

Aquí tienes una simulación técnica de cómo funcionaría este flujo. He diseñado el código en **Python** para representar la lógica del backend (el cerebro del "Gestor de Tareas") y un objeto **JSON** que representa la carga útil (payload) final que se enviaría a la API del Creador de Prompts.

## 1. El Backend: Clase **TaskManager** (Python)

Este script simula cómo el Gestor de Tareas recibe los datos crudos del PDF/Simulación, convoca a los "Titanes" (agentes lógicos) y construye la estructura COPRESET.

Python

```
import json
```

```
class PyrolysisAnalysis:
```

```
    def __init__(self, data):
        self.name = data.get("name")
        self.framework = data.get("framework_type")
        self.mesopore_vol = data.get("mesopore_volume")
        self.coke_resistance = data.get("coke_resistance")
        self.diagnosis = data.get("diagnosis_text")
```

```
class TitanAgent:
```

```
    """Simula un agente especializado (Debate de Titanes)"""
    def analyze(self, context, role):
        # En un sistema real, esto sería una llamada a un LLM
        if role == "Engineer" and context.mesopore_vol < 0.02:
            return {
                "critical_issue": "Bloqueo de poros por difusión limitada",
                "recommendation": "Introducir mesoporosidad jerárquica",
                "visual_metaphor": "Un túnel de autopista tapiado con escombros"
            }
        return None
```

```
class TaskManager:
```

```
    def __init__(self):
        self.titans = [TitanAgent()] # Lista de agentes disponibles

    def generate_copreset_payload(self, analysis_data):
        # 1. Ingesta de Datos (Simulando los datos extraídos del PDF)
        # Datos basados en
        catalyst = PyrolysisAnalysis(analysis_data)

        # 2. El Filtro Controlador (Lógica de Negocio)
```

```
# Detecta que aunque la estabilidad es alta, la porosidad es crítica
status = "CRITICAL" if catalyst.mesopore_vol < 0.05 else "NORMAL"
```

```
# 3. Llamada a los Titanes (Procesamiento Lógico)
# El Titan Ingeniero analiza el fallo estructural
titan_insight = self.titans[0].analyze(catalyst, "Engineer")
```

```
# 4. Construcción del COPRESET (Mapeo de Datos a Narrativa)
```

```
copreset_structure = {
    "source_app": "Pyrolysis_Hub_v2.1",
    "target_module": "Prompt_Creator_Pro",
    "project_id": "Z2_Optimization_Protocol",
    "auto_fill_data": {
        # C - Contexto: Basado en el diagnóstico del PDF
        "context": (
            f"Estamos analizando la {catalyst.name} (tipo {catalyst.framework}). "
            f"Aunque tiene excelente estabilidad térmica, el análisis muestra un volumen "
            f"de mesoporo casi nulo ({catalyst.mesopore_vol} cm³/g), causando un fallo "
            f"por transporte de materia y coquización rápida."
        ),

        # O - Objetivo: Visualizar el problema para justificar la solución
        "objective": (
            f"Crear una representación visual que explique el '{titan_insight['critical_issue']}' "
            "para justificar la necesidad de desilicación post-síntesis."
        ),

        # P - Presentación: Formato visual definido por el Gestor
        "presentation_format": "Render 3D fotorrealista, corte transversal macroscópico.",

        # R - Rol: Asignado al AI Generativo
        "role": "Experto en visualización científica y nanotecnología.",

        # E - Ejemplos/Restricciones: La metáfora del Titan
        "explicit_constraints": (
            f"Usa la metáfora visual de: {titan_insight['visual_metaphor']}. "
            "Colores: Azul eléctrico para la zeolita cristalina (pura), "
            "Negro denso/Gris carbón para el coque bloqueando las entradas."
        ),

        # S - Pasos: Instrucciones secuenciales
        "steps": [
            "1. Generar estructura cristalina AEL perfecta.",
            "2. Añadir capa de bloqueo en la superficie externa.",
```

```

        "3. Mostrar sitios activos internos brillando pero inalcanzables."
    ],

    # E - Emoción/Tono: Urgencia científica
    "tone": "Analítico, Urgente, Revelador.",

    # T - Tarea Final
    "task": "Generar Prompt de Imagen"
}
}

return copreset_structure

# --- EJECUCIÓN SIMULADA ---

# Datos extraídos del PDF "Diagnóstico General"
raw_input_from_hub = {
    "name": "Zeolita Avanzada Z-2",
    "framework_type": "AEL",
    "mesopore_volume": 0.012, #
    "coke_resistance": 58.0, #
    "diagnosis_text": "Ausencia de sistema de poros jerárquico." #
}

manager = TaskManager()
payload = manager.generate_copreset_payload(raw_input_from_hub)

print(json.dumps(payload, indent=4, ensure_ascii=False))

```

---

## 2. El Resultado: Payload JSON (API Request)

Este es el objeto exacto que viaja desde el **TaskManager** hacia el **Prompt Creator**. Cuando el usuario abre el Creador, estos campos pueblan automáticamente la interfaz.

```

JSON
{
    "source_app": "Pyrolysis_Hub_v2.1",
    "target_module": "Prompt_Creator_Pro",
    "project_id": "Z2_Optimization_Protocol",
    "timestamp": "2025-12-10T23:05:00Z",
    "auto_fill_data": {

```

```

    "context": "Estamos analizando la Zeolita Avanzada Z-2 (tipo AEL). Aunque tiene excelente estabilidad térmica, el análisis muestra un volumen de mesoporo casi nulo (0.012 cm³/g), causando un fallo por transporte de materia y coquización rápida.",
    "objective": "Crear una representación visual que explique el 'Bloqueo de poros por difusión limitada' para justificar la necesidad de desilicación post-síntesis.",
    "presentation_format": "Render 3D fotorrealista, corte transversal macroscópico.",
    "role": "Experto en visualización científica y nanotecnología.",
    "explicit_constraints": "Usa la metáfora visual de: Un túnel de autopista tapiado con escombros. Colores: Azul eléctrico para la zeolita cristalina (pura), Negro denso/Gris carbón para el coque bloqueando las entradas.",
    "steps": [
      "1. Generar estructura cristalina AEL perfecta.",
      "2. Añadir capa de bloqueo en la superficie externa.",
      "3. Mostrar sitios activos internos brillando pero inalcanzables."
    ],
    "tone": "Analítico, Urgente, Revelador.",
    "task": "Generar Prompt de Imagen"
  },
  "metadata": {
    "analysis_source": "Reporte_Dr_Pirolis.pdf",
    "titan_consensus": "Architecture_Failure_Detected",
    "suggested_action": "Hierarchical_Pore_Generation"
  }
}

```

## ¿Por qué funciona esto para tu "Nexo Sinérgico"?

1. **Trazabilidad:** El JSON incluye **metadata** que rastrea el origen de la idea (el reporte del Dr. Pirolis ), cerrando el ciclo entre el dato científico y la salida creativa.
2. **Automatización Inteligente:** El usuario no tiene que copiar y pegar "0.012 cm³/g". El sistema entiende que ese número bajo implica "Bloqueo" y lo traduce a lenguaje natural en el campo **context**.
3. **Flexibilidad:** Si el debate de los Titanes concluyera otra cosa (ej. "Acidez insuficiente"), el campo **explicit\_constraints** cambiaría dinámicamente, pidiendo visualizar enlaces químicos rotos en lugar de túneles bloqueados.