DEMUESTRA - Guide complet d'installation et d'utilisation

Le projet complet **DEMUESTRA** peut être téléchargé ou consulté sur GitHub, sous l'utilisateur **quark-cha**, dépôt **DEMUESTRA**: https://github.com/quark-cha/DEMUESTRA.

1. Ce qu'est DEMUESTRA

DEMUESTRA est un environnement permettant de soumettre une demonstration mathematique a un processus d'analyse structurelle et de verification formelle au moyen de **Lean 4**.

Le chercheur part de sa propre demonstration ecrite dans un fichier Markdown `.md`.

Pendant le processus, une **IA gratuite peut etre utilisee comme outil auxiliaire** pour des taches comme:

- ecrire correctement les formules en LaTeX;
- transformer la demonstration en JSON;
- transformer les propositions mathematiques en Lean;
- localiser les erreurs;
- relier une erreur au point correspondant de la demonstration;
- aider a corriger des problemes de syntaxe ou de formalisation.

L'IA **n'intervient pas dans le processus creatif de la demonstration**.

L'IA n'apporte pas l'idee mathematique et ne remplace pas le raisonnement du chercheur: elle agit seulement comme un outil pour ajuster formellement, organiser, traduire des formats et aider a exprimer avec precision une demonstration qui appartient deja a l'auteur.

La proposition, ses hypotheses, definitions, raisonnements et conclusions appartiennent a l'auteur.

DEMUESTRA ne pretend pas non plus decider, au moyen d'un simple score, si une theorie scientifique est vraie ou fausse.

Son objectif est beaucoup plus concret:

rendre explicite la chaine deductive et verifier formellement que les etapes mathematiques qui ont ete portees dans Lean sont correctement construites a partir de leurs premisses.

---

2. Auteur, licence et usage ethique

Ce projet fait partie du developpement de **Victor Estrada Diaz**.

Vous etes libre d'utiliser ces utilitaires dans votre propre projet a condition de respecter mon statut d'auteur et de ne pas les utiliser a des fins commerciales.

Vous etes egalement libre de les distribuer et de les diffuser a condition d'indiquer clairement la partie dont je suis l'auteur.

La licence applicable est:

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

L'utilisation de toute partie de ce developpement est interdite dans des environnements non ethiques, notamment:

- les guerres;
- les usages qui ne respectent pas les Droits de l'Homme;
- l'exploitation ou l'abus de personnes ou d'animaux;
- les usages destines a humilier ou a devaloriser d'autres personnes.

Tout usage de ce type enfreint ma licence.

---

3. Depot mis a jour

Le projet **DEMUESTRA** dispose d'un depot public depuis lequel vous pouvez telecharger la version mise a jour et acceder a tous ses fichiers.

Utilisateur GitHub:

```
quark-cha
```

Depot:

```
DEMUESTRA
```

PARTIE I - INSTALLATION

4. Ce que vous devez installer

Avant d'utiliser DEMUESTRA, vous devez preparer l'ordinateur.

Obligatoire

Installez:

1. **Python**
2. **Lean 4**
3. le depot **DEMUESTRA**

Recommande

Nous recommandons aussi:

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

Visual Studio Code n'est pas indispensable, mais il facilite beaucoup l'edition des fichiers `.md`, `.json` et `.lean`.

---

5. Telecharger DEMUESTRA

Telechargez ou clonez le depot **DEMUESTRA** dans votre dossier habituel de developpement.

Par exemple, vous pouvez avoir une structure generale semblable a:

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

Vos projets, **PUBLICAR** si vous l'utilisez, et **DEMUESTRA** se trouvent donc au meme niveau.

DEMUESTRA n'a pas besoin de modifier directement vos projets originaux.

PUBLICAR est optionnel. Il n'est pas necessaire d'installer PUBLICAR pour utiliser DEMUESTRA. PUBLICAR sert seulement si vous voulez aussi utiliser cet autre systeme de generation de PDF et de publication.

PARTIE II - PREPARER UNE DEMONSTRATION

6. Le document original

Supposons que vous avez un projet appele:

```
MI_TEORIA/
```

et qu'a l'interieur vous avez ecrit une demonstration:

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

Ce `.md` est votre document original.

Nous n'allons pas utiliser directement ce fichier pendant les tests.

Nous faisons d'abord une copie afin de travailler avec elle dans DEMUESTRA.

---

7. Copier le Markdown dans DEMUESTRA

Copiez le fichier `.md` que vous voulez demontrer dans la zone de projets de DEMUESTRA.

Conceptuellement:

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

A partir de ce moment, il faut distinguer clairement deux documents:

```
MI_TEORIA/demostracion.md
```

est le document qui reste dans votre projet original.

Tandis que:

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

est la **copie de travail**.

Regle importante

Pendant tout le processus de verification:

travaillez et effectuez les corrections sur le fichier `.md` situe dans DEMUESTRA.

Vous n'avez pas besoin de modifier continuellement le projet original.

Lorsque tout le processus aura ete termine de maniere satisfaisante, vous pourrez copier le fichier `.md` corrige depuis DEMUESTRA vers votre projet original.

PARTIE III - COMMENT ECRIRE LE MARKDOWN

8. Le fichier `.md`

Markdown est simplement un format de texte brut.

Vous pouvez l'ecrire avec n'importe quel editeur de texte, meme si Visual Studio Code facilite le travail.

Les formules mathematiques s'ecrivent en utilisant LaTeX.

Formules dans une ligne

Dans votre fichier Markdown reel, une formule dans une ligne est delimitee par le caractere $ au debut et un autre caractere $ a la fin.

Elle doit etre ecrite ainsi, comme texte:

$formula$

Par exemple:

L'energie est donnee par $E=mc^2$.

Formules independantes

Dans votre fichier Markdown reel, une formule independante dans son propre paragraphe est delimitee par deux caracteres $$ au debut et deux caracteres $$ a la fin.

Elle doit etre ecrite ainsi, comme texte:

$$formula$$

Par exemple:

$$E=mc^2$$

Vous pouvez demander a une IA de reviser le document et de placer correctement les formules et leurs delimitateurs.

PARTIE IV - LE PDF

9. Dois-je generer un PDF?

Non.

Le PDF n'est pas necessaire pour realiser la demonstration.

Le document fondamental de travail est le fichier `.md`.

Pour commencer a travailler, vous avez seulement besoin de ce fichier `.md`.

Avec le fichier `.md` et l'aide d'une IA, vous pouvez extraire les derivations arithmetiques et deductives qui seront ensuite portees dans le fichier Lean.

Vous pouvez generer un PDF si vous voulez disposer d'une version destinee a la lecture, a la distribution ou a la publication.

Vous pouvez le faire avec une IA ou avec n'importe quel outil specifique de conversion.

Dans mon cas, j'utilise un autre projet appele **PUBLICAR**.

PUBLICAR convertit mes projets en PDF et automatise leur publication dans Zenodo et sur mon site web.

Mais PUBLICAR utilise ma propre structure de projets.

Par consequent:

PUBLICAR est optionnel et n'est pas necessaire pour utiliser DEMUESTRA.

Vous ne devez pas installer PUBLICAR si vous voulez uniquement verifier une demonstration avec DEMUESTRA.

PARTIE V - ANALYSE STRUCTURELLE

10. Convertir le Markdown en JSON

Une demonstration peut sembler evidente pour la personne qui l'a developpee et contenir pourtant des concepts qui n'ont pas ete definis explicitement ou des relations qui ont ete supposees.

C'est pourquoi nous realisons d'abord une analyse structurelle.

Vous pouvez demander a une IA gratuite de transformer votre demonstration Markdown dans la structure JSON requise par l'analyseur de **Gianluca R. Pisano**.

Le processus est:

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

```
      v
demostracion.json
```

Ici, l'IA effectue une **traduction de format et de structure**.

Elle ne doit pas inventer de nouvelles premisses ni modifier votre demonstration pour qu'elle passe l'analyse.

---

11. Passer le JSON dans l'analyseur

Introduisez le JSON obtenu dans l'analyseur de Gianluca R. Pisano.

L'analyseur peut detecter des problemes tels que:

- des symboles qui apparaissent sans avoir ete definis;
- des concepts insuffisamment specifies;
- des dependances qui ne sont pas claires;
- des propositions qui utilisent des resultats anterieurs sans les identifier;
- des sauts dans la chaine deductive;
- des hypotheses qui doivent etre explicitees;
- des problemes de structure formelle.

---

12. Une erreur de l'analyseur ne signifie pas necessairement que vous avez tort

Ce point est fondamental.

Si l'analyseur detecte un probleme, **cela ne signifie pas automatiquement que l'affirmation mathematique est fausse**.

Cela peut signifier, par exemple, que vous savez parfaitement pourquoi un resultat decoule du precedent, mais que vous ne l'avez pas ecrit explicitement.

Dans ce cas, le probleme se trouve dans l'exposition formelle de la demonstration.

Mais il ne faut pas non plus ignorer l'erreur.

Il faut comprendre exactement ce qu'elle signale.

PARTIE VI - CORRIGER LA DEMONSTRATION

13. Toujours revenir au Markdown

Le document que nous devons corriger est:

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

Nous ne devons pas faire du JSON notre document principal.

L'origine de la demonstration reste toujours le Markdown situe dans DEMUESTRA.

Vous pouvez utiliser une IA pour lui demander:

> Quelle partie de mon fichier Markdown correspond a cette erreur detectee par l'analyseur?

Ensuite, vous etudiez le probleme et vous decidez ce qui doit etre corrige.

---

14. Generer de nouveau le JSON

Apres avoir modifie le Markdown:

```
 Markdown corrige
        |
        v
nouveau JSON
        |
        v
analyseur
```

Si de nouveaux problemes apparaissent:

```
 analyseur
     |
     v
localiser le probleme
     |
     v
corriger .md
     |
     v
generer JSON
     |
```

```
    v
analyser a nouveau
```

Ce cycle peut etre repete autant de fois que necessaire.

Le principe est:

ne pas modifier la demonstration simplement pour satisfaire l'analyseur; comprendre d'abord ce qu'il signale et corriger ce qui doit reellement etre explicite ou corrige.

PARTIE VII - ENTRER DANS DEMUESTRA

15. Quand utiliser Lean

Lorsque la chaine mathematique est suffisamment definie et structuree, nous passons a la verification formelle au moyen de Lean 4.

Nous devons maintenant creer un fichier:

```
.lean
```

correspondant a notre demonstration.

---

16. Creer le fichier Lean

Vous pouvez utiliser une IA gratuite pour realiser une premiere traduction des propositions mathematiques du Markdown vers le langage Lean.

Vous devez lui demander de porter dans le fichier Lean:

- definitions;
- variables;
- domaines;
- hypotheses;
- propositions;
- lemmes;
- theoremes;
- dependances entre resultats.

L'objectif n'est pas que l'IA invente une demonstration differente.

L'objectif est **de formaliser la demonstration qui existe deja dans le Markdown**.

Pour la demonstration mathematique finale des deductions, DEMUESTRA a seulement besoin du fichier `.lean`.

La relation doit pouvoir etre suivie:

```
Markdown
  |
  ├── definition
  ├── proposition 1
  ├── proposition 2
  ├── lemme
```

```
   └── theoreme
          |
          v
        Lean
          |
   ├── definition
   ├── proposition 1
   ├── proposition 2
   ├── lemme
   └── theoreme
```

De cette facon, si Lean trouve un probleme, nous pouvons revenir au point correspondant du Markdown.

PARTIE VIII - EXECUTER DEMUESTRA

17. Executer le programme

Ouvrez une console ou un terminal.

Placez-vous dans le dossier racine correspondant et executez:

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

DEMUESTRA cherchera les projets places dans son dossier de travail:

```
DEMUESTRA/proyectos/
```

et traitera les projets qu'il y trouvera.

C'est pourquoi il ne travaille pas directement sur vos projets originaux.

Il travaille sur les copies que vous avez decide de placer dans `DEMUESTRA/proyectos` .

---

18. Ce que fait DEMUESTRA

DEMUESTRA utilise Lean 4 pour verifier les fichiers Lean associes aux projets qui se trouvent dans sa zone de travail.

Le programme indiquera lesquels compilent correctement et lesquels contiennent des erreurs.

Une compilation correcte signifie que Lean a pu verifier la construction formelle fournie.

Une compilation incorrecte signifie qu'il y a quelque chose a etudier.

PARTIE IX - QUAND LEAN TROUVE UNE ERREUR

19. Ne pas corriger a l'aveugle

Si Lean produit une erreur, nous ne devons pas simplement demander a une IA:

> Fais en sorte que cela compile.

Cela pourrait cacher precisement le probleme que nous essayons de decouvrir.

Il faut localiser:

```
erreur Lean
    |
    v
proposition Lean affectee
    |
    v
proposition correspondante du Markdown
    |
    v
raisonnement mathematique
```

Vous pouvez utiliser une IA pour vous aider a localiser cette correspondance.

---

20. Deux types fondamentaux de problemes

Une erreur peut etre due simplement a la formalisation.

Par exemple:

- syntaxe Lean incorrecte;
- variable non declaree;
- type incorrect;
- hypothese qui n'a pas ete transferee;
- definition incomplete.

Dans ce cas, nous corrigeons le fichier `.lean`.

Mais il peut aussi arriver que Lean revele qu'un certain resultat **ne peut pas etre obtenu a partir des premisses que nous avons formalisees**.

Alors nous devons revenir au fichier `.md` et etudier cette etape.

Une justification peut manquer.

Une hypothese peut manquer.

Ou il peut reellement exister une erreur mathematique.

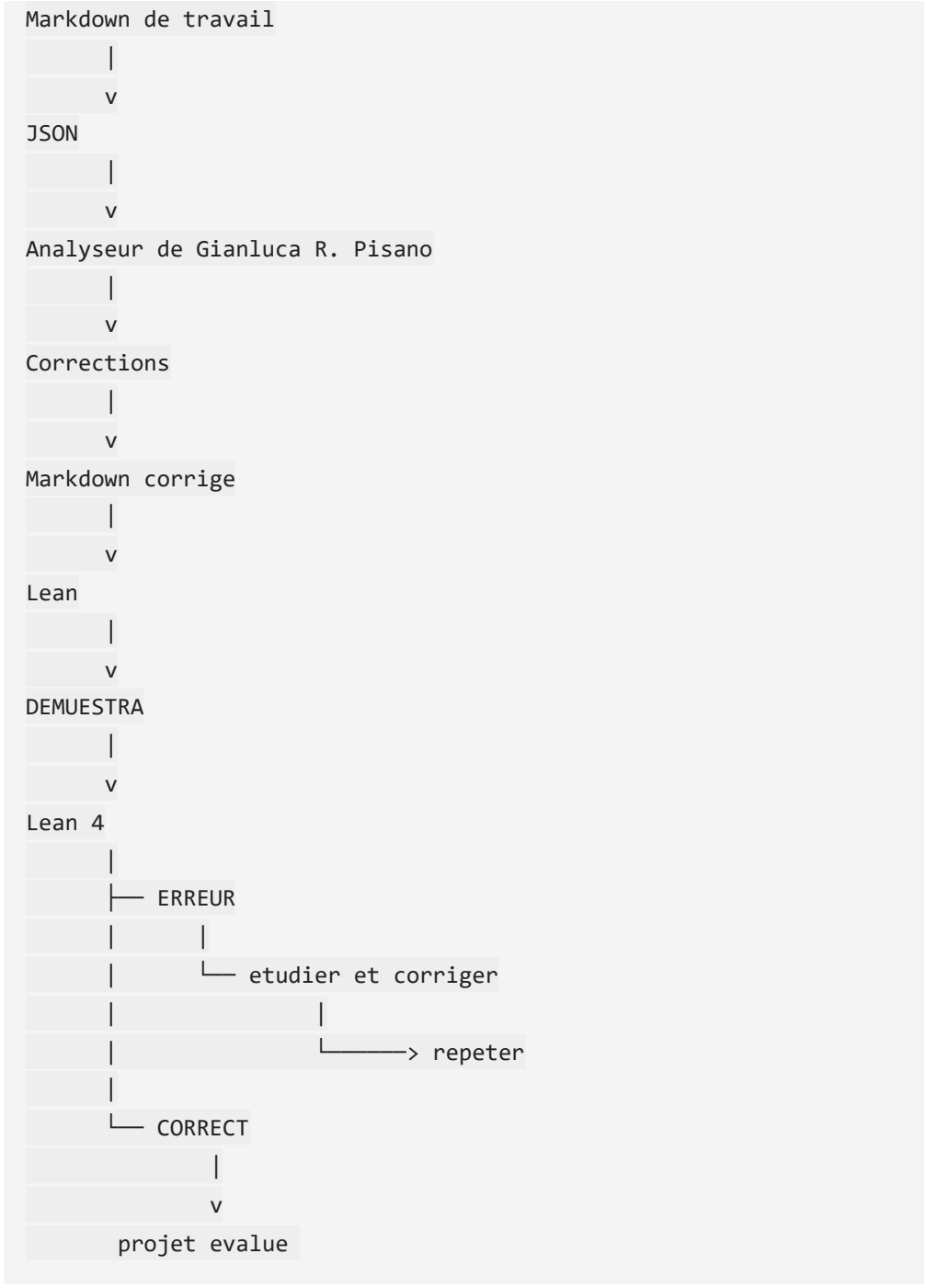
DEMUESTRA est utile precisement parce qu'il oblige a distinguer ces situations.

PARTIE X - REPETER LE PROCESSUS

21. Cycle de verification

La procedure complete peut etre representee ainsi:

```
PROJET ORIGINAL
      |
      | copier .md
      v
DEMUESTRA/proyectos
      |
      v
```


Markdown de travail
JSON
Analyseur de Gianluca R. Pisano
Corrections
Markdown corrige
Lean
DEMUESTRA
Lean 4
ERREUR
etudier et corriger
repeter
CORRECT
projet evalue

PARTIE XI - PROJETS DEJA EVALUES

22. Eviter de retraiter un projet

Une fois le processus termine, DEMUESTRA peut deplacer le projet, **si vous le souhaitez**, depuis la zone des projets en attente vers la zone destinee aux projets deja evalues.

Cela evite qu'il soit traite inutilement de nouveau lors de l'execution suivante.

La logique est:

```
DEMUESTRA/
|
├── proyectos/
|   ├── projet en attente
|   └── projet en evaluation
|
└── evaluados/
    └── projet deja evalue
```

Ainsi:

```
proyectos
```

contient le travail en attente ou actif.

Et:

```
evaluados
```

contient les projets qui ont deja termine le processus.

PARTIE XII - RENVOYER LE RESULTAT AU PROJET ORIGINAL

23. Le Markdown definitif

Pendant tout ce processus, nous avons travaille sur la copie existant dans DEMUESTRA.

Notre projet original reste separe.

Lorsque nous sommes satisfaits du resultat, nous recuperons le Markdown corrige:

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

ou le fichier correspondant conserve par DEMUESTRA apres l'evaluation.

Nous le recopions vers:

```
MI_TEORIA/demostracion.md
```

Notre projet original incorpore maintenant toutes les corrections apparues pendant le processus d'analyse et de formalisation.

PARTIE XIII - CE QUE LEAN DEMONTRE ET CE QU'IL NE DEMONTRE PAS

24. Ce que signifie le fait que Lean compile

Ce point doit etre compris correctement.

Si Lean accepte la formalisation, nous avons une verification mecanique que les resultats formalises se deduisent correctement des definitions, axiomes, hypotheses et theoremes utilises dans cette formalisation.

C'est beaucoup plus fort qu'une simple revision de style ou qu'une opinion d'une IA.

Lean agit comme **verificateur formel**.

---

25. Ce que cela ne signifie pas

Le fait qu'un fichier Lean compile ne demontre pas automatiquement que toutes les premisses utilisees decrivent correctement la nature.

Par exemple, une theorie physique peut avoir une chaine mathematique interne correcte et contenir une hypothese physique qui devra ensuite etre testee experimentalement.

Cela ne signifie pas non plus qu'une formalisation incomplete valide ce qui n'a pas ete formalise.

C'est pourquoi il est essentiel que le fichier Lean represente reellement toutes les propositions necessaires de la demonstration.

PARTIE XIV - PHILOSOPHIE DE DEMUESTRA

26. L'objectif

DEMUESTRA ne pretend pas remplacer le chercheur.

Il ne pretend pas non plus remplacer l'evaluation scientifique ni la verification experimentale lorsque celle-ci est necessaire.

Son but est de fournir quelque chose de different:

une chaine mathematique explicite, reproductible et susceptible de verification formelle.

L'IA peut nous aider a traduire.

L'analyseur peut nous aider a decouvrir des deficiences dans l'exposition.

Lean peut verifier les relations mathematiques formalisees.

Mais l'idee, les hypotheses et la responsabilite de la demonstration continuent d'appartenir au chercheur.

Le principe de travail peut se resumer ainsi:

```
 IDEE HUMAINE
      v
DEMONSTRATION HUMAINE
      v
MARKDOWN
      v
ANALYSE STRUCTURELLE
      v
CORRECTION
      v
FORMALISATION LEAN
      v
VERIFICATION LEAN 4
      v
RESULTAT REPRODUCTIBLE
```

DEMUESTRA ne demande pas qu'une demonstration soit crue.

Il permet qu'elle soit examinee pas a pas.