

# Programación con GPUs

## Guía 00 - Antes de Comenzar

Segundo Semestre, 2025

---

### 1.1 Introducción

#### ¡Bienvenidos!

Esta es una guía de instalación y configuración del entorno de trabajo para la materia, no es una guía de ejercicios o teoría de la materia. Te va a ayudar a configurar y entender cómo funciona el entorno de desarrollo de la materia.

¡Buena suerte!, no dudes en consultar cualquier duda que tengas.

Para poder trabajar en la materia vas a necesitar un entorno de desarrollo con CUDA que te permita correr los ejemplos y ejercicios localmente en tu computadora. No hay una única forma de configurar el entorno de desarrollo, así que te vamos a dar opciones para que elijas la que más que te guste, te sea más familiar o cómoda.

Toda la materia está en un repositorio git llamado Programación con GPUs, que incluye todo el material incluyendo: **teóricas**, **prácticas**, **soluciones**, código de ejemplo y un entorno de desarrollo **Docker** para poder correr los ejemplos y ejercicios de la materia, si no vas a instalar CUDA localmente.

### 1.2 GIT

No es obligatorio, pero te va a facilitar mucho las cosas que tengas instalado un cliente de git en tu computadora, sin embargo si no lo instalas tendrás que copiar algunos archivos del repositorio de forma manual. De todas formas `git` es una herramienta que seguramente ya utilices o utilizarás en el futuro, ya que es un sistema de control de versiones muy utilizado en la industria del software, que te permitirá conseguir las últimas actualizaciones de la materia, como para versionar código fuente. En este link podrás encontrar una lista de clientes de git, aunque editores como Visual Studio Code ya tienen formas de integrar git en su interfaz, o sino podrás usar el cliente de git de consola.

### 1.3 Entorno de desarrollo

El entorno de desarrollo CUDA para la materia se puede instalar de dos formas diferentes:

- **con Docker:** Recomendada, porque no requiere más que tener Docker Desktop o Colima
- **con instalación local:** Hacer una instalación local utilizando la guía oficial de CUDA de Nvidia.

#### 1.3.1 Instalación con Docker

Docker es una herramienta que te permite crear, desplegar y ejecutar aplicaciones en contenedores. Un contenedor es una unidad estandarizada que aísla una aplicación y todas sus

dependencias (sistema de archivos, bibliotecas, etc.) para que pueda ejecutarse de manera consistente en cualquier entorno. Esto significa que, en teoría, se puede ejecutar la misma aplicación en diferentes sistemas operativos independientemente de las bibliotecas y herramientas que estén previamente instaladas en el sistema *host*, ya que las dependencias de cada aplicación se instalan al construir el contenedor de docker.

Además, el *overhead* de Docker para correr aplicaciones es mínimo ya que no hace una virtualización completa del sistema como puede ser una máquina virtual, sino que utiliza la arquitectura subyacente del sistema operativo. Esto permite que los contenedores sean más ligeros y rápidos, lógicamente, la desventaja es que al no virtualizar no se puede correr cualquier sistema operativo en el contenedor.

Para correr el container de docker deberás instalar alguna de las alternativas para correr docker como son: Docker Desktop o Colima para correr el container de docker provisto por la cátedra.

### 1.3.2 Makefile y comandos de docker

Hemos provisto un archivo **Makefile** para poder levantar el container de docker con facilidad, sin embargo, es posible que tu sistema no tenga el comando **make** instalado. Y nuevamente hay dos opciones aquí. Por un lado se pueden correr los comandos de **docker** por separado, o se puede usar el comando **make** para correr las opciones disponibles en el **Makefile**.

#### Dockerfile

El archivo **Dockerfile** no es muy importante para la materia, pero es bueno que sepas qué es. El **Dockerfile** es un archivo de texto que contiene una serie de instrucciones para construir una imagen de docker. Allí se especifica qué comandos deben ejecutarse para construir cada container a partir de una imagen base.

#### Makefile

Adicionalmente también te recomendamos instalar el comando **make** para poder correr los comandos de docker ya que los comandos están escritos en un **Makefile** y no tendrás que escribirlos a mano.

El archivo **Makefile** es un archivo que te permite ejecutar scripts de forma sencilla. En este caso, el **Makefile** contiene una serie de instrucciones para construir el container de docker y ejecutar los comandos necesarios para entrar al shell del container.

Si el comando **make** está instalado en tu computadora entonces podrás correr:

```

1 % make
2
3 -----
4             UP | Introduccion a la Programacion Paralela (CUDA)
5                   CUDA docker container
6 -----
7
8 make build           # Construye el container
9 make shell          # Abre un shell en el container

```

Caso contrario deberás leer el archivo **Makefile** y correr los comandos de docker a mano. Por ejemplo, para construir el container de docker deberás correr (en Linux o MacOS):

```

1 % docker run -ti -v $(PWD):/app --rm up-cuda /bin/bash

```

O si estás en Windows:

```

1 % docker run -ti -v %cd%:/app --rm up-cuda /bin/bash

```

Si vas a correr los comandos manualmente deberás leer el archivo `Makefile` para ver los comandos disponibles y la sintaxis de cada uno.

## 1.4 Instalación local

Si tu preferencia es instalar todo localmente, deberás seguir la guía oficial de CUDA de NVIDIA, ya que la instalación de CUDA depende del sistema operativo que estés utilizando.