

Aquí tienes la pieza final de la tríada: el **Módulo Académico** (o "El Investigador").

Este script no se preocupa por el dinero (como el Auditor Económico), se preocupa por la **Verdad Estadística**. Su función es determinar si la diferencia entre tus reactores es ciencia real o simple casualidad (ruido aleatorio).

Utilizaremos la biblioteca estándar científica de Python ([scipy](#)) para realizar un **T-Test (Prueba T de Student)**, que es el estándar académico para comparar dos grupos experimentales (Tu Grupo A vs. Grupo C).

El Script: [academic_researcher.py](#)

Python

```
import numpy as np
from scipy import stats

class AcademicResearcher:
    def __init__(self, confidence_level=0.95):
        self.alpha = 1 - confidence_level # Nivel de significancia (usualmente 0.05)

    def analyze_experiment(self, group_control, group_test):
        """
        Realiza un análisis estadístico comparativo entre dos grupos.
        """

        # 1. Estadística Descriptiva
        mean_control = np.mean(group_control['data'])
        std_control = np.std(group_control['data'], ddof=1) # Desviación estándar muestral

        mean_test = np.mean(group_test['data'])
        std_test = np.std(group_test['data'], ddof=1)

        # 2. Estadística Inferencial (T-Test de muestras independientes)
        # Asumimos varianzas iguales para simplificar la simulación, pero se podría usar Welch
        t_stat, p_value = stats.ttest_ind(group_control['data'], group_test['data'])

        # 3. Interpretación del P-Value
        # Si p_value < 0.05, rechazamos la hipótesis nula (La diferencia ES real)
        significant = p_value < self.alpha

        delta_mean = mean_control - mean_test # Diferencia de medias (Ahorro energético
                                                promedio)

        return {
            "hypothesis": f"El {group_control['name']} consume menos energía que
{group_test['name']}",
        }
```

```

    "control_stats": f"Media: {mean_control:.2f} (±{std_control:.2f})",
    "test_stats": f"Media: {mean_test:.2f} (±{std_test:.2f})",
    "delta": delta_mean,
    "t_statistic": t_stat,
    "p_value": p_value,
    "is_significant": significant,
    "academic_abstract": self._draft_abstract(group_control['name'], group_test['name'],
delta_mean, p_value, significant)
}

def _draft_abstract(self, name_a, name_b, delta, p, is_sig):
    """Redacta un párrafo en lenguaje académico formal."""
    significance_text = "estadísticamente significativa" if is_sig else "no concluyente"

    return (
        f"Se observó una diferencia {significance_text} (p={p:.4f}) en el consumo energético."
        f"El protocolo '{name_a}' demostró una reducción media de {abs(delta):.2f} unidades "
        f"en comparación con el '{name_b}'. Estos resultados sugieren que la optimización
térmica "
        f"en el grupo experimental es efectiva para mitigar pérdidas energéticas."
    )

# --- SIMULACIÓN CON TUS DATOS (Del Screenshot) ---

# Grupo A (Modo Gaia): R-001 y R-003
group_a_data = {
    "name": "Modo de Carbonización Gaia",
    "data": [828.90, 803.76]
}

# Grupo C (Pirólisis Rápida): R-002 y R-004
group_c_data = {
    "name": "Pirólisis Rápida",
    "data": [810.56, 868.37]
}

# --- EJECUCIÓN ---

researcher = AcademicResearcher()
results = researcher.analyze_experiment(group_a_data, group_c_data)

# --- SALIDA AL GESTOR DE TAREAS ---

print("--- INFORME CIENTÍFICO AUTOMATIZADO ---")

```

```

print(f"Comparativa: {group_a_data['name']} vs {group_c_data['name']}") 
print(f"Grupo A (Control): {results['control_stats']}") 
print(f"Grupo C (Test): {results['test_stats']}") 
print("-" * 30) 
print(f"Valor P (P-Value): {results['p_value']:.4f}") 
print(f"¿Es significativo?: {'Sí' if results['is_significant'] else 'NO (Requiere más muestras)'}") 
print("\n--- BORRADOR PARA PAPER (Abstract) ---") 
print(results['academic_abstract'])

# --- INYECCIÓN COPRESET (CONTEXTO ACADÉMICO) --- 
# Esto va al Creador de Prompt para tareas de documentación o divulgación científica

copreset_academic_payload = { 
    "trigger": "Scientific_Validation", 
    "auto_fill_data": { 
        "context": ( 
            f"Validación científica completada. Los datos confirman con un p-value de {results['p_value']:.3f} " 
            f"que el '{group_a_data['name']}' es superior. " 
            f"La desviación estándar en el grupo de Pirólisis Rápida fue alta, indicando inestabilidad." 
        ), 
        "role": "Editor de revista científica (Nature/Science).", 
        "tone": "Formal, Objetivo, Basado en Evidencia.", 
        "task": "Redactar Conclusiones del Experimento" 
    } 
}

```

Cómo se integra esto en tu Flujo de Trabajo (Workflow)

Ahora tienes las tres patas de la mesa. Así es como el **Gestor de Tareas** orquesta todo cuando termina un experimento:

1. **Termina el Experimento:** R-003 gana.
2. **El Gestor consulta al Auditor Económico:**
 - **Respuesta:** "Ahorramos \$61,000 al año."
 - **Acción:** Crea Tarea Prioridad Alta -> "Reporte ROI para Inversores".
3. **El Gestor consulta al Investigador Académico:**
 - **Respuesta:** "El p-value es 0.45 (alto). Aunque R-003 ganó, la varianza es mucha. No podemos publicar esto como un hecho irrefutable todavía."
 - **Acción:** Crea Tarea -> "Diseñar Experimento de Validación (Aumentar N muestras)".
 - **Acción Creativa:** Configura el Creador de Prompt para generar una gráfica de "Barras de Error" para visualizar la incertidumbre.

Visualización del "Nexo Sinérgico" Completo

Imagina tu dashboard central ahora:

Módulo	Input (Dato Crudo)	Procesamiento (Script)	Output (Acción/Prompt)
Pirólisis Hub	803.76 (Energía)	Motor de Simulación	Dato Experimental
Auditor Económico	803.76 vs 868.37	economic_auditor.py	Prompt: "Visualizar Montaña de Dinero Ahorrado"
Investigador Acad.	Listas [A] vs [C]	academic_researcher.py	Prompt: "Redactar Abstract Técnico"
Gestor de Tareas	Resultados de ambos	Lógica de Decisión	Asigna tareas a usuarios y rellena COPRESET