

Guida per MulticoreTemplate notebook

[Axel Rubini]

November 27, 2025

Contents

1 Descrizione

Questa guida fornisce istruzioni dettagliate per l'uso del notebook `MulticoreTemplate` in Google Colab. I notebook sono già presenti nella repository dove è presente questa guida.

2 Prerequisiti

- Account Google (per accedere a Google Colab)
- GitHub (opzionale)
- Accesso a Google Drive

3 Setup Iniziale

3.1 Apertura del Notebook

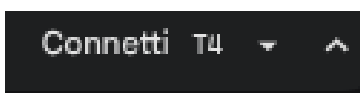
1. Vai su <https://colab.research.google.com/>
2. Caricare il notebook dal computer:
 - Clicca su **File > Carica notebook**
 - Seleziona la scheda **Carica** e scegli il file `MulticoreTemplate.ipynb` dal tuo computer

3.2 Configurazione Runtime

1. Vai su **Runtime > Cambia tipo di runtime**
2. Seleziona le seguenti impostazioni:
 - **Tipo di runtime:** Python 3
 - **Acceleratore hardware:** [GPU - T4]

3.3 Connessione al Runtime

- Clicca su **Connetti** in alto a destra
- Attendi che il runtime sia pronto (icona verde con checkmark)



4 Struttura del Notebook

4.1 Sezione 1: [Configurazione Iniziale e Esercizi Futuri]

Scopo: [In questa sezione del notebook sono presenti gli script che creano i repository per gli esercizi, ovvero verrà creata una main directory che conterrà le sottodirectory per l'esercizio attuale e per i futuri esercizi.]

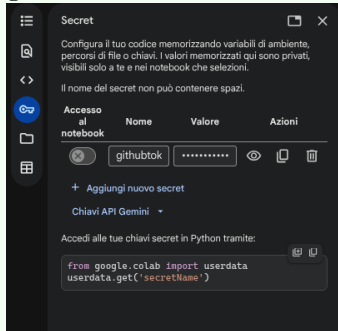
Celle da eseguire:

- [1] - [La prima cella server per creare la main directory se non è già presente. ed a creare le repository git per l'esercizio attuale]
- [2] - [In questa cella è presente uno script che permette di visualizzare le caratteristiche della GPU.]

- [3 (opzionale)] - [Questa cella configura git e crea una repository sul tuo account GitHub. L'accesso a GitHub deve essere eseguito tramite token personale, da inserire ogni volta che viene fatta eseguire la cella per la prima volta nel runtime e rimane valido fino a che il runtime non viene riavviato.]

i Nota

Se volete usare GitHub consiglio di creare un token e di salvarvelo dentro l'apposito gestore di chiavi di colab



4.2 Sezione 2: [Caricare il proprio codice]

Scopo: [Caricare il proprio file.cu all'interno della directory dell'esercizio attuale.]

Celle da eseguire:

- [4] - [In questa cella e' presente uno script che permette di caricare il proprio file.cu all'interno della directory dell'esercizio attuale, oppure utilizzare gia' i file esistenti all'interno del progetto.]

i Nota

Se si vuole caricare un file che ha lo stesso nome di uno attuale e' consigliato eliminare il vecchio file a mano attraverso il terminale di lavoro o con il side tree presente su CoLab.

4.3 Sezione 3: [Eseguire il codice e visualizzare i risultati]

Scopo: [Le celle successive servono per compilare, pulire, eseguire e visualizzare i risultati del proprio codice all'interno del notebook.]

Celle da eseguire:

- [5] - [Compila il codice sorgente .cu i parametri pre configurati sono quelli per il run su GPU T4, se si vuole cambiare tipo di GPU o altri parametri e' possibile modificare la cella.]
- [6] - [Esegue il clean della directory di lavoro, eliminando i file di output generati dalle esecuzioni precedenti.]
- [7] - [Esegue il codice compilato.]

4.4 Sezione 4: [Salvataggio dei risultati e caricamento su GitHub]

Scopo: [Le celle successive servono per generare un po' di documentazione e sincronizzare la repository su GitHub la descrizione e' omessa per brevità.]