DEMUESTRA - Guida completa di installazione e utilizzo

Il progetto completo **DEMUESTRA** può essere scaricato o consultato su GitHub, utente **quark-cha**, repository **DEMUESTRA**: https://github.com/quark-cha/DEMUESTRA.

1. Che cos'e DEMUESTRA

DEMUESTRA e un ambiente per sottoporre una dimostrazione matematica a un processo di analisi strutturale e verifica formale mediante **Lean 4**.

Il ricercatore parte dalla propria dimostrazione scritta in un file Markdown `.md`.

Durante il processo si puo utilizzare una **IA gratuita come strumento ausiliario** per compiti come:

- scrivere correttamente le formule in LaTeX;
- trasformare la dimostrazione in JSON;
- trasformare le proposizioni matematiche in Lean;
- localizzare errori;
- collegare un errore al punto corrispondente della dimostrazione;
- aiutare a correggere problemi di sintassi o di formalizzazione.

La IA **non interviene nel processo creativo della dimostrazione**.

La IA non apporta l'idea matematica e non sostituisce il ragionamento del ricercatore: agisce solo come strumento per aggiustare formalmente, ordinare, tradurre formati e aiutare a esprimere con precisione una dimostrazione che appartiene gia all'autore.

La proposta, le sue ipotesi, definizioni, ragionamenti e conclusioni appartengono all'autore.

DEMUESTRA non pretende nemmeno di decidere, mediante un semplice punteggio, se una teoria scientifica sia vera o falsa.

Il suo obiettivo e molto piu concreto:

rendere esplicita la catena deduttiva e verificare formalmente che i passaggi matematici portati in Lean siano costruiti correttamente a partire dalle loro premesse.

---

2. Autore, licenza e uso etico

Questo progetto fa parte dello sviluppo di **Victor Estrada Diaz**.

Sei libero di usare queste utilita nel tuo progetto, purche tu rispetti la mia paternita dell'opera e non le usi a fini commerciali.

Sei anche libero di distribuirle e diffonderle, purche tu indichi chiaramente la parte di cui sono autore.

La licenza applicabile e:

```
CC BY-SA-NC
```

L'uso di qualsiasi parte di questo sviluppo e proibito in ambienti non etici, inclusi:

- guerre;
- usi che non rispettano i Diritti Umani;
- sfruttamento o abuso di persone o animali;
- usi destinati a umiliare o svalutare altre persone.

Qualsiasi uso di questo tipo viola la mia licenza.

---

3. Repository aggiornato

Il progetto **DEMUESTRA** ha un repository pubblico da cui puoi scaricare la versione aggiornata e accedere a tutti i suoi file.

Utente GitHub:

```
quark-cha
```

Repository:

```
DEMUESTRA
```

PARTE I - INSTALLAZIONE

4. Che cosa devi installare

Prima di utilizzare DEMUESTRA devi preparare il computer.

Obbligatorio

Installa:

1. **Python**
2. **Lean 4**
3. il repository **DEMUESTRA**

Consigliato

Consigliamo anche:

4. **Anaconda**
5. **Visual Studio Code**

Visual Studio Code non e indispensabile, ma facilita molto la modifica dei file `.md`, `.json` e `.lean`.

---

5. Scaricare DEMUESTRA

Scarica o clona il repository **DEMUESTRA** nella tua cartella abituale di sviluppo.

Per esempio, puoi avere una struttura generale simile a:

```
MIS_PROYECTOS/
│
├── PROYECTO_A/
├── PROYECTO_B/
├── PUBLICAR/
└── DEMUESTRA/
```

I tuoi progetti, **PUBLICAR** se lo usi, e **DEMUESTRA** si trovano quindi allo stesso livello.

DEMUESTRA non ha bisogno di modificare direttamente i tuoi progetti originali.

PUBLICAR e opzionale. Non e necessario installare PUBLICAR per utilizzare DEMUESTRA. PUBLICAR serve solo se vuoi usare anche quell'altro sistema di generazione di PDF e pubblicazione.

PARTE II - PREPARARE UNA DIMOSTRAZIONE

6. Il documento originale

Supponiamo che tu abbia un progetto chiamato:

```
MI_TEORIA/
```

e che al suo interno tu abbia scritto una dimostrazione:

```
MI_TEORIA/
└── demostracion.md
```

Questo `.md` e il tuo documento originale.

Non useremo direttamente questo file durante le prove.

Prima facciamo una copia per lavorarci dentro DEMUESTRA.

---

7. Copiare il Markdown in DEMUESTRA

Copia il file `.md` che vuoi dimostrare nell'area dei progetti di DEMUESTRA.

Concettualmente:

```
MI_TEORIA/
└── demostracion.md
          |
          | COPIARE
          v
DEMUESTRA/
└── proyectos/
    └── MI_TEORIA/
        └── demostracion.md
```

Da questo momento bisogna distinguere chiaramente due documenti:

```
MI_TEORIA/demostracion.md
```

e il documento che rimane nel tuo progetto originale.

Mentre:

```
DEMUESTRA/proyectos/MI_TEORIA/demostracion.md
```

e la **copia di lavoro**.

Regola importante

Durante tutto il processo di verifica:

lavora ed effettua le correzioni sul file `.md` situato dentro DEMUESTRA.

Non hai bisogno di modificare continuamente il progetto originale.

Quando avrai completato soddisfacentemente tutto il processo, potrai copiare il file `.md` corretto da DEMUESTRA al tuo progetto originale.

PARTE III - COME SCRIVERE IL MARKDOWN

8. Il file `.md`

Markdown e semplicemente un formato di testo semplice.

Puoi scriverlo con qualsiasi editor di testo, anche se Visual Studio Code facilita il lavoro.

Le formule matematiche si scrivono utilizzando LaTeX.

Formule all'interno di una riga

Nel tuo file Markdown reale, una formula all'interno di una riga si delimita con il carattere $ all'inizio e un altro carattere $ alla fine.

Deve essere scritta cosi, come testo:

$formula$

Per esempio:

L'energia e data da $E=mc^2$.

Formule indipendenti

Nel tuo file Markdown reale, una formula indipendente nel proprio paragrafo si delimita con due caratteri $$ all'inizio e due caratteri $$ alla fine.

Deve essere scritta cosi, come testo:

$$formula$$

Per esempio:

$$E=mc^2$$

Puoi chiedere a una IA di revisionare il documento e collocare correttamente le formule e i loro delimitatori.

PARTE IV - IL PDF

9. Devo generare un PDF?

No.

Il PDF non e necessario per realizzare la dimostrazione.

Il documento fondamentale di lavoro e il file `.md`.

Per iniziare a lavorare hai bisogno solo di quel file `.md`.

Con il file `.md` e l'aiuto di una IA puoi estrarre le derivazioni aritmetiche e deduttive che saranno poi portate nel file Lean.

Puoi generare un PDF se vuoi disporre di una versione destinata alla lettura, alla distribuzione o alla pubblicazione.

Puoi farlo mediante una IA o mediante qualsiasi strumento specifico di conversione.

Nel mio caso utilizzo un altro progetto chiamato **PUBLICAR**.

PUBLICAR converte i miei progetti in PDF e automatizza la loro pubblicazione su Zenodo e sul mio sito web.

Pero PUBLICAR utilizza la mia propria struttura di progetti.

Per questo:

PUBLICAR e opzionale e non e necessario per utilizzare DEMUESTRA.

Non devi installare PUBLICAR se vuoi soltanto verificare una dimostrazione mediante DEMUESTRA.

PARTE V - ANALISI STRUTTURALE

10. Convertire il Markdown in JSON

Una dimostrazione puo sembrare evidente a chi l'ha sviluppata e, tuttavia, contenere concetti che non sono stati definiti esplicitamente o relazioni date per scontate.

Per questo realizziamo prima un'analisi strutturale.

Puoi chiedere a una IA gratuita di trasformare la tua dimostrazione Markdown nella struttura JSON richiesta dall'analizzatore di **Gianluca R. Pisano**.

Il processo e:

```
demostracion.md
      |
      v
     IA
      |
      v
demostracion.json
```

Qui la IA sta realizzando una **traduzione di formato e struttura**.

Non deve inventare nuove premesse ne modificare la tua dimostrazione per farla passare l'analisi.

---

11. Passare il JSON nell'analizzatore

Inserisci il JSON ottenuto nell'analizzatore di Gianluca R. Pisano.

L'analizzatore puo rilevare problemi come:

- simboli che appaiono senza essere stati definiti;
- concetti specificati in modo insufficiente;
- dipendenze non chiare;
- proposizioni che utilizzano risultati precedenti senza identificarli;
- salti nella catena deduttiva;
- ipotesi che devono essere rese esplicite;
- problemi di struttura formale.

---

12. Un errore dell'analizzatore non significa necessariamente che tu abbia torto

Questo punto e fondamentale.

Se l'analizzatore rileva un problema, **non significa automaticamente che l'affermazione matematica sia falsa**.

Puo significare, per esempio, che sai perfettamente perche un risultato segue dal precedente, ma non lo hai scritto esplicitamente.

In quel caso il problema e nell'esposizione formale della dimostrazione.

Ma non dobbiamo nemmeno ignorare l'errore.

Bisogna capire esattamente che cosa sta segnalando.

PARTE VI - CORREGGERE LA DIMOSTRAZIONE

13. Tornare sempre al Markdown

Il documento che dobbiamo correggere e:

```
DEMUESTRA/proyectos/MI_TEORIA/demostracion.md
```

Non dobbiamo trasformare il JSON nel nostro documento principale.

L'origine della dimostrazione rimane sempre il Markdown situato dentro DEMUESTRA.

Puoi utilizzare una IA per chiederle:

> Quale parte del mio file Markdown corrisponde a questo errore rilevato dall'analizzatore?

Poi studi il problema e decidi che cosa deve essere corretto.

---

14. Generare di nuovo il JSON

Dopo aver modificato il Markdown:

```
 Markdown corretto
        |
        v
nuovo JSON
        |
        v
analizzatore
```

Se compaiono nuovi problemi:

```
 analizzatore
     |
     v
localizzare problema
     |
     v
correggere .md
     |
     v
generare JSON
     |
     v
analizzare di nuovo
```

Questo ciclo puo ripetersi tutte le volte che e necessario.

Il principio e:

non modificare la dimostrazione semplicemente per soddisfare l'analizzatore; comprendere prima che cosa sta segnalando e correggere cio che deve davvero essere esplicitato o corretto.

PARTE VII - ENTRARE IN DEMUESTRA

15. Quando utilizzare Lean

Quando la catena matematica e sufficientemente definita e strutturata, passiamo alla verifica formale mediante Lean 4.

Ora dobbiamo creare un file:

```
.lean
```

corrispondente alla nostra dimostrazione.

---

16. Creare il file Lean

Puoi utilizzare una IA gratuita per realizzare una prima traduzione delle proposizioni matematiche del Markdown nel linguaggio Lean.

Devi chiederle di portare nel file Lean:

- definizioni;

- variabili;
- domini;
- ipotesi;
- proposizioni;
- lemmi;
- teoremi;
- dipendenze tra risultati.

L'obiettivo non e che la IA inventi una dimostrazione diversa.

L'obiettivo e **formalizzare la dimostrazione che esiste gia nel Markdown**.

Per la dimostrazione matematica finale delle deduzioni, DEMUESTRA ha bisogno solo del file `.lean`.

La relazione deve poter essere seguita:

```
Markdown
   |
   ├── definizione
   ├── proposizione 1
   ├── proposizione 2
   ├── lemma
   └── teorema
          |
          v
        Lean
          |
   ├── definizione
   ├── proposizione 1
   ├── proposizione 2
   ├── lemma
   └── teorema
```

In questo modo, se Lean trova un problema, possiamo tornare al punto corrispondente del Markdown.

PARTE VIII - ESEGUIRE DEMUESTRA

17. Eseguire il programma

Apri una console o un terminale.

Posizionati nella cartella radice corrispondente ed esegui:

```
python PckDemuestra\demuestra.py
```

DEMUESTRA cerchera i progetti depositati nella sua cartella di lavoro:

```
DEMUESTRA/proyectos/
```

e processera i progetti che trovera li.

Per questo non lavora direttamente sui tuoi progetti originali.

Lavora sulle copie che tu hai deciso di collocare dentro `DEMUESTRA/proyectos` .

---

18. Che cosa fa DEMUESTRA

DEMUESTRA utilizza Lean 4 per verificare i file Lean associati ai progetti che si trovano nella sua zona di lavoro.

Il programma indichera quali compilano correttamente e quali contengono errori.

Una compilazione corretta significa che Lean ha potuto verificare la costruzione formale fornita.

Una compilazione errata significa che c'e qualcosa da studiare.

PARTE IX - QUANDO LEAN TROVA UN ERRORE

19. Non correggere alla cieca

Se Lean produce un errore, non dobbiamo chiedere semplicemente a una IA:

> Fai in modo che compili.

Questo potrebbe nascondere proprio il problema che stiamo cercando di scoprire.

Bisogna localizzare:

```
 errore Lean
      |
      v
proposizione Lean interessata
      |
      v
proposizione corrispondente del Markdown
      |
      v
ragionamento matematico
```

Puoi utilizzare una IA per aiutarti a localizzare questa corrispondenza.

---

20. Due classi fondamentali di problemi

Un errore puo dipendere semplicemente dalla formalizzazione.

Per esempio:

- sintassi Lean errata;
- variabile non dichiarata;
- tipo errato;
- ipotesi che non e stata trasferita;
- definizione incompleta.

In questo caso correggiamo il file `.lean`.

Pero puo anche accadere che Lean riveli che un determinato risultato **non puo essere ottenuto dalle premesse che abbiamo formalizzato**.

Allora dobbiamo tornare al file `.md` e studiare quel passaggio.

Puo mancare una giustificazione.

Puo mancare un'ipotesi.

Oppure puo esistere realmente un errore matematico.

DEMUESTRA e utile precisamente perche obbliga a distinguere queste situazioni.

PARTE X - RIPETERE IL PROCESSO

21. Ciclo di verifica

La procedura completa puo essere rappresentata cosi:

```
 PROGETTO ORIGINALE
      |
      | copiare .md
      v
DEMUESTRA/proyectos
      |
      v
Markdown di lavoro
      |
      v
JSON
      |
      v
Analizzatore di Gianluca R. Pisano
      |
```

```
      v
Correzioni
      |
      v
Markdown corretto
      |
      v
Lean
      |
      v
DEMUESTRA
      |
      v
Lean 4
      |
      ├── ERRORE
      |      |
      |      └── studiare e correggere
      |                |
      |                └──────> ripetere
      |
      └── CORRETTO
             |
             v
       progetto valutato
```

PARTE XI - PROGETTI GIA VALUTATI

22. Evitare di riprocessare un progetto

Una volta terminato il processo, DEMUESTRA puo spostare il progetto, **se lo desideri**, dalla zona dei progetti in attesa alla zona destinata ai progetti gia valutati.

Questo evita che venga processato inutilmente di nuovo nella successiva esecuzione.

La logica e:

```
DEMUESTRA/
|
├── proyectos/
|   ├── progetto in attesa
|   └── progetto in valutazione
|
└── evaluados/
    └── progetto gia valutato
```

Cosi:

```
proyectos
```

contiene il lavoro in attesa o attivo.

E:

```
evaluados
```

contiene i progetti che hanno gia terminato il processo.

PARTE XII - RESTITUIRE IL RISULTATO AL PROGETTO ORIGINALE

23. Il Markdown definitivo

Durante tutto questo processo abbiamo lavorato sulla copia esistente dentro DEMUESTRA.

Il nostro progetto originale rimane separato.

Quando siamo soddisfatti del risultato, recuperiamo il Markdown corretto:

```
DEMUESTRA/proyectos/MI_TEORIA/demostracion.md
```

o il file corrispondente conservato da DEMUESTRA dopo la valutazione.

Lo copiamo di nuovo in:

```
MI_TEORIA/demostracion.md
```

Ora il nostro progetto originale incorpora tutte le correzioni emerse durante il processo di analisi e formalizzazione.

PARTE XIII - CHE COSA DIMOSTRA E CHE COSA NON DIMOSTRA LEAN

24. Che cosa significa che Lean compili

Questo punto deve essere compreso correttamente.

Se Lean accetta la formalizzazione, abbiamo una verifica meccanica del fatto che i risultati formalizzati derivano correttamente dalle definizioni, assiomi, ipotesi e teoremi utilizzati all'interno di quella formalizzazione.

Questo e molto piu forte di una semplice revisione di stile o di un'opinione di una IA.

Lean agisce come **verificatore formale**.

---

25. Che cosa non significa

Il fatto che un file Lean compili non dimostra automaticamente che tutte le premesse utilizzate descrivano correttamente la natura.

Per esempio, una teoria fisica puo avere una catena matematica internamente corretta e contenere un'ipotesi fisica che dovra poi essere verificata sperimentalmente.

Non significa nemmeno che una formalizzazione incompleta validi cio che non e stato formalizzato.

Per questo e essenziale che il file Lean rappresenti realmente tutte le proposizioni necessarie della dimostrazione.

PARTE XIV - FILOSOFIA DI DEMUESTRA

26. L'obiettivo

DEMUESTRA non pretende di sostituire il ricercatore.

Non pretende nemmeno di sostituire la revisione scientifica ne la verifica sperimentale quando questa sia necessaria.

Il suo scopo e fornire qualcosa di diverso:

una catena matematica esplicita, riproducibile e suscettibile di verifica formale.

La IA puo aiutarci a tradurre.

L'analizzatore puo aiutarci a scoprire carenze nell'esposizione.

Lean puo verificare le relazioni matematiche formalizzate.

Ma l'idea, le ipotesi e la responsabilita sulla dimostrazione continuano ad appartenere al ricercatore.

Il principio di lavoro puo essere riassunto cosi:

```
 IDEA UMANA
     v
DIMOSTRAZIONE UMANA
     v
MARKDOWN
     v
ANALISI STRUTTURALE
     v
CORREZIONE
     v
FORMALIZZAZIONE LEAN
     v
VERIFICA LEAN 4
     v
RISULTATO RIPRODUCIBILE
```

DEMUESTRA non chiede che una dimostrazione venga creduta.

Permette che possa essere esaminata passo dopo passo.