



**TECNOLÓGICO
NACIONAL DE MÉXICO®**



Ing. En Sistemas Computacionales
Inteligencia Artificial

Maestro:

Zuriel Dathan Mora Felix

Tarea 4:

“Sistema Experto”

Alumnos:

Grande Espinoza Víctor Ramon

20170684

Montero López Yahel Alejandro

21170401

10 / 05 / 2025

Sistema Experto

KnowledgeBase Class explicación:

Esta clase almacena todos los hechos de la base de conocimiento.

Cada tipo de hecho (dispositivo, sintoma, causa, implica, asociado_a, contradice, patron_sintomas) se representa como un arreglo o una lista de tuplas.

implication_facts: Lista de tuplas (síntoma, causa).

association_facts: Lista de tuplas (síntoma, causa, probabilidad).

contradiction_facts: Lista de tuplas (síntoma, causa_incompatible).

pattern_facts: Lista de tuplas, donde cada tupla contiene una lista de síntomas (List[str]) y la causa asociada (str).

Hemos añadido algunos tipos de dispositivos, síntomas y causas adicionales para hacer un ejemplo mas realista.

```
from typing import List, Dict, Tuple, Any

# Definición de La Base de Conocimiento
class KnowledgeBase:
    def __init__(self):
        # --- Hechos Fundamentales ---
        self.device_types: List[str] = [
            'smartphone',
            'computadora',
            'televisor',
            'consola_videojuegos',
            'tablet',
            'router_wifi',
            'impresora',
            'monitor',
        ]

        self.symptom_types: List[str] = [
            'apagado_inesperado',
            'lentitud_general',
            'pantalla_negra',
            'no_reconoce_teclado',
            'no_conecta_wifi',
            'sonidos_extraños',
            'sobrecalentamiento',
            'errores_en_pantalla',
            'no_enciende',
            'periferico_no_funciona',
```

```

        'luz_indicadora_parpadea',
        'imagen_distorsionada',
        'no_imprime',
        'mensaje_de_error',
        'se_reinicia_solo',
        'no_hay_sonido',
        'sin_senal',
        'bateria_completamente_cargada',
        'teclado_funciona_en_otro_dispositivo',
        'impresora_imprime_pagina_de_prueba',
        'otros_dispositivos_se_conectan_bien_a_la_misma_red_wifi',
        'dispositivo_funciona_normalmente',
    ]

    self.cause_types: List[str] = [
        'bateria_agotada',
        'sobrecalentamiento',
        'error_software',
        'fallo_hardware_interno',
        'problema_alimentacion',
        'driver_incompatible',
        'problema_red',
        'fallo_componente_especifico',
        'conexion_defectuosa',
        'configuracion_incorrecta',
        'sistema_operativo_corrupto',
        'cable_danado',
        'puerto_danado',
        'problema_compatibilidad',
        'cartuchos_vacios',
        'atasco_papel',
        'problema_configuracion_red_dispositivo',
        'problema_firmware',
        'virus_malware',
        'problema_conexion_externa',
        'altavoces_desconectados',
        'problema_ventilacion',
        'problema_red_general',
    ]

    # --- Relaciones Síntoma-Causa ---
    # implica(Sintoma, Causa)
    self.implication_facts: List[Tuple[str, str]] = [
        ('pantalla_negra', 'fallo_hardware_interno'),
        ('pantalla_negra', 'error_software'),
    ]

```

```

        ('no_conecta_wifi', 'problema_red'),
        ('no_conecta_wifi', 'configuracion_incorrecta'),
        ('no_enciende', 'problema_alimentacion'),
        ('no_enciende', 'fallo_hardware_interno'),
        ('no_reconoce_teclado', 'conexion_defectuosa'),
        ('no_reconoce_teclado', 'driver_incompatible'),
        ('periferico_no_funciona', 'conexion_defectuosa'),
        ('periferico_no_funciona', 'driver_incompatible'),
        ('sonidos_extraños', 'fallo_hardware_interno'),
        ('imagen_distorsionada', 'fallo_hardware_interno'),
        ('imagen_distorsionada', 'driver_incompatible'),
        ('no_imprime', 'cartuchos_vacios'),
        ('no_imprime', 'atasco_papel'),
        ('no_imprime', 'problema_configuracion_red_dispositivo'),
        ('no_imprime', 'driver_incompatible'),
        ('no_hay_sonido', 'altavoces_desconectados'),
        ('sin_senal', 'problema_conexion_externa'),
        ('mensaje_de_error', 'error_software'),
        ('apagado_inesperado', 'bateria_agotada'),
        ('apagado_inesperado', 'sobrecalentamiento'),
        ('apagado_inesperado', 'error_software'),
        ('apagado_inesperado', 'problema_alimentacion'),
        ('se_reinicia_solo', 'error_software'),
        ('se_reinicia_solo', 'fallo_hardware_interno'),
        ('lentitud_general', 'error_software'),
        ('lentitud_general', 'sobrecalentamiento'),
        ('lentitud_general', 'fallo_hardware_interno'),
    ]

```

```

# asociado_a(Sintoma, Causa, Probabilidad)
self.association_facts: List[Tuple[str, str, float]] = [
    ('apagado_inesperado', 'bateria_agotada', 0.9),
    ('apagado_inesperado', 'sobrecalentamiento', 0.6),
    ('apagado_inesperado', 'error_software', 0.5),
    ('apagado_inesperado', 'problema_alimentacion', 0.7),
    ('se_reinicia_solo', 'error_software', 0.6),
    ('se_reinicia_solo', 'fallo_hardware_interno', 0.8),
    ('lentitud_general', 'error_software', 0.8),
    ('lentitud_general', 'sobrecalentamiento', 0.6),
    ('lentitud_general', 'fallo_hardware_interno', 0.5),
    ('lentitud_general', 'virus_malware', 0.75),
    ('no_conecta_wifi', 'problema_red', 0.7),
    ('no_conecta_wifi', 'configuracion_incorrecta', 0.5),
    ('no_conecta_wifi', 'problema_firmware', 0.4),
    ('pantalla_negra', 'cable_danado', 0.4),

```

```

        ('pantalla_negra', 'puerto_danado', 0.3),
        ('pantalla_negra', 'fallo_hardware_interno', 0.7),
        ('errores_en_pantalla', 'driver_incompatible', 0.8),
        ('errores_en_pantalla', 'fallo_hardware_interno', 0.6),
        ('no_reconoce_teclado', 'conexion_defectuosa', 0.8),
        ('no_reconoce_teclado', 'driver_incompatible', 0.4),
    ]

    # contradice(Sintoma, Causa)
    self.contradiction_facts: List[Tuple[str, str]] = [
        ('no_enciende', 'dispositivo_funciona_normalmente'),
        ('apagado_inesperado', 'bateria_completamente_cargada'),
        ('no_reconoce_teclado', 'teclado_funciona_en_otro_dispositivo'),
        ('no_imprime', 'impresora_imprime_pagina_de_prueba'),
        ('no_conecta_wifi',
'otros_dispositivos_se_conectan_bien_a_la_misma_red_wifi'),
    ]

    # patron_sintomas(Lista_Sintomas, Causa)
    self.pattern_facts: List[Tuple[List[str], str]] = [
        (['apagado_inesperado', 'se_reinicia_solo'],
'fallo_alimentacion_interno'),
        (['lentitud_general', 'sobrecalentamiento'],
'problema_ventilacion'),
        (['no_conecta_wifi', 'no_conecta_ethernet'],
'problema_red_general'),
        (['no_reconoce_teclado', 'no_reconoce_mouse'], 'puerto_danado'),
        (['no_hay_sonido', 'imagen_normal'],
'fallo_componente_especifico'),
    ]

```

InferenceEngine Class explicacion:

Se inicializa con una instancia de KnowledgeBase.

Con listas para almacenar los resultados intermedios de cada regla (possible_causes_direct, possible_causes_probabilistic, possible_causes_pattern, causes_to_exclude).

refined_causes: Un diccionario para almacenar las causas probabilísticas con su probabilidad combinada.

reset_diagnostics(): Limpia los resultados de una ejecución anterior.

diagnose(device, observed_symptoms): Este es el método principal que simula la ejecución del motor de inferencia:

Itera a través de los síntomas observados y aplica la lógica de cada regla:

Regla 1 (Directa): Recorre `implication_facts`. Si un síntoma observado coincide con el síntoma en un hecho implica, agrega la causa correspondiente a `possible_causes_direct`.

Regla 2 (Asociación): Recorre `association_facts`. Si un síntoma observado coincide con el síntoma en un hecho asociado_a, agrega la causa y su probabilidad a `possible_causes_probabilistic`.

Regla 4 (Exclusión): Recorre `contradiction_facts`. Si un síntoma observado coincide con el síntoma en un hecho contradice, agrega la causa incompatible a `causes_to_exclude`.

Regla 5 (Patrón): Recorre `pattern_facts`. Para cada patrón, verifica si todos los síntomas en la lista del patrón están presentes en la lista `observed_symptoms`. Si es así, agrega la causa del patrón a `possible_causes_pattern`. Se usa `all(symptom in observed_symptoms for symptom in pattern_symptoms_list)` para esta verificación.

Regla 3 (Refinamiento): Primero, se agrupan todas las probabilidades encontradas por la Regla 2 para cada causa. Luego, itera sobre estas causas agrupadas y, si una causa tiene múltiples probabilidades asociadas, calcula una probabilidad combinada (en este ejemplo simple, usamos el promedio). Almacena el resultado en `self.refined_causes`.

Consolidación de Resultados: Combina los resultados de las diferentes reglas en el diccionario `final_possible_causes`, manejando posibles duplicados y agregando información de probabilidad si está disponible.

Exclusión: Elimina las causas de `final_possible_causes` que fueron marcadas para exclusión por la Regla 4.

Presentación: Imprime la lista final de posibles causas, opcionalmente ordenadas por probabilidad combinada.

```
class InferenceEngine:
    def __init__(self, kb: KnowledgeBase):
        self.kb = kb
        self.possible_causes_direct: List[Tuple[str, str]] = []
        self.possible_causes_probabilistic: List[Tuple[str, str, float]] =
[]
        self.possible_causes_pattern: List[Tuple[str, str]] = []
        self.causes_to_exclude: List[Tuple[str, str]] = []
        self.refined_causes: Dict[str, float] = {}
```

```

def reset_diagnostics(self):
    """Reinicia los resultados del diagnóstico."""
    self.possible_causes_direct = []
    self.possible_causes_probabilistic = []
    self.possible_causes_pattern = []
    self.causes_to_exclude = []
    self.refined_causes = {}

def diagnose(self, device: str, observed_symptoms: List[str]):
    """
    Ejecuta el proceso de diagnóstico basado en los síntomas observados
    para un dispositivo específico.
    """
    self.reset_diagnostics()

    print(f"--- Diagnóstico para '{device}' con síntomas:
{observed_symptoms} ---")

    if device not in self.kb.device_types:
        print(f"Advertencia: Tipo de dispositivo '{device}' no
reconocido en la base de conocimiento.")

    for symptom in observed_symptoms:
        if symptom not in self.kb.symptom_types:
            print(f"Advertencia: Síntoma '{symptom}' no reconocido en
la base de conocimiento.")

    print("\nAplicando Regla 1 (Diagnóstico Directo):")
    for observed_symptom in observed_symptoms:
        for implication_symptom, cause in self.kb.implication_facts:
            if observed_symptom == implication_symptom:
                self.possible_causes_direct.append((device, cause))
                print(f"  Síntoma '{observed_symptom}' implica ->
Posible Causa: '{cause}'")

    print("\nAplicando Regla 2 (Diagnóstico por Asociación):")
    for observed_symptom in observed_symptoms:
        for association_symptom, cause, prob in
self.kb.association_facts:
            if observed_symptom == association_symptom:
                self.possible_causes_probabilistic.append((device,
cause, prob))
                print(f"  Síntoma '{observed_symptom}' asociado a ->
Posible Causa: '{cause}' con probabilidad {prob}")

```

```

        print("\nAplicando Regla 4 (Exclusión de Causas):")
        for observed_symptom in observed_symptoms:
            for contradiction_symptom, excluded_cause in
self.kb.contradiction_facts:
                if observed_symptom == contradiction_symptom:
                    self.causes_to_exclude.append((device, excluded_cause))
                    print(f" Síntoma '{observed_symptom}' contradice ->
Excluir Causa: '{excluded_cause}'")

        print("\nAplicando Regla 5 (Agrupación de Síntomas en Patrones):")
        for pattern_symptoms_list, cause in self.kb.pattern_facts:

            if all(symptom in observed_symptoms for symptom in
pattern_symptoms_list):
                self.possible_causes_pattern.append((device, cause))
                print(f" Patrón de síntomas {pattern_symptoms_list}
observado -> Posible Causa: '{cause}'")

        print("\nAplicando Regla 3 (Refinamiento por Múltiples Síntomas):")
        cause_probabilities: Dict[str, List[float]] = {}
        for _, cause, prob in self.possible_causes_probabilistic:
            if cause not in cause_probabilities:
                cause_probabilities[cause] = []
            cause_probabilities[cause].append(prob)

        for cause, probs in cause_probabilities.items():
            if len(probs) > 1:
                combined_prob = sum(probs) / len(probs)
                self.refined_causes[cause] = combined_prob
                print(f" Causa '{cause}' sugerida por múltiples síntomas
con probabilidades {probs} -> Probabilidad Combinada: {combined_prob:.2f}")
            elif len(probs) == 1:
                self.refined_causes[cause] = probs[0]

        print("\n--- Resultados del Diagnóstico Preliminar ---")
        final_possible_causes: Dict[str, Any] = {}

        for _, cause in self.possible_causes_direct:
            if cause not in final_possible_causes:
                final_possible_causes[cause] = {'tipo': 'Directa'}

        for _, cause in self.possible_causes_pattern:
            if cause not in final_possible_causes:

```



```

        final_possible_causes[cause] = {'tipo': 'Patrón'}
    else:
        final_possible_causes[cause]['tipo'] =
f"{final_possible_causes[cause]['tipo']}, Patrón"

    for cause, prob in self.refined_causes.items():
        if cause not in final_possible_causes:
            final_possible_causes[cause] = {'tipo': 'Probabilística',
'probabilidad': prob}
        else:
            final_possible_causes[cause]['probabilidad'] = prob
            final_possible_causes[cause]['tipo'] =
f"{final_possible_causes[cause]['tipo']}, Probabilística"

    causes_to_remove = [cause for _, cause in self.causes_to_exclude]
    for cause in causes_to_remove:
        if cause in final_possible_causes:
            del final_possible_causes[cause]
            print(f" -> Causa '{cause}' excluida por síntoma
contradictorio.")

    if not final_possible_causes:
        print(f"No se encontraron posibles causas claras para {device}
con los síntomas proporcionados.")
    else:
        print("Posibles Causas Sugeridas:")
        sorted_causes = sorted(
            final_possible_causes.items(),
            key=lambda item: item[1].get('probabilidad', -1),
            reverse=True
        )

        for cause, details in sorted_possible_causes:
            detail_str = f" - {cause} (Tipo: {details['tipo']})"
            if 'probabilidad' in details:
                detail_str += f", Probabilidad/Peso Combinado:
{details['probabilidad']:.2f}"
            detail_str += ")"
            print(detail_str)

```

Clase para la ejecución del programa:

Se crea una instancia de KnowledgeBase y de InferenceEngine pasándole la instancia de KnowledgeBase.

Se definen listas de observed_symptoms simulando lo que un usuario reportaría.

Se llama al método diagnose del motor de inferencia para cada escenario de síntomas.

```
if __name__ == "__main__":
    kb = KnowledgeBase()

    engine = InferenceEngine(kb)

    symptoms_user1_smartphone = [
        'apagado_inesperado',
        'no_conecta_wifi',
        'sobrecalentamiento',
    ]

    symptoms_user2_computadora = [
        'lentitud_general',
        'errores_en_pantalla',
        'sobrecalentamiento',
        'mensaje_de_error'
    ]

    symptoms_user3_smartphone_pattern = [
        'apagado_inesperado',
        'se_reinicia_solo'
    ]

    engine.diagnose("mi_smartphone", symptoms_user1_smartphone)

    print("\n" + "="*30 + "\n")

    engine.diagnose("mi_computadora", symptoms_user2_computadora)

    print("\n" + "="*30 + "\n")

    engine.diagnose("mi_smartphone", symptoms_user3_smartphone_pattern)

    print("\n" + "="*30 + "\n")
```

Capturas de la ejecución para smartphone con problemas:

```
--- Diagnóstico para 'mi_smartphone' con síntomas: ['apagado_inesperado', 'no_conecta_wifi', 'sobrecalentamiento'] ---
Advertencia: Tipo de dispositivo 'mi_smartphone' no reconocido en la base de conocimiento.
```

Aplicando Regla 1 (Diagnóstico Directo):

```
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'bateria_agotada'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'sobrecalentamiento'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'error_software'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'problema_alimentacion'
Síntoma 'no_conecta_wifi' implica -> Posible Causa: 'problema_red'
Síntoma 'no_conecta_wifi' implica -> Posible Causa: 'configuracion_incorrecta'
```

Aplicando Regla 2 (Diagnóstico por Asociación):

```
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'bateria_agotada' con probabilidad 0.9
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'sobrecalentamiento' con probabilidad 0.6
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'error_software' con probabilidad 0.5
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'problema_alimentacion' con probabilidad 0.7
Síntoma 'no_conecta_wifi' asociado a -> Posible Causa: 'problema_red' con probabilidad 0.7
Síntoma 'no_conecta_wifi' asociado a -> Posible Causa: 'configuracion_incorrecta' con probabilidad 0.5
Síntoma 'no_conecta_wifi' asociado a -> Posible Causa: 'problema_firmware' con probabilidad 0.4
```

Aplicando Regla 4 (Exclusión de Causas):

```
Síntoma 'apagado_inesperado' contradice -> Excluir Causa: 'bateria_completamente_cargada'
Síntoma 'no_conecta_wifi' contradice -> Excluir Causa: 'otros_dispositivos_se_conectan_bien_a_la_misma_red_wifi'
```

Aplicando Regla 5 (Agrupación de Síntomas en Patrones):

Aplicando Regla 3 (Refinamiento por Múltiples Síntomas):

```
--- Resultados del Diagnóstico Preliminar ---
```

Posibles Causas Sugeridas:

- bateria_agotada (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.90)
- problema_alimentacion (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.70)
- problema_red (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.70)
- sobrecalentamiento (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.60)
- error_software (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.50)
- configuracion_incorrecta (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.50)
- problema_firmware (Tipo: Probabilística, Probabilidad/Peso Combinado: 0.40)

```
=====
```

Capturas de la ejecución para computadora con problemas:

```
--- Diagnóstico para 'mi_computadora' con síntomas: ['lentitud_general', 'errores_en_pantalla', 'sobrecalentamiento', 'mensaje_de_error']
---
Advertencia: Tipo de dispositivo 'mi_computadora' no reconocido en la base de conocimiento.

Aplicando Regla 1 (Diagnóstico Directo):
Síntoma 'lentitud_general' implica -> Posible Causa: 'error_software'
Síntoma 'lentitud_general' implica -> Posible Causa: 'sobrecalentamiento'
Síntoma 'lentitud_general' implica -> Posible Causa: 'fallo_hardware_interno'
Síntoma 'mensaje_de_error' implica -> Posible Causa: 'error_software'

Aplicando Regla 2 (Diagnóstico por Asociación):
Síntoma 'lentitud_general' asociado a -> Posible Causa: 'error_software' con probabilidad 0.8
Síntoma 'lentitud_general' asociado a -> Posible Causa: 'sobrecalentamiento' con probabilidad 0.6
Síntoma 'lentitud_general' asociado a -> Posible Causa: 'fallo_hardware_interno' con probabilidad 0.5
Síntoma 'lentitud_general' asociado a -> Posible Causa: 'virus_malware' con probabilidad 0.75
Síntoma 'errores_en_pantalla' asociado a -> Posible Causa: 'driver_incompatible' con probabilidad 0.8
Síntoma 'errores_en_pantalla' asociado a -> Posible Causa: 'fallo_hardware_interno' con probabilidad 0.6

Aplicando Regla 4 (Exclusión de Causas):

Aplicando Regla 5 (Agrupación de Síntomas en Patrones):
Patrón de síntomas ['lentitud_general', 'sobrecalentamiento'] observado -> Posible Causa: 'problema_ventilacion'

Aplicando Regla 3 (Refinamiento por Múltiples Síntomas):
Causa 'fallo_hardware_interno' sugerida por múltiples síntomas con probabilidades [0.5, 0.6] -> Probabilidad Combinada: 0.55
```

--- Resultados del Diagnóstico Preliminar ---

Posibles Causas Sugeridas:

- error_software (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.80)
- driver_incompatible (Tipo: Probabilística, Probabilidad/Peso Combinado: 0.80)
- virus_malware (Tipo: Probabilística, Probabilidad/Peso Combinado: 0.75)
- sobrecalentamiento (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.60)
- fallo_hardware_interno (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.55)
- problema_ventilacion (Tipo: Patrón)

=====

Capturas de la ejecución para segundo smarthphone con problemas:

```
--- Diagnóstico para 'mi_smartphone' con síntomas: ['apagado_inesperado', 'se_reinicia_solo'] ---
Advertencia: Tipo de dispositivo 'mi_smartphone' no reconocido en la base de conocimiento.

Aplicando Regla 1 (Diagnóstico Directo):
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'bateria_agotada'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'sobrecalentamiento'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'error_software'
Síntoma 'apagado_inesperado' implica -> Posible Causa: 'problema_alimentacion'
Síntoma 'se_reinicia_solo' implica -> Posible Causa: 'error_software'
Síntoma 'se_reinicia_solo' implica -> Posible Causa: 'fallo_hardware_interno'

Aplicando Regla 2 (Diagnóstico por Asociación):
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'bateria_agotada' con probabilidad 0.9
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'sobrecalentamiento' con probabilidad 0.6
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'error_software' con probabilidad 0.5
Síntoma 'apagado_inesperado' asociado a -> Posible Causa: 'problema_alimentacion' con probabilidad 0.7
Síntoma 'se_reinicia_solo' asociado a -> Posible Causa: 'error_software' con probabilidad 0.6
Síntoma 'se_reinicia_solo' asociado a -> Posible Causa: 'fallo_hardware_interno' con probabilidad 0.8

Aplicando Regla 4 (Exclusión de Causas):
Síntoma 'apagado_inesperado' contradice -> Excluir Causa: 'bateria_completamente_cargada'

Aplicando Regla 5 (Agrupación de Síntomas en Patrones):
Patrón de síntomas ['apagado_inesperado', 'se_reinicia_solo'] observado -> Posible Causa: 'fallo_alimentacion_interno'

Aplicando Regla 3 (Refinamiento por Múltiples Síntomas):
Causa 'error_software' sugerida por múltiples síntomas con probabilidades [0.5, 0.6] -> Probabilidad Combinada: 0.55
```

```
--- Resultados del Diagnóstico Preliminar ---
Posibles Causas Sugeridas:
- bateria_agotada (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.90)
- fallo_hardware_interno (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.80)
- problema_alimentacion (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.70)
- sobrecalentamiento (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.60)
- error_software (Tipo: Directa, Probabilística, Probabilidad/Peso Combinado: 0.55)
- fallo_alimentacion_interno (Tipo: Patrón)
```

```
=====
```