

# CUDA

O Motor Paralelo para Computação de Alta  
Performance da NVIDIA



nVIDIA

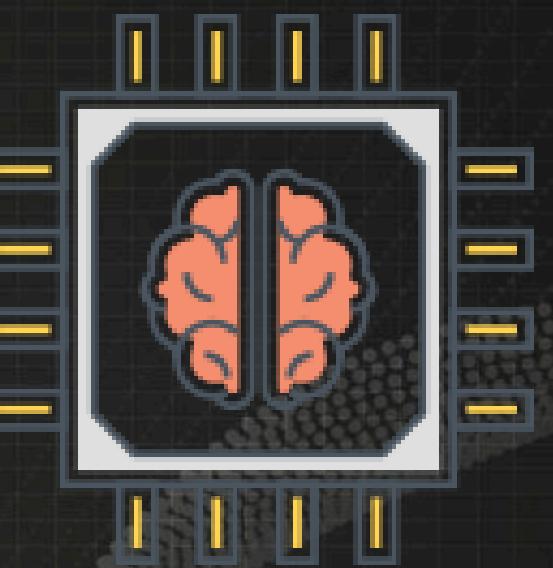
# A Necessidade de Mais Poder Computacional

- Aumento exponencial de dados (Big Data).
- Simulações cada vez mais complexas (Clima, Física, Biologia).
- Inteligência Artificial exige processamento massivo.
- CPUs tradicionais chegam aos seus limites para tarefas paralelas.



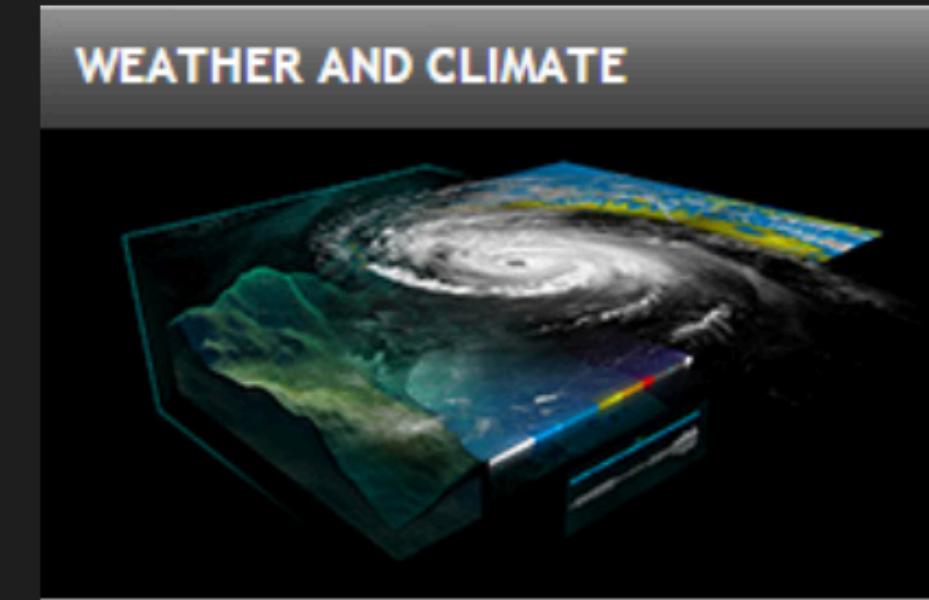
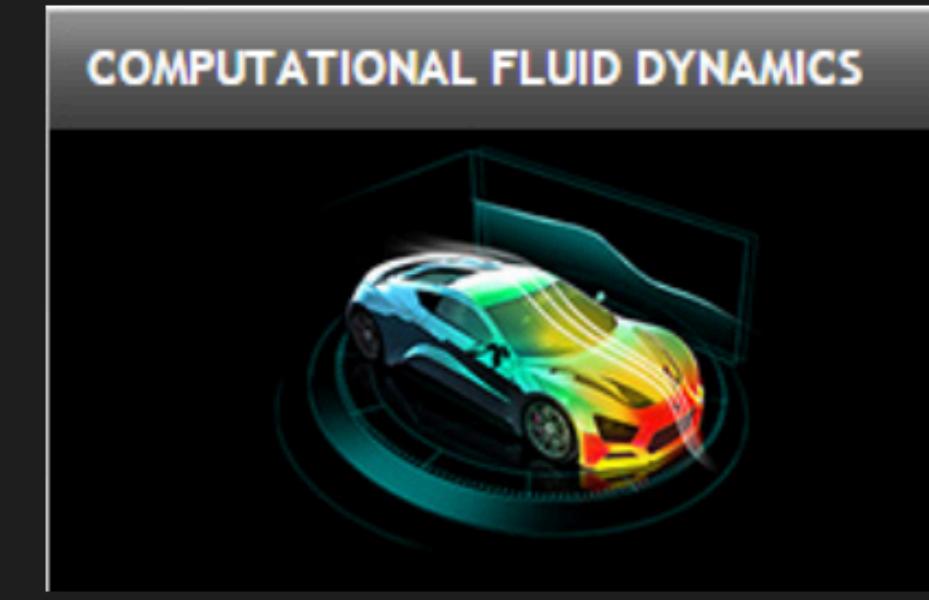
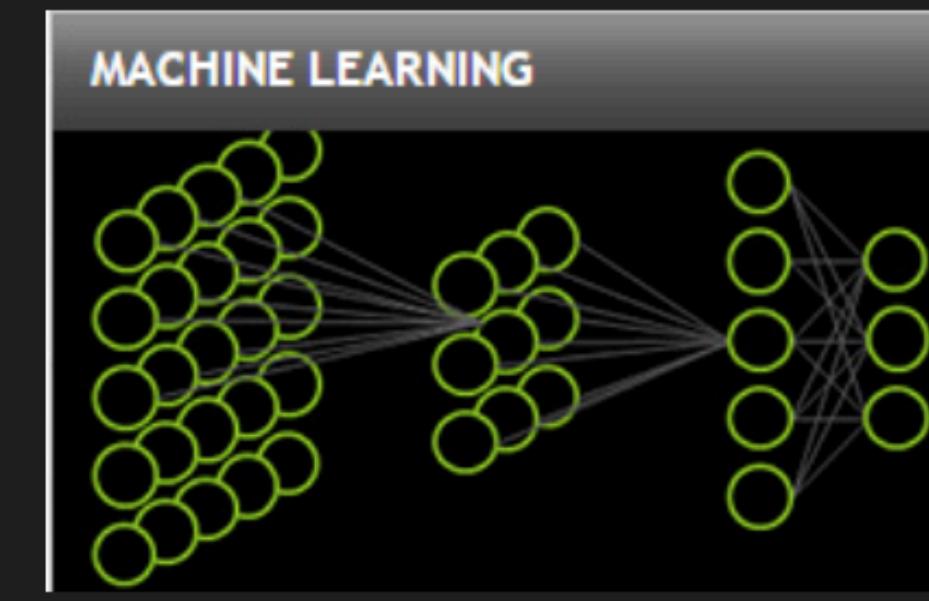
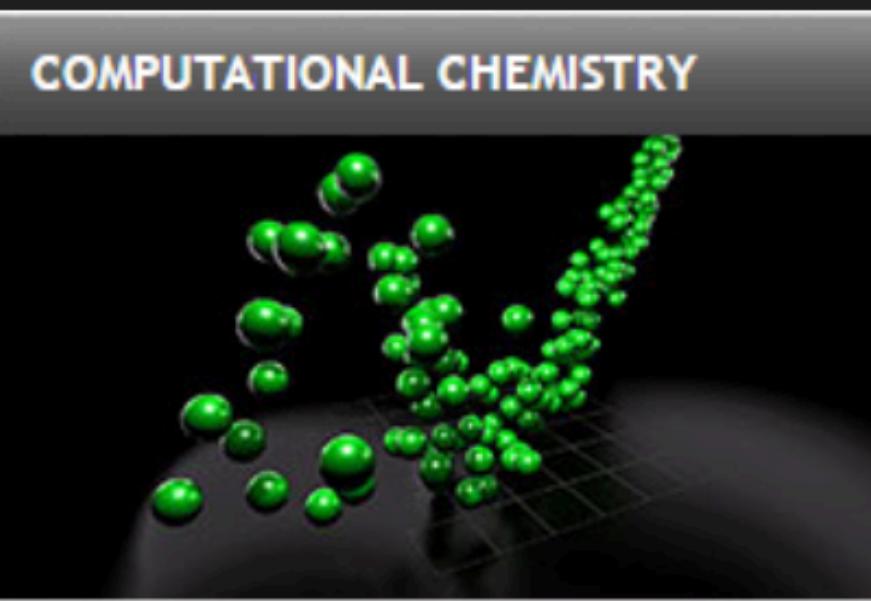
# A Chave para o Superprocessamento

- GPU (Graphics Processing Unit):
  - > Evoluiu dos jogos para a computação geral.
  - > Arquitetura com milhares de núcleos = Mestre do Paralelismo.
- CUDA (Compute Unified Device Architecture):
  - > Plataforma da NVIDIA para programar GPUs.
  - > Libera o poder da GPU para qualquer tarefa.
  - > CUDA = GPU NVIDIA + Programação Paralela

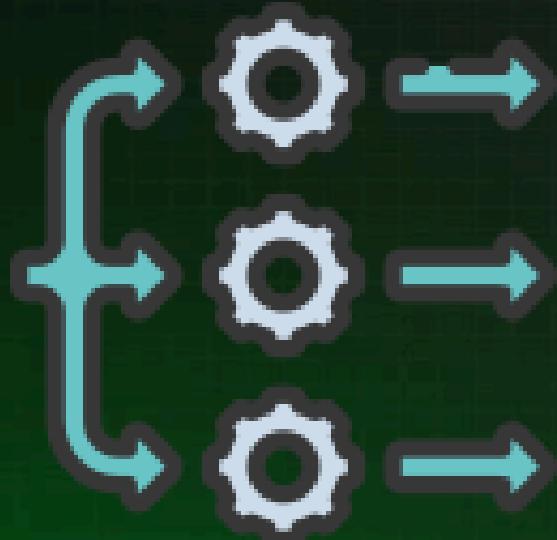


# Domínios com aplicativos acelerados por CUDA

CUDA acelera aplicações em uma ampla gama de domínios, desde processamento de imagens até aprendizado profundo, análise numérica e ciência computacional.



# Programação Paralela: Milhares de Tarefas ao Mesmo Tempo!



- CPU (Host): O Gerente - organiza e distribui o trabalho.
- GPU (Device): O Exército - executa o trabalho pesado.
- Kernel: A sua "função" que roda na GPU.
- Threads: Milhares de "cópias" do seu kernel rodando em paralelo.
- Blocos / Grids: Como organizamos nosso "exército" de threads.
- Resultado: Processamento massivamente paralelo.

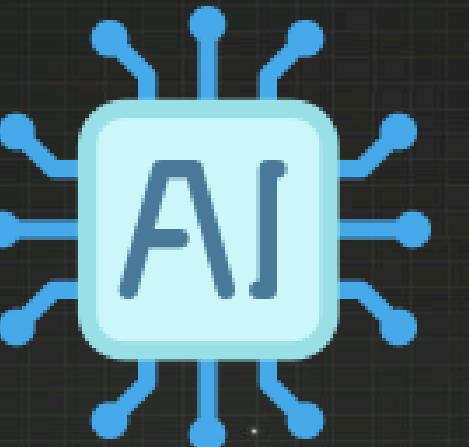
# Os Superpoderes: Velocidade e Eficiência

- **Aceleração Massiva:** Tarefas até 100x mais rápidas que em CPUs.
- **Processa Big Data:** Ideal para analisar grandes volumes de dados.
- **Viabiliza Inovação:** Permite simulações e modelos complexos.
- **Motor da IA:** Essencial para a revolução do Deep Learning.



# CUDA e a Revolução da Inteligência Artificial

- Essencial para Treinamento e Inferência de Redes Neurais.
- Aplicações: Visão computacional, processamento de linguagem, diagnósticos médicos.
- Bibliotecas: cuDNN acelera drasticamente o Deep Learning.
- (GPUs vs. NPUs):
  - > GPUs (CUDA): Flexíveis e poderosas para treino e inferência.
  - > NPUs: Superespecializadas para inferência com baixo consumo (celulares).



# Ciência, Jogos e Indústria Turbinados

- Ciência: Simulações 3D (Física, Química, Clima).
- Jogos: Física realista, IA avançada, efeitos visuais.
- Finanças: Análise de risco, modelos de mercado.
- Engenharia: CAD, simulações de fluidos.
- Mídia: Edição e renderização de vídeo 8K.



# CUDA: uma Plataforma Completa!

**CUDA não é uma "nova linguagem":**

**Ele estende linguagens que já conhecemos, como Phyton, C ou C++, com comandos especiais para se comunicar com a GPU.**

**Ecossistema CUDA, onde várias peças trabalham juntas:**

- 1. Modelo de Programação:** A forma de pensar e escrever o código paralelo (**kernels, threads, blocos, grids**).
- 2. Hardware Específico:** As GPUs da NVIDIA, com seus "**CUDA Cores**" feitos para esse tipo de tarefa.
- 3. Software (O CUDA Toolkit):** O pacote de desenvolvimento fundamental, que nos fornece o **compilador, bibliotecas e as ferramentas necessárias para criar e otimizar programas para as GPUs NVIDIA**.

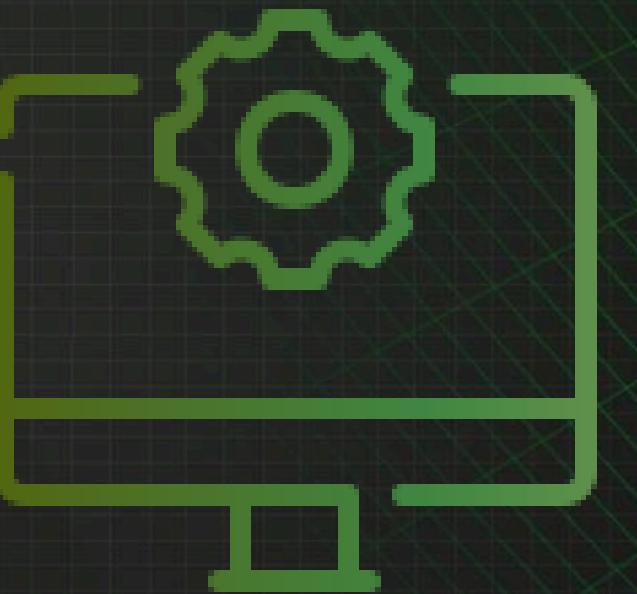
# "Hello, CUDA!"

```
// Função executada por CADA thread na GPU
__global__ void addVectors(float* a, float* b, float* c, int n) {
    // Calcula o índice único para esta thread
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    // Cada thread soma UM elemento
    if (idx < n) {
        c[idx] = a[idx] + b[idx];
    }
}

// Na CPU, você chamaria:
// addVectors<<<numBlocks, threadsPerBlock>>>(a_gpu, b_gpu, c_gpu, N);
```

# Principais Ferramentas e Bibliotecas

- CUDA Toolkit: Compilador (NVCC), APIs, Ferramentas.
  - > Bibliotecas
  - > cuDNN: Redes Neurais.
  - > cuBLAS: Álgebra Linear.
  - > cuFFT: Transformadas de Fourier.
  - > Thrust: Algoritmos C++.
  - > CuPy: integrando CUDA no Python.
- Linguagens: C/C++, Python, Fortran e mais.



# O Universo CUDA (Exemplo com Python/CuPy)

```
import cupy as cp
```

# 1. Cria 'arrays' DIRETAMENTE na GPU

```
a = cp.array([1.0, 2.0, 3.0, 4.0])
```

```
b = cp.array([10.0, 20.0, 30.0, 40.0])
```

# 2. Soma A + B = C na GPU!

# (CuPy usa CUDA por baixo dos panos aqui)

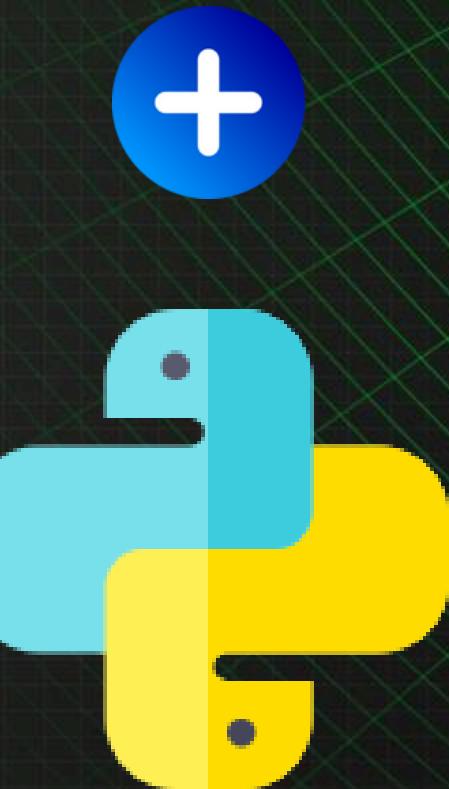
```
c = a + b
```

# 3. Mostra o resultado (que ainda está na GPU,

# mas o print o traz para a CPU para visualização)

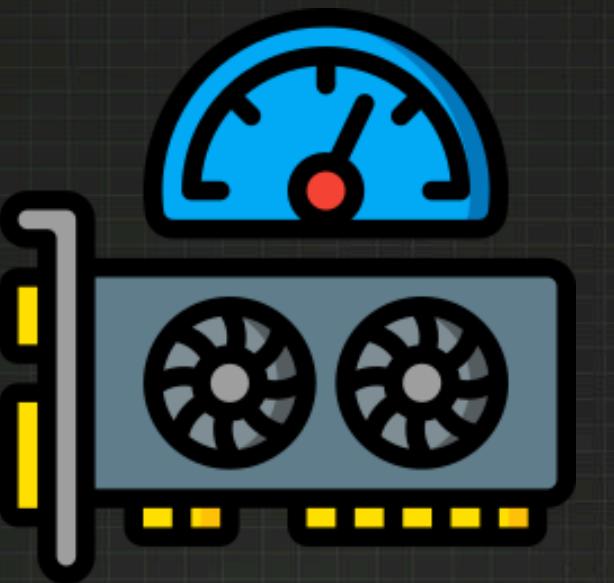
```
print("CuPy (A+B):", c)
```

# Saída Esperada: CuPy (A+B): [11. 22. 33. 44.]



# O Futuro é Paralelo e Acelerado por GPU

- CUDA transforma GPUs NVIDIA em poderosas ferramentas computacionais.
- É um pilar fundamental para avanços em IA, Ciência e Tecnologia.
- A computação paralela está mudando o mundo.
- O poder da GPU está ao seu alcance!



# Referências

- Guia de Programação CUDA C++. Disponível em: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/>
- Domínios com aplicativos acelerados por CUDA. Disponível em: <https://developer.nvidia.com/cuda-zone>
- Vídeos, palestras e documentação do CUDA. Disponível em: <https://www.nvidia.com/en-us/on-demand/search/?q=cuda&searchPath=%2Fen-us%2Fon-demand%2F&sort=relevance>

# Dúvidas? Pergunte para a gente!

Obrigado!

## Integrantes:

- Davi Santos | [davi.seabra.121@ufrn.edu.br](mailto:davi.seabra.121@ufrn.edu.br)
- Gustavo Martins | [gustavo.martins.119@ufrn.edu.br](mailto:gustavo.martins.119@ufrn.edu.br)
- Samuel Nascimento | [samuel321fernandes@gmail.com](mailto:samuel321fernandes@gmail.com)

Tads  
EAJ/UFRN