Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc6948571)

[1.1 Autores 2](#_Toc6948572)

[1.2 Planificación 2](#_Toc6948573)

[1.3 Entrega 2](#_Toc6948574)

[2. Requisitos del prototipo a implementar 3](#_Toc6948575)

[2.1 Requisitos funcionales 4](#_Toc6948576)

[2.2 Otros requisitos 4](#_Toc6948577)

[3. Criterios de comparación en la implementación 5](#_Toc6948578)

[3.1 Criterio 1: Nombre del criterio 5](#_Toc6948579)

[3.2 Criterio 2: Nombre del criterio 5](#_Toc6948580)

[3.N Criterio N: Nombre del criterio 5](#_Toc6948581)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 6](#_Toc6948582)

[4.1 Documentación de diseño 6](#_Toc6948583)

[4.2 Documentación de construcción 6](#_Toc6948584)

[4.3 Documentación de pruebas 6](#_Toc6948585)

[4.4 Documentación de instalación 6](#_Toc6948586)

[4.5 Manual de usuario 6](#_Toc6948587)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 7](#_Toc6948588)

[5.1 Documentación de diseño 7](#_Toc6948589)

[5.2 Documentación de construcción 7](#_Toc6948590)

[5.3 Documentación de pruebas 7](#_Toc6948591)

[5.4 Documentación de instalación 7](#_Toc6948592)

[5.5 Manual de usuario 7](#_Toc6948593)

[6. Comparación de las dos implementaciones 8](#_Toc6948594)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 8](#_Toc6948595)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 8](#_Toc6948596)

[7. Comparación de la implementación de las tecnologías 9](#_Toc6948597)

[8. Conclusiones 11](#_Toc6948598)

# 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:

## 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 grafico (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 ente 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 cumplir los requisitos mínimos de Gráfica |
| RF02 | El sistema deberá cumplir los requisitos mínimos de RAM |
| RF03 | El sistema deberá cumplir los requisitos mínimos de Procesador |
| RF04 | El sistema deberá tener instalado sistema operativo adecuado y SDK/Herramientas adecuadas para la tecnología |
| RF05 | El sistema será capaz de ejecutar el IDE escogido |
| RF06 | El sistema será capaz de compilar y ejecutar sin errores el código |
| RF07 | El sistema mostrará los resultados de forma adecuada |

## 2.2 Otros requisitos

Se pueden incluir aquí otros requisitos para el prototipo que no puedan considerarse como funcionales. Por ejemplo, requisitos de datos, de seguridad, de interfaz de usuario, de rendimientos, etc.

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

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01 | El sistema deberá completar las pruebas en un tiempo razonable |
| R02 | El sistema deberá permanecer fiable bajo condiciones de carga |
| R03 |  |

# 3. Criterios de comparación en la implementación

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este 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.

Se trata de criterios del tipo” “horas empleadas en el desarrollo del sistema”, “velocidad de funcionamiento del sistema”, “recursos necesarios”, etc.

## 3.1 Criterio 1: Requisitos de Hardware

Nombre del criterio: Requisitos hardware

Descripción: Condiciones que debe tener el sistema para operar con el hardware.

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: Escalabilidad

Nombre del criterio: Escalabilidad.

Descripción: Capacidad del sistema para reaccionar y adaptarse sin perder la calidad, manejar el crecimiento continuo o para hacerse más grande sin perder calidad.

Tipo de valor: Escala 1-5, 1 menor 5 mayor.

## 3.4 Criterio 4: Rendimiento

Nombre del criterio: Rendimiento.

Descripción: Capacidad de un sistema para llevar a cabo una tarea.

Tipo de valor: Texto libre.

## 3.5 Criterio 5: Comunidad de ayuda

Nombre del criterio: Comunidad.

Descripción: Es un espacio interactivo donde se comparte con otras personas sus inquietudes, sus ideas, sus sugerencias, hacer comentarios, reportar problemas y opinar.

Tipo de valor: Booleano (Si/No).

## 3.6 Criterio 6: Windows

Nombre del criterio: Windows.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

## 3.7 Criterio 7: Linux

Nombre del criterio: Linux.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

## 3.8 Criterio 8: Mac

Nombre del criterio: Mac.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

## 3.9 Criterio 9: IDEs

Nombre del criterio: IDEs.

Descripción: Servicios que facilitan el desarrollo de software.

Tipo de valor: Texto libre.

## 3.10 Criterio 10: Idiomas

Nombre del criterio: Idiomas.

Descripción: Idiomas disponibles para los sistemas.

Tipo de valor: Texto libre.

## 3.11 Criterio 11: Madurez

Nombre del criterio: Madurez.

Descripción: Calidad y ventaja competitiva respecto otros sistemas.

Tipo de valor: Texto libre.

## 3.12 Criterio 12: Popularidad

Nombre del criterio: Popularidad.

Descripción: Es el grado de aceptación de los sistemas.

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

## 3.13 Criterio 13: Soporte

Nombre del criterio: Soporte.

Descripción: Capacidad de solucionar problemas del sistema.

Tipo de valor: Booleano (Si/No).

## 3.14 Criterio 14: Mantenimiento

Nombre del criterio: Mantenimiento.

Descripción: El conjunto de operaciones necesarias para el funcionamiento correcto del sistema.

Tipo de valor: Booleano (Si/No).

## 3.15 Criterio 15: Licencia

Nombre del criterio: Licencia

Descripción: Autorización para hacer una serie de tareas o procesos con el software.

Tipo de valor: Texto libre.

# 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.

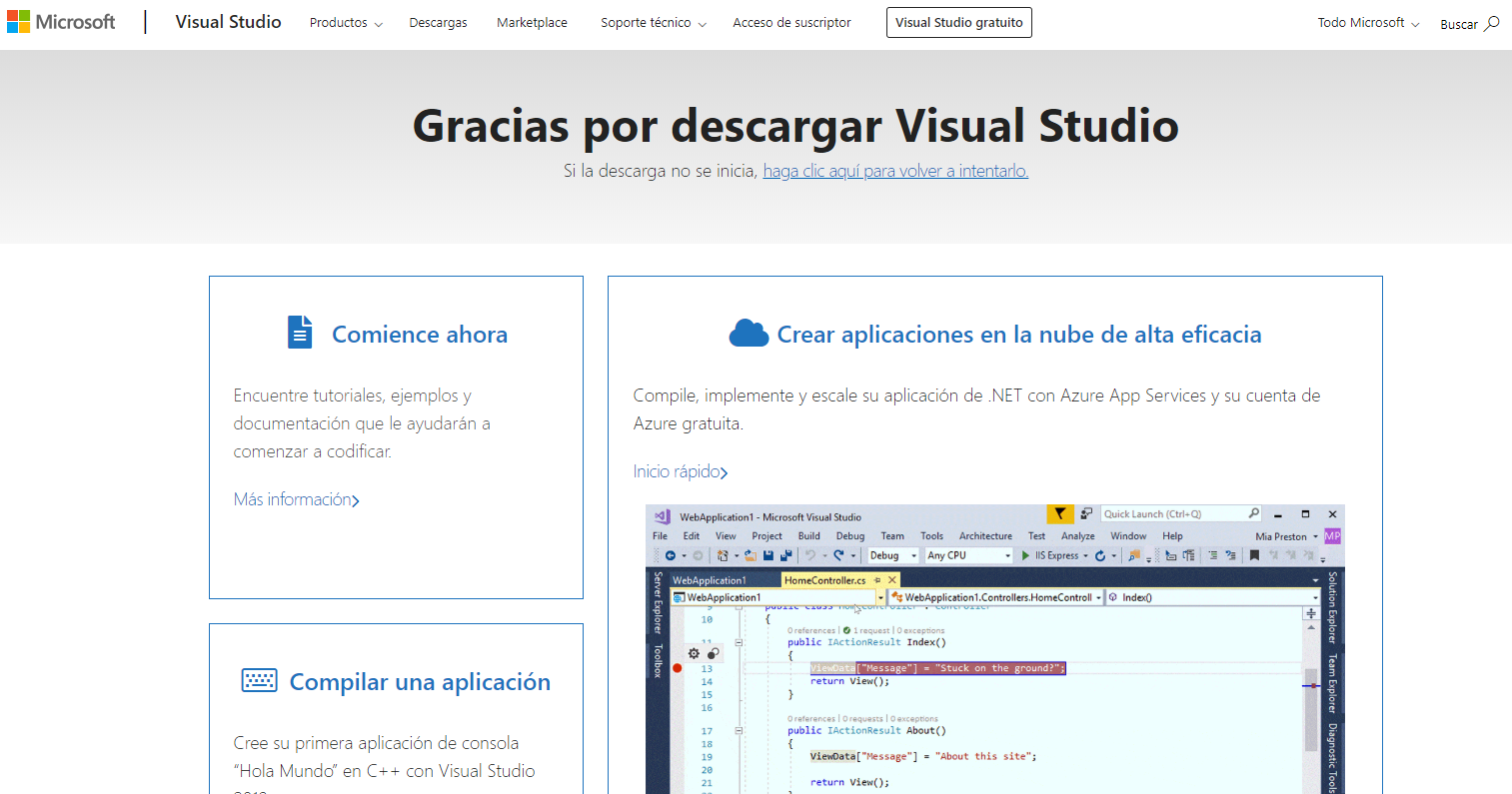
|  |  |
| --- | --- |
| CAPA SOFTWARE | * Drivers * SDK * CUDA/OpenCL driver * IDE * Testing Tools * Scientific Software |
| CAPA SISTEMA OPERATIVO | * Sistema (Windows/Mac/Linux) * Actualizaciones |
| CAPA HARDWARE | * Procesador > SSE2 * Ram > 4Gb * Gráfica compatible |

## 4.3 Documentación de pruebas

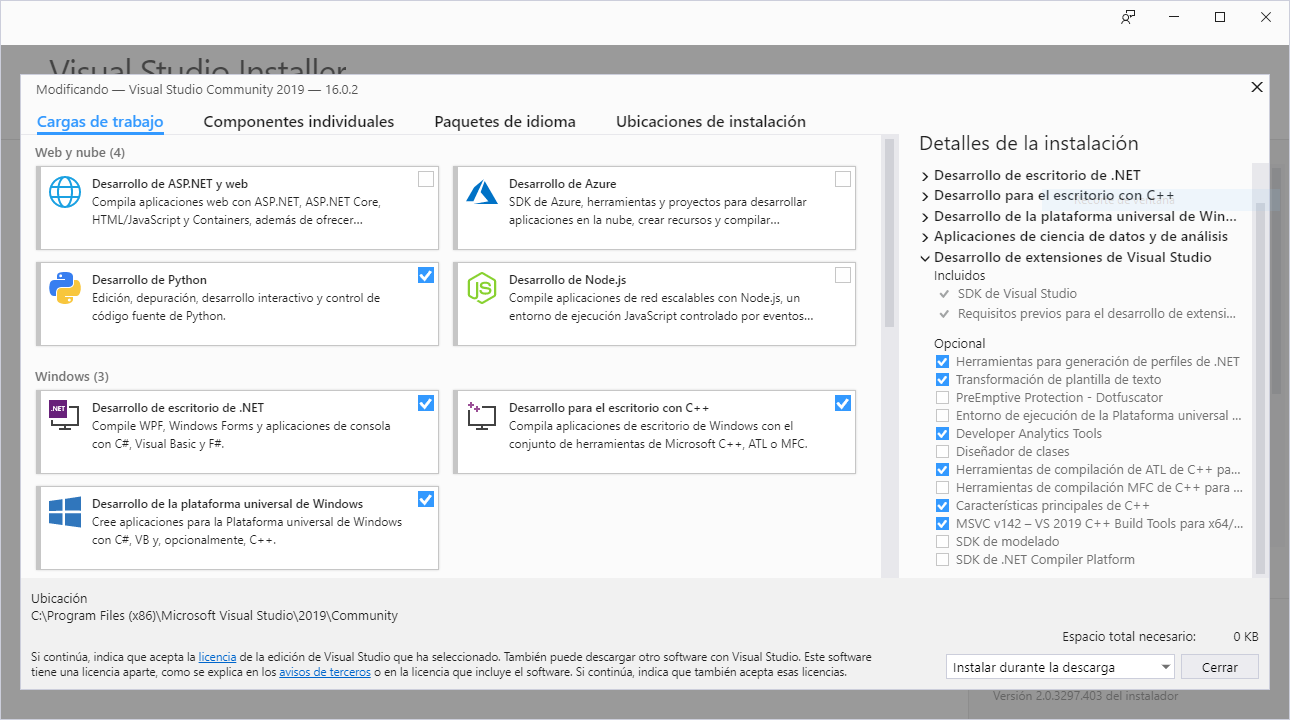
## 4.4 Documentación de instalación

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>



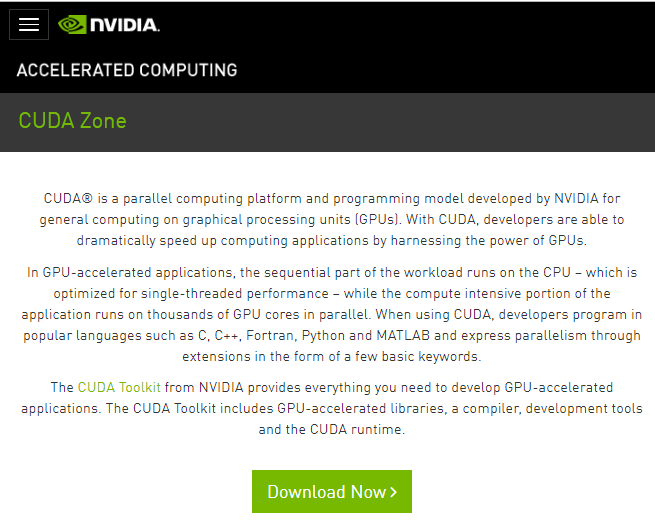
Cuando se abra y se instale el instalador nos aparecerá la siguiente pantalla:



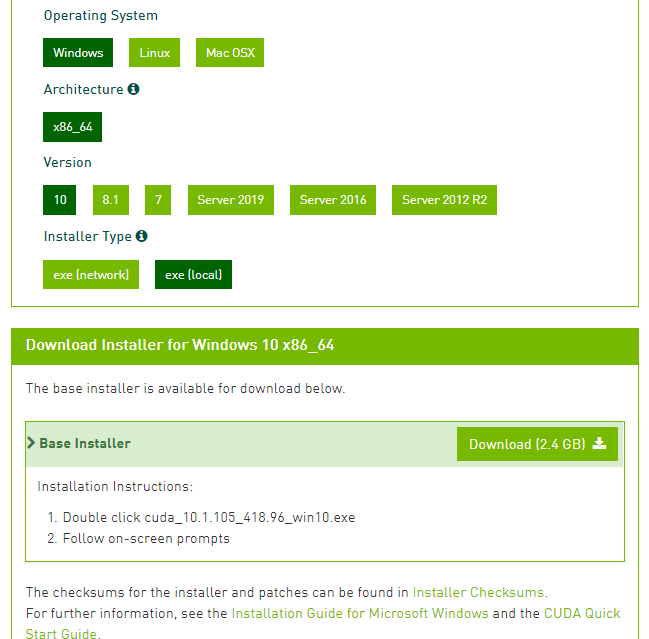
Se deberá escoger los entornos de trabajo que se le quieren asignar para poder programar en Visual Studio dependiendo de lo que se quiera hacer y qué herramientas vamos a necesitar. En este caso con la instalación de CUDA ya vienen las librerías y el entorno de trabajo de esta para poder programar.

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>

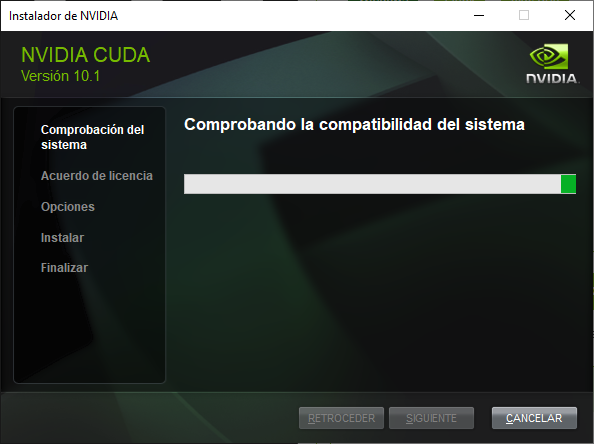
Tras este enlace nos llevará a la siguiente página:

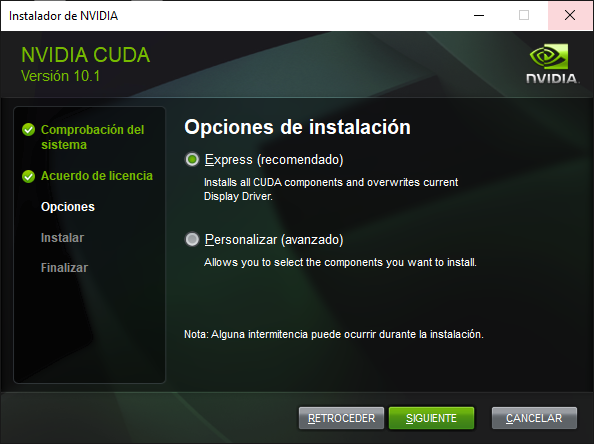


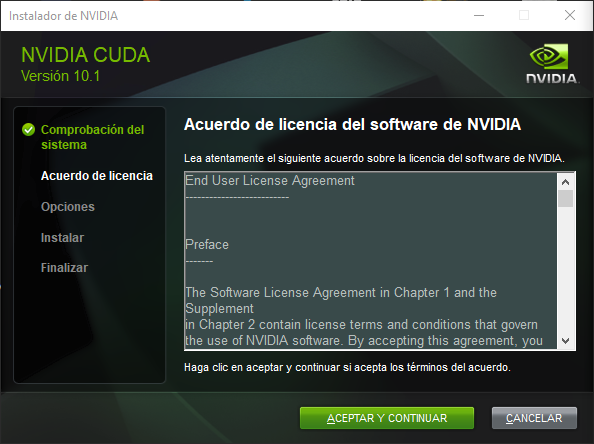
Siguiendo los pasos podremos descargarnos el instalador dependiendo de las características que cuenta nuestro dispositivo. En este caso nos instalaremos los controladores correspondientes a Windows 10 x86\_64 de forma local.

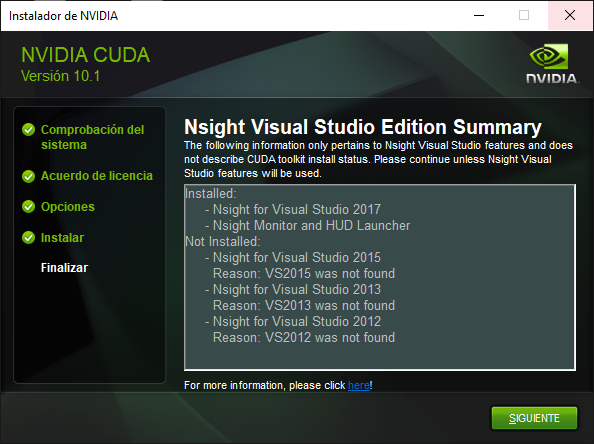


Tras esto abrimos el instalador y seguimos todos los pasos que se nos indica:

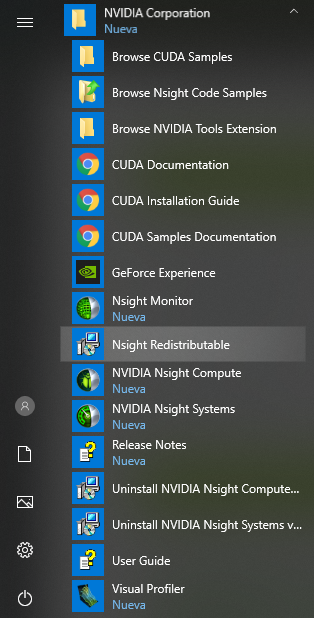








Tras esto se obtendrá un repertorio de aplicaciones en las cuales nos ayudarán a la hora de querer implementar el código por Visual Studio correctamente:



## 4.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 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 | * Drivers * SDK * CUDA/OpenCL driver * IDE * Testing Tools * Scientific Software |
| CAPA SISTEMA OPERATIVO | * Sistema (Windows/Mac/Linux) * Actualizaciones |
| CAPA HARDWARE | * Procesador > SSE2 * Ram > 4Gb * Gráfica compatible |

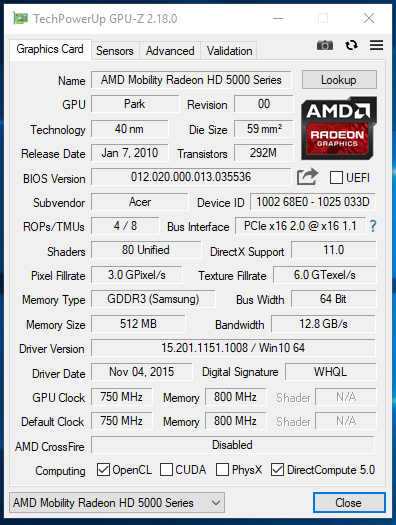
## 5.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los casos de prueba.

## 5.4 Documentación de instalación

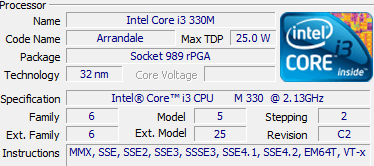
Para comenzar la instalación, lo primero es contar con un equipo adecuado. Y dentro del equipo adecuado, tener una gráfica compatible con la tecnología que queramos implementar, como hemos visto en el trabajo anterior, es vital.

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

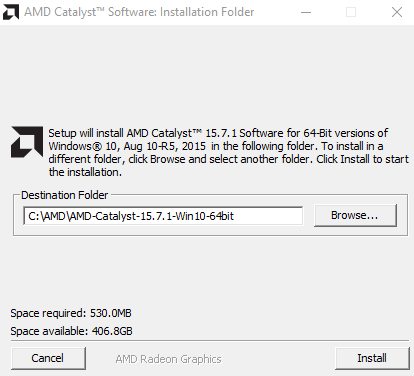


En nuestro caso, observamos que es compatible con la tecnología OpenCL pero no CUDA al ser una gráfica AMD.

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

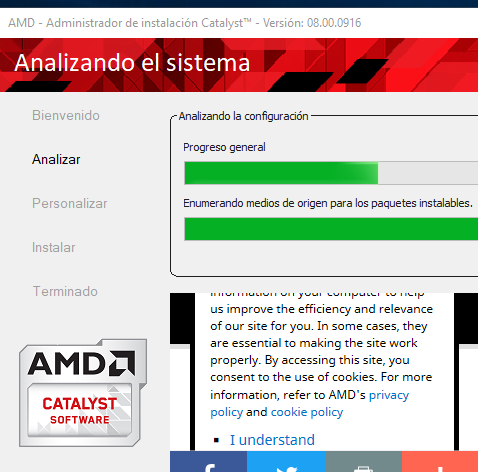


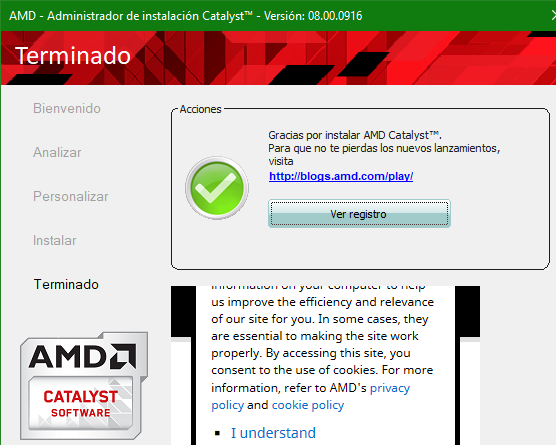
Lo siguiente será instalar los drivers adecuados, en nuestro caso AMD Catalyst:



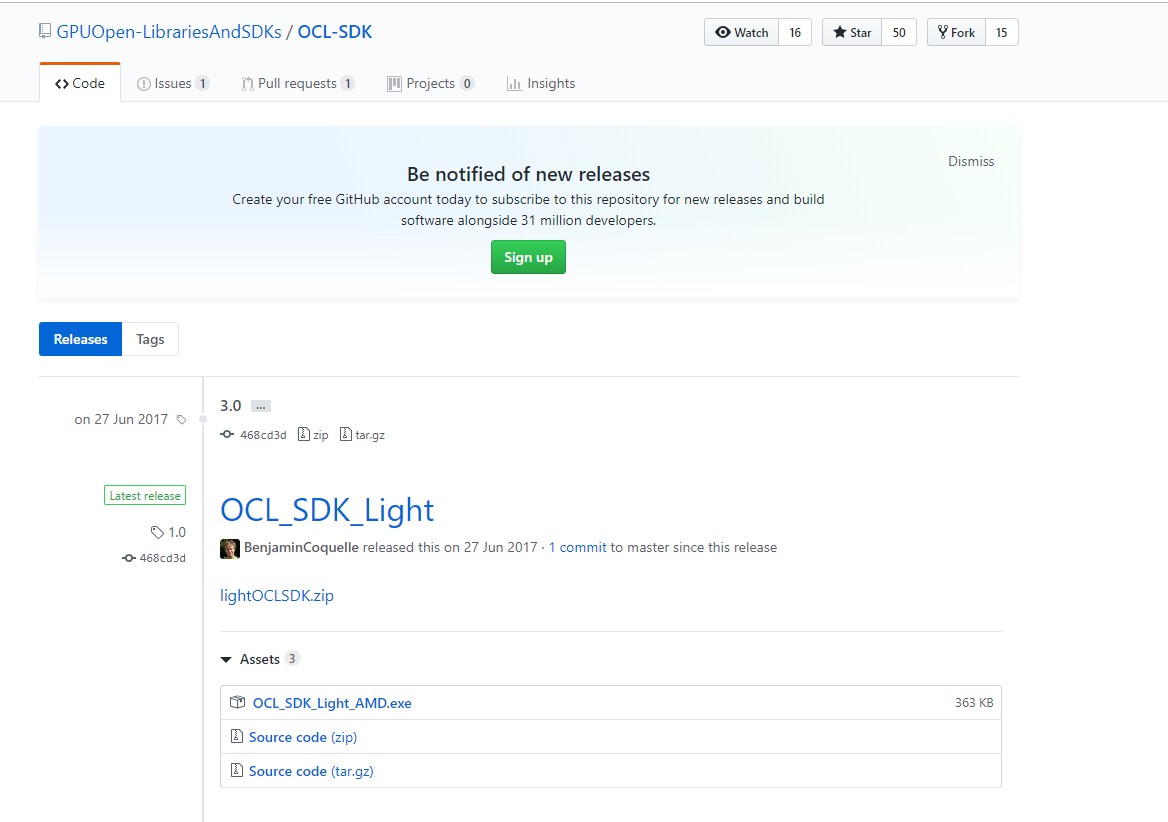


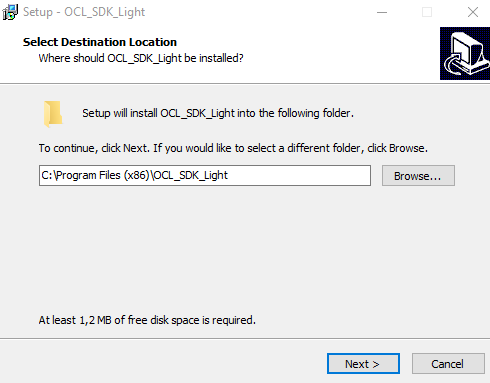
Ahora ya, se inicia el proceso de instalación propiamente dicho:



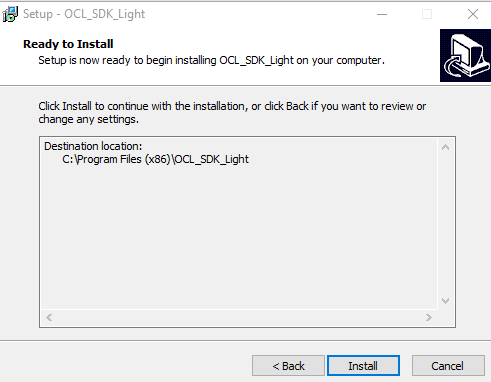


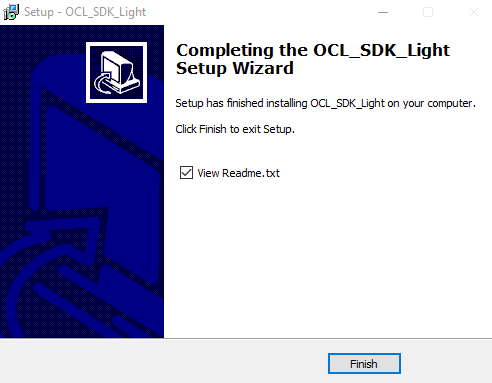
Ahora, tenemos que descargar del repositorio Github oficial, el SDK Bluelight, que es una versión ligera del SDK, y luego habrá que instalar la completa:



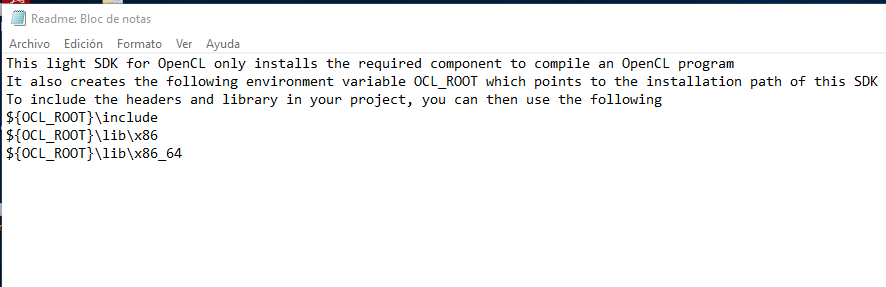


Procedemos a instalarlo:

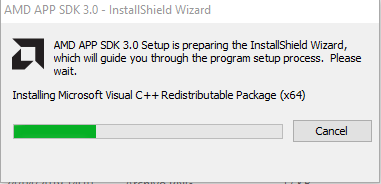


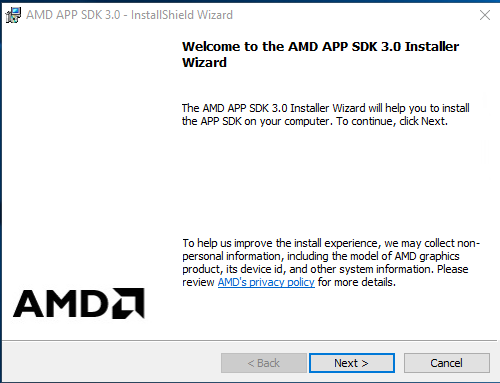


Vemos que se ha instalado correctamente:

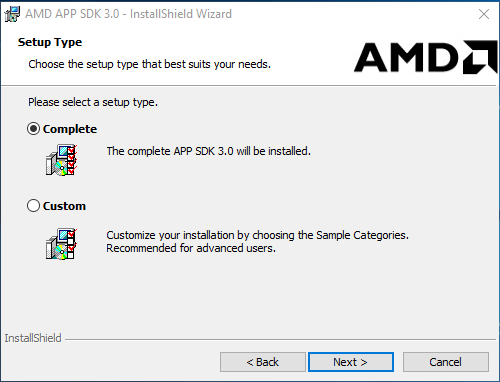


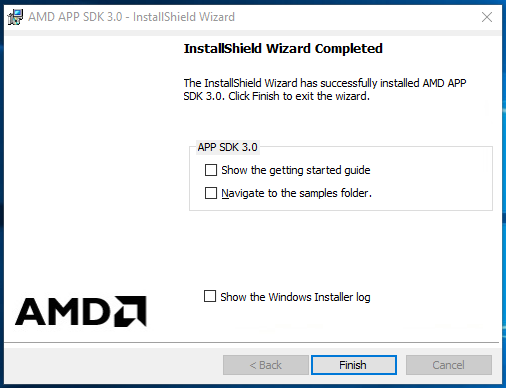
Ahora, tenemos que instalar el SDK APP V3.0 de AMD. Su instalación es sencilla, viene contenido todo lo necesario en un instalador oficial de AMD, de unos 200MB:



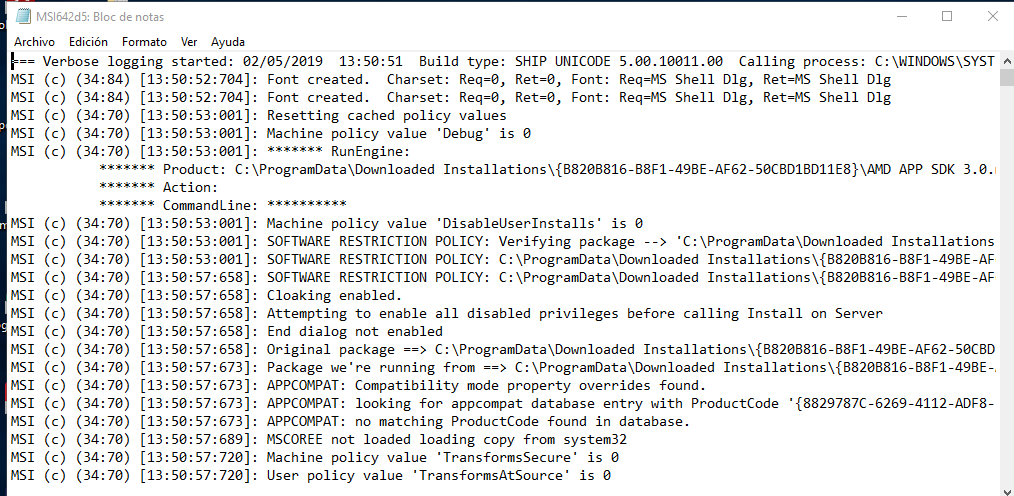


Elegimos instalación completa, para que verifique que están instalados los drivers de OpenCL:

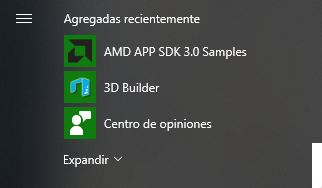




Comprobamos que está instalado correctamente seleccionando “Show the Windows Installer log”:



Y vemos que en Windows 10 aparece instalado:



## 5.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 6. Comparación de las dos implementaciones

Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

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

Debe incluir al menos una tabla con la siguiente estructura.

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Criterio 1 |  |
| Criterio 2 |  |
| … |  |
| Criterio N |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

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

# 7. Comparación de la implementación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

| **CRITERIOS** | **TECNOLOGÍA A** | **TECNOLOGÍA B** | **COMENTARIOS** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| … |  |  |  |
| N |  |  |  |

# 8. Conclusiones

A partir de la información incluida en el apartado 7 y de la experiencia al realizar el trabajo, el grupo debe estar en condiciones de manifestar su opinión sobre la implementación del sistema utilizando ambas tecnologías, y debe plasmarla en este apartado, indicando las ventajas e inconvenientes más relevantes de utilizar una u otra tecnología para implementar el sistema.

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)