# DEMUESTRA - Guia completa de instalacion y uso

El proyecto completo **DEMUESTRA** se puede descargar o consultar en GitHub, en el usuario **quark-cha**, repositorio **DEMUESTRA**: https://github.com/quark-cha/DEMUESTRA.

## 1. Que es DEMUESTRA

**DEMUESTRA** es un entorno para someter una demostracion matematica a un proceso de analisis estructural y comprobacion formal mediante **Lean 4**.

El investigador parte de su propia demostracion escrita en un fichero Markdown `.md`.

Durante el proceso puede utilizarse una **IA gratuita como herramienta auxiliar** para tareas como:

- escribir correctamente las formulas en LaTeX;
- transformar la demostracion a JSON;
- transformar las proposiciones matematicas a Lean;
- localizar errores;
- relacionar un error con el punto correspondiente de la demostracion;
- ayudar a corregir problemas de sintaxis o formalizacion.

La IA **no interviene en el proceso creativo de la demostracion**.

La IA no aporta la idea matematica ni sustituye el razonamiento del investigador: solo actua como herramienta para ajustar formalmente, ordenar, traducir formatos y ayudar a expresar con precision una demostracion que ya pertenece al autor.

La propuesta, sus hipotesis, definiciones, razonamientos y conclusiones pertenecen al autor.

DEMUESTRA tampoco pretende decidir mediante una simple puntuacion si una teoria cientifica es verdadera o falsa.

Su objetivo es mucho mas concreto:

**hacer explicita la cadena deductiva y comprobar formalmente que los pasos matematicos que han sido llevados a Lean estan correctamente construidos a partir de sus premisas.**

---

# 2. Autoria, licencia y uso etico

Este proyecto forma parte del desarrollo de **Victor Estrada Diaz**.

Eres libre de usar estas utilidades en tu propio proyecto siempre que respetes mi autoria y no las uses con fines comerciales.

Tambien eres libre de distribuirlas y difundirlas siempre que indiques claramente la parte de la que soy autor.

La licencia aplicable es:

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

El uso de cualquier parte de este desarrollo queda prohibido en entornos no eticos, incluyendo:

- guerras;
- usos que no cumplan los Derechos Humanos;
- explotacion o abuso de personas o animales;
- usos destinados a humillar o infravalorar a otras personas.

Cualquier uso de este tipo incumple mi licencia.

---

## 3. Repositorio actualizado

El proyecto **DEMUESTRA** tiene un repositorio publico desde el que puedes descargar la version actualizada y acceder a todos sus archivos.

Usuario de GitHub:

```
quark-cha
```

Repositorio:

```
DEMUESTRA
```

---

# PARTE I - INSTALACION

## 4. Que necesitas instalar

Antes de utilizar DEMUESTRA necesitas preparar el ordenador.

### Obligatorio

Instala:

1. **Python**
2. **Lean 4**
3. el repositorio **DEMUESTRA**

### Recomendado

Tambien recomendamos:

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

Visual Studio Code no es imprescindible, pero facilita mucho la edicion de los ficheros `.md`, `.json` y `.lean`.

---

# 5. Descargar DEMUESTRA

Descarga o clona el repositorio **DEMUESTRA** en tu carpeta habitual de desarrollo.

Por ejemplo, puedes tener una estructura general semejante a:

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

Tus proyectos, **PUBLICAR** si lo usas, y **DEMUESTRA** estan, por tanto, al mismo nivel.

DEMUESTRA no necesita modificar directamente tus proyectos originales.

**PUBLICAR es opcional.** No es necesario instalar PUBLICAR para utilizar DEMUESTRA. PUBLICAR solo sirve si ademas quieres usar ese otro sistema de generacion de PDF y publicacion.

---

# PARTE II - PREPARAR UNA DEMOSTRACION

## 6. El documento original

Supongamos que tienes un proyecto llamado:

```
MI_TEORIA/
```

y dentro de el has escrito una demostracion:

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

Este `.md` es tu documento original.

No vamos a utilizar directamente ese fichero durante las pruebas.

Primero hacemos una copia para trabajar con ella dentro de DEMUESTRA.

---

# 7. Copiar el Markdown a DEMUESTRA

Copia el `.md` que quieres demostrar a la zona de proyectos de DEMUESTRA.

Conceptualmente:

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

A partir de este momento hay que distinguir claramente dos documentos:

```
MI_TEORIA/demostracion.md
```

es el documento que permanece en tu proyecto original.

Mientras que:

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

es la **copia de trabajo**.

## Regla importante

Durante todo el proceso de comprobacion:

**trabaja y realiza las correcciones sobre el `.md` situado dentro de DEMUESTRA.**

No necesitas modificar continuamente el proyecto original.

Cuando hayas terminado satisfactoriamente todo el proceso, podras copiar el `.md` corregido desde DEMUESTRA a tu proyecto original.

---

# PARTE III - COMO ESCRIBIR EL MARKDOWN

## 8. El fichero `.md`

Markdown es simplemente un formato de texto plano.

Puedes escribirlo con cualquier editor de texto, aunque Visual Studio Code facilita el trabajo.

Las formulas matematicas se escriben utilizando LaTeX.

## Formulas dentro de una linea

En tu fichero Markdown real, una formula dentro de una linea se delimita con el caracter $ al principio y otro caracter $ al final.

Debe quedar escrita asi, como texto:

$formula$

Por ejemplo:

La energia viene dada por $E=mc^2$.

## Formulas independientes

En tu fichero Markdown real, una formula independiente en su propio parrafo se delimita con dos caracteres $$ al principio y dos caracteres $$ al final.

Debe quedar escrita asi, como texto:

$$formula$$

Por ejemplo:

$$E=mc^2$$

Puedes pedir a una IA que revise el documento y coloque correctamente las formulas y sus delimitadores.

---

# PARTE IV - EL PDF

## 9. Necesito generar un PDF

**No.**

El PDF no es necesario para realizar la demostracion.

El documento fundamental de trabajo es el `.md`.

Para empezar a trabajar solo necesitas ese fichero `.md`.

Con el `.md` y la ayuda de una IA puedes extraer las derivaciones aritmeticas y deductivas que despues se llevaran al fichero Lean.

Puedes generar un PDF si quieres disponer de una version destinada a lectura, distribucion o publicacion.

Puedes hacerlo mediante una IA o mediante cualquier herramienta especifica de conversion.

En mi caso utilizo otro proyecto llamado **PUBLICAR**.

PUBLICAR convierte mis proyectos a PDF y automatiza su publicacion en Zenodo y en mi web.

Pero PUBLICAR utiliza mi propia estructura de proyectos.

Por ello:

**PUBLICAR es opcional y no es necesario para utilizar DEMUESTRA.**

No debes instalar PUBLICAR si unicamente quieres comprobar una demostracion mediante DEMUESTRA.

---

# PARTE V - ANALISIS ESTRUCTURAL

## 10. Convertir el Markdown a JSON

Una demostracion puede parecer evidente para quien la ha desarrollado y, sin embargo, contener conceptos que no han sido definidos explicitamente o relaciones que se han dado por supuestas.

Por eso realizamos primero un analisis estructural.

Puedes pedir a una IA gratuita que transforme tu demostracion Markdown a la estructura JSON requerida por el analizador de **Gianluca R. Pisano**.

El proceso es:

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

La IA esta realizando aqui una **traduccion de formato y estructura**.

No debe inventar nuevas premisas ni modificar tu demostracion para conseguir que pase el analisis.

---

## 11. Pasar el JSON por el analizador

Introduce el JSON obtenido en el analizador de Gianluca R. Pisano.

El analizador puede detectar problemas como:

- simbolos que aparecen sin haber sido definidos;
- conceptos insuficientemente especificados;
- dependencias que no estan claras;
- proposiciones que utilizan resultados anteriores sin identificarlos;
- saltos en la cadena deductiva;
- hipotesis que necesitan hacerse explicitas;
- problemas de estructura formal.

---

## 12. Un error del analizador no significa necesariamente que estes equivocado

Este punto es fundamental.

Que el analizador detecte un problema **no significa automaticamente que la afirmacion matematica sea falsa**.

Puede significar, por ejemplo, que sabes perfectamente por que un resultado se sigue del anterior pero no lo has escrito explicitamente.

En ese caso el problema esta en la exposicion formal de la demostracion.

Pero tampoco debemos ignorar el error.

Hay que averiguar exactamente que esta senalando.

---

# PARTE VI - CORREGIR LA DEMOSTRACION

## 13. Volver siempre al Markdown

El documento que debemos corregir es:

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

No debemos convertir el JSON en nuestro documento principal.

El origen de la demostracion sigue siendo siempre el Markdown situado dentro de DEMUESTRA.

Puedes utilizar una IA para preguntarle:

> Que parte de mi fichero Markdown corresponde a este error detectado por el analizador?

Despues estudias el problema y decides que debe corregirse.

# 14. Generar nuevamente el JSON

Despues de modificar el Markdown:

```
 Markdown corregido
        |
        v
nuevo JSON
        |
        v
analizador
```

Si aparecen nuevos problemas:

```
 analizador
     |
     v
localizar problema
     |
     v
corregir .md
     |
     v
generar JSON
     |
     v
analizar otra vez
```

Este ciclo puede repetirse tantas veces como sea necesario.

El principio es:

**no modificar la demostracion simplemente para satisfacer al analizador; comprender primero que esta senalando y corregir aquello que realmente necesite ser explicitado o corregido.**

---

# PARTE VII - ENTRAR EN DEMUESTRA

## 15. Cuando utilizar Lean

Cuando la cadena matematica esta suficientemente definida y estructurada, pasamos a la comprobacion formal mediante Lean 4.

Ahora necesitamos crear un fichero:

```
.lean
```

correspondiente a nuestra demostracion.

---

## 16. Crear el fichero Lean

Puedes utilizar una IA gratuita para realizar una primera traduccion de las proposiciones matematicas del Markdown al lenguaje Lean.

Debes pedirle que lleve al fichero Lean:

- definiciones;

- variables;
- dominios;
- hipotesis;
- proposiciones;
- lemas;
- teoremas;
- dependencias entre resultados.

El objetivo no es que la IA invente una demostracion diferente.

El objetivo es **formalizar la demostracion que ya existe en el Markdown**.

Para la demostracion matematica final de las deducciones, DEMUESTRA solo necesita el fichero `.lean`.

La relacion debe poder seguirse:

```
Markdown
   |
   ├── definicion
   ├── proposicion 1
   ├── proposicion 2
   ├── lema
   └── teorema
          |
          v
        Lean
          |
   ├── definicion
   ├── proposicion 1
   ├── proposicion 2
   ├── lema
   └── teorema
```

De esta forma, si Lean encuentra un problema, podemos volver al punto correspondiente del Markdown.

---

# PARTE VIII - EJECUTAR DEMUESTRA

## 17. Ejecutar el programa

Abre una consola o terminal.

Situate en la carpeta raiz correspondiente y ejecuta:

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

DEMUESTRA buscara los proyectos depositados en su carpeta de trabajo:

```
DEMUESTRA/proyectos/
```

y procesara los proyectos que encuentre alli.

Por eso no trabaja directamente sobre tus proyectos originales.

**Trabaja sobre las copias que tu has decidido colocar dentro de `DEMUESTRA/proyectos` .**

---

## 18. Que hace DEMUESTRA

DEMUESTRA utiliza Lean 4 para comprobar los ficheros Lean asociados a los proyectos que se encuentran en su zona de trabajo.

El programa informara de cuales compilan correctamente y cuales contienen errores.

Una compilacion correcta significa que Lean ha podido verificar la construccion formal proporcionada.

Una compilacion incorrecta significa que hay algo que estudiar.

---

# PARTE IX - CUANDO LEAN ENCUENTRA UN ERROR

## 19. No corregir a ciegas

Si Lean produce un error, no debemos pedir simplemente a una IA:

> Haz que compile.

Eso podria ocultar precisamente el problema que estamos intentando descubrir.

Hay que localizar:

```
error Lean
    |
    v
```

```
proposicion Lean afectada
     |
     v
proposicion correspondiente del Markdown
     |
     v
razonamiento matematico
```

Puedes utilizar una IA para ayudarte a localizar esa correspondencia.

---

# 20. Dos clases fundamentales de problemas

Un error puede deberse simplemente a la formalizacion.

Por ejemplo:

- sintaxis Lean incorrecta;
- variable no declarada;
- tipo incorrecto;
- hipotesis que no se ha trasladado;
- definicion incompleta.

En ese caso corregimos el `.lean`.

Pero tambien puede ocurrir que Lean revele que determinado resultado **no puede obtenerse de las premisas que hemos formalizado**.

Entonces debemos volver al `.md` y estudiar ese paso.

Puede faltar una justificacion.

Puede faltar una hipotesis.

O puede existir realmente un error matematico.

DEMUESTRA es util precisamente porque obliga a distinguir estas situaciones.

---

# PARTE X - REPETIR EL PROCESO

## 21. Ciclo de comprobacion

El procedimiento completo puede representarse asi:

```
 PROYECTO ORIGINAL
       |
       | copiar .md
       v
DEMUESTRA/proyectos
       |
       v
Markdown de trabajo
       |
       v
JSON
       |
       v
Analizador de Gianluca R. Pisano
       |
       v
Correcciones
```

```
      |
      v
Markdown corregido
      |
      v
Lean
      |
      v
DEMUESTRA
      |
      v
Lean 4
      |
      ├── ERROR
      |      |
      |      └── estudiar y corregir
      |                |
      |                └──────> repetir
      |
      └── CORRECTO
             |
             v
       proyecto evaluado
```

---

# PARTE XI - PROYECTOS YA EVALUADOS

## 22. Evitar reprocesar un proyecto

Una vez terminado el proceso, DEMUESTRA puede mover el proyecto, **si asi lo deseas**, desde la zona de proyectos pendientes a la zona destinada a los proyectos ya evaluados.

Esto evita que vuelva a procesarse innecesariamente en la siguiente ejecucion.

La logica es:

```
DEMUESTRA/
|
├── proyectos/
|   ├── proyecto pendiente
|   └── proyecto en evaluacion
|
└── evaluados/
    └── proyecto ya evaluado
```

Asi:

```
proyectos
```

contiene el trabajo pendiente o activo.

Y:

```
evaluados
```

contiene los proyectos que ya han terminado el proceso.

---

# PARTE XII - DEVOLVER EL RESULTADO AL PROYECTO ORIGINAL

## 23. El Markdown definitivo

Durante todo este proceso hemos trabajado sobre la copia existente dentro de DEMUESTRA.

Nuestro proyecto original continua separado.

Cuando estamos satisfechos con el resultado, recuperamos el Markdown corregido:

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

o el correspondiente fichero conservado por DEMUESTRA despues de la evaluacion.

Lo copiamos nuevamente a:

```
MI_TEORIA/demostracion.md
```

Ahora nuestro proyecto original incorpora todas las correcciones que han surgido durante el proceso de analisis y formalizacion.

---

# PARTE XIII - QUE DEMUESTRA Y QUE NO DEMUESTRA LEAN

## 24. Que significa que Lean compile

Este punto debe entenderse correctamente.

Si Lean acepta la formalizacion, tenemos una comprobacion mecanica de que los resultados formalizados se derivan correctamente de las definiciones, axiomas, hipotesis y teoremas utilizados dentro de esa formalizacion.

Esto es mucho mas fuerte que una simple revision de estilo o una opinion de una IA.

Lean actua como **verificador formal**.

---

## 25. Que no significa

Que un fichero Lean compile no demuestra automaticamente que todas las premisas utilizadas describan correctamente la naturaleza.

Por ejemplo, una teoria fisica puede tener una cadena matematica internamente correcta y contener una hipotesis fisica que posteriormente deba contrastarse experimentalmente.

Tampoco significa que una formalizacion incompleta valide aquello que no se ha formalizado.

Por eso es esencial que el fichero Lean represente realmente todas las proposiciones necesarias de la demostracion.

---

# PARTE XIV - FILOSOFIA DE DEMUESTRA

## 26. El objetivo

DEMUESTRA no pretende sustituir al investigador.

Tampoco pretende sustituir la revision cientifica ni la comprobacion experimental cuando esta sea necesaria.

Su proposito es proporcionar algo diferente:

**una cadena matematica explicita, reproducible y susceptible de comprobacion formal.**

La IA puede ayudarnos a traducir.

El analizador puede ayudarnos a descubrir deficiencias en la exposicion.

Lean puede comprobar las relaciones matematicas formalizadas.

Pero la idea, las hipotesis y la responsabilidad sobre la demostracion continuan perteneciendo al investigador.

El principio de trabajo puede resumirse asi:

```
 IDEA HUMANA
      v
DEMOSTRACION HUMANA
      v
MARKDOWN
      v
ANALISIS ESTRUCTURAL
      v
CORRECCION
      v
FORMALIZACION LEAN
      v
VERIFICACION LEAN 4
      v
RESULTADO REPRODUCIBLE
```

**DEMUESTRA no pide que una demostracion sea creida.**

**Permite que pueda ser examinada paso a paso.**