Suma de dos arreglos (bloques)

Una de las formas más sencillas de paralelizar es **replicando** el código que se escribe en un *kernel*.

En la instrucción

```
call add_dv<<<N,1>>>( dev_a, dev_b, dev_c )
```

se especifica que el código en `add_dv` se replique `N` veces, pero solamente en **un solo** hilo por bloque.

A este se le conoce, en inglés, como *embarrasingly parallel* y es una referencia a distribuir la misma tarea a muchos *agentes* de cómputo. En este caso, los bloques en la GPU, y la tarea siendo sumar dos números.

Suma masivamente paralela

Esencialmente, el esquema es el siguiente:

> A cada bloque se le da un par de números. Cada bloque los suma, y nos esperamos hasta que todos concluyan. Al final, se junta el resultado de cada bloque, y esto concluye el proceso.

Para **repartir** los trabajos, necesitamos asignar un *índice* del bloque a cada uno. Esto implica replicar el código del *kernel* para cada bloque.

De esta forma, se tiene el siguiente código

```
if ( tid <= n ) c(tid) = a(tid) + b(tid)
```

donde se espera que `tid` sea menor que el número total de elementos en los arreglos.

A este tipo de llamadas se les conocen coloquialmente como kernels **monolíticos**: se espera que exista un número muy grande de bloques e hilos (si se usan) para procesar todos los elementos de forma simultánea.