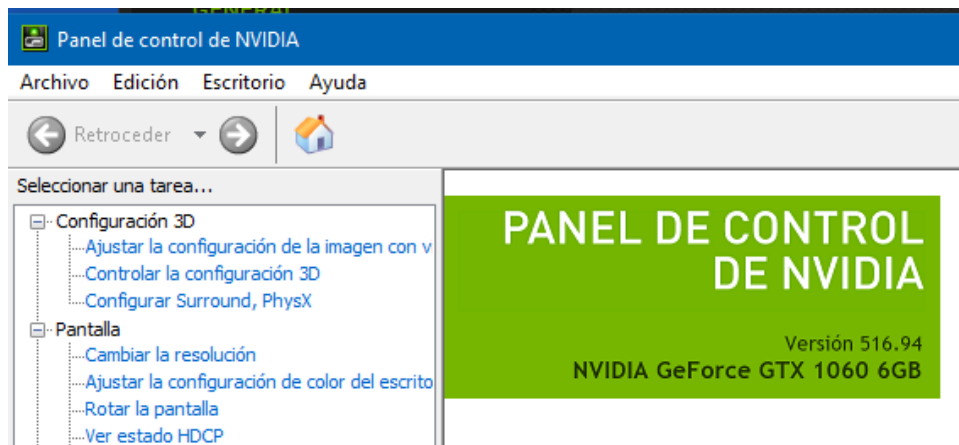
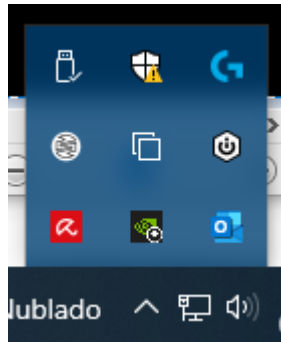


# Guía de Instalación CUDA

Descubrir la GPU que tenemos



Ir aquí: <https://developer.nvidia.com/cuda-gpus> Y buscar nuestra GPU. Generalmente, estará en el apartado "GeForce and Titan". La columna "Notebook" implica GPUs de portátil.

## GeForce and TITAN Products

GPU	Compute Capability
GeForce GTX 1060	6.1

Con esto ya sabemos la capacidad de cómputo (“*compute capability*”) de nuestra GPU. Es lo que nos marcará la versión de CUDA que podremos instalar.

**TABLA 3.** Relación entre la versión de CUDA y la capacidad de cómputo requerida.

Versión de CUDA	Capacidad de cómputo soportada
6.5	1.0 – 5.3
7.5	2.0 – 5.3
8.0	2.0 – 6.2
9.2	3.0 – 7.2
10.2	3.0 – 7.5
11.0	5.0 – 8.0
¿?	¿?

Aquí <https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html> podemos ver la versión de los Drivers que tenemos que tener, como mínimo, instalado.

El IDE que usaremos será **Visual Studio**. No podemos instalar la versión que nos venga en gana. La versión está relacionada con la versión de CUDA. Una vez determinada la versión de CUDA que vamos a instalar, ir a <https://developer.nvidia.com/cuda-toolkit-archive>. Buscar la versión que queremos instalar. Seleccionar “*Versioned Online Documentation*”. Abrir la sección “*Installation Guide Windows*”. Una de las tablas indicará las versiones de VS soportadas.

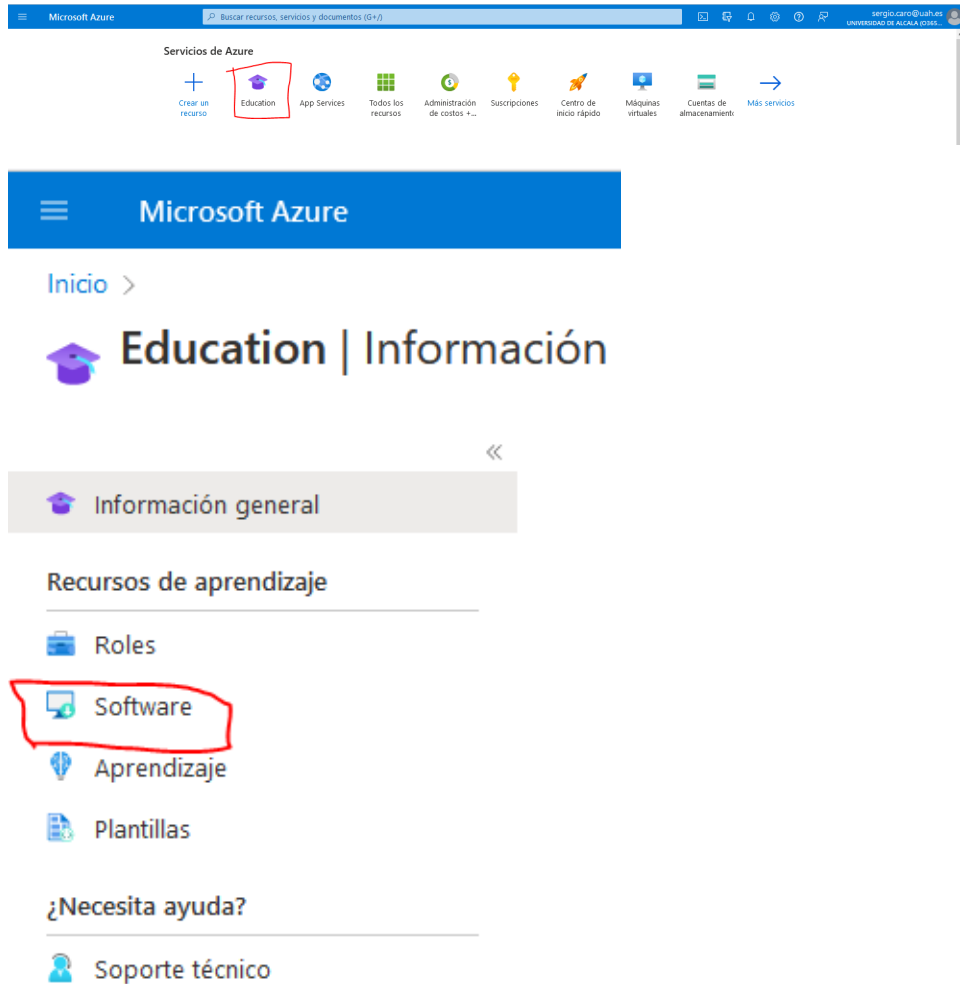
**Table 2. Windows Compiler Support in CUDA 10.2**

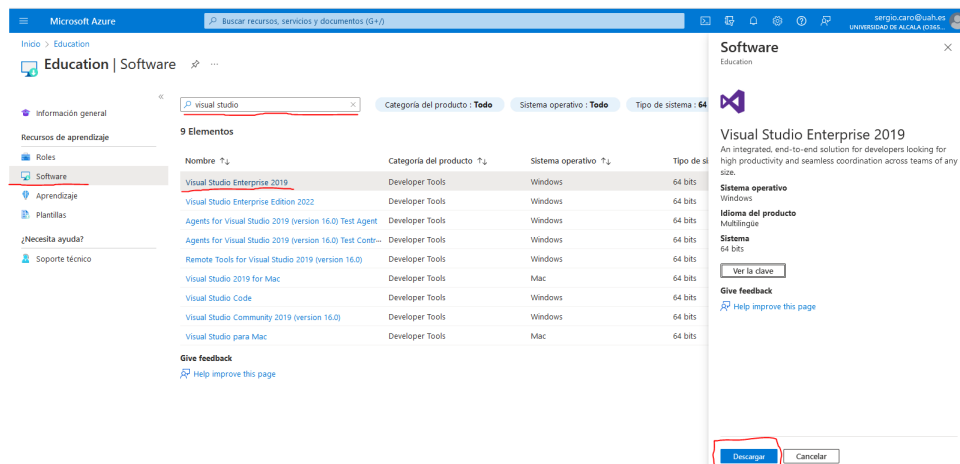
Compiler*	IDE	Native x86_64	Cross (x86_32 on x86_64)
MSVC Version 192x	Visual Studio 2019 16.x (Preview releases)	YES	NO
MSVC Version 191x	Visual Studio 2017 15.x (RTW and all updates)	YES	NO
MSVC Version 1900	Visual Studio 2015 14.0 (RTW and updates 1, 2, and 3)	YES	NO
	Visual Studio Community 2015	YES	NO
MSVC Version 1800	Visual Studio 2013 12.0	YES	YES
MSVC Version 1700	Visual Studio 2012 11.0	YES	YES

Como resumen, VS2015/2017 hasta la versión de CUDA 10 (inclusive). Posterior a eso, optar por VS2019.

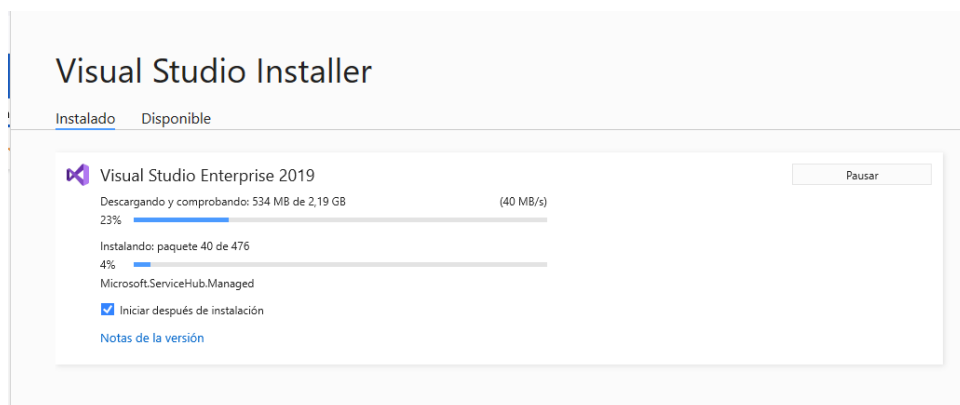
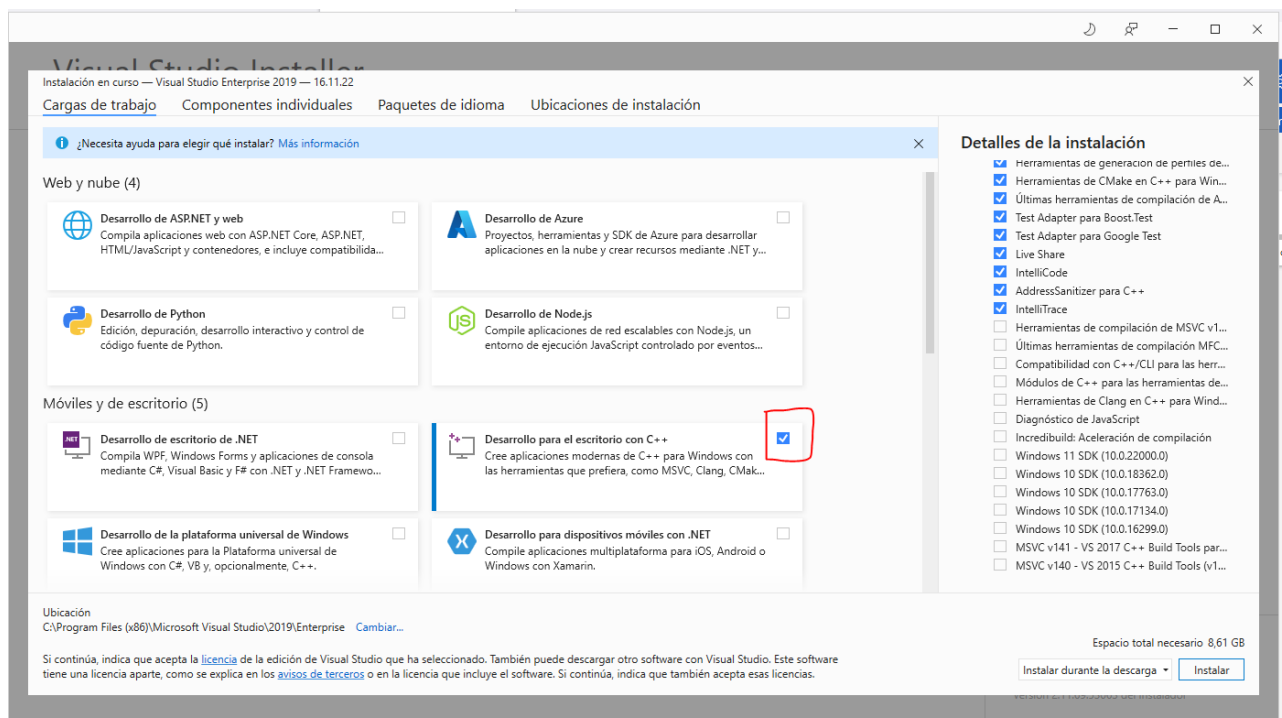
Las versiones previas de Visual Studio están aquí: <https://visualstudio.microsoft.com/es/vs/older-downloads/>.

VS 2019 se puede obtener en el Portal de Azure (<https://portal.azure.com>). Sección “Education”. En el panel lateral, seleccionar “software”. Se muestra una lista con todos los programas que se pueden descargar. Filtrar en el campo de búsqueda con “visual studio”. Seleccionar “Visual Studio Enterprise 2019”. En el panel lateral, descargar.





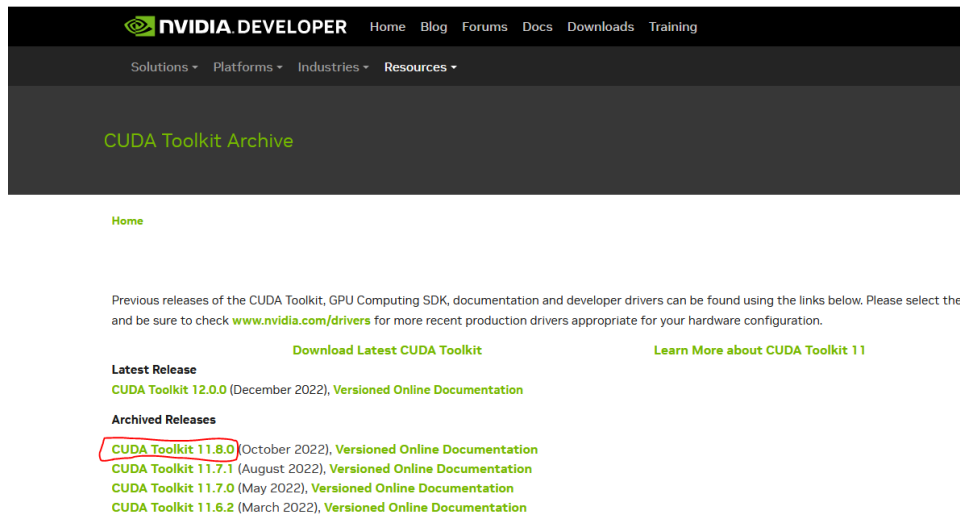
En el proceso de instalación, marcar la opción “Desarrollo para el escritorio con C++”. Eso nos instalará el compilador MSVC.



Nota: Parece que desde Visual Studio 2019 se puede instalar los MSVCs de VS 2015 y VS 2017. Desde aquí NO lo hemos probado, por lo que su idoneidad es desconocida.

Una vez instalado Visual Studio, pedirá reiniciar el equipo.

Regresar a la web de versiones de CUDA, <https://developer.nvidia.com/cuda-toolkit-archive>. Buscar la versión que queramos instalar y acceder al Toolkit deseado.



The screenshot shows the NVIDIA Developer website's CUDA Toolkit Archive page. The header includes the NVIDIA Developer logo and navigation links: Home, Blog, Forums, Docs, Downloads, and Training. Below the header, there are dropdown menus for Solutions, Platforms, Industries, and Resources. The main heading is "CUDA Toolkit Archive". A "Home" link is visible. A paragraph explains that previous releases of the CUDA Toolkit, GPU Computing SDK, documentation, and developer drivers can be found using the links below, and directs users to check [www.nvidia.com/drivers](https://www.nvidia.com/drivers) for more recent production drivers. There are two main sections: "Latest Release" and "Archived Releases". Under "Latest Release", there is a link for "Download Latest CUDA Toolkit" and "Learn More about CUDA Toolkit 11". Under "Archived Releases", there is a list of versions: "CUDA Toolkit 12.0.0 (December 2022), Versioned Online Documentation", "CUDA Toolkit 11.8.0 (October 2022), Versioned Online Documentation", "CUDA Toolkit 11.7.1 (August 2022), Versioned Online Documentation", "CUDA Toolkit 11.7.0 (May 2022), Versioned Online Documentation", and "CUDA Toolkit 11.6.2 (March 2022), Versioned Online Documentation". The "CUDA Toolkit 11.8.0" link is highlighted with a red box.

Primero se nos solicita el Sistema Operativo. Seleccionamos Windows.



The screenshot shows the NVIDIA Developer website's CUDA Toolkit 10.2 Download page. The header includes the NVIDIA Developer logo and navigation links: Home, Blog, Forums, Docs, Downloads, and Training. Below the header, there are dropdown menus for Solutions, Platforms, Industries, and Resources. The breadcrumb trail is: Home > High Performance Computing > CUDA Toolkit > CUDA Toolkit Archive > CUDA Toolkit 10.2 Download. The main heading is "CUDA Toolkit 10.2 Download". Below the heading, there is a section titled "Select Target Platform". A paragraph explains that users should click on the green buttons that describe their target platform, and only supported platforms will be shown. There are three buttons: "Operating System", "Windows", "Linux", and "Mac OSX". The "Windows" button is highlighted with a red box.

Tras esto, seleccionar la arquitectura. Que principalmente será x64. Y de “versión” seleccionar la versión de Windows que tengamos instalada.

## CUDA Toolkit 10.2 Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture	x86_64					
Version	10	8.1	7	Server 2019	Server 2016	Server 2012 R2

Ahora podremos descargar el instalador. La opción “network” descarga un .exe ligero y el resto de los componentes los descargará (según se necesiten) durante la instalación. La opción “local”, que ocupa más espacio, es un .exe con todos los componentes posibles.

## CUDA Toolkit 10.2 Download

Select Target Platform




Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture	x86_64					
Version	10	8.1	7	Server 2019	Server 2016	Server 2012 R2
Installer Type	exe (network) exe (local)					

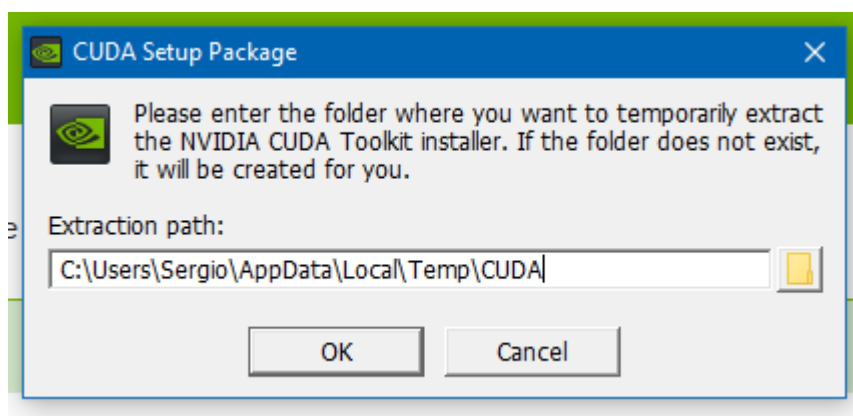
Por ahora, nos descargaremos el “Base Installer”. Los parches podrán instalarse posteriormente.

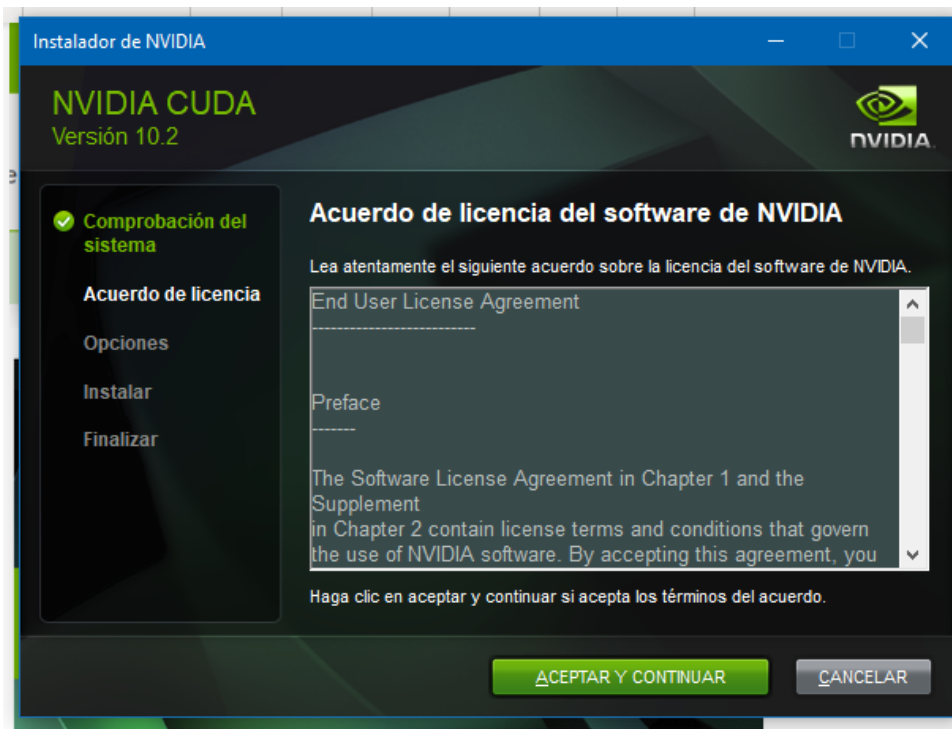
**Download Installers for Windows 10 x86\_64**

The base installer is available for download below.  
There are 2 patches available. These patches require the base installer to be installed first.

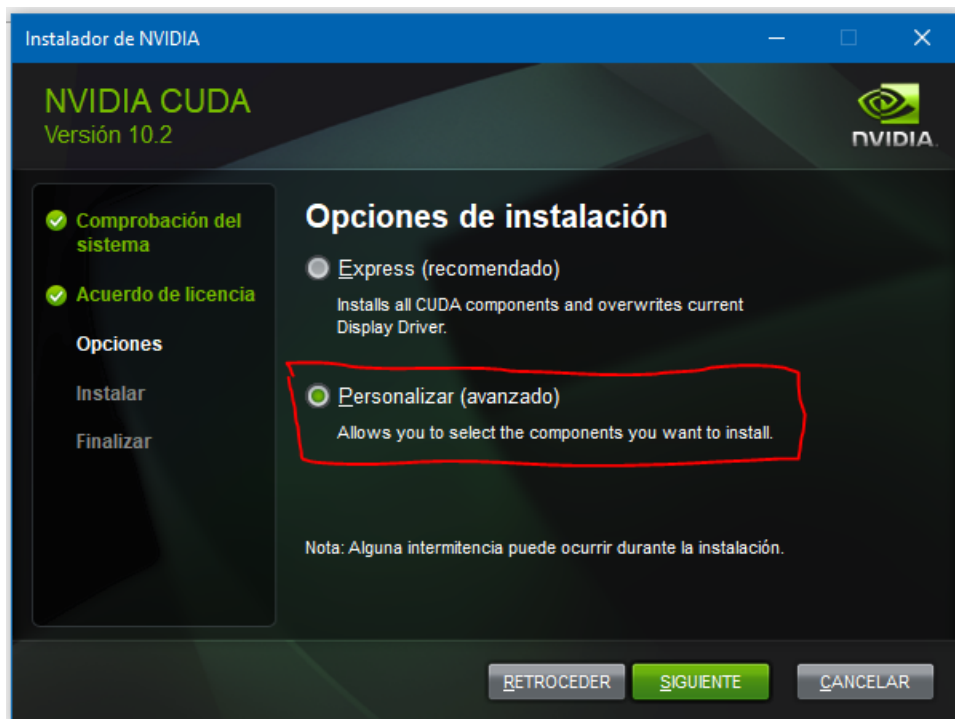
<b>&gt; Base Installer</b>	<b>Download (19.7 MB)</b> 
Installation Instructions: 1. Double click cuda_10.2.89_win10_network.exe 2. Follow on-screen prompts	
<b>&gt; Patch 1 (Released Aug 26, 2020)</b>	<b>Download (47.4 MB)</b> 
This patch fixes an issue in the cuBLAS library bundled in CUDA 10.2 which caused silent corruption of data in uncommon edge cases.	
<b>&gt; Patch 2 (Released Nov 17, 2020)</b>	<b>Download (47.5 MB)</b> 
This patch fixes an issue in cuBLAS library batched GEMM APIs which caused silent corruption of data in uncommon cases with large batch counts in mixed precision and fast math.	

Tras la descarga, procedemos a la instalación.





Seleccionamos la opción de instalación personalizada.

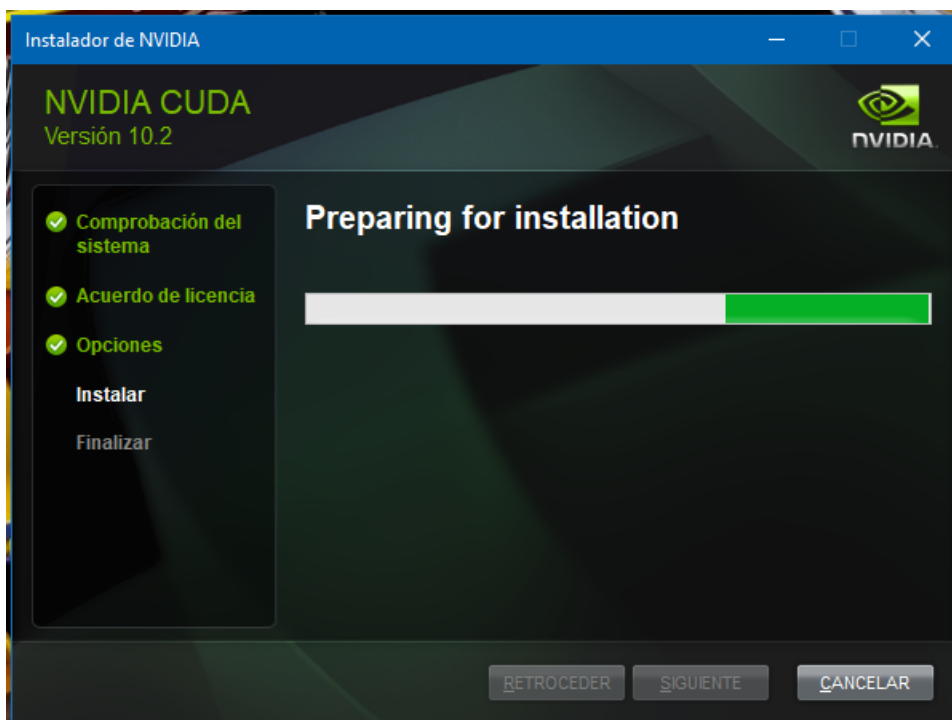
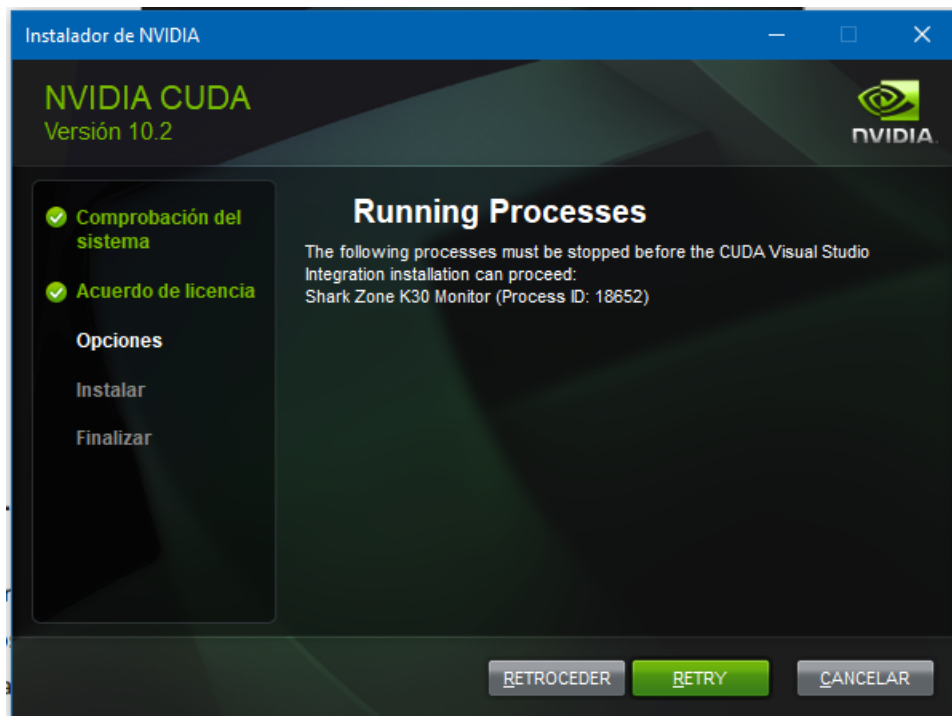


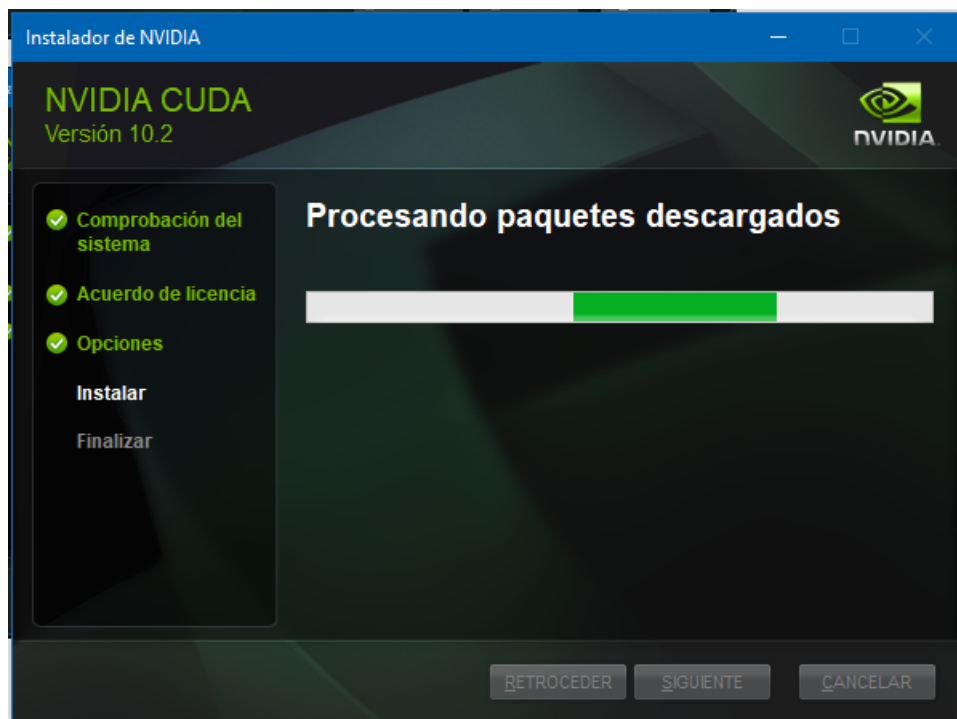
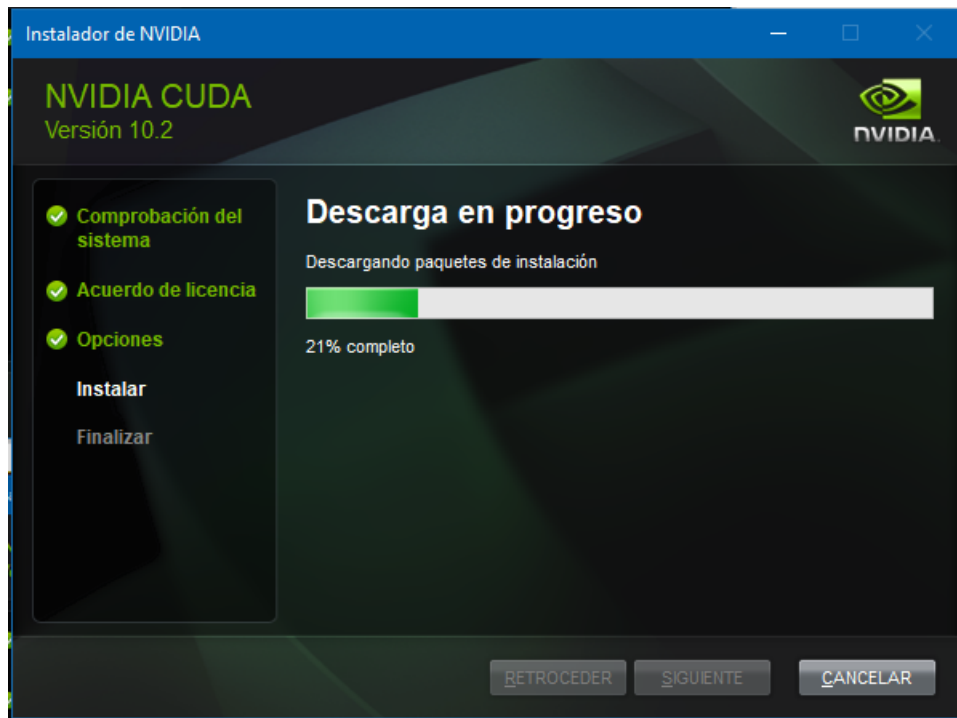
De aquellos elementos que tengamos una versión superior instalada, los desmarcaremos. Serán, principalmente, los controladores (drivers) de la GPU.





Puede que nos pida parar algún proceso abierto antes de instalar.



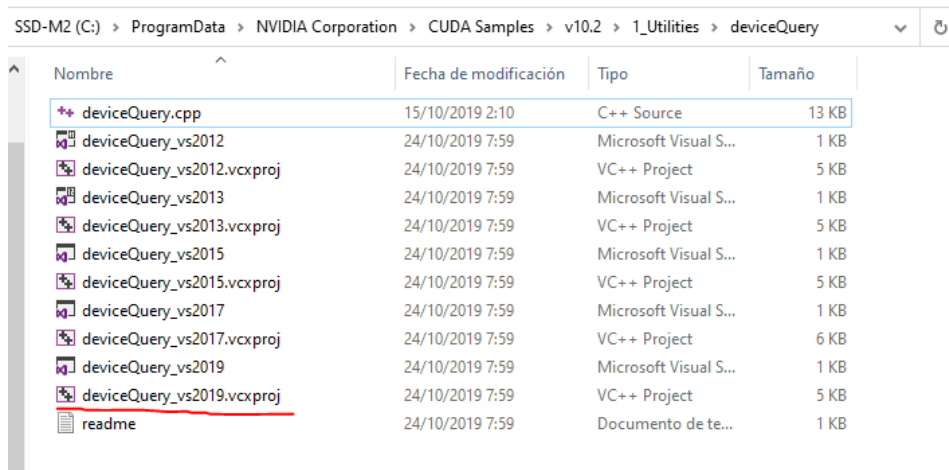


Al finalizar, procede a instalarnos componentes para Visual Studio.

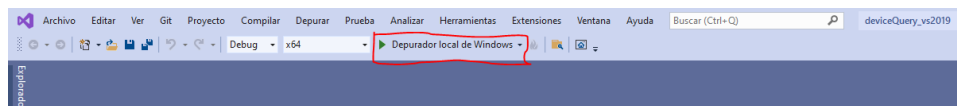


Los ejemplos de código los tenemos en la ubicación de instalación. En el ejemplo seguido aquí, "C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2"

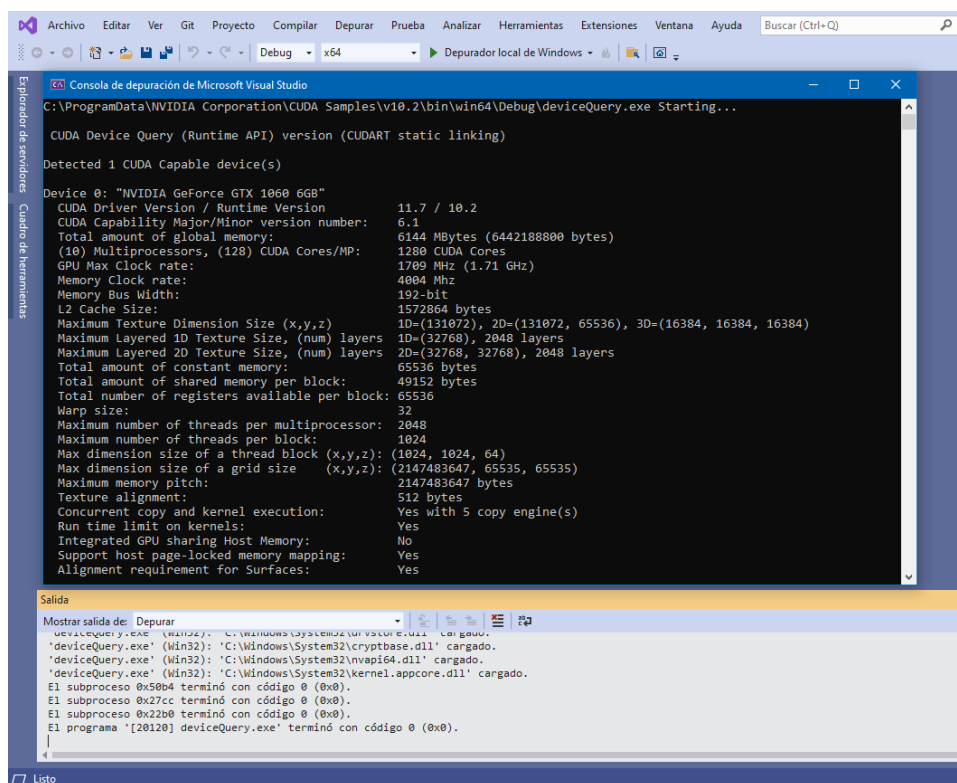
Abriremos uno para probar el funcionamiento. Abrir la carpeta, dentro de los códigos de ejemplo, "\CUDA Samples\v[xx]\1\_Uilities\deviceQuery". Seleccionar el fichero .vcxproj correspondiente a la versión de Visual Studio instalada.



Por ahora, nos dan igual las opciones. En la ventana del IDE, en la parte superior, seleccionar la opción de ejecución "> Depurador local de Windows".



Compilará y mostrará una ventana CMD con los resultados. En este caso, con las propiedades CUDA de nuestra GPU.



Puede que se muestren errores en el código. Esto es porque la herramienta de corrección (syntax highlight) no reconoce bien CUDA. Si ejecutamos no tendremos los mencionados problemas. Podemos añadir "#include<device\_launch\_parameters.h>" para que el corrector funcione correctamente. Aún así, es posible que persistan el marcado de algunos "errores".