

Esta es una de las partes más críticas del **Nexo Sinérgico**. No queremos que la IA genere imágenes aleatorias de "fábricas". Queremos que visualice **tus datos**.

Para lograr esto, necesitamos una función que traduzca **Valores Numéricos (Hard Data)** en **Descriptores Estéticos (Soft Data)**.

Aquí tienes la propuesta técnica de la función `generate_visual_metaphor_prompt` escrita en Python. Esta función está diseñada para vivir en tu backend y alimentar la API de generación de imágenes (DALL-E 3, Midjourney API o Stable Diffusion).

La Lógica: "Traductor de Semántica Industrial"

El objetivo es convertir el dato `0,06 €/kWh` (barato/eficiente) en conceptos visuales como *"aerodinámico"*, *"flujo sin fricción"* o *"luz azul limpia"*.

Python

```
import json
```

```
class VisualMetaphorEngine:
```

```
    def __init__(self):
```

```
        # Configuración de estilos base según la audiencia
```

```
        self.styles = {
```

```
            "INVESTOR": "isometric 3D render, unreal engine 5, corporate minimalism, clean white background, golden accents",
```

```
            "ENGINEER": "blueprint schematic style, cyanotype, technical drawing, precise lines, white grid on blue",
```

```
            "MARKETING": "cinematic hyperrealistic photography, depth of field, golden hour lighting, futuristic sustainability"
```

```
        }
```

```
    def determine_visual_mood(self, efficiency_rate):
```

```
        """
```

```
        Traduce la eficiencia (precio/kWh) en 'Estado de Ánimo' visual.
```

```
        Dato de entrada ejemplo: 0.06 (de tu imagen)
```

```
        """
```

```
        # Benchmark hipotético de mercado: 0.10 €/kWh
```

```
        MARKET_BENCHMARK = 0.10
```

```
        if efficiency_rate <= 0.07:
```

```
            # Caso: Muy eficiente (Tu caso actual)
```

```
            return {
```

```
                "adjectives": "frictionless, aerodynamic, glowing, crystalline, levitating",
```

```
                "color_palette": "electric blue, pure white, and holographic silver",
```

```
                "chaos_level": "--s 50" # Estilizado, ordenado
```

```

    }
elif efficiency_rate > 0.12:
    # Caso: Ineficiente / Costoso
    return {
        "adjectives": "heavy, complex, dense, industrial smoke, friction, sparks",
        "color_palette": "warning orange, dark grey, rusty metal",
        "chaos_level": "--s 250" # Más caótico visualmente
    }
else:
    # Caso: Estándar
    return {
        "adjectives": "solid, metallic, structured, balanced",
        "color_palette": "steel grey and navy blue",
        "chaos_level": "--s 100"
    }

def generate_prompt(self, data_context, audience="INVESTOR"):
    """
    Construye el Prompt Final ensamblando las piezas.
    """

    # 1. Extracción de Datos de tu Interfaz (Imagen 5)
    power_demand = data_context.get('demand_power_kw', 0) # 385 kW
    annual_cost = data_context.get('annual_cost_eur', 0) # 184,800 €
    elec_rate = data_context.get('electricity_rate_eur_kwh', 0) # 0.06 €

    # 2. Determinar la Metáfora Central (El Objeto)
    # Si la demanda es alta (>200kW), usamos objetos de "Poder"
    if power_demand > 200:
        core_object = "a massive pulsing energetic turbine core"
    else:
        core_object = "a compact efficient battery cell"

    # 3. Obtener el 'Mood' basado en la eficiencia
    mood_settings = self.determine_visual_mood(elec_rate)

    # 4. Representación del Dinero (Costo Anual)
    # Si el costo es alto (>100k), lo representamos como estructura sólida
    financial_element = ""
    if annual_cost > 100000:
        financial_element = "supported by a solid foundation of stacked golden geometric blocks representing financial value"

    # 5. CONSTRUCCIÓN DEL PROMPT FINAL

```

Estructura: [Sujeto] + [Acción/Contexto] + [Estilo Artístico] + [Iluminación/Color]

```
prompt = (  
    f"Subject: {mood_settings['adjectives']} {core_object}, {financial_element}. "  
    f"Context: visualizing industrial energy efficiency. "  
    f"Style: {self.styles[audience]}. "  
    f"Colors: {mood_settings['color_palette']}. "  
    f"Lighting: volumetric lighting, ray tracing, 8k resolution. "  
    "No text, no letters."  
)  
  
return prompt
```

--- EJEMPLO DE EJECUCIÓN CON TUS DATOS ---

Datos simulados extraídos de tu captura de pantalla (Imagen 5)

```
input_data = {  
    "demand_power_kw": 385,  
    "electricity_rate_eur_kwh": 0.06,  
    "annual_cost_eur": 184800  
}
```

```
engine = VisualMetaphorEngine()  
final_prompt = engine.generate_prompt(input_data, audience="INVESTOR")  
  
print(final_prompt)
```

El Resultado (Output Generado)

Si ejecutamos este código con los datos de tu imagen, este sería el prompt resultante que tu sistema enviaría automáticamente a la IA de imagen:

Prompt Generado: "Subject: frictionless, aerodynamic, glowing, crystalline, levitating a massive pulsing energetic turbine core, supported by a solid foundation of stacked golden geometric blocks representing financial value. Context: visualizing industrial energy efficiency. Style: isometric 3D render, unreal engine 5, corporate minimalism, clean white background, golden accents. Colors: electric blue, pure white, and holographic silver. Lighting: volumetric lighting, ray tracing, 8k resolution. No text, no letters."

Análisis de la Transformación (Data-to-Art)

1. **Del Dato 0.06 (Eficiencia):** La función inyectó las palabras "*frictionless, levitating*" (sin fricción, levitando). Visualmente, esto comunica que el proceso es barato y fluye bien, sin necesidad de poner números.
2. **Del Dato 385 kW (Potencia):** Se transformó en "*massive pulsing core*" (núcleo masivo pulsante), dando la sensación de escala industrial.
3. **Del Dato 184.800 € (Valor):** Se convirtió en "*foundation of stacked golden geometric blocks*" (cimientos de bloques dorados), convirtiendo el gasto en una imagen de solidez financiera (inversión) en lugar de pérdida.

¿Cómo implementarlo en tu UI?

En tu pantalla de resultados (Imagen 5), al lado del valor **184.800,00 €**, podrías añadir un pequeño icono de "Varita Mágica".

- **Tooltip:** "Visualizar Rentabilidad".
- **Acción:** Ejecuta esta función, llama a la API, y en 10 segundos muestra una ventana modal con la imagen generada y un botón de "Descargar para Presentación".