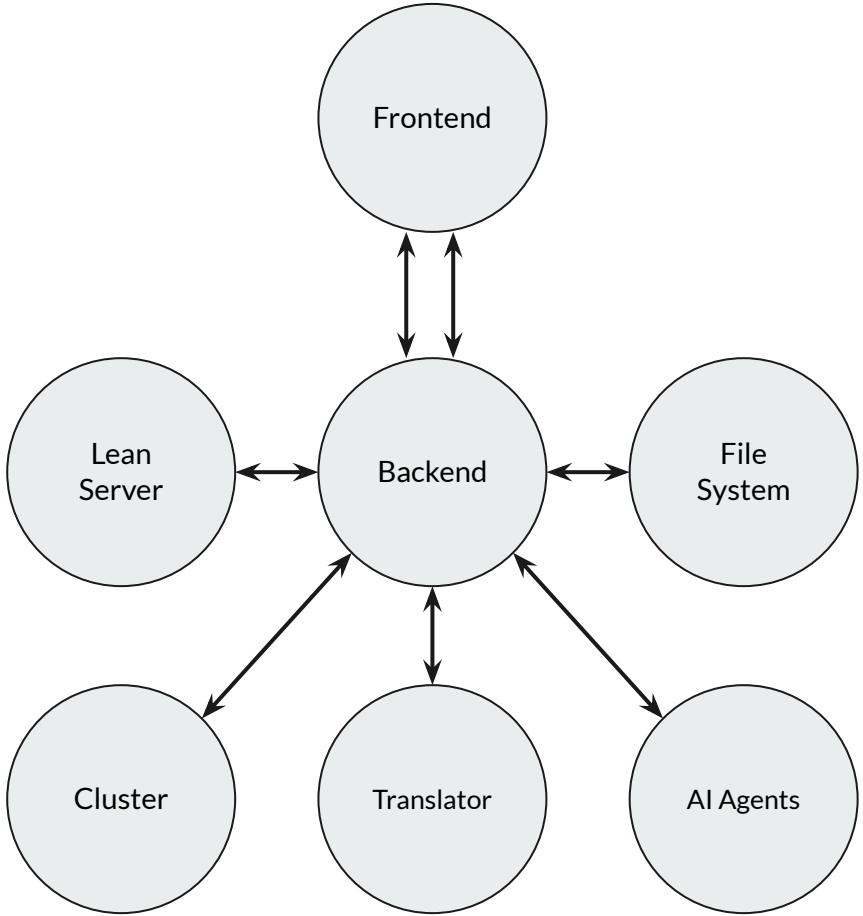
CoProof

Actualizaciones Febrero 12, 2026

David Alejandro López Torres - 22310432
Daniel Tejeda Saavedra - 22310431
Emiliano Flores Márquez - 22110044

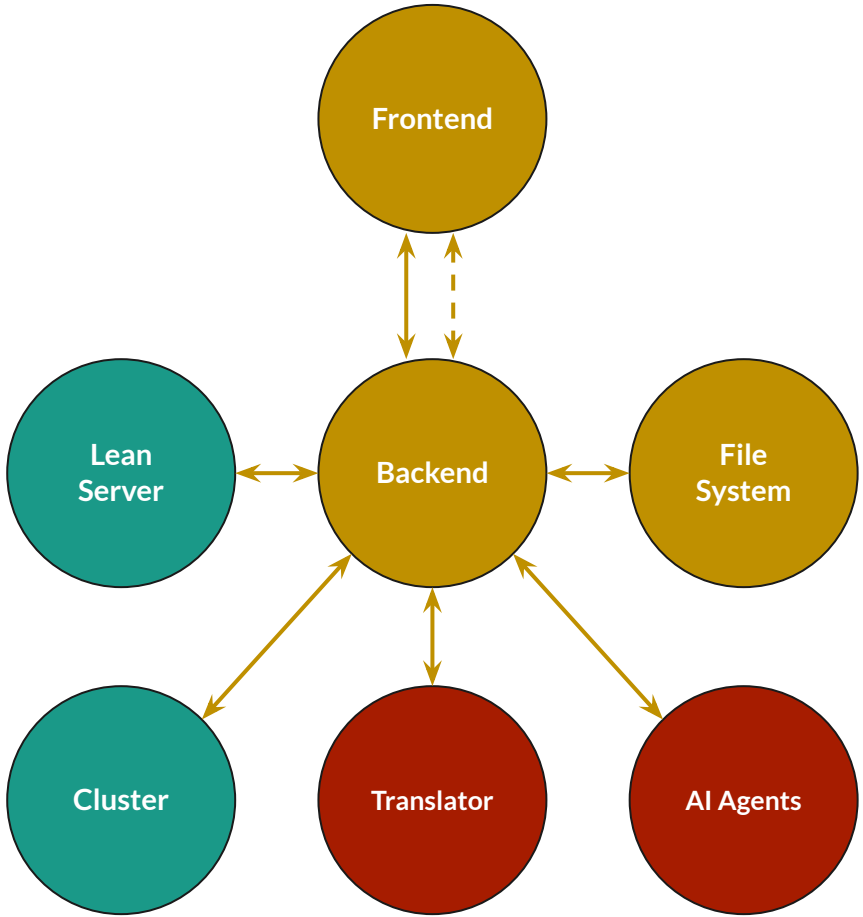
Arquitectura

- Backend (Python)
- Frontend (Angular)
- LeanServer (Lean + API)
- Cluster (OHPC en RPI4 + API)
- Filesystem + DB (Git + SQL)
- Translator NL2FL (Python)
- AI Agents (Python)



Arquitectura

- Backend (Python)
- Frontend (Angular)
- LeanServer (Lean + API)
- Cluster (OHPC en RPI4 + API)
- Filesystem + DB (Git + SQL)
- Translator NL2FL (Python)
- AI Agents (Python)



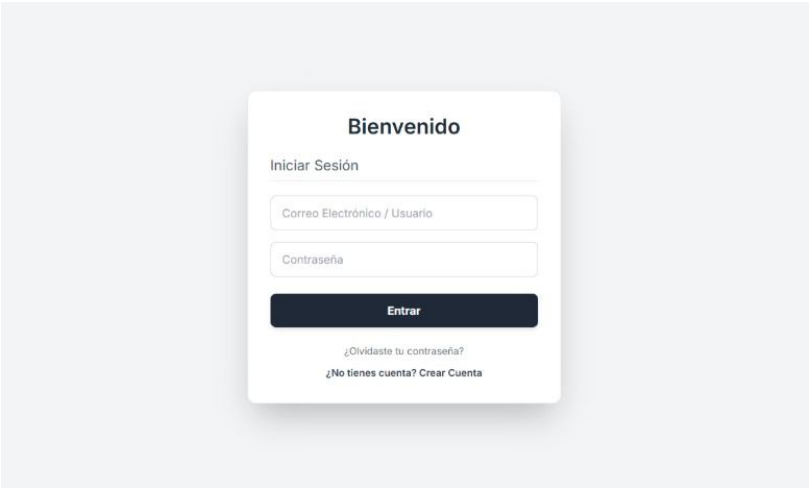
Frontend (Angular)

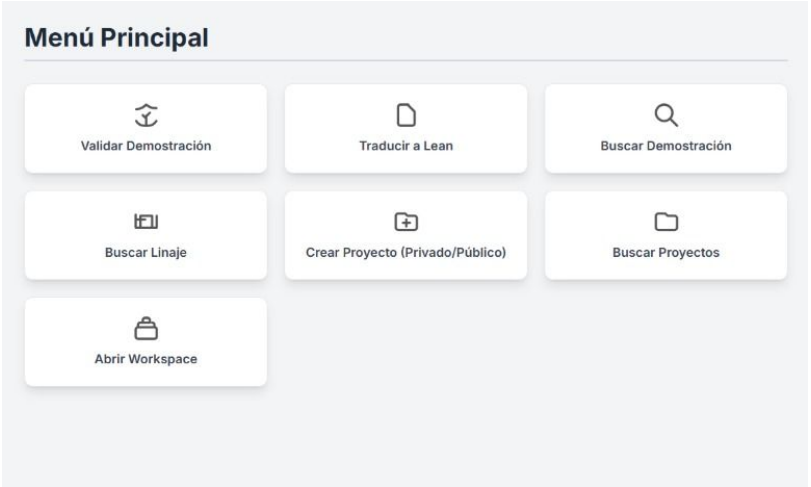
Hecho:

- Vistas preliminares funcionales (HTML)
- Contenedor de desarrollo

Por hacer:

- Crear componentes en Angular para recrear las vistas preliminares
- Gestionar solicitudes al backend (REST API & Websocket)
- Contenerización


Bienvenido
Iniciar Sesión
Correo Electrónico / Usuario
Contraseña
Entrar
¿Olvidaste tu contraseña?
¿No tienes cuenta? Crear Cuenta


Menú Principal
Validar Demostración
Traducir a Lean
Buscar Demostración
Buscar Linaje
Crear Proyecto (Privado/Público)
Buscar Proyectos
Abrir Workspace

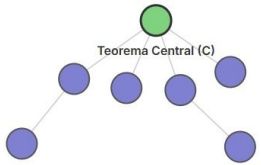

Validación de Demostración
Subir Archivo (.lean, .txt, .pdf)
Procesar Validación
Resultado: VALIDADO Exitosamente
Vincular a Proyecto
Demostración Original
Versión en Lean
Lenguaje Natural
1. Demostración Subida
// Archivo del usuario (código, texto o PDF).
/* Ejemplo de entrada de usuario */
Proposición: Demostrar que para todo número
Método: Inducción.
2. Versión Estructurada en Lean
// Código Lean generado y verificado.
import tactic
theorem paridad_n_cuadrado_mas_n (n : nat)
begin
-- Demostración por inducción, simplifica
by_cases h_par : even n,
{ -- Caso n es par: n=2k. n^2+n = 4k^2+2k
exact even_add.mpr (even_pow.mpr h_par,
{ -- Caso n es impar: n^2 es impar. impar
exact even_add.mpr ((odd_iff_not_even.m
end
3. Explicación en Lenguaje Natural
// Explicación clara de la demostración formal.
La demostración formal establece que $n^2 + n$ es par al factorizar la expresión como $n(n+1)$. Dado que n y $n+1$ son números consecutivos, uno de ellos siempre debe ser par. El producto de cualquier número por un número par es siempre par, completando la prueba.
Detalles de la Validación:
• **Coherencia Lógica:** Confirmada.
• **Requisitos de Lean:** Cumplidos (import tactic).
• **Tiempo de Verificación:** 0.23 segundos.

Teorema Central (C)

Ancestros (Arriba) 3

Descendencia (Abajo) 2

Grafo de Linaje de Dependencias



Detalle del Nodo Seleccionado

Haga click en un nodo (círculo) en el grafo para ver su descripción aquí.

Teorema Central (C)

Teorema

Mostrar/Ocultar Vistas:

Lenguaje Natural

Formalismo Lean

Definición (Natural)

Toda potencia de un par es par (base de la prueba de n^2). Este teorema combina A1, A2 y A'1.

Código Lean

```
theorem pow_of_even ...
```

Buscar Proyectos de Formalización

Buscar por nombre del proyecto

Resultados

- Teoría de la Computación — Público
- Geometría Algebraica Av... — Privado
- Fundamentos de la Lógic... — Público
- Cálculo Integral para Ing... — Privado
- Teoría de Grafos y Cone... — Público
- Ecuaciones Diferenciale... — Privado

Geometría Algebraica Avanzada

Privado

Colaboradores

User_Current · EvaristeGalois · SophieGermain

Progreso General de Metas

Demostrado: 8 Por Demostrar: 22

27% del total de nodos de la formalización completado.

Crear Nuevo Proyecto de Formalización

Nombre del Proyecto *

Twin Prime Conjecture

Grado de Visibilidad

○ Público (Visible para todos) ◉ Privado (Solo colaboradores)

☑ Importar premisas de un proyecto existente (Opcional)

Teoría de Categorías

Se copiarán todos los teoremas y definiciones del proyecto seleccionado como base.

Añadir Colaboradores (Nombres de Usuario)

toto × adán ×

Escribe el nombre de usuario y presiona Enter — Añadir

Crear Proyecto

Abrir Workspace

Seleccionar Proyecto

Teoría de la Computación (Público)

Tipo de Sesión

○ Individual
Trabajar en el proyecto sin interacción en tiempo real con otros colaboradores.

◉ Nueva Sesión Colaborativa
Invitar a otros colaboradores a editar y demostrar teoremas simultáneamente.

Abrir Workspace

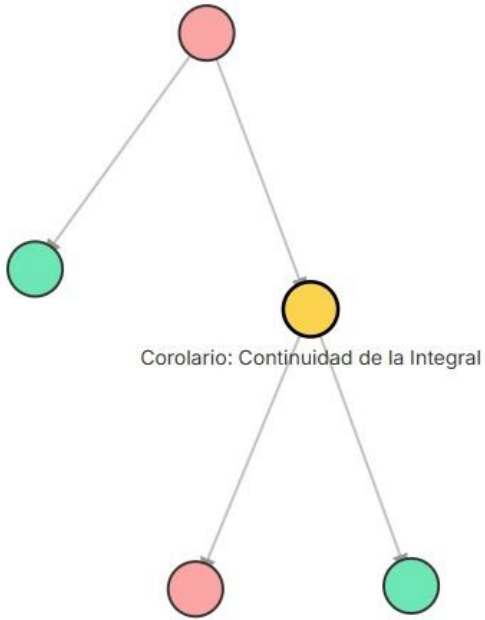
Abriendo Workspace para el proyecto 'Teoría de la Computación' en modo Colaborativo...

Proyecto: Cálculo de Límites (Colab) Sesión Colaborativa

Guardar Cambios (Commit) Activos: User_Current AdaLovelace

Metas del Proyecto

- **Teorema Fundamental del Cálculo (Parte I)**
- Límite de una Sucesión Convergente
- Propiedades de Números Reales



Detalles del Nodo

Nombre:

Corolario: Continuidad de la Integral

Tipo: Corolario

Estado: In progress

Acciones Disponibles

Demostración

Planificación

Datos y Ejemplos

Backend (Python)

Hecho o en proceso:

- **Pruebas unitarias**
- **Estructuras básicas**
- **Contenerización**
- **Conexiones REST API**
- **Conexión con Websocket**

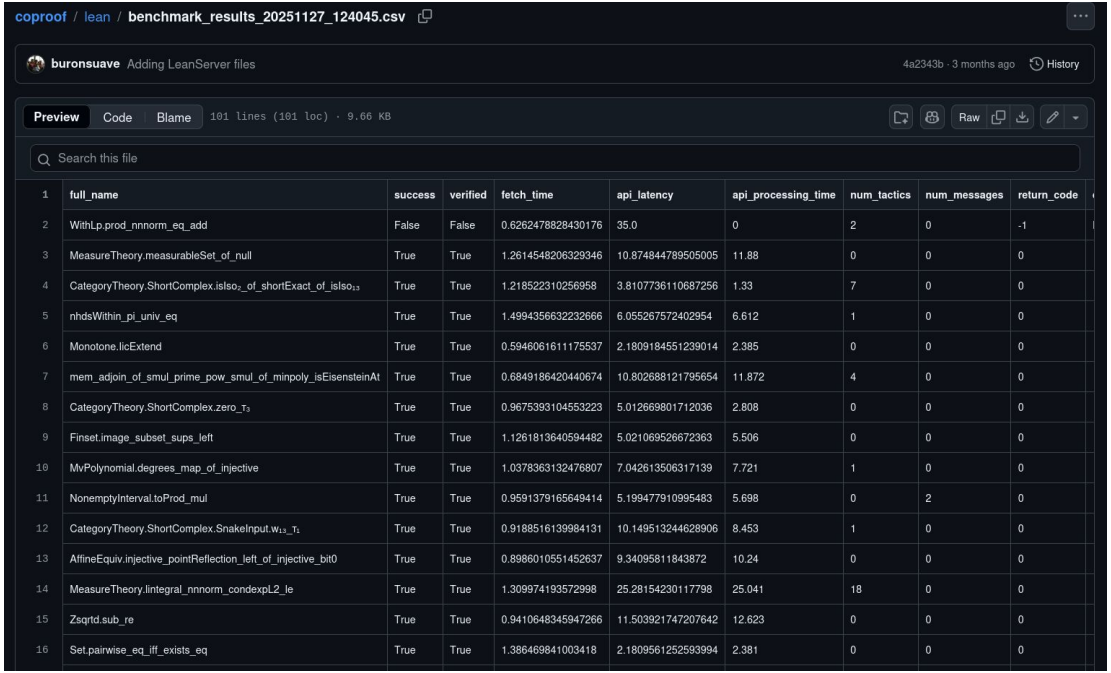
Por hacer:

- **Implementar casos específicos**

Lean Server (LEAN + API)

Hecho o en proceso:

- Pruebas funcionales
- Estructuras básicas
- Contenerización
- Conexión REST API



coproof / lean / benchmark_results_20251127_124045.csv

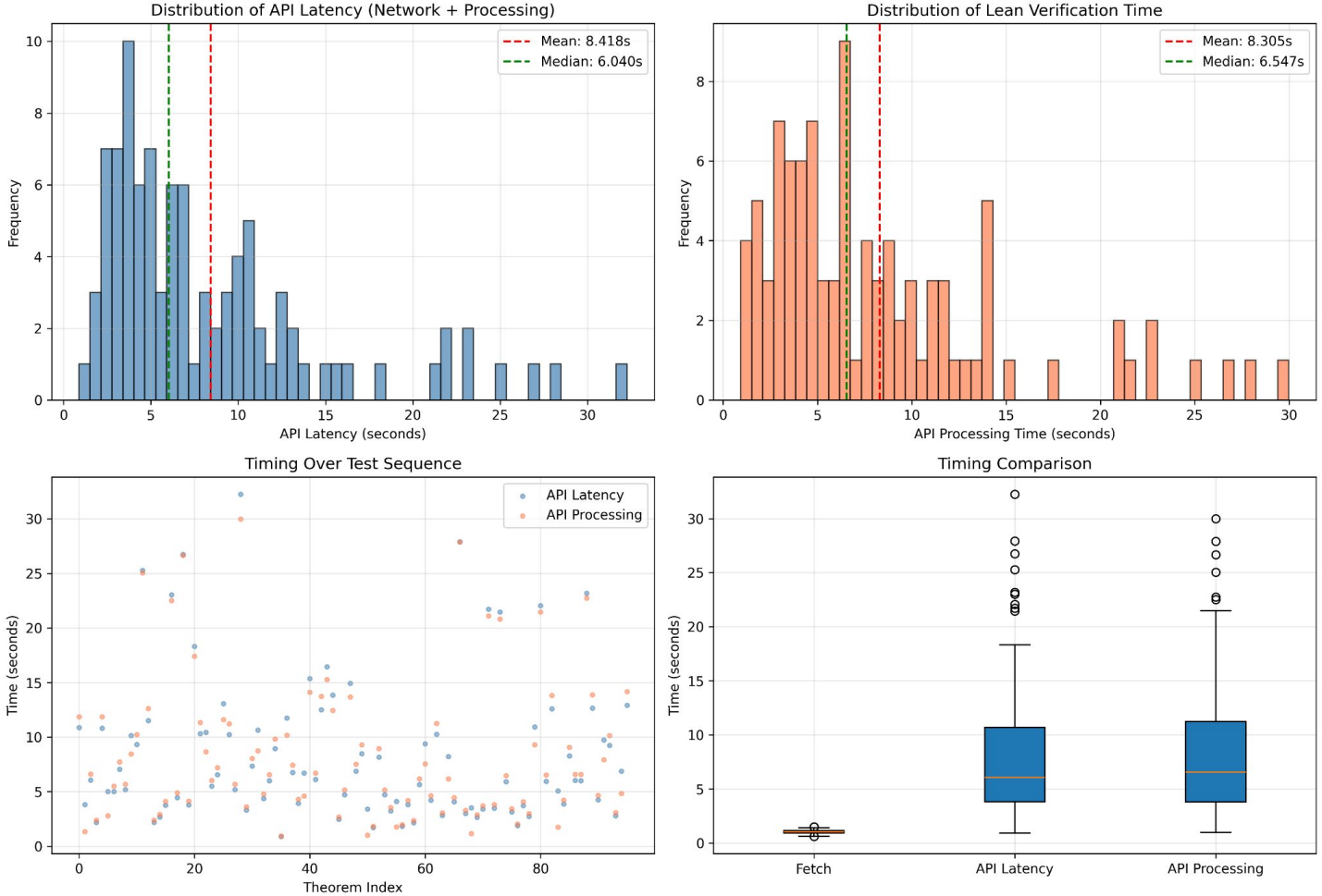
buronsuave Adding LeanServer files — 4a2343b · 3 months ago — History

Preview | Code | Blame — 101 lines (101 loc) · 9.66 KB — Raw

Search this file

	full_name	success	verified	fetch_time	api_latency	api_processing_time	num_tactics	num_messages	return_code
2	WithLp.prod_nnnorm_eq_add	False	False	0.6262478828430176	35.0	0	2	0	-1
3	MeasureTheory.measurableSet_of_null	True	True	1.2614548206329346	10.874844789505005	11.88	0	0	0
4	CategoryTheory.ShortComplex.isIso$_2$_of_shortExact_of_isIso$_{13}$	True	True	1.218522310256958	3.8107736110687256	1.33	7	0	0
5	nhdsWithin_pi_univ_eq	True	True	1.4994356632232666	6.055267572402954	6.612	1	0	0
6	Monotone.licExtend	True	True	0.5946061611175537	2.1809184551239014	2.385	0	0	0
7	mem_adjoin_of_smul_prime_pow_smul_of_minpoly_isEisensteinAt	True	True	0.6849186420440674	10.802688121795654	11.872	4	0	0
8	CategoryTheory.ShortComplex.zero$_{T_3}$	True	True	0.9675393104553223	5.012669801712036	2.808	0	0	0
9	Finset.image_subset_sups_left	True	True	1.1261813640594482	5.021069526672363	5.506	0	0	0
10	MvPolynomial.degrees_map_of_injective	True	True	1.0378363132476807	7.042613506317139	7.721	1	0	0
11	NonemptyInterval.toProd_mul	True	True	0.9591379165649414	5.19947791099548 3	5.698	0	2	0
12	CategoryTheory.ShortComplex.SnakeInput.w$_{13_T_1}$	True	True	0.9188516139984131	10.149513244628906	8.453	1	0	0
13	AffineEquiv.injective_pointReflection_left_of_injective_bit0	True	True	0.8986010551452637	9.34095811843872	10.24	0	0	0
14	MeasureTheory.lintegral_nnnorm_condexpL2_le	True	True	1.309974193572998	25.28154230117798	25.041	18	0	0
15	Zsqrtd.sub_re	True	True	0.9410648345947266	11.503921747207642	12.623	0	0	0
16	Set.pairwise_eq_iff_exists_eq	True	True	1.386469841003418	2.1809561252593994	2.381	0	0	0

Lean 4 Verification API - Timing Analysis



Cluster (OHPC + API)

Hecho o en proceso:

- Acondicionamiento de las RPI4
- Instalación de OHPC
- Pruebas funcionales
- Conexión REST API

```c
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>

// 1200 is divisible by 12 cores.
// Math = 1200^3 = ~1.7 Billion operations (Heavy!)
// RAM  = ~35 MB (Light!)
#define N 600

int main(int argc, char **argv) {
    int rank, size;
    double *A = NULL;       // Master Matrix A (Rank 0 only)
    double *B = NULL;       // Master Matrix B (Everyone needs this)
    double *C = NULL;       // Master Result C (Rank 0 only)
    double *local_A = NULL; // Rows of A for this process
    double *local_C = NULL; // Calculated rows of C for this process

    int rows_per_proc;
    double start_time, end_time;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    // 0. Safety Check
    if (rank == 0) printf("--- Starting Matrix-Matrix Mult with N=%d ---\n", N);

    if (N % size != 0) {
        if (rank == 0) printf("Error: N (%d) must be divisible by cores (%d)\n", N, size);
        MPI_Finalize();
        return 1;
    }

    rows_per_proc = N / size;

    // 1. Allocate Memory
    // Everyone needs the full Matrix B to multiply their rows against
    B = (double *)malloc(N * N * sizeof(double));
```

Welcome | cluster_request.py

test > cluster_request.py > ...

```python
# THE C CODE
# We are searching numbers 1 to 100,000,000
c_source = r"""
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>

// Search range: 100 Million
#define LIMIT 100000000ULL

int main(int argc, char** argv) {
    int rank, size;
    unsigned long long i, start, end;
    unsigned long long local_max_steps = 0;
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
Rank 5 found max steps: 734 (Number: 46564287) in 12.652716 seconds
Rank 6 found max steps: 742 (Number: 55187303) in 12.672971 seconds
Rank 4 found max steps: 744 (Number: 36791535) in 12.564677 seconds
Rank 0 found max steps: 664 (Number: 6649279) in 11.056504 seconds

--- FINAL RESULT ---
The number < 100000000 with the longest Collatz sequence is: 63728127
Sequence Length: 949 steps
Rank 3 found max steps: 705 (Number: 31466382) in 12.324661 seconds
Rank 7 found max steps: 949 (Number: 63728127) in 12.782399 seconds
Rank 10 found max steps: 797 (Number: 86010015) in 12.952342 seconds

--- Stderr ---
[2025-12-04T19:31:20.331] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:20.337] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:20.385] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:19.437] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:20.340] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:20.344] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:20.348] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
[2025-12-04T19:31:19.444] error: couldn't chdir to `/home/mpiuser/webserver': No such file or directory: going to /tmp instead
```

Filesystem + DB (Git + SQL)

Hecho o en proceso:

- Pruebas unitarias
- Modelos
- Contenerización
- Conexiones REST API

Por hacer:

- Implementar casos específicos

- **Translator (Python)**
- **AI Agents (Python)**

Hecho o en proceso:

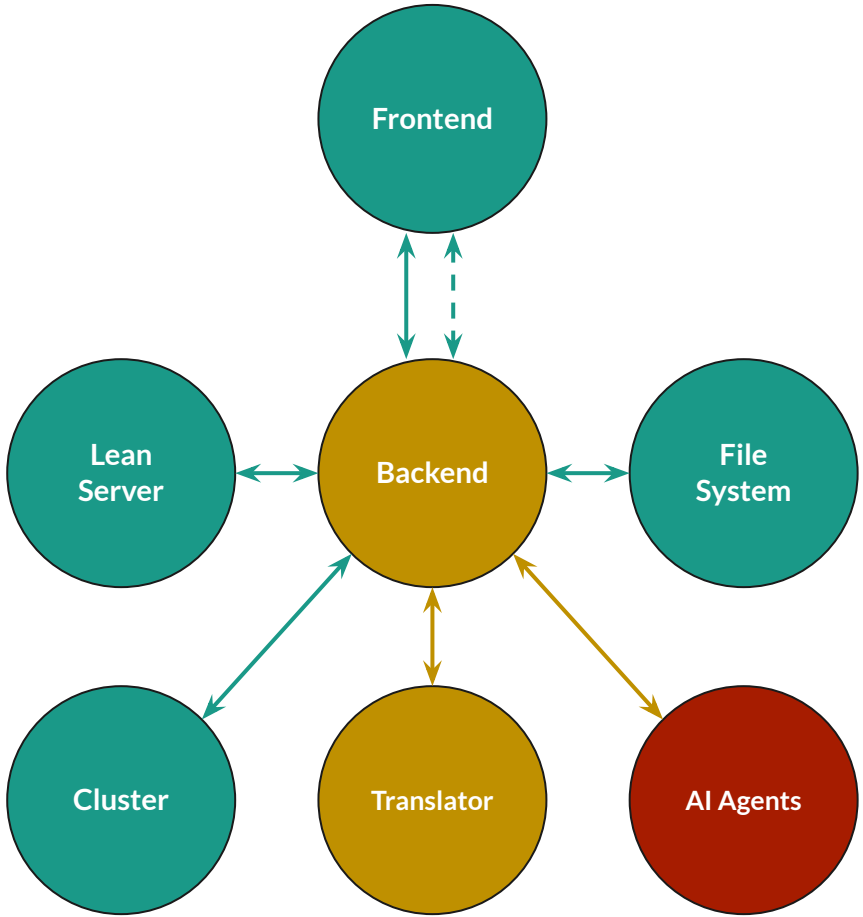
- **Investigación**

Por hacer:

- **Implementación**
- **Conexión con REST API con el Backend**
- **Pruebas**

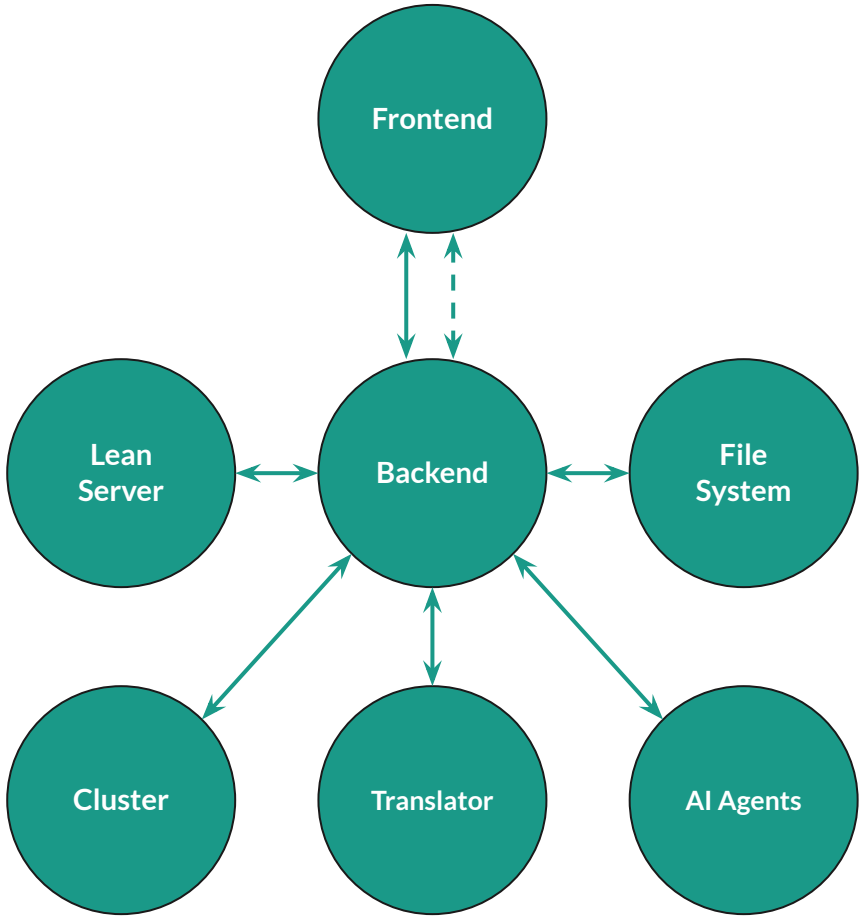
Roadmap: 1er Parcial - Marzo 17

- Backend (Python)
- Frontend (Angular)
- LeanServer (Lean + API)
- Cluster (OHPC en RPI4 + API)
- Filesystem + DB (Git + SQL)
- Translator NL2FL (Python)
- AI Agents (Python)



Roadmap: 2do Parcial - Mayo 8

- Backend (Python)
- Frontend (Angular)
- LeanServer (Lean + API)
- Cluster (OHPC en RPI4 + API)
- Filesystem + DB (Git + SQL)
- Translator NL2FL (Python)
- AI Agents (Python)



CoProof

¡Gracias por su atención!