

Enunciado practica 1 MPI-CUDA

- **Apartado 1: Hola mundo.**

Implementar función en CUDA que imprima por pantalla "helloWorld"

- Para declarar una función en CUDA debe empezar por `__global__`
- Para llamar a una función de CUDA: ***función<<<nBloques,nThreads>>>(args)***
- Para sincronizar hasta que terminen las llamadas a CUDA:
cudaDeviceSynchronize()
- Compilar con ***nvcc -arch=sm_30 -gencode arch=compute_30,code=sm_30 -g -G <nombre>.cu -o <nombre>***
- Probar con varias combinaciones de threads y de bloques, recordar que existe un limite

- **Apartado 2: Identificadores**

Implementar función en CUDA que imprima por pantalla "helloWorld", así como el identificador de thread, de bloque y numero de threads de cada bloque.

- Recordar las variables ***threadIdx, blockIdx, blockDim***
- Limitar la función para que solo imprima con los 10 primeros threads de cada bloque

- **Apartado 3: Suma**

Desarrollar un programa que realice una suma entre dos arrays de enteros, y guarde el resultado en un tercero

- Utilizar funciones:
 - ***cudaMalloc(void** array, int size)***
 - ***cudaMemcpy(void* dst, void* src, int size, int operation)***
 - operations: ***cudaMemcpyHostToDevice, cudaMemcpyDeviceToHost***
 - ***cudaMemset(void* array, int size)***

- **Apartado 4: Suma afín**

Implementar función en CUDA que, para cada posición de un array de enteros, realice una suma entre este valor y el de la posición anterior y posterior del array

- Utilizar la función ***__syncthreads()*** dentro del kernel