

Este es el "puente de oro" de tu sistema. Si este Payload está bien diseñado, la IA creativa (el LLM detrás del Creador de Prompt) no necesitará adivinar; tendrá **conciencia situacional**.

Para el **Nexo Sinérgico**, no basta con enviar números crudos (**temp: 500**). Necesitamos enviar **Significado**.

Aquí tienes el diseño del JSON Schema, estructurado en capas de abstracción: **Datos Duros (Ciencia)** y **Datos Semánticos (Narrativa)**.

El Payload JSON: **ContextTransferObject**

JSON

```
{
  "meta": {
    "version": "1.0",
    "timestamp": "2025-10-27T14:30:00Z",
    "trace_id": "sim-998877-bridge",
    "origin_system": "PyrolysisHub_Core_v2"
  },
  "project_identity": {
    "project_id": "uuid-del-proyecto-analitico",
    "project_name": "Valorización de Algas Marinas - Fase 3",
    "user_role": "Academico"
  },
  "technical_core": {
    "inputs": {
      "feedstock_type": "Microalgas (Spirulina)",
      "moisture_content_percent": 12.5,
      "process_temperature_celsius": 550,
      "residence_time_seconds": 45
    },
    "outcomes": {
      "yield_distribution": {
        "biochar_percent": 25.0,
        "biooil_percent": 55.0,
        "syngas_percent": 20.0
      },
      "energy_efficiency_percent": 78.4,
      "carbon_sequestration_potential": "High"
    }
  },
  "semantic_bridge": {
    "primary_insight": "Rendimiento de Bio-oil excepcionalmente alto optimizado para biocombustibles.",
```

```

"key_success_metrics": [
  "Eficiencia energética superior al 75%",
  "Baja producción de residuos"
],
"suggested_keywords": [
  "Sostenibilidad Azul",
  "Biocombustible Avanzado",
  "Economía Circular",
  "Energía Renovable"
],
"visual_cues": {
  "dominant_color": "Verde esmeralda (alga) a Oro negro (aceite)",
  "process_state": "Fluido, dinámico, transformación limpia",
  "elements_to_show": ["Matraces de laboratorio", "Océano abstracto", "Gota de combustible brillante"]
}
},
"creative_defaults": {
  "suggested_tone": "Innovador / Científico-Optimista",
  "target_audience_hint": "Inversores de Green Tech",
  "prompt_starter_text": "Crea una narrativa visual que destaque la transformación eficiente de algas en energía limpia, enfatizando..."
}
}

```

Desglose de la Estructura (Por qué funciona)

1. **technical_core** (La Verdad Científica)

Aquí residen los datos puros que vienen de tu motor de simulación.

- **¿Para qué sirve?** Si el usuario pide en el Creador de Prompt: *"Escribe un reporte técnico"*, la IA tiene los números exactos (**550°C**, **78.4%**) para no alucinar datos falsos.

2. **semantic_bridge** (El Traductor)

Esta es la sección más innovadora. El *Pyrolysis Hub* pre-procesa los datos para darles significado.

- **primary_insight**: En lugar de que la IA creativa tenga que juzgar si un 55% de bio-oil es bueno, el sistema científico se lo dice: *"Es excepcionalmente alto"*.

- **visual_cues**: Aquí el sistema científico sugiere metáforas visuales. Sabe que la materia prima es "Algas", por lo tanto sugiere colores "Verde esmeralda". Si fuera "Lodos de depuradora", sugeriría tonos tierra/oscuras. Esto mejora drásticamente la generación de imágenes (DALL-E / Midjourney).

3. **creative_defaults** (La Configuración UI)

Pre-configura la interfaz del Creador de Prompt para ahorrar tiempo al usuario.

- Si la eficiencia es >70%, el tono sugerido es "Optimista/Triunfal".
- Si la eficiencia es <30%, el tono sugerido podría ser "Analítico/Búsqueda de mejoras".

Implementación en Pydantic (Python)

Para integrar esto en tu backend FastAPI, este es el modelo de datos que validará la entrada cuando el botón sea presionado:

Python

```
from pydantic import BaseModel, Field
from typing import List, Dict, Optional
```

Sub-modelos para mantener el código limpio

```
class TechnicalInputs(BaseModel):
    feedstock_type: str
    moisture_content_percent: float
    process_temperature_celsius: int
```

```
class TechnicalOutcomes(BaseModel):
    yield_distribution: Dict[str, float] # {"biochar": 20.5, ...}
    energy_efficiency_percent: float
```

```
class SemanticBridge(BaseModel):
    primary_insight: str
    suggested_keywords: List[str]
    visual_cues: Dict[str, str] # Color, State, Elements
```

```
class ContextPayload(BaseModel):
    """
    El objeto oficial de transferencia entre Pyrolysis Hub y Prompt Creator
    """
    meta: Dict[str, str]
    project_identity: Dict[str, str]
    technical_core: dict = Field(..., description="Datos duros de la simulación")
```

```
    semantic_bridge: SemanticBridge = Field(..., description="Interpretación narrativa de los  
datos")  
    creative_defaults: Optional[Dict[str, str]] = None
```

```
# Ejemplo de uso en el endpoint (receptor)  
@app.post("/creative/receive-context")  
def ingest_context(payload: ContextPayload):  
    # Aquí la IA recibe 'payload.semantic_bridge.primary_insight'  
    # y lo inyecta en el System Prompt del LLM.  
    return {"status": "Contexto asimilado. Listo para generar."}
```

Visualización del Flujo de Datos