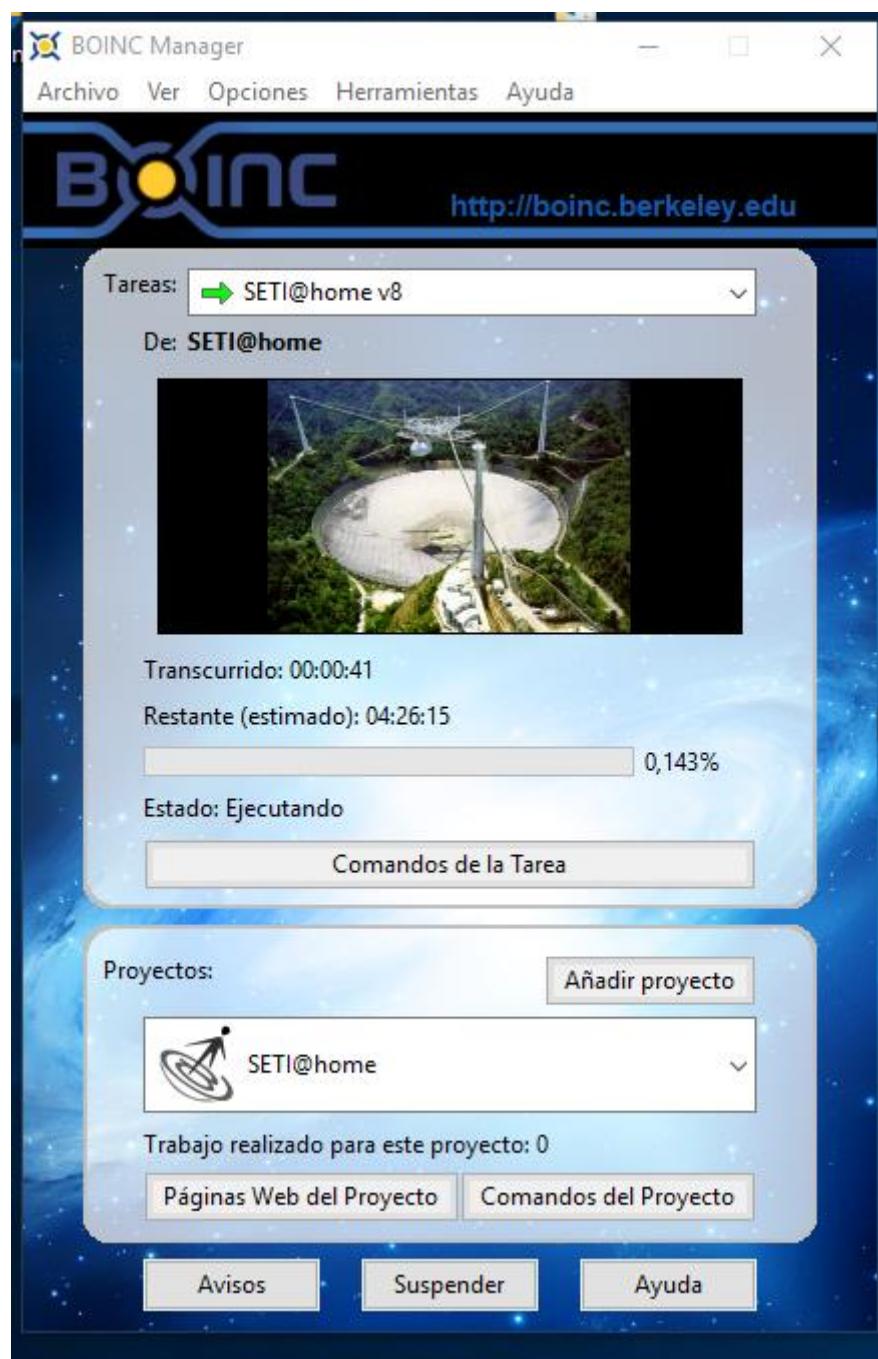
#### ----- Pruebas científicas -----

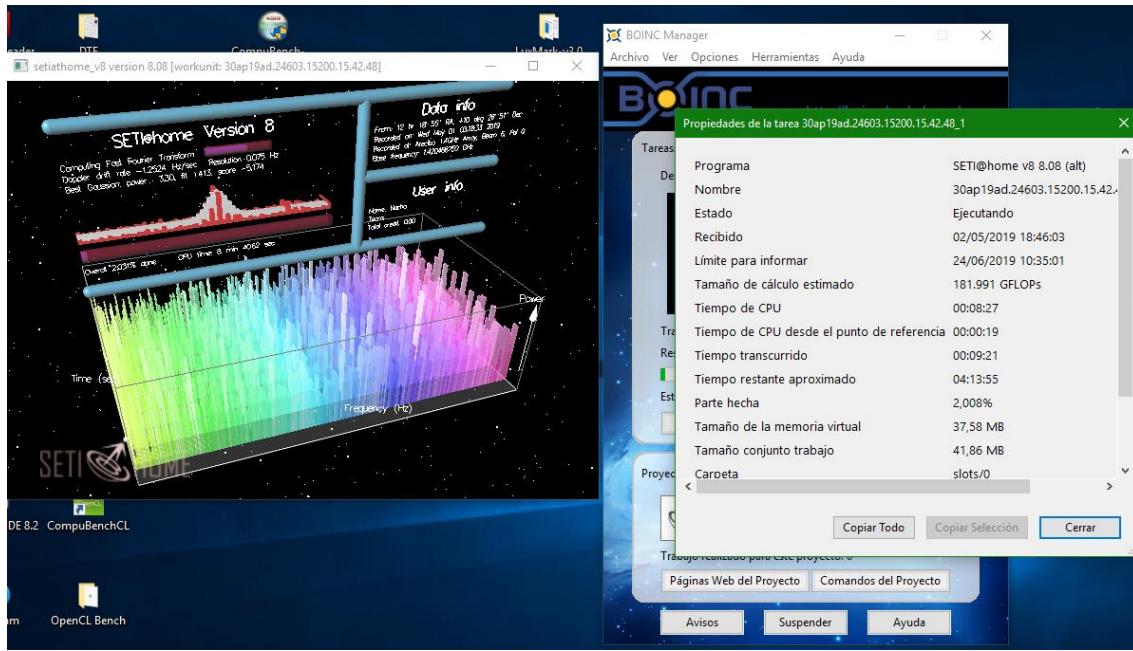
Ahora, vamos a hacer las pruebas en el proyecto seti@home. Vamos a adjuntar pantallazos también de pruebas que hicimos intentando unirnos a los proyectos GPUGRID y Denis.





Seleccionamos el proyecto a unirnos. Nos deja sin problema. Intentamos ejecutar carga de trabajo en nuestro ordenador: Ejecuta un test de rendimiento. Vemos que lo pasa, e inicia el test, prueba superada:

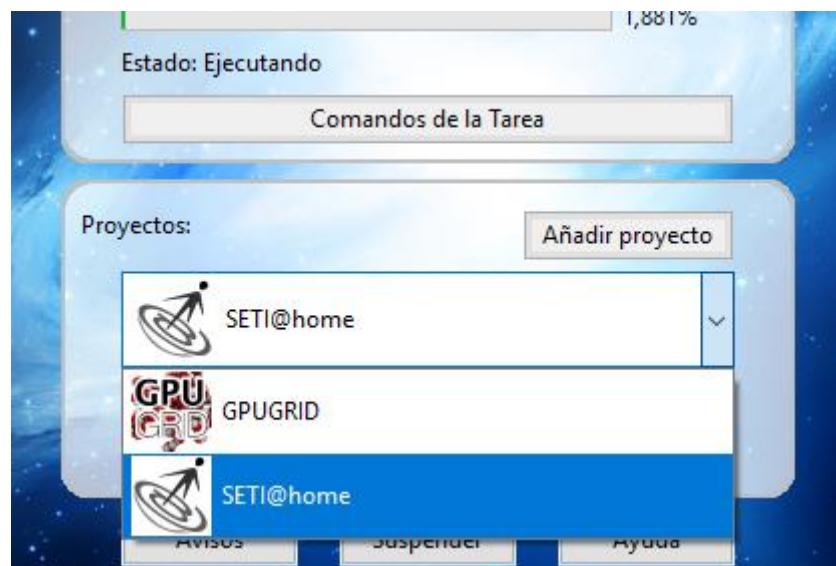
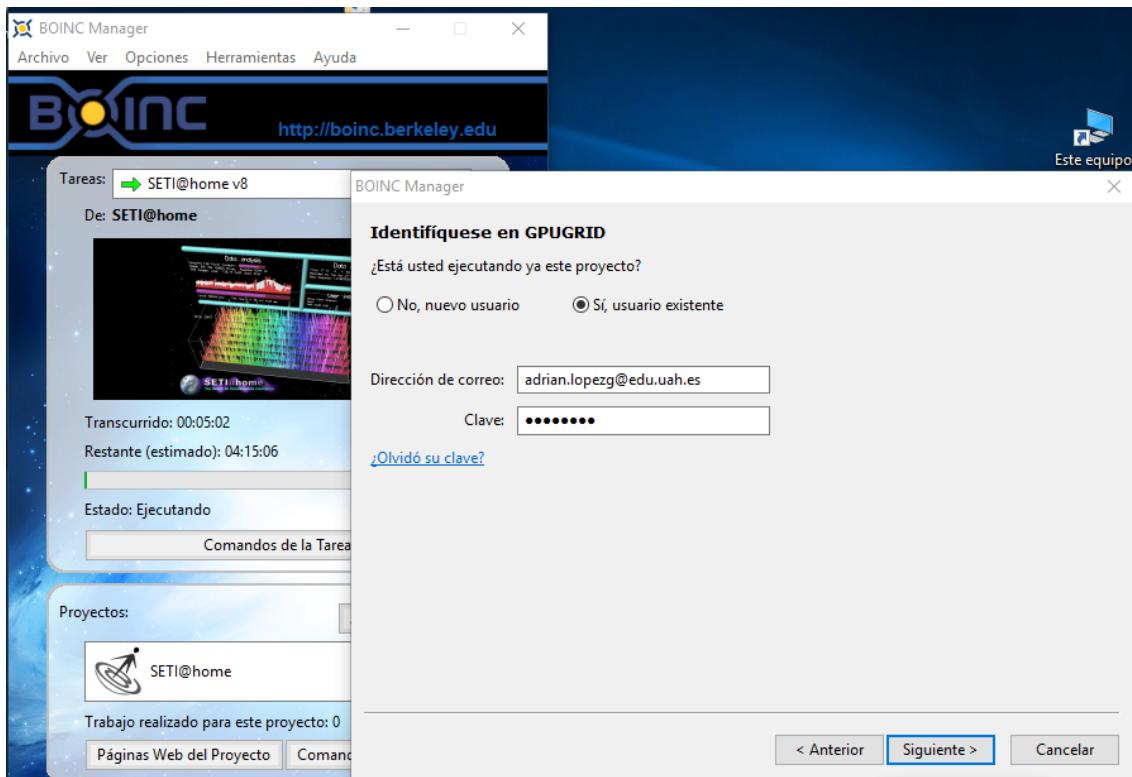




Además, observamos que está usando la tecnología OpenCL usando la gráfica también:

02/05/2019 18:42:47	Data directory: C:\ProgramData\BOINC
02/05/2019 18:42:47	Running under account Nacho
02/05/2019 18:42:48	CAL: ATI GPU 0: ATI Radeon HD 5400/R5 210 series (Cedar) (CAL ver...
02/05/2019 18:42:48	OpenCL: AMD/ATI GPU 0: ATI Radeon HD 5400/R5 210 series (Cedar...
02/05/2019 18:42:48	Creating new client state file
02/05/2019 18:42:48	Host name: DESKTOP-OVHMBOP
02/05/2019 18:42:48	Processor: 4 GenuineIntel Intel(R) Core(TM) i3 CPU M 330 @ 2.13GH...
02/05/2019 18:42:48	Processor features: fpu vme de pse tsc msr pae mce cx8 apic sep mt...
02/05/2019 18:42:48	OS: Microsoft Windows 10: Core x64 Edition, (10.00.16299.00)
02/05/2019 18:42:48	Memory: 3.86 GB physical, 4.93 GB virtual
02/05/2019 18:42:48	Disk: 464.45 GB total, 406.81 GB free
02/05/2019 18:42:48	Local time is UTC +2 hours
02/05/2019 18:42:48	No WSL found.
02/05/2019 18:42:48	Last benchmark was 18018 days 16:42:46 ago
02/05/2019 18:42:53	No general preferences found - using defaults
02/05/2019 18:42:53	Preferences:
02/05/2019 18:42:53	max memory usage when active: 1978.25 MB
02/05/2019 18:42:53	max memory usage when idle: 3560.85 MB
02/05/2019 18:42:53	max disk usage: 406.71 GB
02/05/2019 18:42:53	don't use GPU while active

El resultado es satisfactorio. Sin embargo, no con todos los programas fue así. Con gpugrid@home y con Denis@home no pudimos conectarnos, esto es debido a que eran proyectos obsoletos.

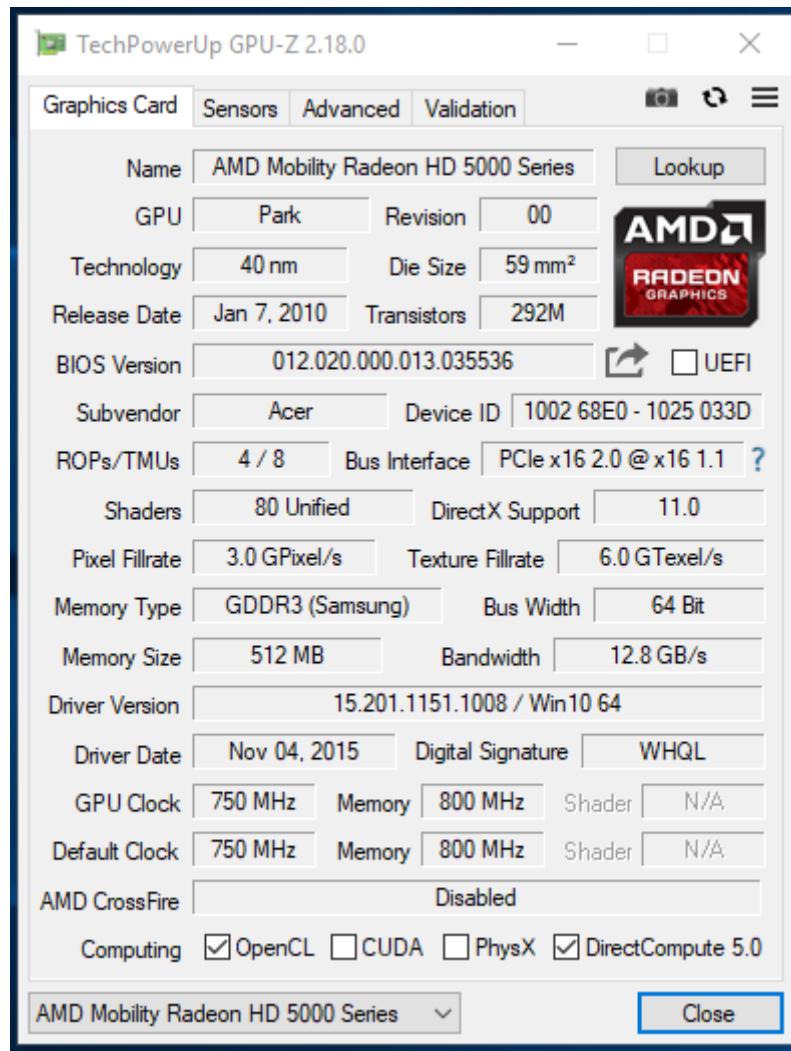


El error consistía en que nunca llegaban lotes de trabajo de los dos anteriores proyectos, siempre llegaban de Seti@home. Ello es porque ya no hay forma de colaborar, son proyectos que datan de principios del 2000 y dependen de universidades públicas. Sin embargo, el proyecto SETI tiene detrás a la Universidad de Berkeley, en USA, y de momento se prevé que sigan aportando apoyo.

## 5.4 Documentación de instalación

Para comenzar la instalación, lo primero es contar con un equipo adecuado. Y dentro del equipo adecuado, tener una gráfica compatible con la tecnología que queramos implementar, como hemos visto en el trabajo anterior, es vital.

Nuestra gráfica es una AMD HD5470, como vemos totalmente compatible:

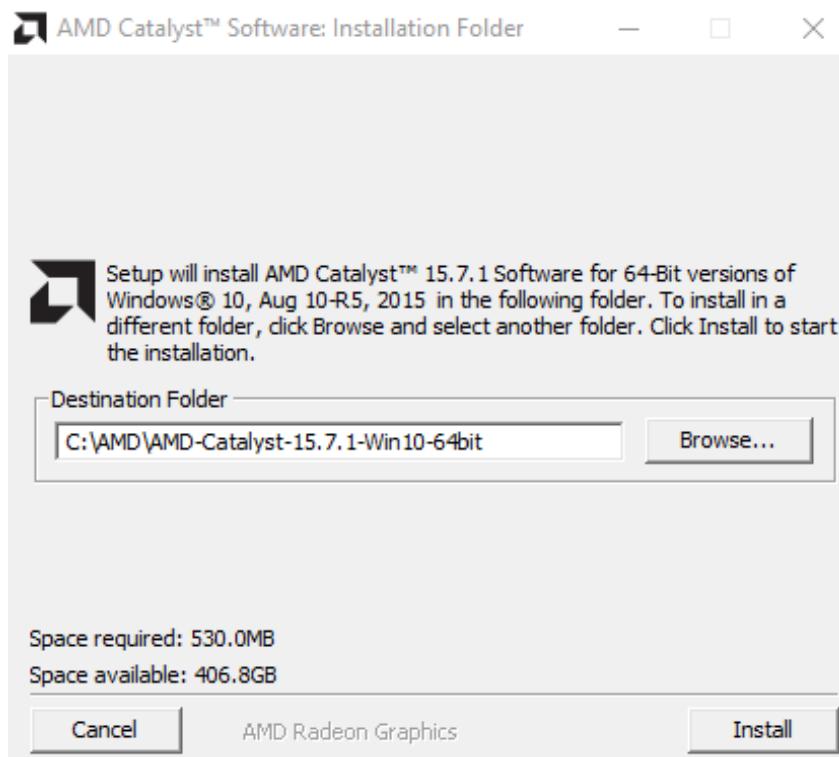


En nuestro caso, observamos que es compatible con la tecnología OpenCL pero no CUDA al ser una gráfica AMD.

Lo siguiente será ver si nuestra CPU es compatible con las instrucciones necesarias (SSE2 mínimo):

Name	Intel Core i3 330M			
Code Name	Arrandale	Max TDP 25.0 W		
Package	Socket 989 rPGA			
Technology	32 nm	Core Voltage		
Specification	Intel® Core™ i3 CPU M 330 @ 2.13GHz			
Family	6	Model	5	Stepping 2
Ext. Family	6	Ext. Model	25	Revision C2
Instructions	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x			

Lo siguiente será instalar los drivers adecuados, en nuestro caso AMD Catalyst:



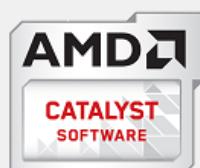


## Bienvenido

Bienvenido

Bienvenido

El Administrador de instalación Catalyst™ se utiliza para instalar y actualizar el software para sus productos gráficos



Soporte de idioma

¿Qué idioma desea que muestre el Administrador de Instalación Catalyst™?

Español (Spanish)

<http://www.amd.com>

Siguiente >

Cancelar

Ahora ya, se inicia el proceso de instalación propiamente dicho:

## Analizando el sistema

Bienvenido

Analizando la configuración

Analizar

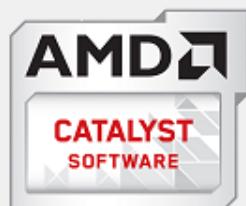
Progreso general

Personalizar

Enumerando medios de origen para los paquetes instalables.

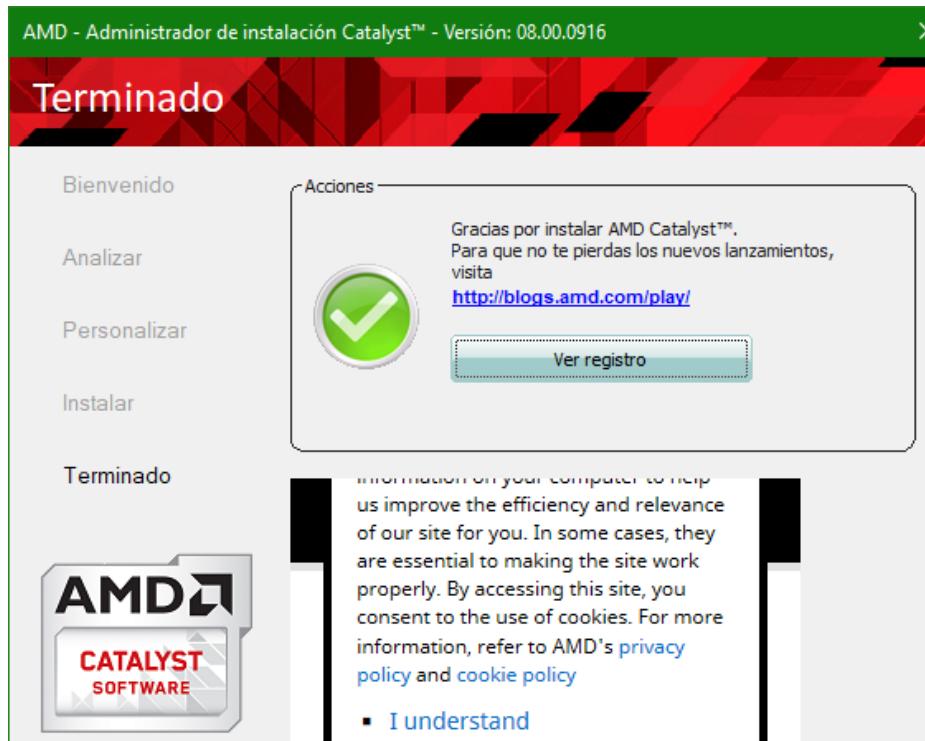
Instalar

Terminado



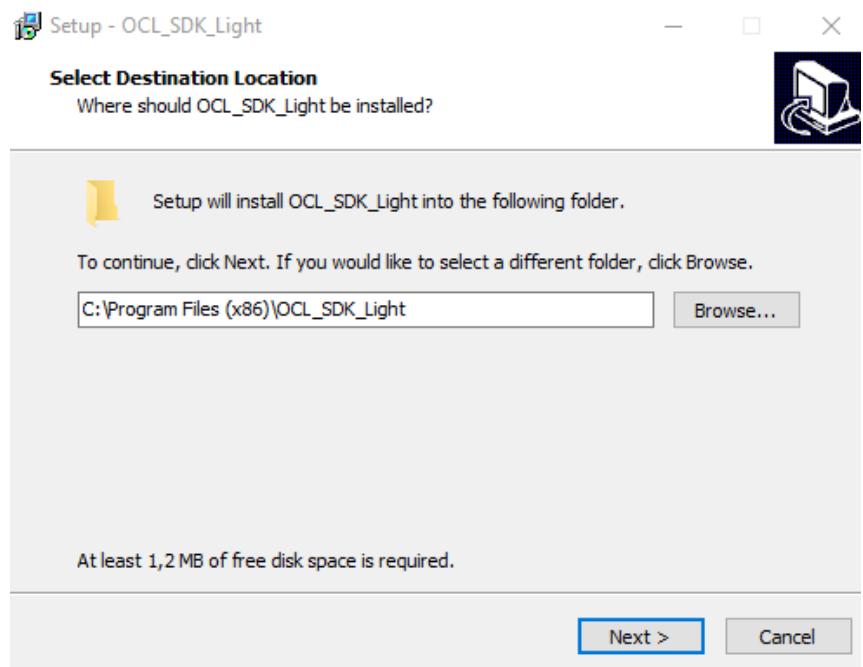
information on your computer to help us improve the efficiency and relevance of our site for you. In some cases, they are essential to making the site work properly. By accessing this site, you consent to the use of cookies. For more information, refer to AMD's [privacy policy](#) and [cookie policy](#)

I understand

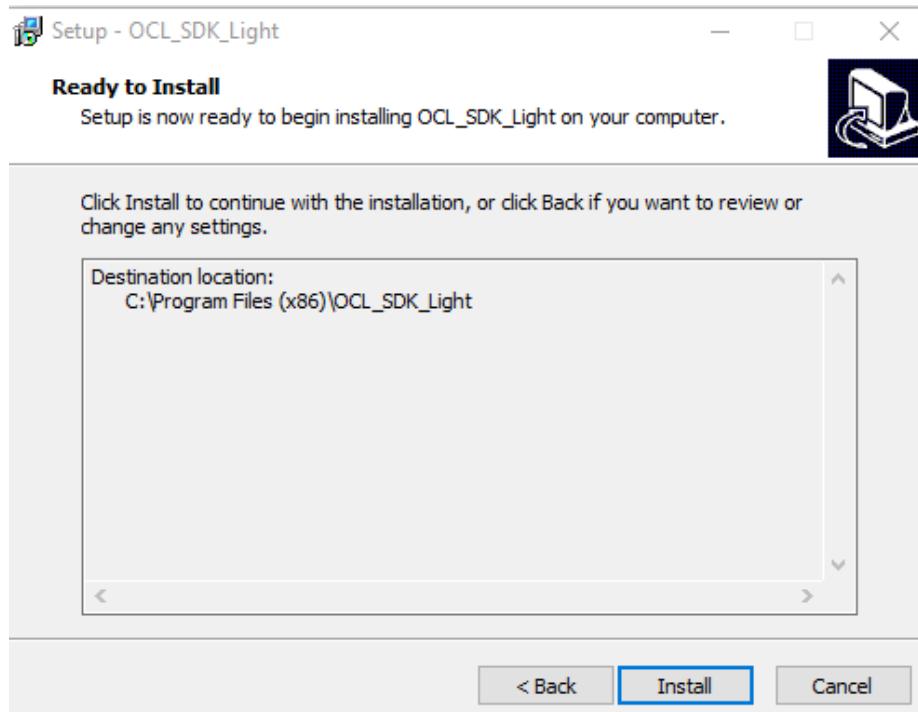


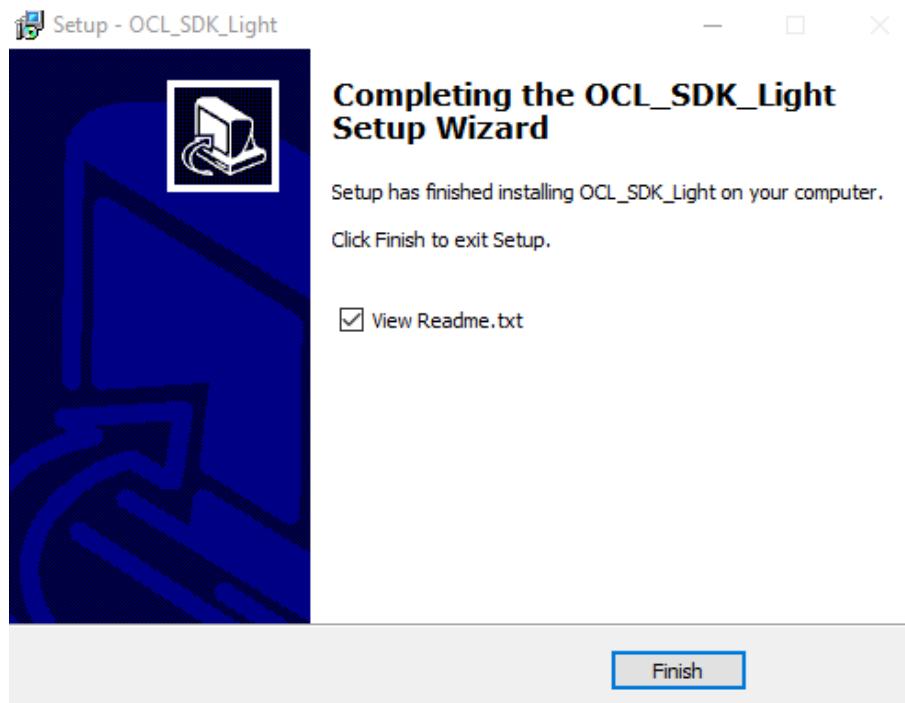
Ahora, tenemos que descargar del repositorio Github oficial, el SDK Bluelight, que es una versión ligera del SDK, y luego habrá que instalar la completa:

The screenshot shows the GitHub repository page for `GPUOpen-LibrariesAndSDKs / OCL-SDK`. The repository has 16 watchers, 50 stars, and 15 forks. A modal window is open, prompting the user to "Be notified of new releases" by creating a free GitHub account. The main content area shows the "Releases" tab selected, displaying the "Latest release" (version 3.0, released on 27 Jun 2017). The release notes mention "BenjaminCoquelle released this on 27 Jun 2017 · 1 commit to master since this release". The download link for the release is [lightOCLSDK.zip](#). Below the release, there is a section for "Assets" containing three items: `OCL_SDK_Light_AMD.exe`, `Source code (zip)`, and `Source code (tar.gz)`.

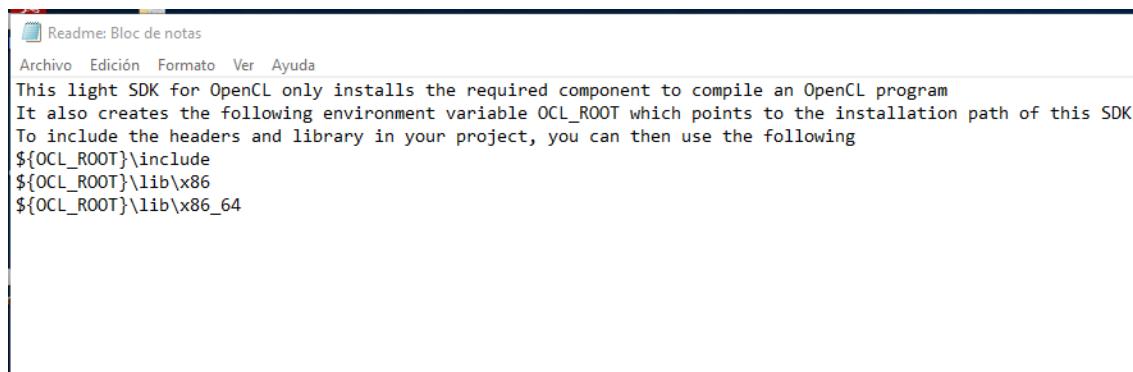


Procedemos a instalarlo:

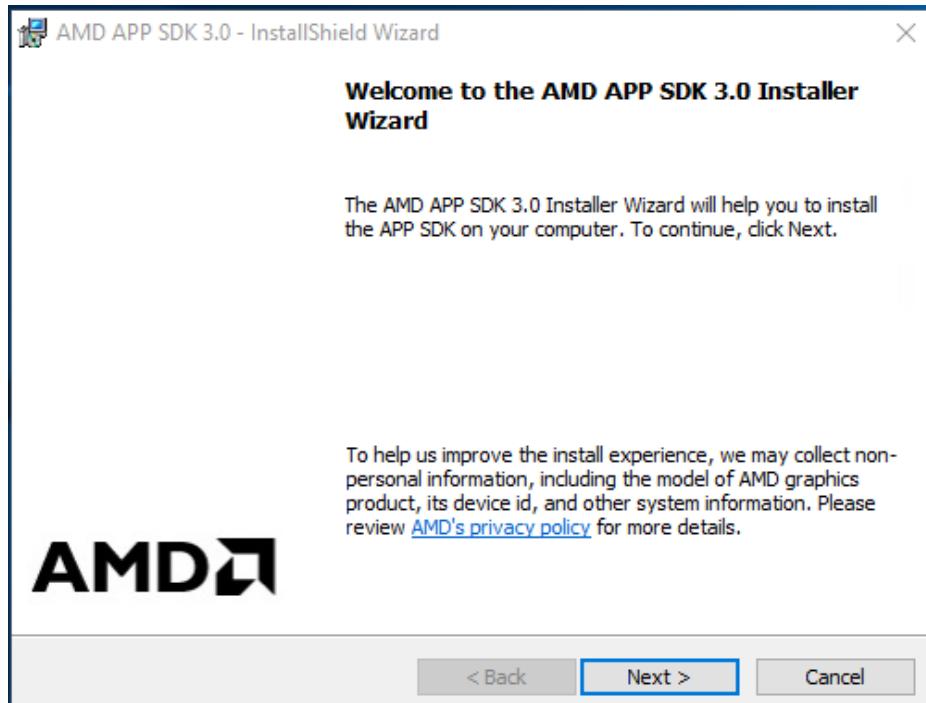
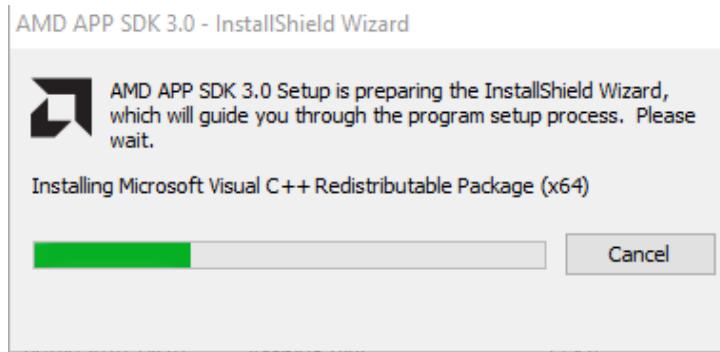




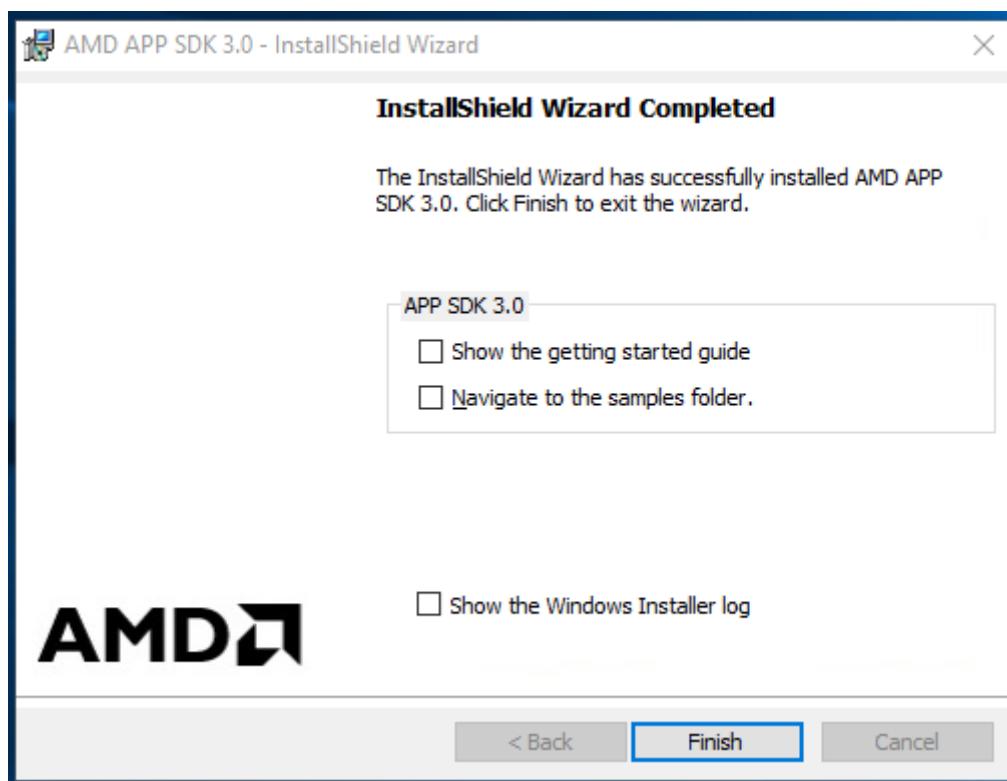
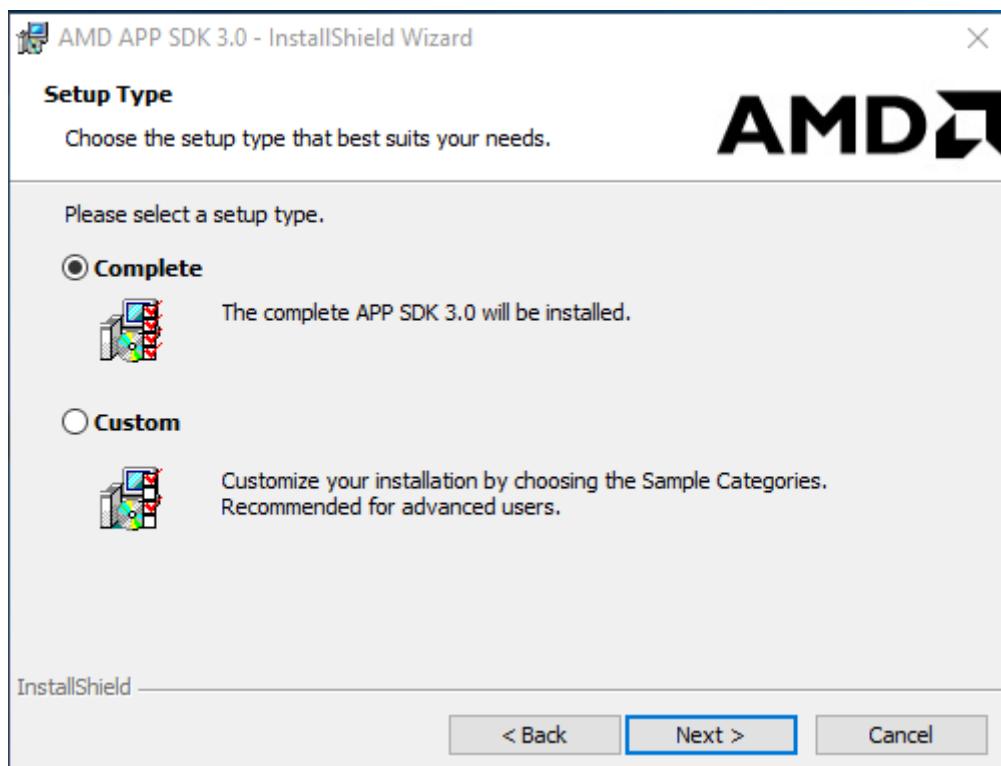
Vemos que se ha instalado correctamente:



Ahora, tenemos que instalar el SDK APP V3.0 de AMD. Su instalación es sencilla, viene contenido todo lo necesario en un instalador oficial de AMD, de unos 200MB:



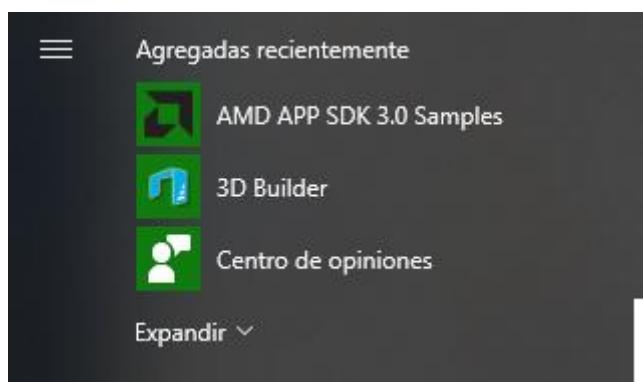
Elegimos instalación completa, para que verifique que están instalados los drivers de OpenCL:



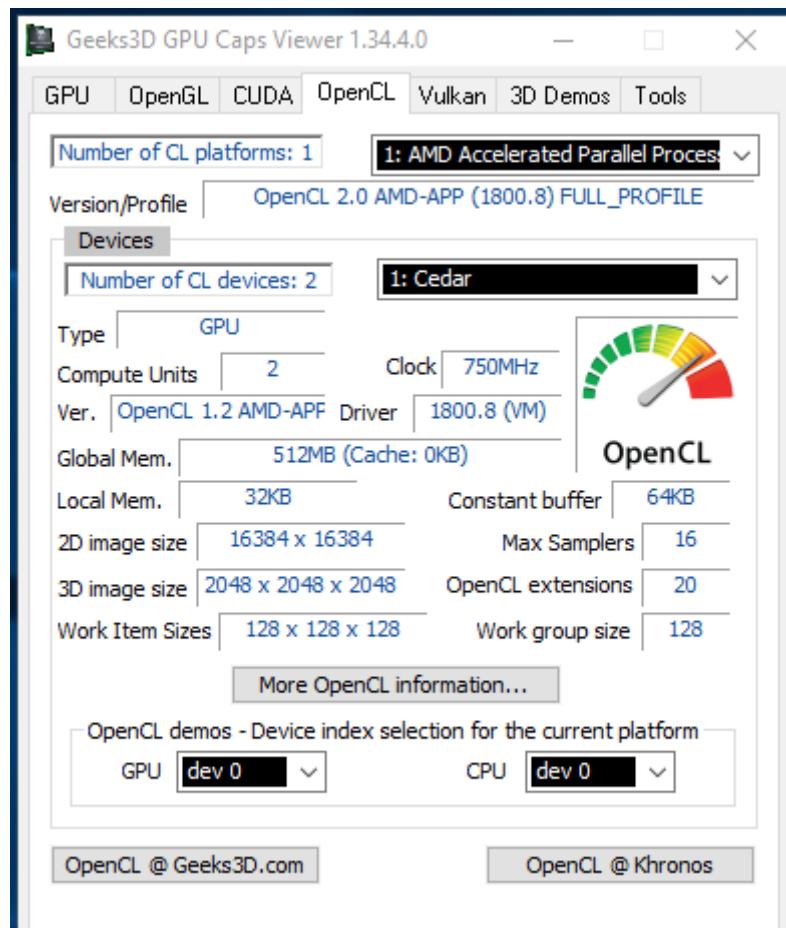
Comprobamos que está instalado correctamente seleccionando "Show the Windows Installer log":

```
MSI642d5: Bloc de notas
Archivo Edición Formato Ver Ayuda
== Verbose logging started: 02/05/2019 13:50:51 Build type: SHIP UNICODE 5.00.10011.00 Calling process: C:\WINDOWS\SYST
MSI (c) (34:84) [13:50:52:704]: Font created. Charset: Req=0, Ret=0, Font: Req=MS Shell Dlg, Ret=MS Shell Dlg
MSI (c) (34:84) [13:50:52:704]: Font created. Charset: Req=0, Ret=0, Font: Req=MS Shell Dlg, Ret=MS Shell Dlg
MSI (c) (34:70) [13:50:53:001]: Resetting cached policy values
MSI (c) (34:70) [13:50:53:001]: Machine policy value 'Debug' is 0
MSI (c) (34:70) [13:50:53:001]: ***** RunEngine:
***** Product: C:\ProgramData\Downloaded Installations\{B820B816-B8F1-49BE-AF62-50CBD1BD11E8}\AMD APP SDK 3.0.1
***** Action:
***** Commandline: *****
MSI (c) (34:70) [13:50:53:001]: Machine policy value 'DisableUserInstalls' is 0
MSI (c) (34:70) [13:50:53:001]: SOFTWARE RESTRICTION POLICY: Verifying package --> 'C:\ProgramData\Downloaded Installations
MSI (c) (34:70) [13:50:53:001]: SOFTWARE RESTRICTION POLICY: C:\ProgramData\Downloaded Installations\{B820B816-B8F1-49BE-AF
MSI (c) (34:70) [13:50:57:658]: SOFTWARE RESTRICTION POLICY: C:\ProgramData\Downloaded Installations\{B820B816-B8F1-49BE-AF
MSI (c) (34:70) [13:50:57:658]: Cloaking enabled.
MSI (c) (34:70) [13:50:57:658]: Attempting to enable all disabled privileges before calling Install on Server
MSI (c) (34:70) [13:50:57:658]: End dialog not enabled
MSI (c) (34:70) [13:50:57:658]: Original package ==> C:\ProgramData\Downloaded Installations\{B820B816-B8F1-49BE-AF62-50CBD
MSI (c) (34:70) [13:50:57:673]: Package we're running from ==> C:\ProgramData\Downloaded Installations\{B820B816-B8F1-49BE-AF62-50CBD
MSI (c) (34:70) [13:50:57:673]: APPCOMPAT: Compatibility mode property overrides found.
MSI (c) (34:70) [13:50:57:673]: APPCOMPAT: looking for appcompat database entry with ProductCode '{8829787C-6269-4112-ADF8-000000000000'
MSI (c) (34:70) [13:50:57:673]: APPCOMPAT: no matching ProductCode found in database.
MSI (c) (34:70) [13:50:57:689]: MSCREEE not loaded loading copy from system32
MSI (c) (34:70) [13:50:57:720]: Machine policy value 'TransformsSecure' is 0
MSI (c) (34:70) [13:50:57:720]: User policy value 'TransformsAtSource' is 0
```

Y vemos que en Windows 10 aparece instalado:



GPUCaps nos confirma que estamos ejecutando OpenCL:



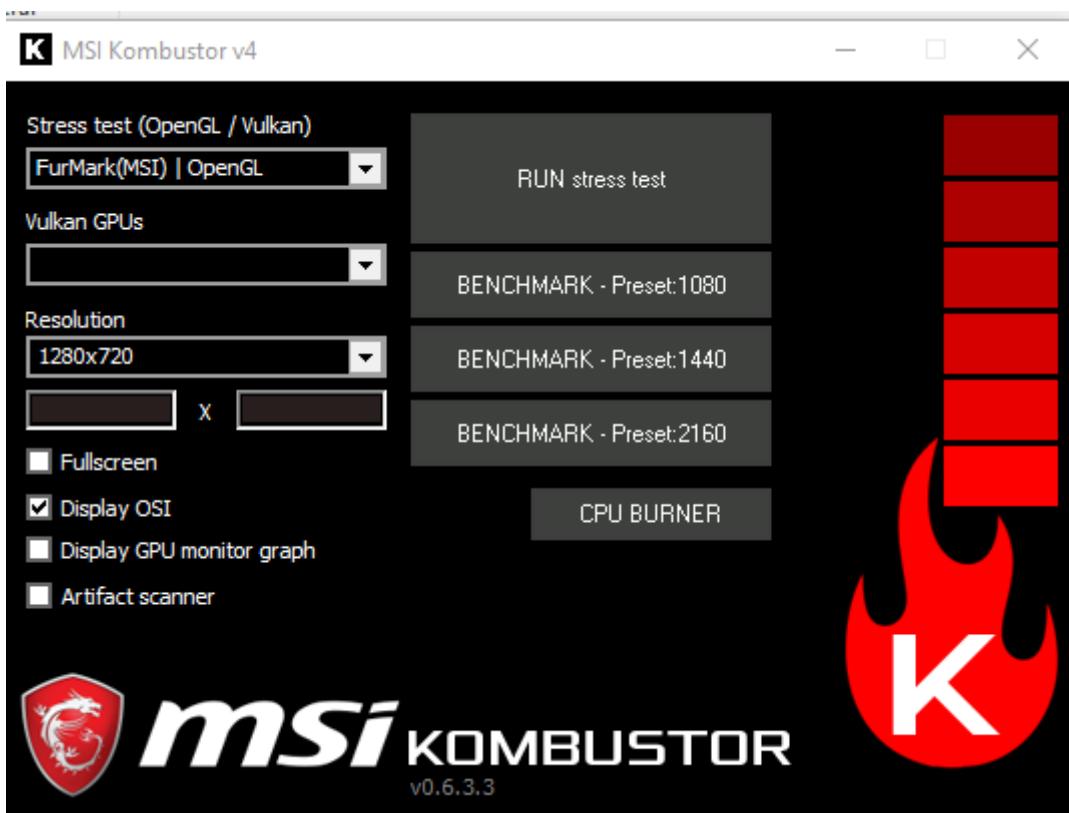
## 5.5 Manual de usuario

Una vez supuesto la instalación del hardware, vamos a ver cómo instalar y qué son los diferentes programas que hemos usado en el apartado anterior.

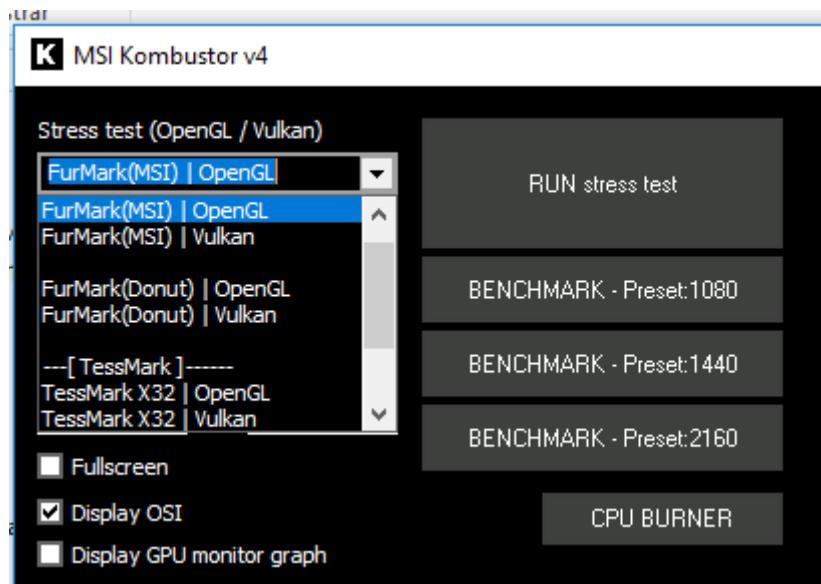
Vamos a comenzar instalando la utilidad MSI Kombustor. Es una suite software que incluye varios programas para realizar test de estrés. Podemos bajarlo gratuitamente de la siguiente página:

- <https://geeks3d.com/furmark/kombustor/>

Lo abrimos, tras instalarlo, y seleccionamos el test que queremos. La pantalla principal es la siguiente:



Tenemos que seleccionar la resolución que queremos, y el test a ejecutar. Para estresar GPU + CPU, lo mejor es usar el test Furmark, poner la resolución más alta a la que vaya a trabajar el equipo, y pulsamos “RUN stress test”:



Arrojando como resultado el que vimos en los casos de prueba:



Esto es igual para ambas plataformas. Tenemos que ver qué temperatura alcanzan ambos, si se bloquea, y cuál es capaz de alcanzar mayor número de FPS o Frames Por Segundo.

Vamos ahora a observar cómo funciona la suite de pruebas o benchmarking CompuBench. Emplea el desempeño de la gráfica usando OpenCL o CUDA si están los drivers correctamente instalados. Nos da una idea de cómo se comportará nuestro equipo y en qué tareas puede ser empleado.

The screenshot shows the CompuBench Desktop landing page. At the top, there's a navigation bar with links for CompuBench, Results, Compare, The Benchmark, News, About Us, Follow, Register or Login, and a dropdown menu. Below the navigation is a sub-navigation bar with links for CompuBench RS, CompuBench 1.5 Desktop, CompuBench 1.5 Mobile, and CompuBench 2.0 Desktop. The main content area features the title "CompuBenchDesktop" in large green letters, followed by a subtitle: "CompuBench measures the compute performance of your OpenCL and CUDA device." A button labeled "Download from Here" with a checkmark icon is visible. To the right is a photograph of two small, translucent gummy bears (one green, one red) standing on a reflective silver object, possibly a car door handle, with sunlight reflecting off the surface.

Es multiplataforma, multidispositivos, y cuenta con las siguientes pruebas:

The screenshot shows the CompuBench results page for the "Face Detection" test. At the top, there's a header with a lock icon and the URL "https://compubench.com/result.jsp?benchmark=compu15d". Below the header is a navigation bar with categories: Recetas, Programación, BBDD, EEDD, Tracking, Parking, SysAdmin, Pi, Tiendas, Web. The main content area has a sidebar on the left with the text "Choose a test:" and "1 to 20 of 2831". It lists several test configurations with NVIDIA and OpenCL icons. On the right, a large table displays the results for the Face Detection test. The table has columns for GPU (NVIDIA TITAN Xp), API (OpenCL), Date (2017.04.17), and Result (363.602 mPixels/s). The result row for the NVIDIA TITAN Xp is highlighted in green.

Face Detection			
NVIDIA	OpenCL	2019.01.08   NVIDIA GeForce RTX 2080 Ti	
NVIDIA	OpenCL	2017.04.17   NVIDIA TITAN Xp	363.602 mPixels/s

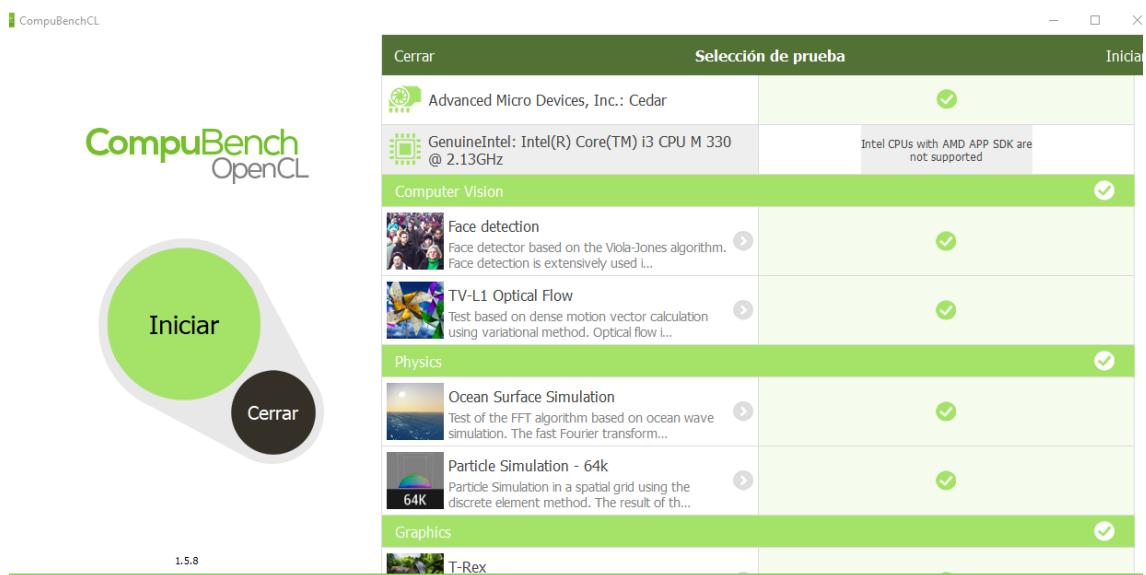
Nos lo bajamos de su sitio oficial, en nuestro caso usaremos la versión gratuita “CompuBench Desktop 1.5”:

- <https://compubench.com/result.jsp?benchmark=compu15d>

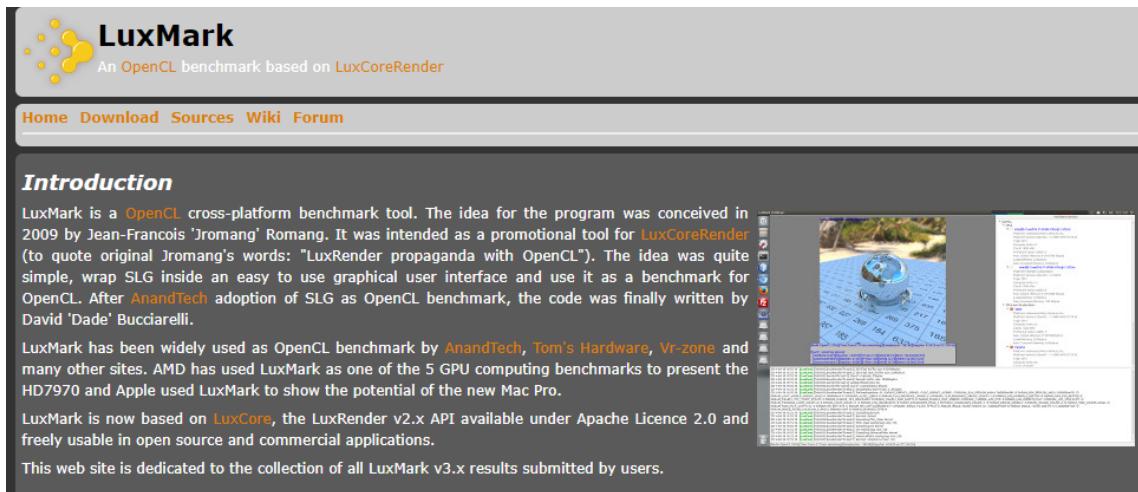
Lo ejecutamos tras instalarlo:



Y vamos a la pantalla principal, donde podremos elegir test a realizar. Damos a iniciar, y comenzará la prueba, con el resultado visto en el documento de pruebas:



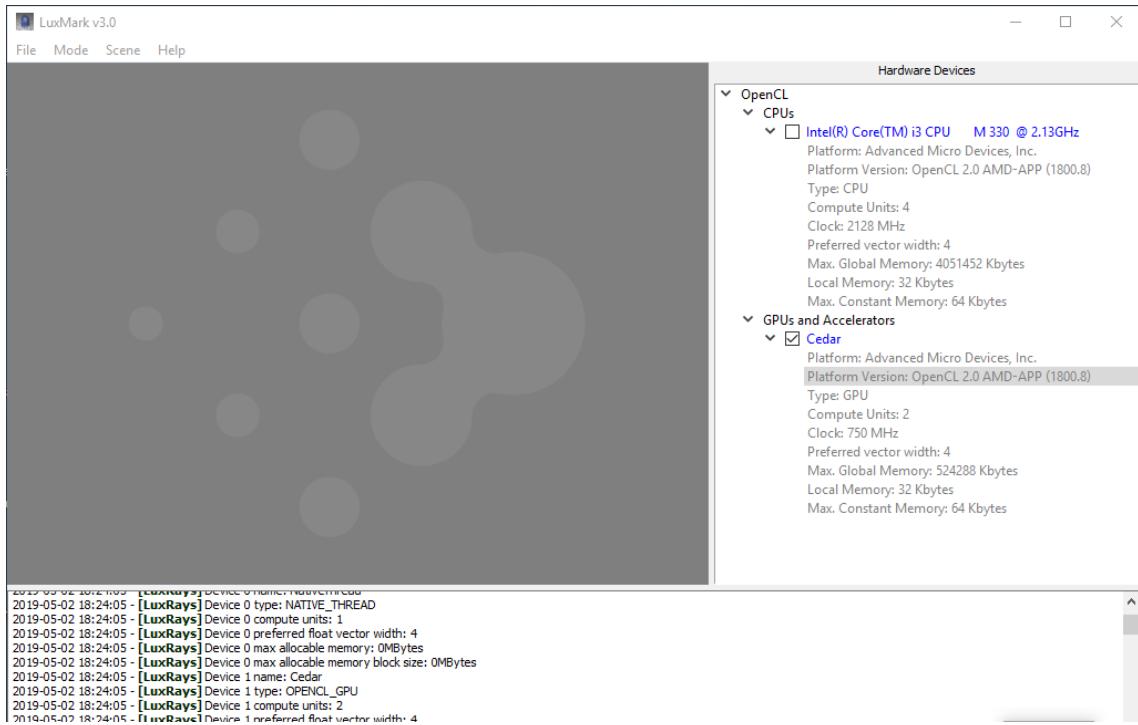
El software siguiente es la suite Luxmark, que mide de una forma más sencilla, ejecutando algoritmos matemáticos, qué equipo es más potente. Emplea también la potencia OpenCL y CUDA de las gráficas. En origen servía para comprobar la potencia OpenCL, pero se fue abriendo a más tecnologías:



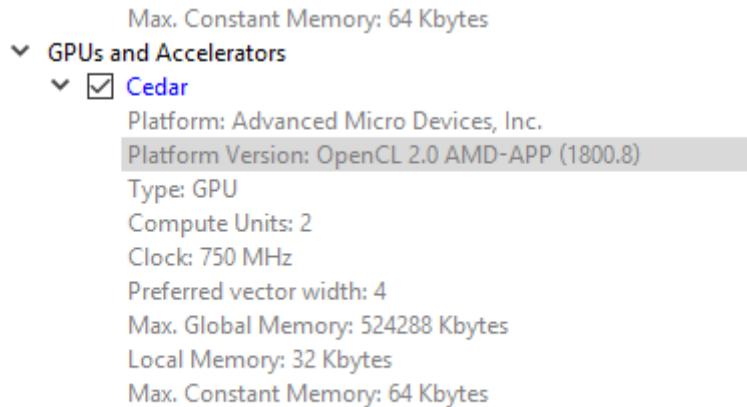
Es multiplataforma. Lo podemos obtener del siguiente servidor oficial:

- <http://wiki.luxcorerender.org/LuxMark#Binaries>

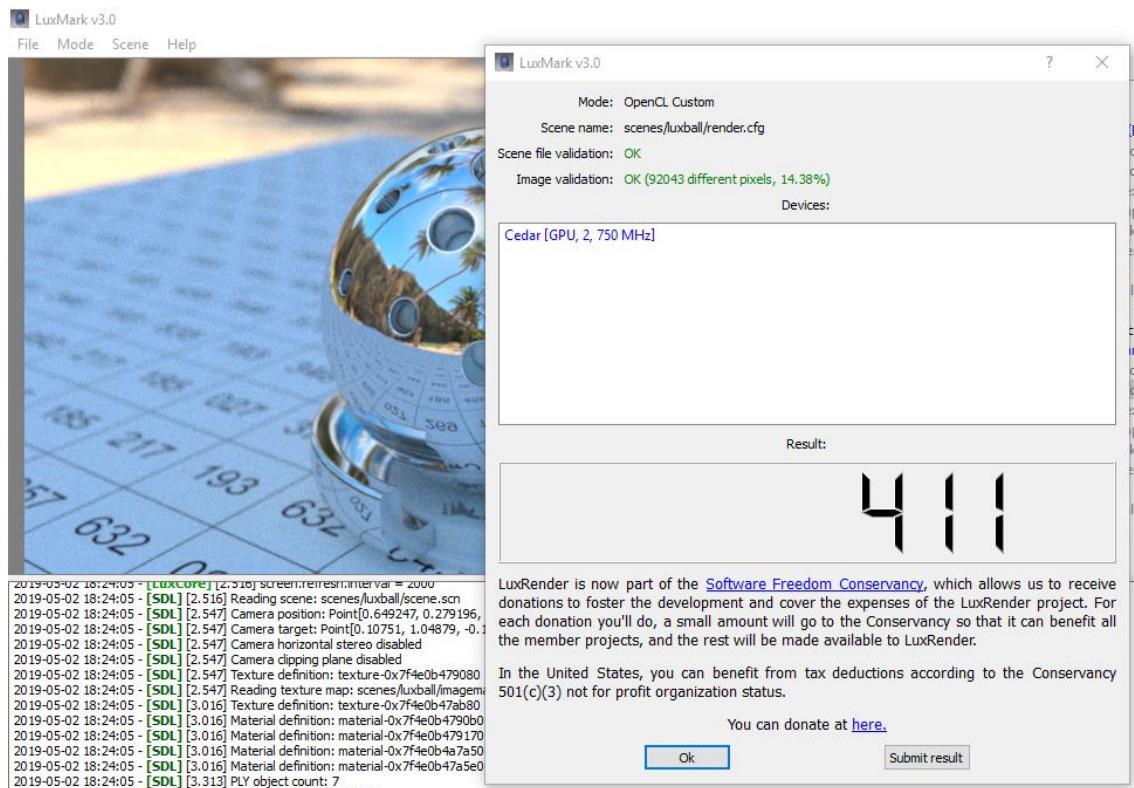
Al ejecutarlo por primera vez, veremos un diseño muy sobrio:



Observamos en el panel izquierdo, que es capaz de detectarnos la gráfica y los drivers OpenCL.



Ahora, solo tenemos que pulsar en Mode -> run y ejecutará las pruebas, dándonos un resultado final .En nuestro caso, hemos testeado solo la gráfica, pues en nuestro caso compiten un i3 y un i7:



Vamos ahora con nuestras pruebas científicas. En nuestro caso, nos vamos a unir a la red de ordenadores distribuidos del BOINC o “Compute for Science Project” iniciado por la universidad de Berkeley, y que nos permite donar poder computacional de nuestro equipo a diversas investigaciones científicas, desde búsqueda extraterrestre a cura del cáncer:

The screenshot shows the BOINC website at <https://boinc.berkeley.edu>. The page title is "Compute for Science". The main content area explains what BOINC is: helping cutting-edge science research using your computer (Windows, Mac, Linux) or Android device. It runs in the background and is safe. It mentions about 30 science projects like Einstein@Home, IBM World Community Grid, and SETI@home. Below this, there are two options: "Choose science areas" (which links to Science United) or "Choose projects" (which links to download BOINC). There are also social sharing icons (Facebook, Twitter, LinkedIn) at the bottom left.

## La búsqueda de aliens de SETI en crisis: el Bitcoin les está dejando sin GPUs para sus radiotelescopios



En nuestro caso, como acabamos de adelantar, hemos decidido emplear ambos equipos en Seti. Vamos a mostrar cómo hacerlo. Primero, ¿qué es Seti?:



## GPU computing

Most computers are equipped with a **Graphics Processing Unit (GPU)** that handles their graphical output, including the 3-D animated graphics used in computer games. The computing power of GPUs has increased rapidly, and they are now often much faster than the computer's main processor, or CPU.

Some BOINC-based projects have applications that run on GPUs. **These applications run from 10X to 200X faster than the CPU-only version depending on the application, CPU and GPU in question.** We urge BOINC participants to use them if possible. Just follow these instructions:

### Check whether your computer has a capable GPU

- Identify the model name of your GPU.
  - On Windows, use GPU-Z found [here](#). (note: The listed capabilities of the card may be inaccurate on multi GPU systems.)
  - On linux, in a console use: `lspci | grep VGA`
  - On Macintosh, Select *About this Mac* from the Apple menu, then click *More Info*. Under *Hardware* select *Graphics/Displays*.
- Note the Adapter Type and Memory Size.
- To find out if your NVIDIA GPU is compatible: check [NVIDIA's list of CUDA-enabled products](#). If your GPU is listed here and has at least 256MB of RAM, it's compatible.
- ATI GPUs: you need a platform based on the AMD R600 or AMD R700 GPU or later. R600 GPUs are found on [ATI Radeon HD2400, HD2600, HD2900 and HD3800](#) graphics board. R700 GPUs are found on HD4350 to HD4890 graphics boards. Check [ATI's list of OpenCL-enabled products](#) to see which GPUs use OpenCL.
- Intel GPUs: Currently [Ivy Bridge](#) and [Haswell](#) are the only Intel CPUs with an OpenCL capable Intel GPU, however future embedded GPUs may also support OpenCL. You will need to install [Intel graphics drivers](#) to enable OpenCL support. It's also needed to add a monitor or VGA dummy-plug before the Intel GPU is recognized.
- AMD APUs: The AMD Kaveri, Brazos, Kabini and Lynx platforms - Zacate and Kabini for tablet and embedded products and Llano, Trinity and Richland APUs for desktops and laptops- are capable of doing GPU work on their own.

*Note: Some projects may have additional requirements. Check with the website of your project of interest for more details.*

### Contents [hide]

- 1 Check whether your computer has a capable GPU
- 2 Get the latest BOINC software
- 3 Get the latest driver
- 4 Attach to projects with GPU applications
  - 4.1 Things to be aware of

Procedemos a instalarlo. Para ello, obtendremos el software oficial de BOINC de su propia página. Los requisitos previos, tener un ordenador que lo soporte con sistema Windows/Linux/Mac, y tener los drivers de la gráfica y CUDA/OpenCL instalados:

- <https://boinc.berkeley.edu/download.php>

## Install BOINC

BOINC es un programa que le permite donar el tiempo en que su computador está inactivo a proyectos científicos como SETI@home, Climateprediction.net, Rosetta@home, World Community Grid y muchos otros. Tras instalar BOINC en su computador, puede conectarlo con tantos proyectos como quiera. We recommend that you also install VirtualBox, so your computer can work on science projects that require it. [Saber más de VirtualBox](#).

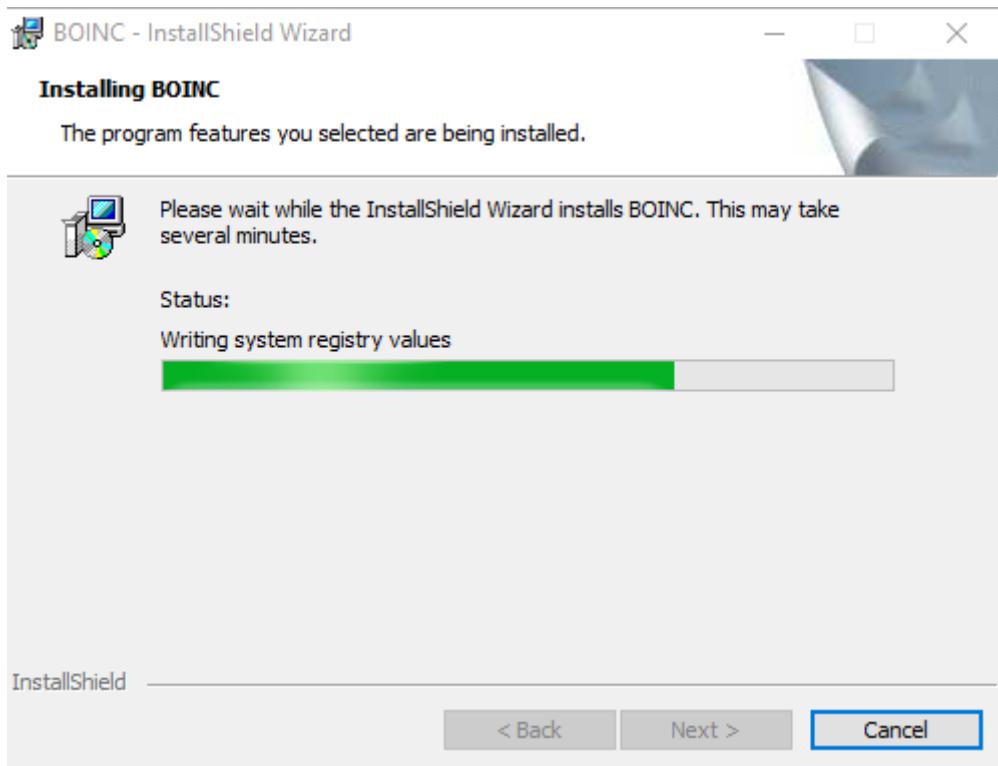
[Bajar BOINC y VirtualBox](#)  
for Windows 64-bit (92.93 MB)  
(BOINC 7.14.2, VirtualBox 5.2.8)

[Descargar BOINC](#)  
for Windows 64-bit (9.66 MB)  
(BOINC 7.14.2)

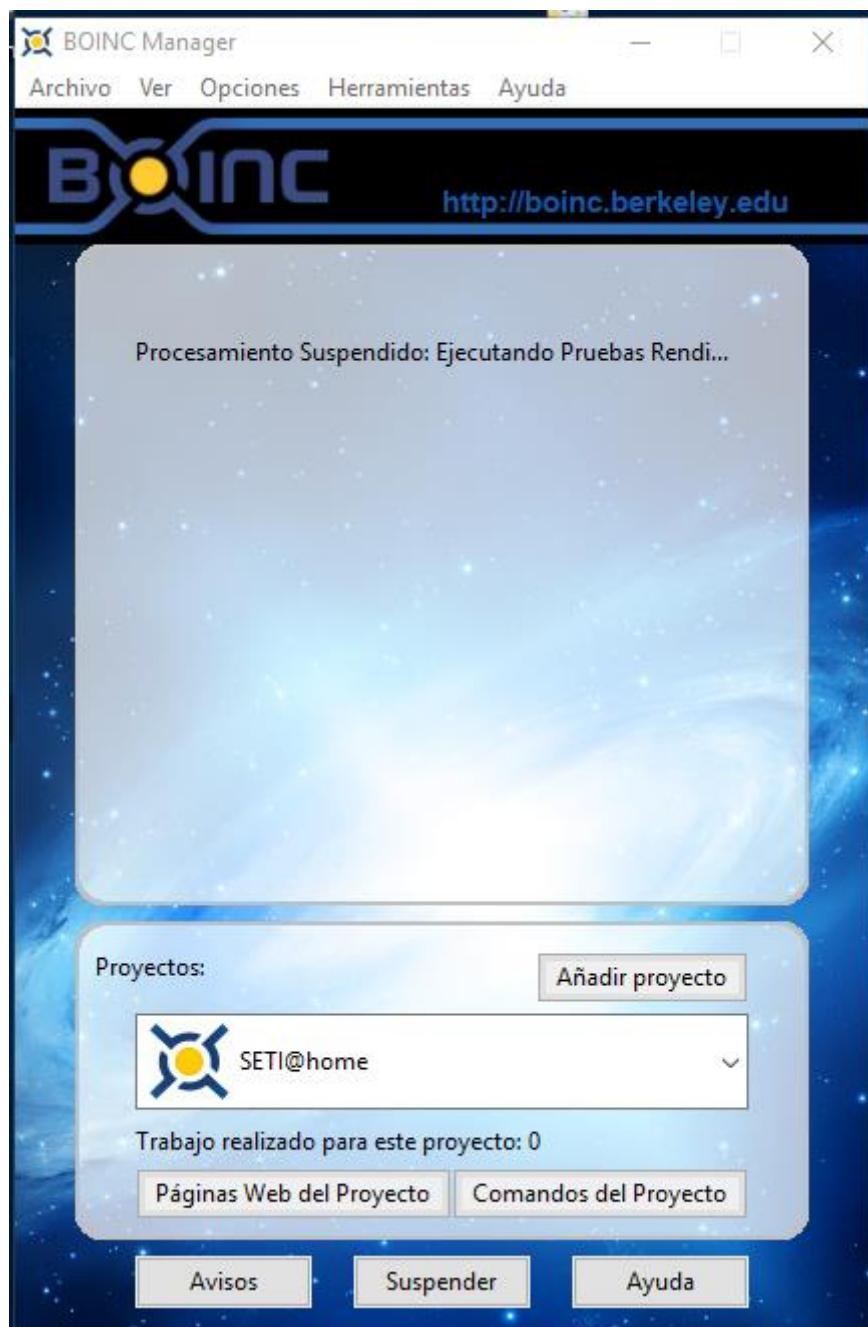
After downloading BOINC you must **install** it: typically this means double-clicking on the file icon when the download is finished.

When you first run BOINC, you will be asked to choose a project. For instructions, see the [BOINC User Manual](#).

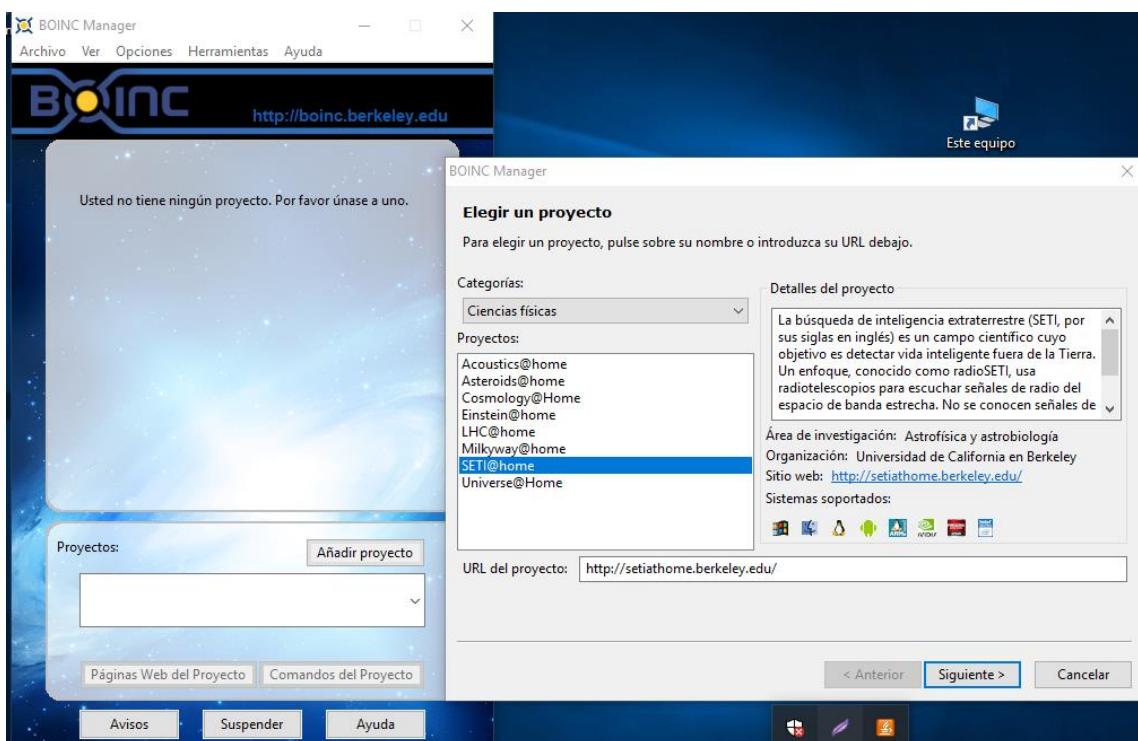
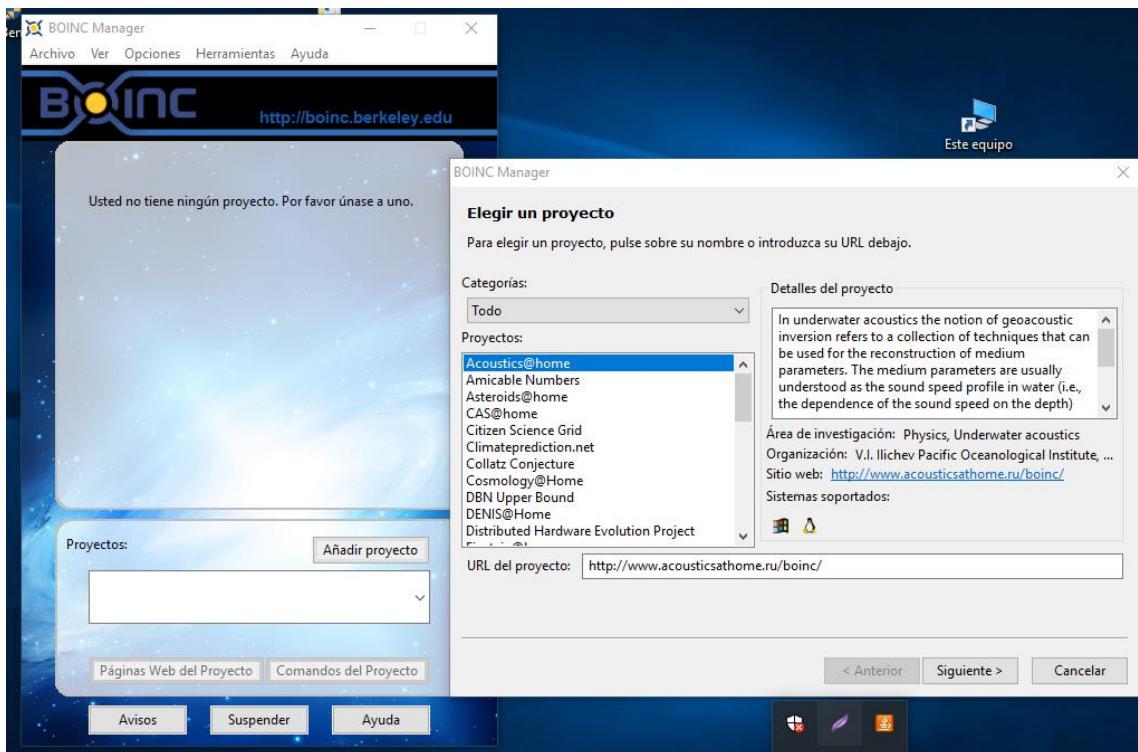
Debemos seleccionar la segunda opción y bajarnos el ejecutable. Ahora, configuraremos por primera vez el programa tras ejecutarlo:



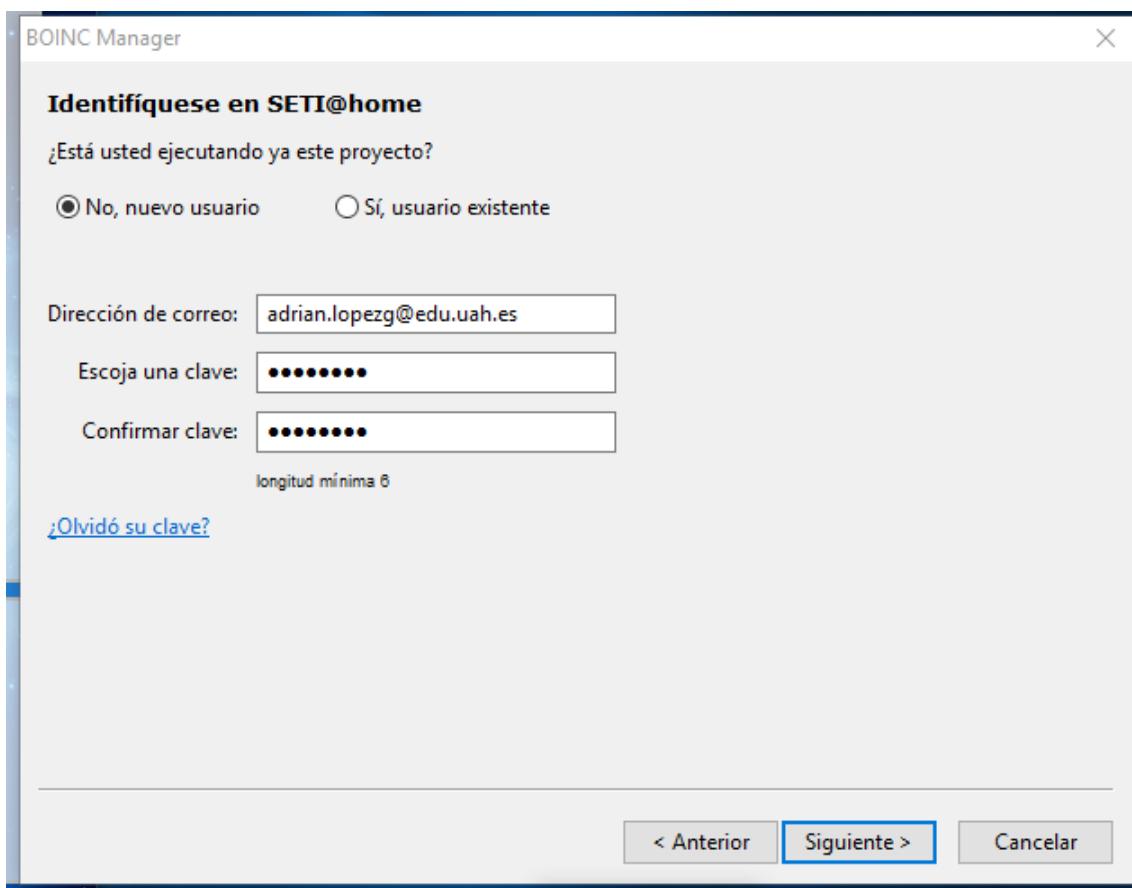
En el menú principal, seleccionamos “añadir proyecto”, y se abre la siguiente pantalla:



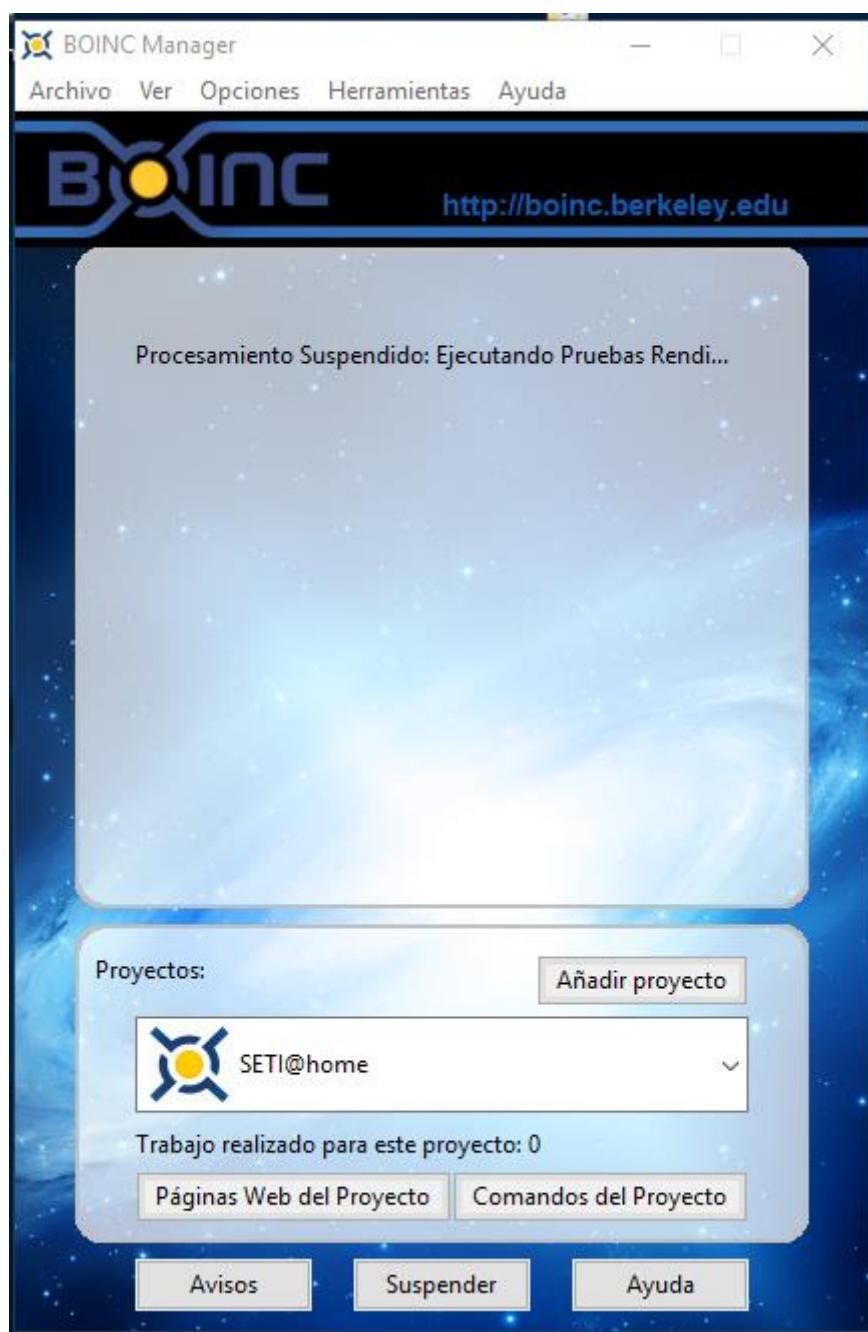
Seleccionamos el proyecto a añadir:

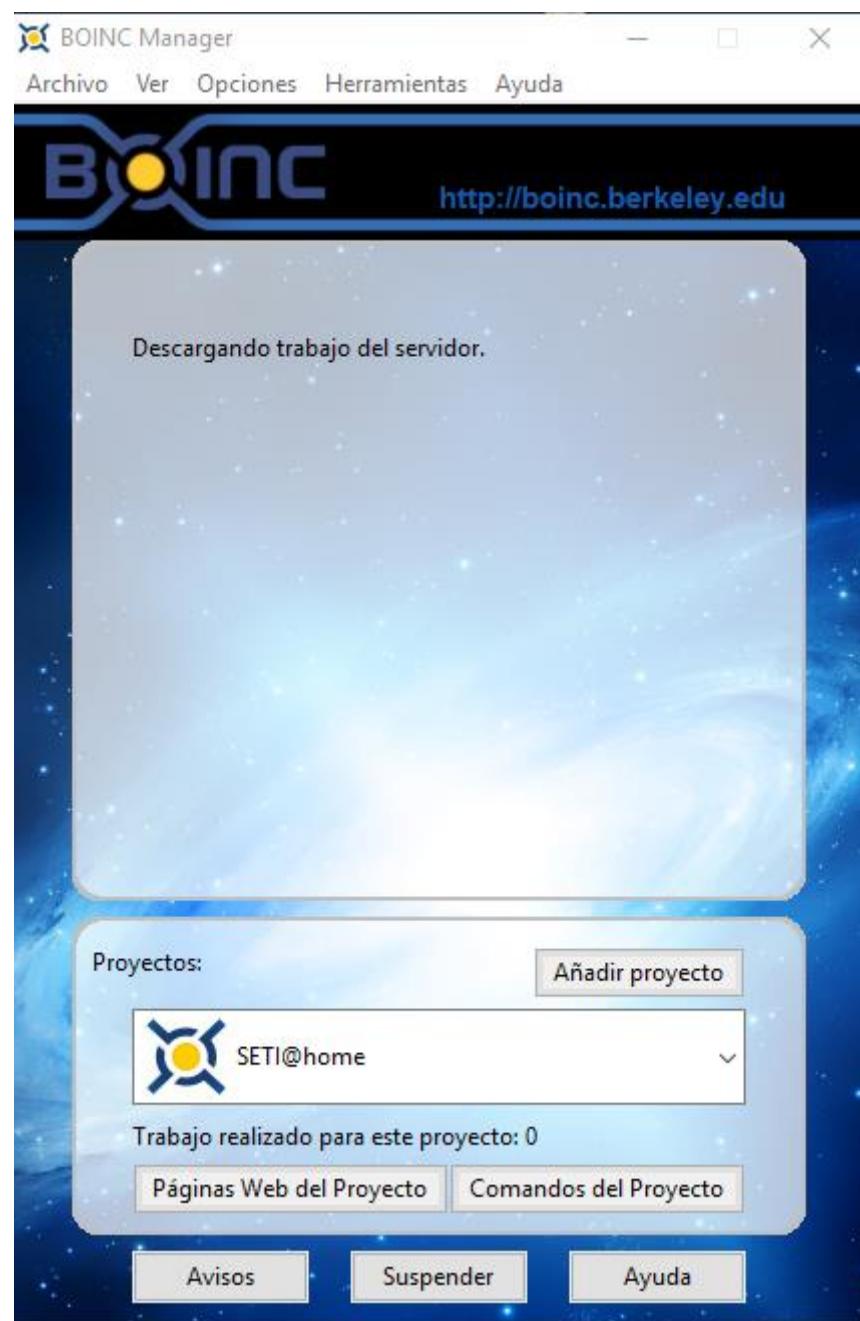


Creamos una cuenta para participar;



Y esperamos a que el servidor mida el rendimiento de nuestro equipo, y nos comience a mandar carga de trabajo:

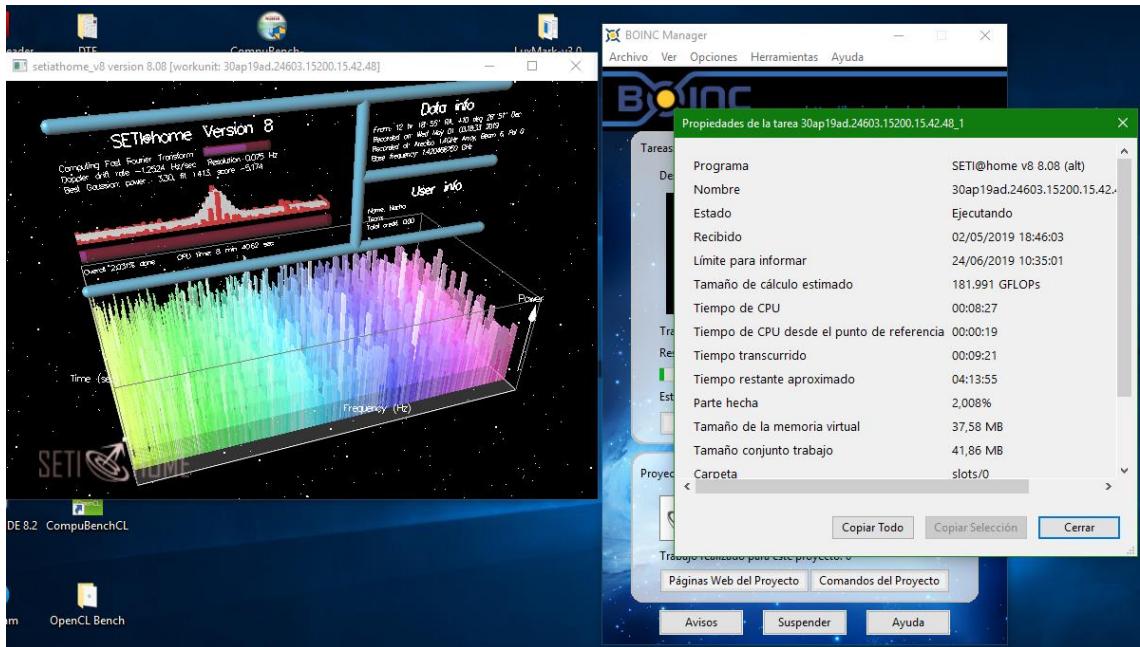




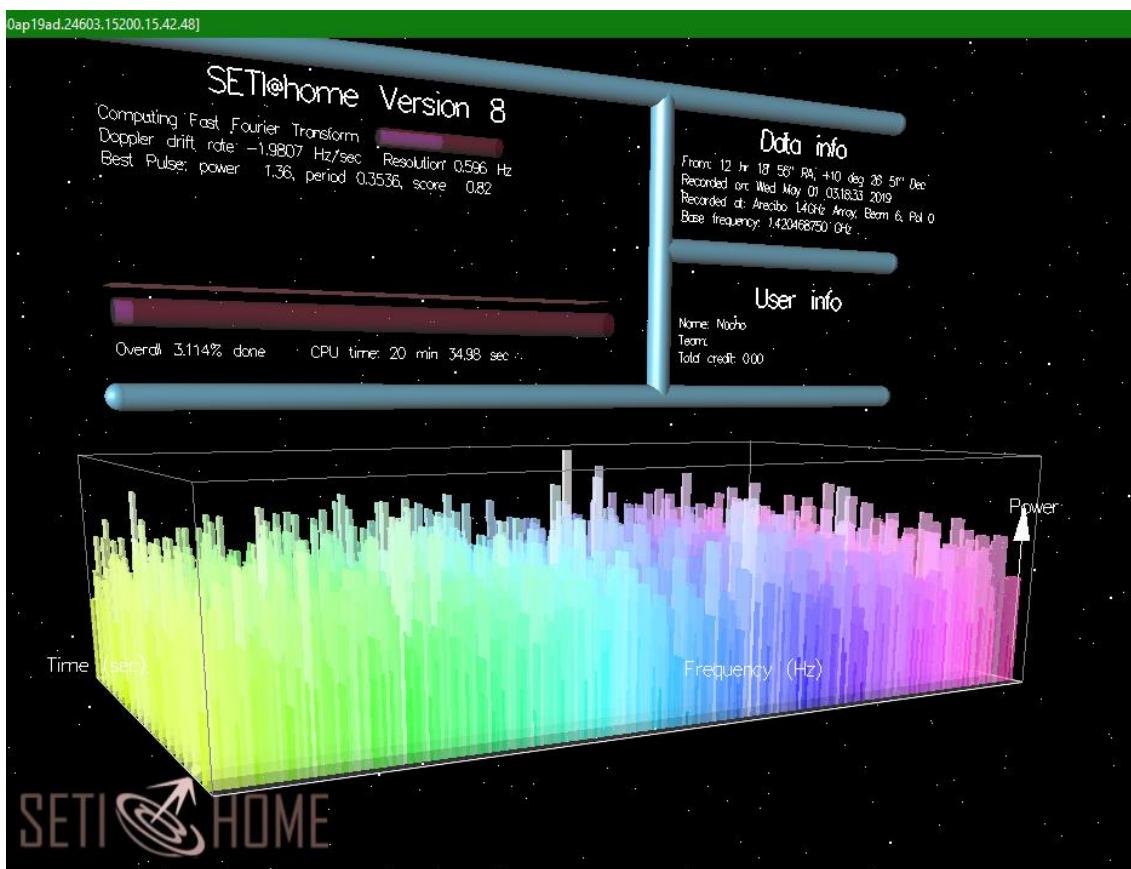
Si seguimos estos simples pasos, ya debemos ser capaces de ver el programa ejecutándose correctamente:



En “Comandos de la Tarea”, seremos capaces de ver cómo ejecuta nuestro equipo el trabajo mandado:



02/05/2019 18:42:47	Data directory: C:\ProgramData\BOINC
02/05/2019 18:42:47	Running under account Nacho
02/05/2019 18:42:48	CAL: ATI GPU 0: ATI Radeon HD 5400/R5 210 series (Cedar) (CAL ver...
02/05/2019 18:42:48	OpenCL: AMD/ATI GPU 0: ATI Radeon HD 5400/R5 210 series (Cedar...
02/05/2019 18:42:48	Creating new client state file
02/05/2019 18:42:48	Host name: DESKTOP-OVHMBOP
02/05/2019 18:42:48	Processor: 4 GenuineIntel Intel(R) Core(TM) i3 CPU M 330 @ 2.13GH...
02/05/2019 18:42:48	Processor features: fpu vme de pse tsc msr pae mce cx8 apic sep mt...
02/05/2019 18:42:48	OS: Microsoft Windows 10: Core x64 Edition, (10.00.16299.00)
02/05/2019 18:42:48	Memory: 3.86 GB physical, 4.93 GB virtual
02/05/2019 18:42:48	Disk: 464.45 GB total, 406.81 GB free
02/05/2019 18:42:48	Local time is UTC +2 hours
02/05/2019 18:42:48	No WSL found.
02/05/2019 18:42:48	Last benchmark was 18018 days 16:42:46 ago
02/05/2019 18:42:53	No general preferences found - using defaults
02/05/2019 18:42:53	Preferences:
02/05/2019 18:42:53	max memory usage when active: 1978.25 MB
02/05/2019 18:42:53	max memory usage when idle: 3560.85 MB
02/05/2019 18:42:53	max disk usage: 406.71 GB
02/05/2019 18:42:53	don't use GPU while active



En la pantalla superior, nos irá indicando cómo es capaz de ejecutar las transformadas de Fourier, y las fórmulas gaussianas en la siguiente captura:



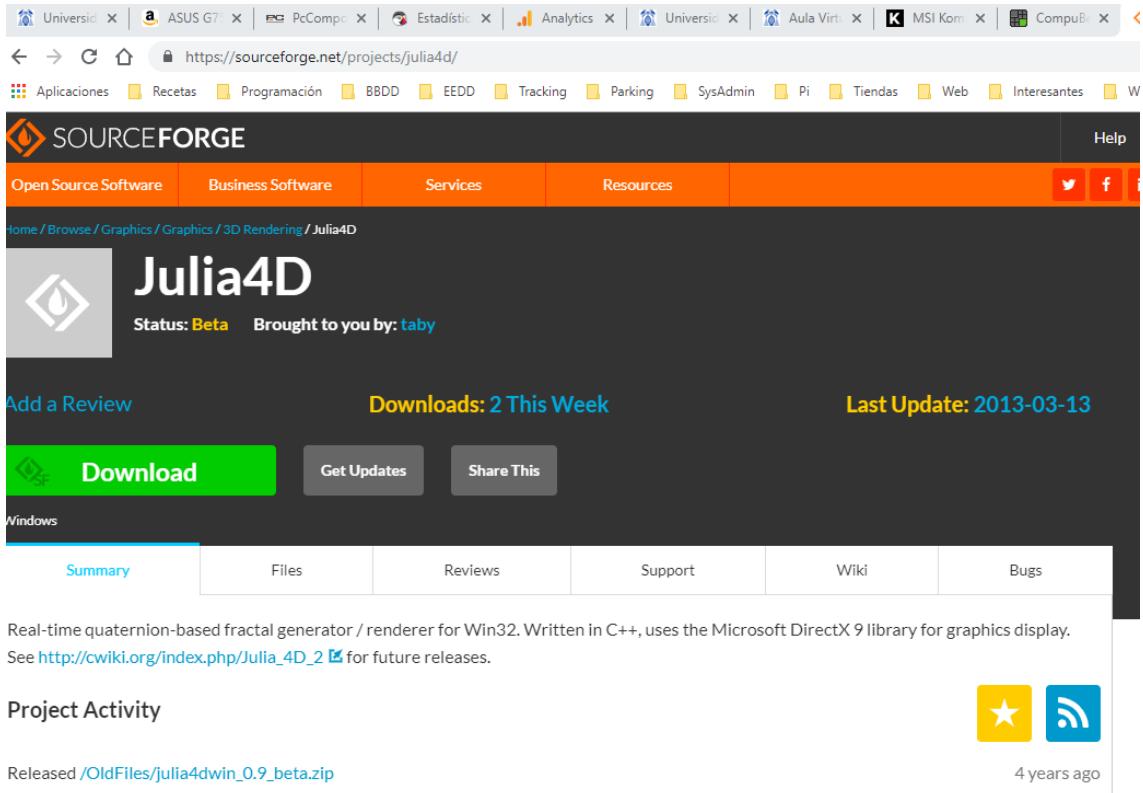
Propiedades de la tarea 30ap19ad.24603.15200.15.42.5\_1

Programa	SETI@home v8.22 (opencl_ati_cat132)
Nombre	30ap19ad.24603.15200.15.42.5
Estado	Tarea suspendida por el usuario
Recibido	02/05/2019 18:46:03
Límite para informar	24/06/2019 10:35:01
Recursos	0.01 CPUs + 1 AMD/ATI GPU
Tamaño de cálculo estimado	181.991 GFLOPs
Tiempo de CPU	00:01:58
Tiempo de CPU desde el punto de referencia	---
Tiempo transcurrido	00:01:23
Tiempo restante aproximado	---
Parte hecha	100,000%
Tamaño de la memoria virtual	103,93 MB
Tamaño conjunto trabajo	88,02 MB
Carpeta	slots/4
Índice de progreso	72,030% por minuto
Ejecutable	setiathome_8.22_windows_intelx86_opencl_ati_sah.exe

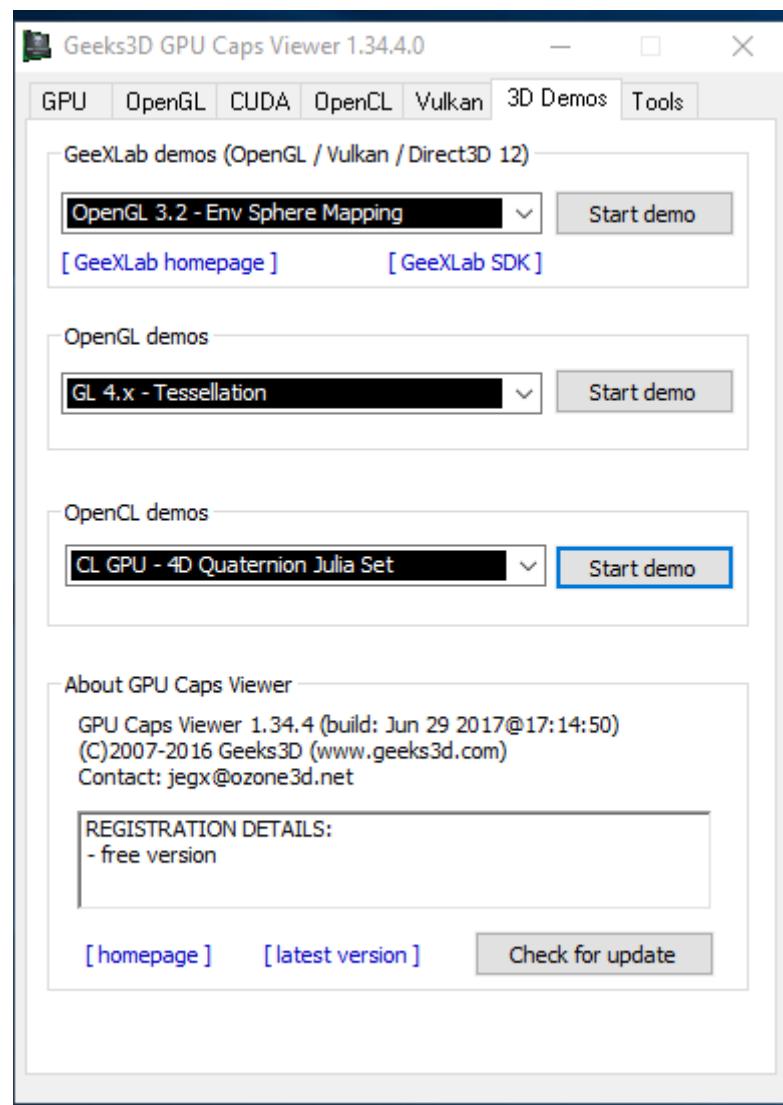
[Copiar Todo](#) [Copiar Selección](#) [Cerrar](#)

Ahora, ejecutamos el software específico para gráficas compatible con OpenCL. El programa elegido es Julia4D, que bajaremos del enlace oficial:

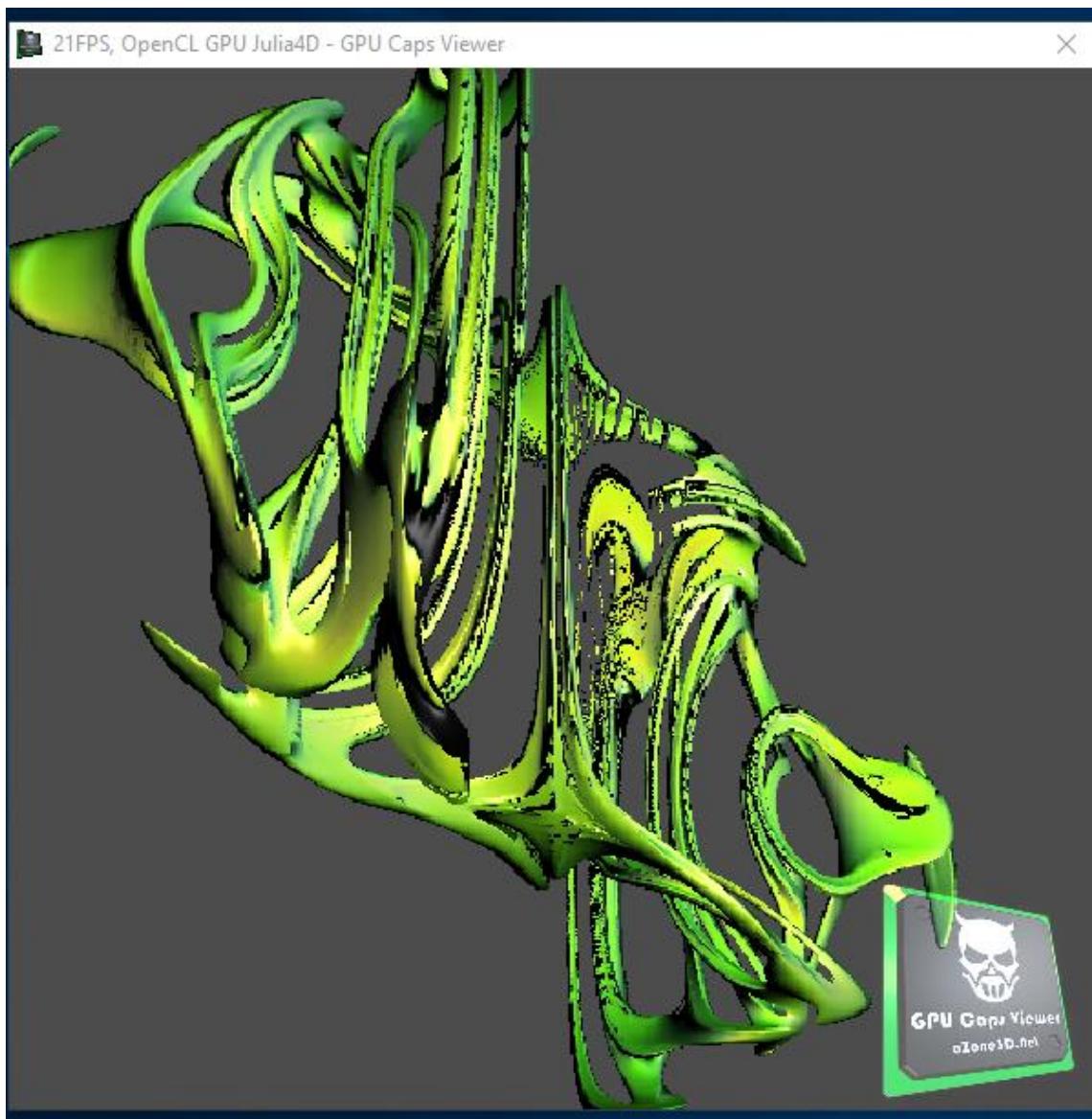
- <https://sourceforge.net/projects/julia4d/>



Lo ejecutamos, y esta será la primera pantalla que veamos:



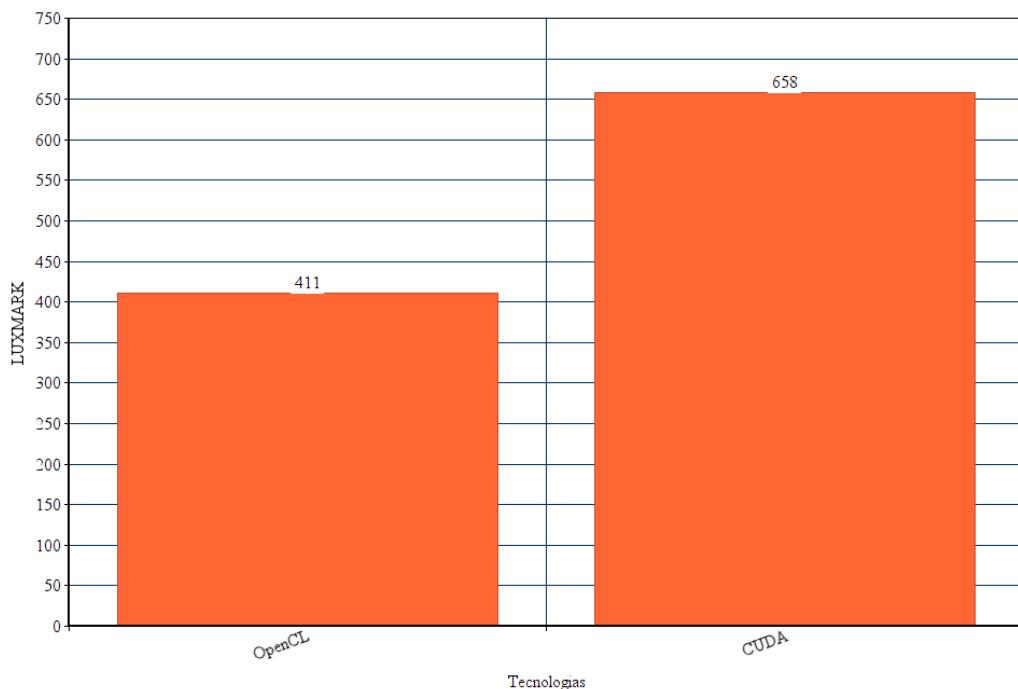
Simplemente tenemos que elegir el test, dentro de Geeks3D GPU Caps, y ejecutarlo. Obtenemos el resultado en Frames por Segundo:



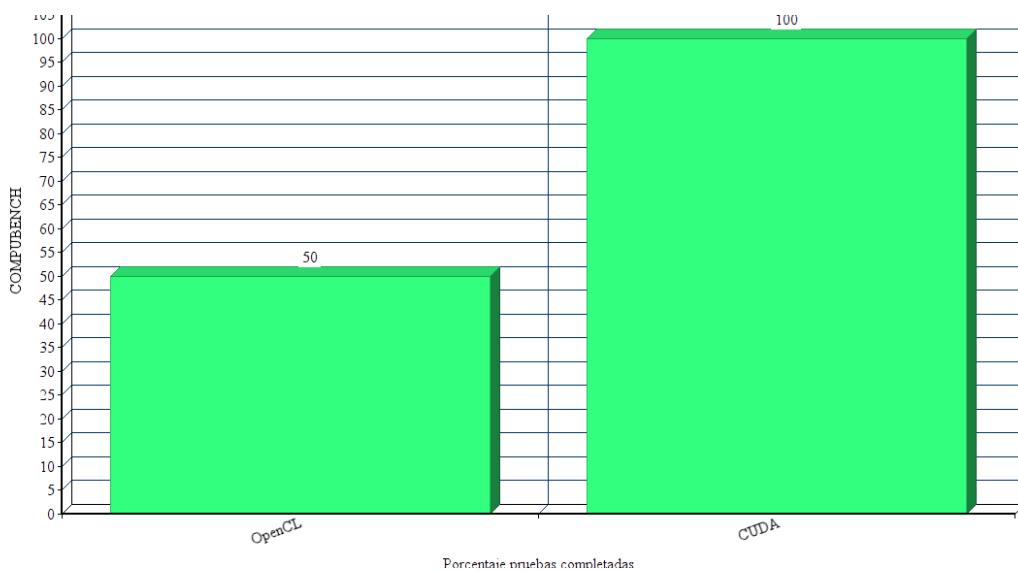
## 6. Comparación de las dos implementaciones

Estableceremos valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos. A no ser que se especifique lo contrario, contra mayor sea el resultado, mejor.

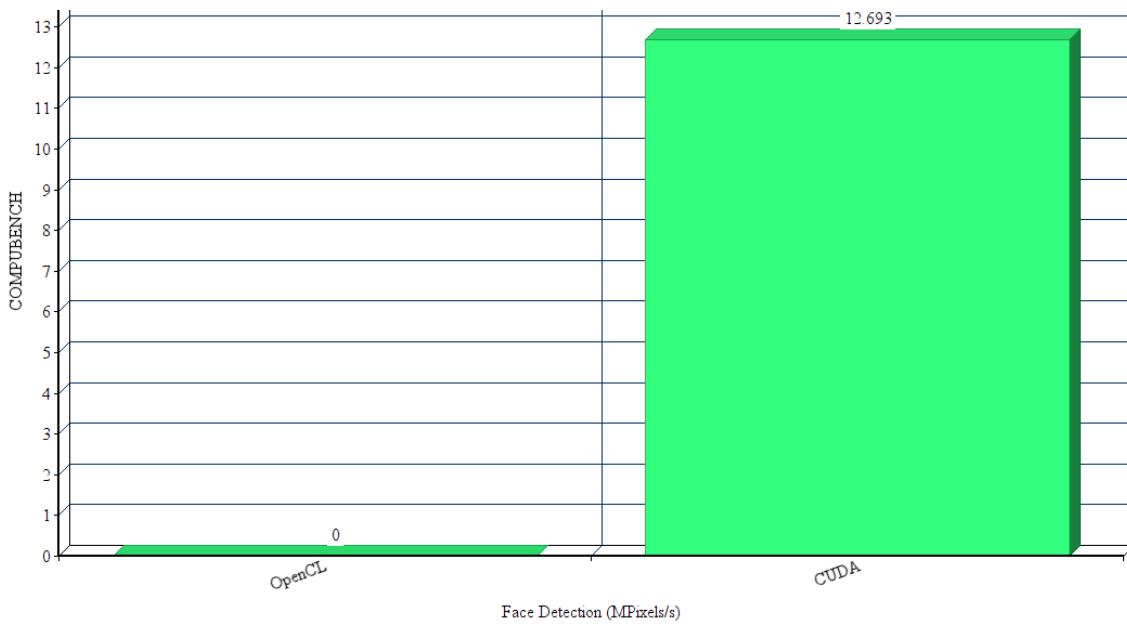
- Gráfica resultados Luxmark:



- Gráfica Compubench (Porcentaje pruebas completas):



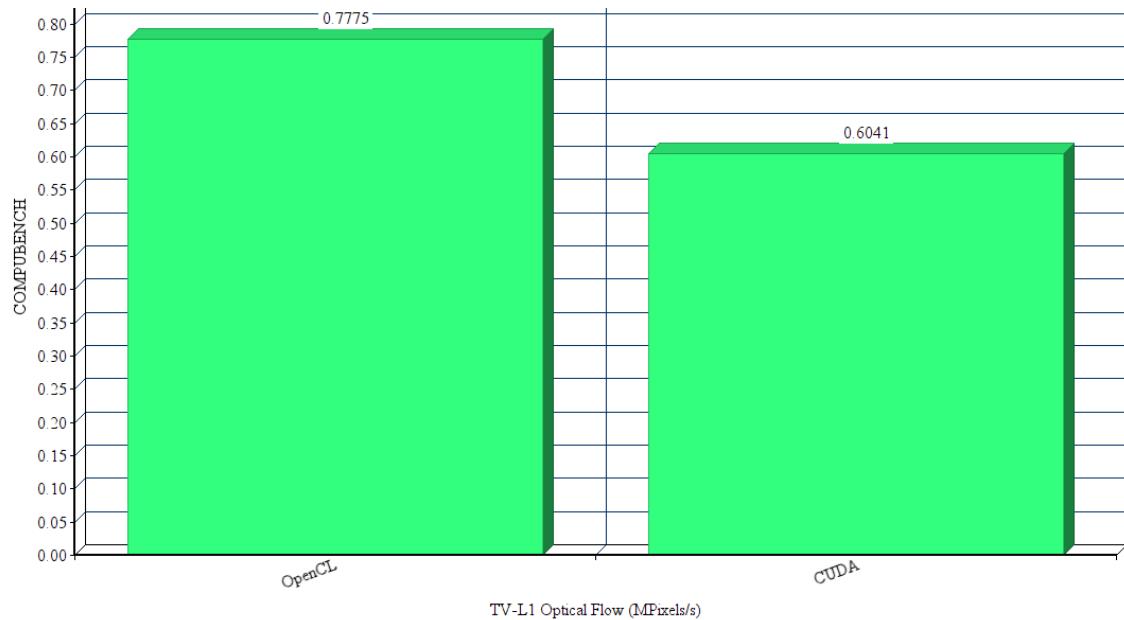
- Prueba Compubench: Face Detection:



Basado en el algoritmo Viola-Jones, empleado en el reconocimiento de patrones.

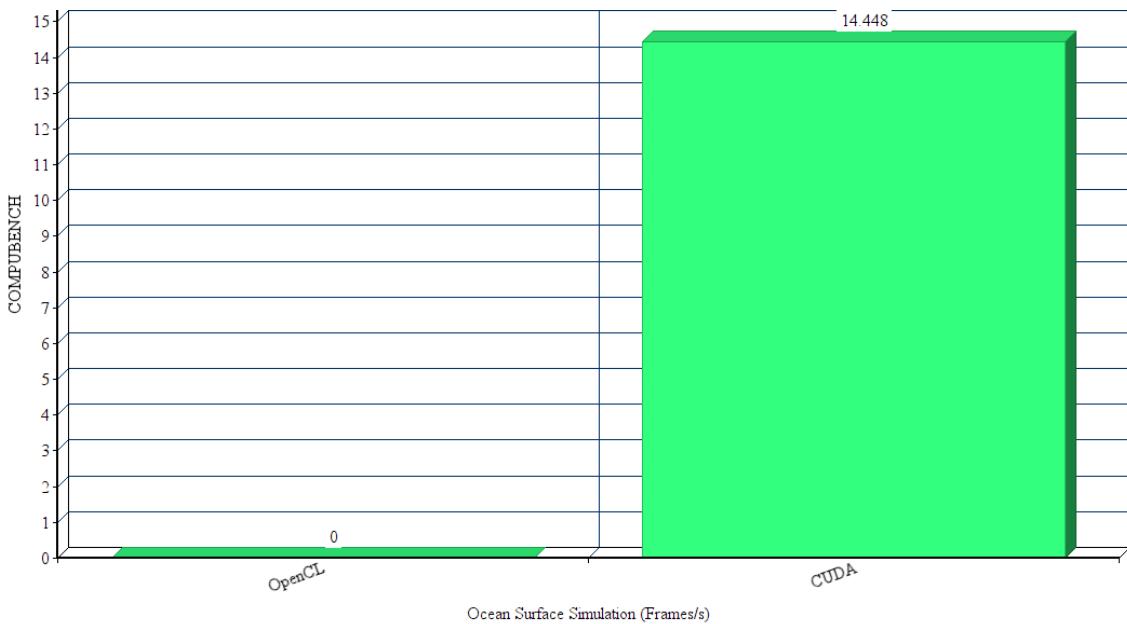
*\*Nota: la gráfica usando OpenCL no es capaz de ejecutar la prueba.*

- Prueba Compubench: TV-L1 Optical Flow:



Test basado en cálculo de vectores densos de movimiento usando método variacional.

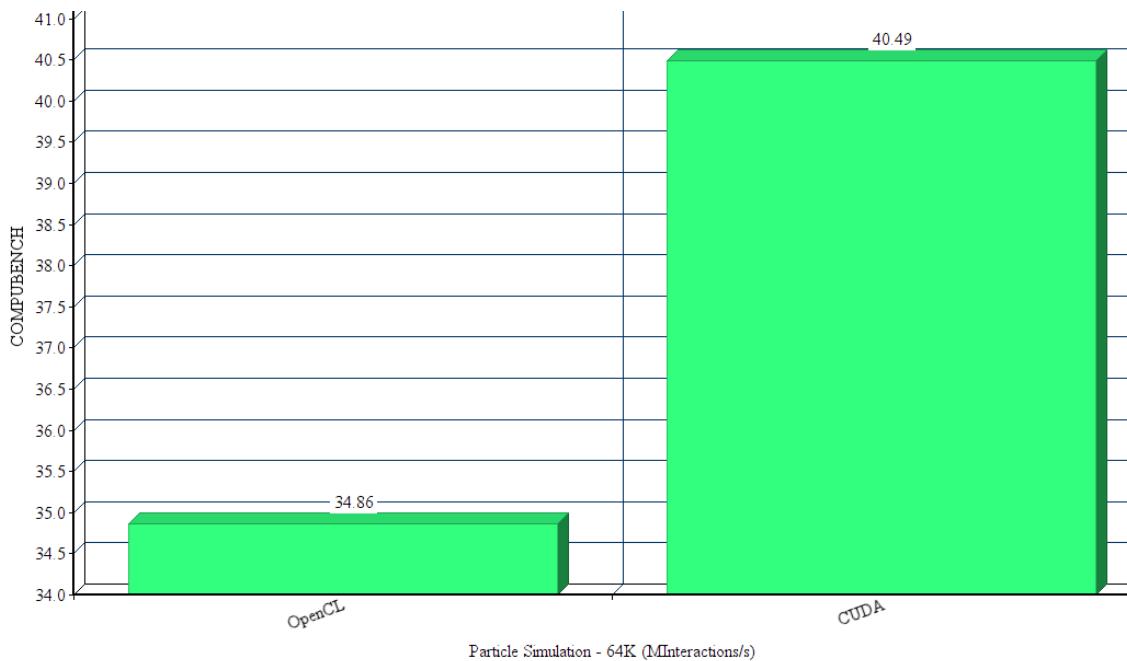
- Prueba Compubench: Ocean Surface Simulation:



Test basado en el algoritmo FFT, que a su vez emula las ondas que describe el mar.

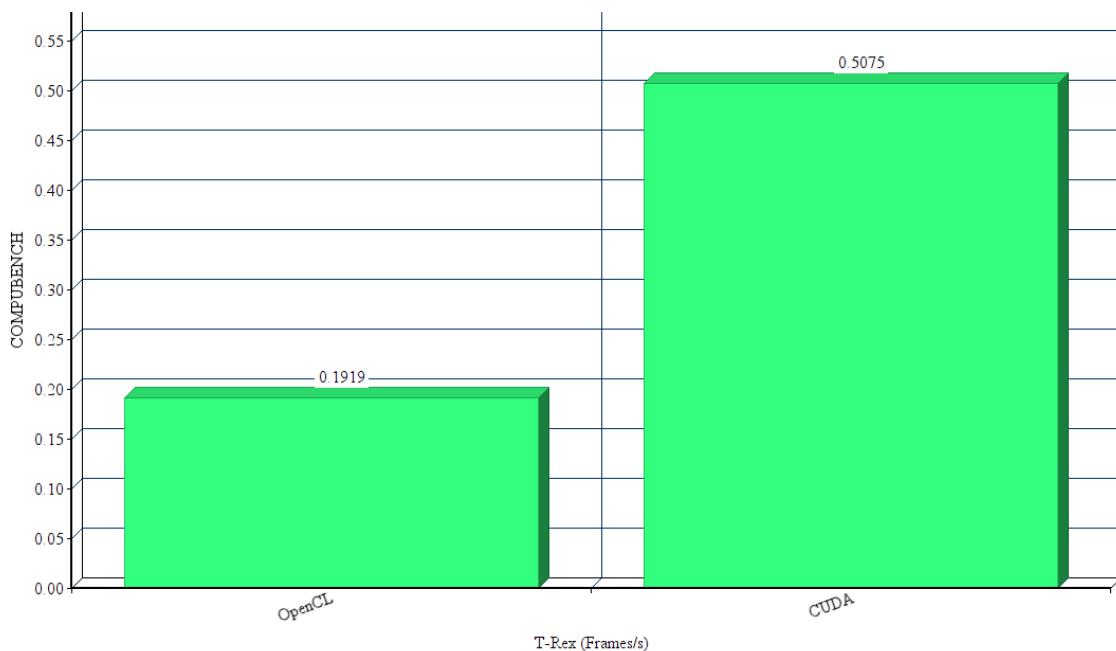
\*Nota: la gráfica usando OpenCL no es capaz de ejecutar la prueba.

- Prueba Compubench: Particle Simulation – 64K:



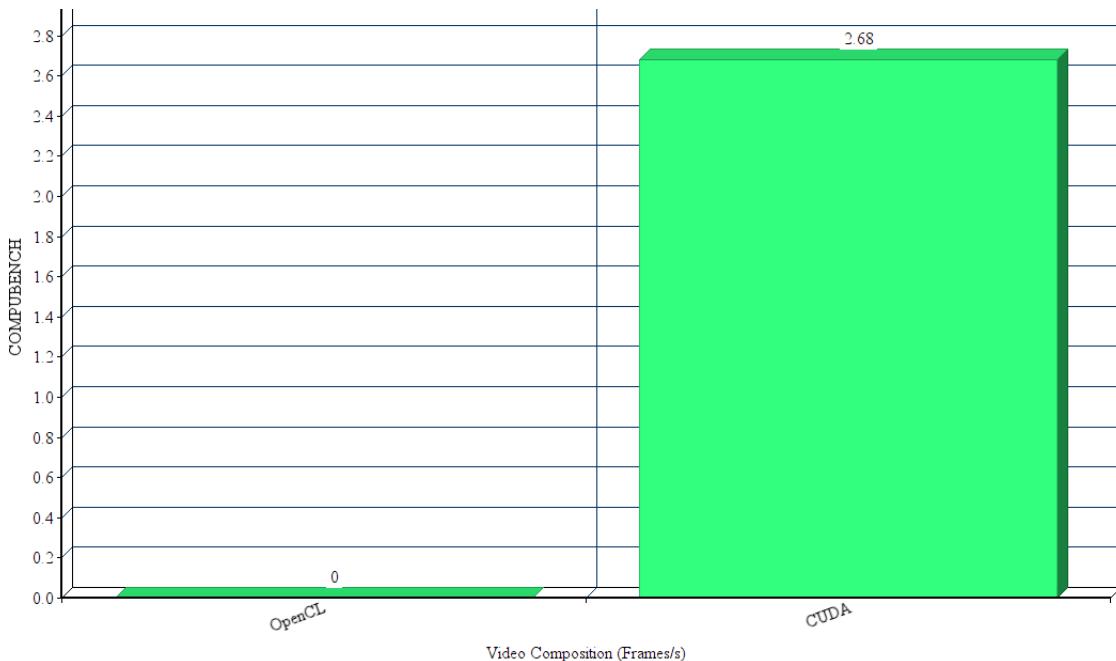
Simulación de partículas basada en el método de rejilla espacial, usando el método matemático discreto.

- **Prueba Compubench:T-Rex:**



Test de dibujado de rutas usando estructuras de aceleración dinámicas e iluminación global.

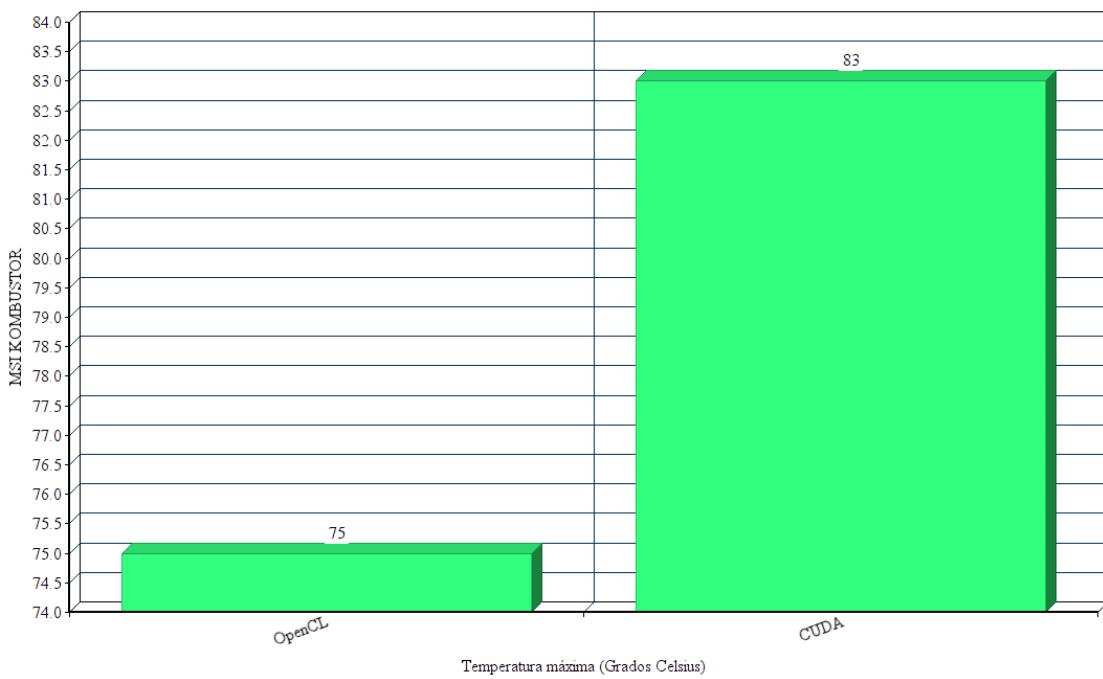
- **Prueba Compubench:Video Composition:**



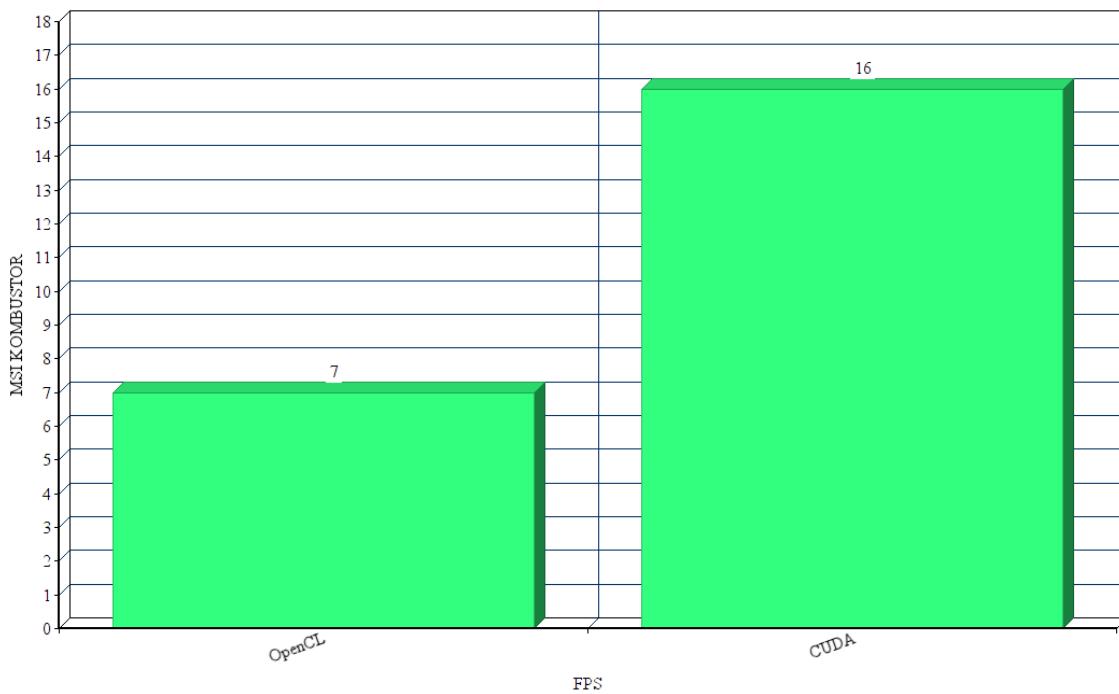
Este test recrea la composición de vídeo típica realizada por los programas modernos.

\*Nota: la gráfica usando OpenCL no es capaz de ejecutar la prueba.

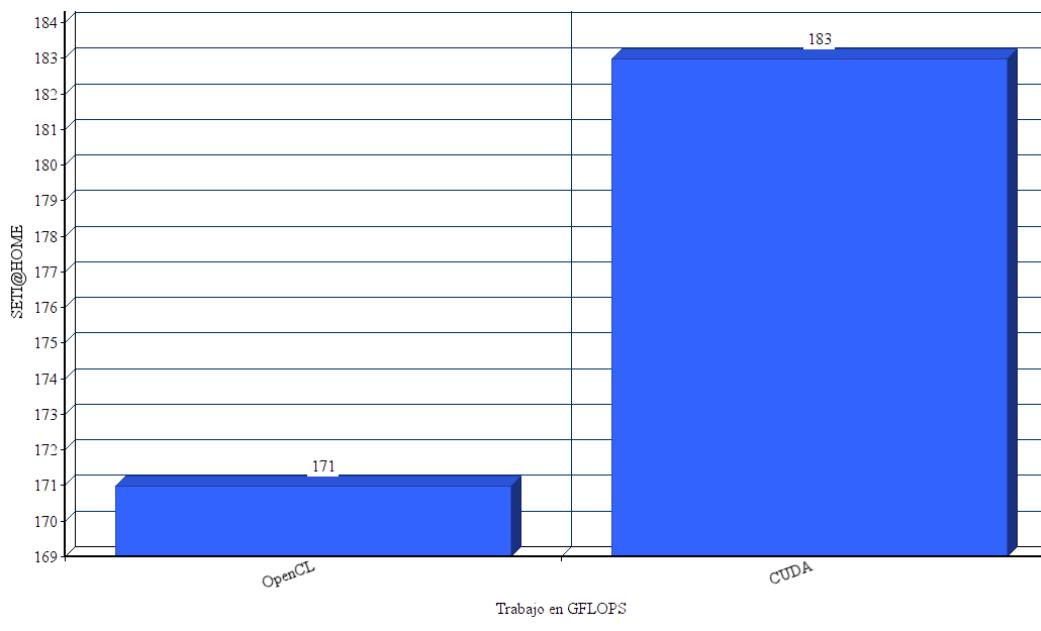
- **MSI Kombustor: temp máxima test Furmark OpenGL (mínimo es mejor)**



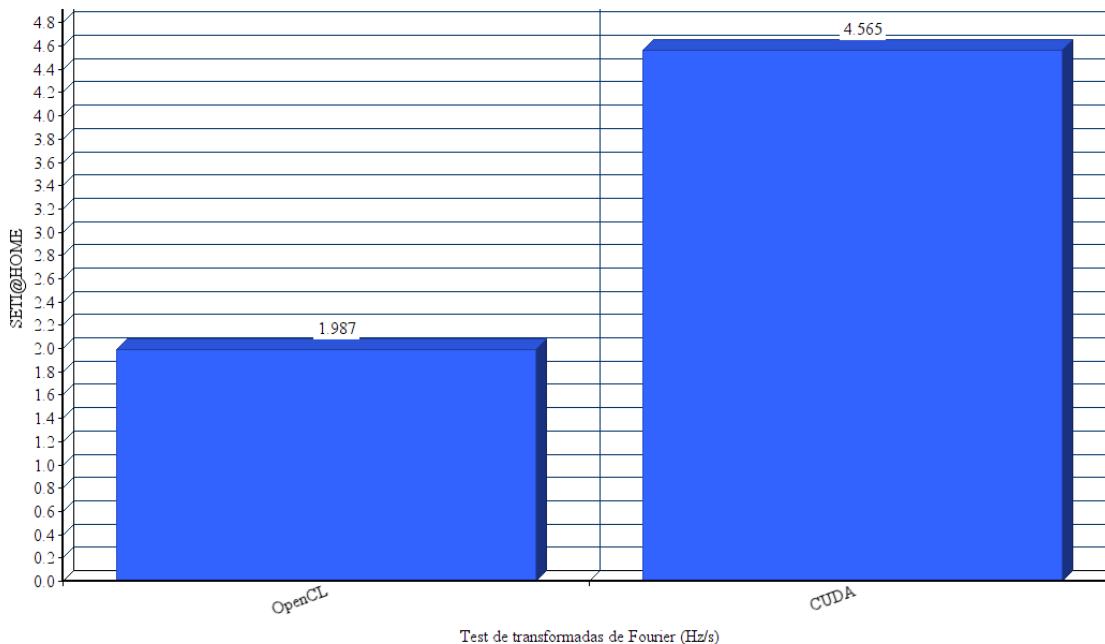
- **MSI Kombustor: FPS máximos Furmark OpenGL**



- **SETI@HOME: Carga de trabajo que recibe cada equipo:**

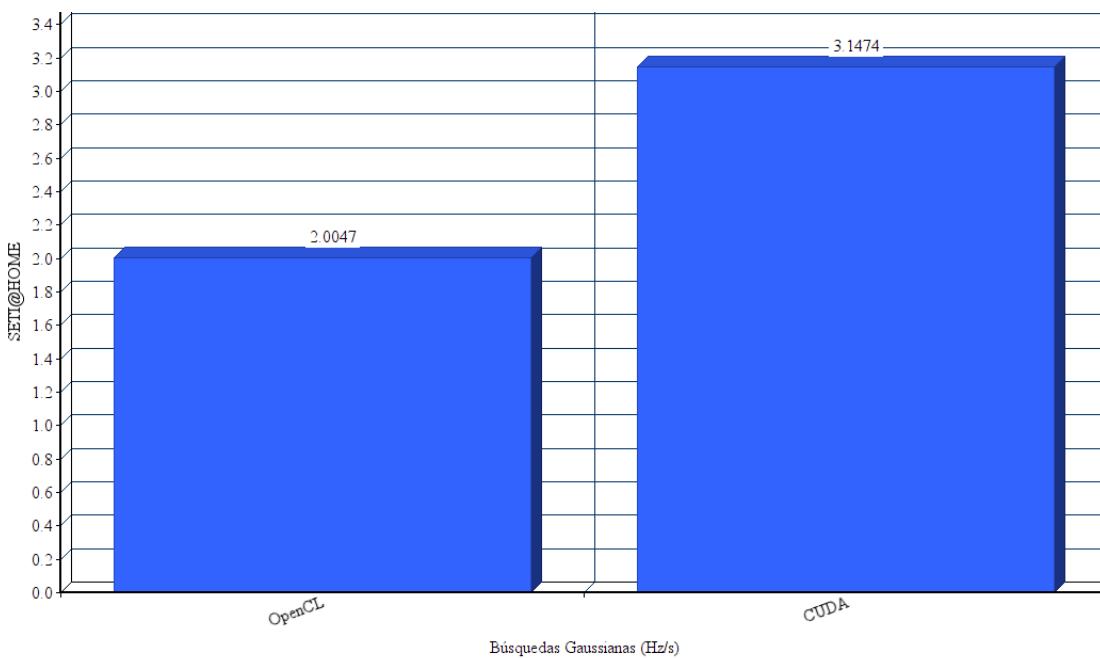


- **SETI@HOME: Test de transformadas de Fourier:**



Una de las principales pruebas de trabajo que se ejecutan en nuestro ordenador cliente, basado en el algoritmo matemático desarrollado a partir de las transformadas de Fourier.

- SETI@HOME: Test de búsquedas Gaussianas:



## 6.1 Evaluación de los criterios en la implementación usando la tecnología CUDA

CRITERIO	EVALUACIÓN
1. Requisitos de Hardware	Requerimientos que se necesita para ejecutar la tecnología.
2. Facilidad de Implementación	Complejidad para instalar el sistema.
3. Rendimiento Test Luxmark	Test que mide el rendimiento de la GPU empleando CUDA.
4. Rendimiento Test Compubench	Test sintético que mide el rendimiento del sistema y su capacidad ante determinadas pruebas.
5. Prueba Test MSI Kombustor	Test de estrés que mide la temperatura y capacidad de bloqueo del sistema.
6. Compatibilidad Software científico	Indica si nuestro software científico es capaz de ejecutarse correctamente
7. Puntuación SETI	Nos muestra la puntuación obtenida al realizar las pruebas científicas, viendo con qué solvencia se resuelven.
8. Software Específico CUDA	Software diseñado para comprobar que se está ejecutando la tecnología correctamente.
9. Multisistema	Capacidad de funcionar en varios sistemas.
10. Escalabilidad	Capacidad de la tecnología de adaptarse en tamaño según surja la necesidad.

## 6.2 Evaluación de los criterios en la implementación usando la tecnología OpenCL

CRITERIO	EVALUACIÓN
1.Requisitos de Hardware	Requerimientos que se necesita para ejecutar la tecnología.
2.Facilidad de Implementación	Complejidad para instalar el sistema.
3.Rendimiento Test Luxmark	Test que mide el rendimiento de la GPU empleando OpenCL.
4.Rendimiento Test Compubench	Test sintético que mide el rendimiento del sistema y su capacidad ante determinadas pruebas.
5.Prueba Test MSI Kombustor	Test de estrés que mide la temperatura y capacidad de bloqueo del sistema.
6.Compatibilidad Software científico	Indica si nuestro software científico es capaz de ejecutarse correctamente
7. Puntuación SETI	Nos muestra la puntuación obtenida al realizar las pruebas científicas, viendo con qué solvencia se resuelven.
8.Software Específico OpenCL	Software diseñado para comprobar que se está ejecutando la tecnología correctamente.
9. Multisistema	Capacidad de funcionar en varios sistemas.
10. Escalabilidad	Capacidad de la tecnología de adaptarse en tamaño según surja la necesidad.

## **7. Comparación de la implementación de las tecnologías**

CRITERIOS	TECNOLOGÍA A (CUDA)	TECNOLOGÍA B (OpenCL)	COMENTARIOS
1.Requisitos hardware			OpenCL es compatible con gráficas más antiguas, CUDA desde gráficas de 2008-2009.
2.Facilidad de implementación			CUDA está más preparado, viene con un autoinstalador con SDK drivers gráficos y CUDA.
3.Rendimiento Test Luxmark			La puntuación es mayor en el caso de CUDA.
4.Rendimiento CompuBench			La gráfica Nvidia, con CUDA, es capaz de ejecutar todos los test incorporados, y de manera más solvente.
5.Test Kombustor			Empate, ambos pasan la prueba
6.Compatibilidad software científico			Ambos son capaces de ejecutar el software BOINC requerido.
7.Puntuación SETI			La gráfica Nvidia alcanza mejores números en la prueba.
8.Software Específico			Ambas pasan la prueba.
9.Multisistema			Ambas son compatibles con los principales sistemas.
10.Escalabilidad			OpenCL es más escalable, además se puede integrar con sistemas Nvidia que fueron usados para CUDA.

## **8. Conclusiones**

A partir de la tabla anterior, que consiste en una síntesis de todas las pruebas que hemos realizado, podemos concluir que, si eliminamos en este caso la variable precio, la gran vencedora es en este caso Nvidia, a pesar de ser dos gráficas de la misma gama.

Nvidia/CUDA gana en rendimiento bruto, lo que le hace ganar la prueba, sin embargo, si buscamos un sistema escalable, que sea multisistema, y con pocos requisitos de implementación, podemos apostar por OpenCL, sobre todo si queremos implementar una red computacional usando, por ejemplo, recursos sobrantes de otros departamentos.

En ese caso, podemos crear una red distribuida con esos PCs que, en conjunto, superen la potencia de un equipo único CUDA, aunque ahí entrarían variables como precio de la luz y espacio.

En cualquier caso, ambas tecnologías pueden considerarse válidas, al haber pasado la mayoría de tests a lo largo de los tres trabajos que hemos ido realizando a lo largo de este cuatrimestre..