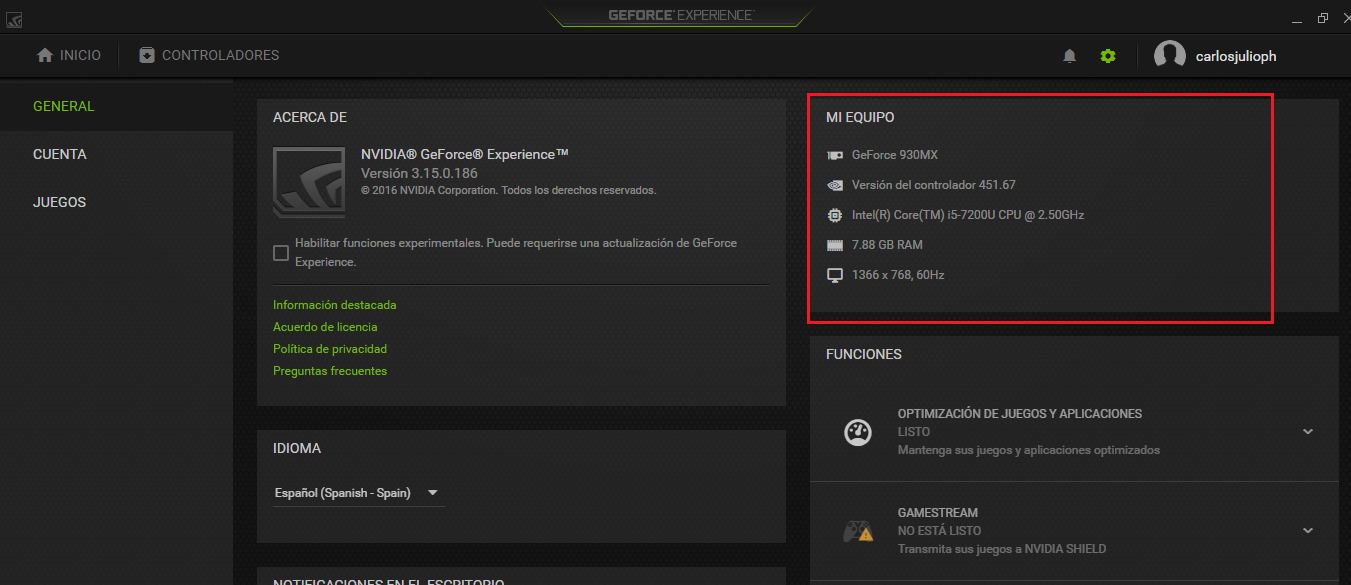
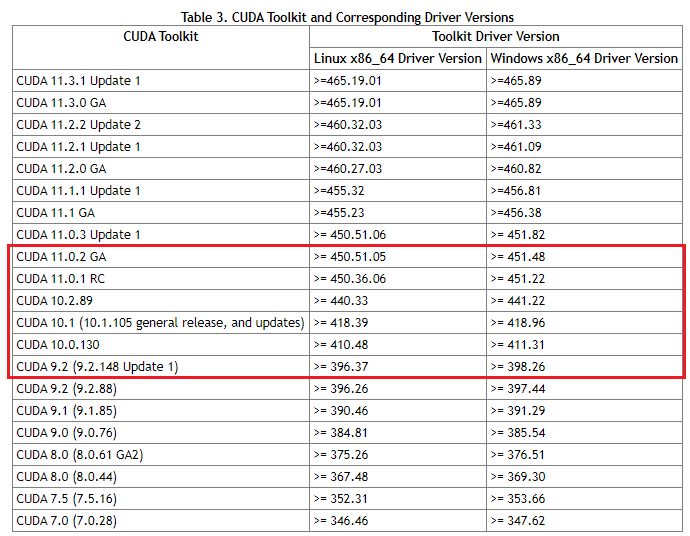
* Se realizó instalación de GeForce Experience v3.15.0.186
* Se verificó versión de driver:



La versión registrada es la 451.67

* Se establece la [versión de CUDA compatible](https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html):

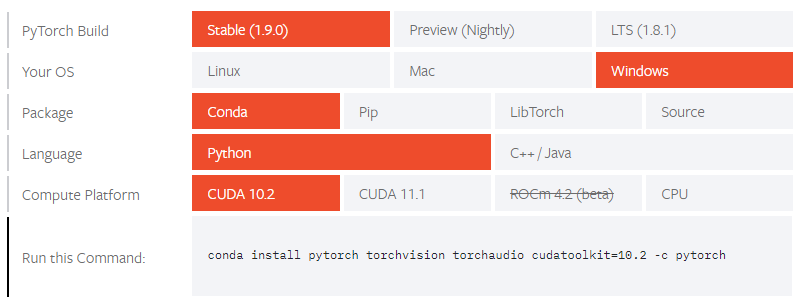


Es decir que podría instalar la versión 10.2 de CUDA (CUDA es un componente de software que se requiere para que la GPU pueda ser capaz de computar la GPU).

**INSTALACIÓN DE PYTORCH + CUDA TOOLKIT**

PyTorch se puede instalar y utilizar en varias distribuciones de Windows. Dependiendo de los requisitos de proceso y del sistema, su experiencia con PyTorch en Windows puede variar en términos de tiempo de procesamiento. Se recomienda, pero no es necesario, que su sistema Windows tenga una GPU NVIDIA para aprovechar toda la potencia del soporte CUDA de PyTorch.

1. Utilice las [instrucciones oficiales](https://pytorch.org/get-started/locally/) de instalación local de PyTorch.
2. Elija su configuración. Tenga en cuenta que no es necesario instalar CUDA por separado, pero debe seleccionar una versión de CUDA si su sistema es compatible con Nvidia GPU.

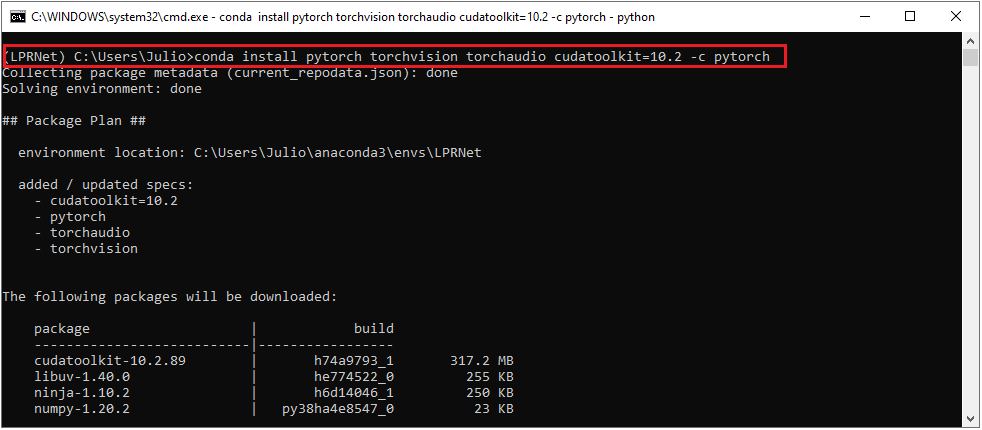


1. Creamos un entorno virtual en Anaconda.
2. Ejecute el comando recomendado en el paso 2 para completar la instalación.

conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch

**NOTA:** Tenga en cuenta que no hay necesidad de instalar CUDA por separado. El software CUDA necesario viene instalado con PyTorch si se selecciona una versión CUDA en el paso 2.

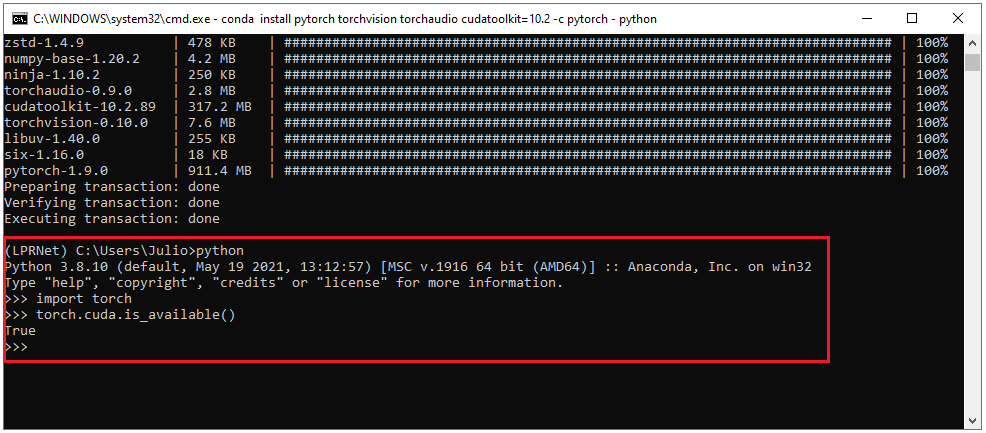
El kit de herramientas CUDA que se instala sólo es visible dentro del entorno en el que instalamos Pytorch con soporte CUDA, que es una gran ventaja. Imagine esta versión del kit de herramientas CUDA arruinando las versiones CUDA de su sistema global. El aislamiento completo es la mayor ventaja de usar un entorno virtual.



Puede comprobar la instalación utilizando como se ha explicado anteriormente. Para verificarlo en Python, utilice las siguientes líneas de código:

import torch

torch.cuda.is\_available()



Adicionalmente, para la verificación de instalación de Pytorch con soporte CUDA, ejecutamos el script Test\_GPU.py:

import torch

x = torch.rand(5, 3)

print(x)

if torch.cuda.is\_available():

print ("Cuda is available")

device\_id = torch.cuda.current\_device()

gpu\_properties = torch.cuda.get\_device\_properties(device\_id)

print("Found %d GPUs available. Using GPU %d (%s) of compute capability %d.%d with "

"%.1fGb total memory.\n" %

(torch.cuda.device\_count(),

device\_id,

gpu\_properties.name,

gpu\_properties.major,

gpu\_properties.minor,

gpu\_properties.total\_memory / 1e9))

else:

print ("Cuda is not available")

El resultado es el siguiente,

