TP7

September 25, 2025

Temas Tratados en el Trabajo Práctico 7

- Teoría de utilidad.
- Toma de decisiones basadas en utilidad.
- Valor de la información.
- Ganancia y entropía.
- Algoritmos basados en la teoría de la decisión.
- Sistemas expertos.

1 Ejercicios Teóricos

1.1 ¿Qué representa una función de utilidad?

En el campo de la Inteligencia Artificial, y particularmente en la teoría de la toma de decisiones, la **función de utilidad** es un concepto central que representa el valor subjetivo o la conveniencia que un agente racional le asigna a un determinado resultado o estado del mundo. En términos más simples, es una forma matemática de cuantificar las preferencias de un agente.

A diferencia del valor monetario o cualquier otra medida objetiva, la utilidad busca capturar la verdadera satisfacción o insatisfacción que un agente experimenta. La premisa fundamental es que un agente no solo busca un resultado, sino que busca el resultado que le proporcione la mayor utilidad esperada, especialmente en escenarios donde hay incertidumbre.

La función de utilidad en la práctica

La función de utilidad es el pilar de la **teoría de la máxima utilidad esperada**. Un agente que sigue este principio elige la acción que maximiza su utilidad esperada (UE), calculada con la siguiente fórmula:

`UE(a) = P(resultado | a) * U(resultado)`

Donde: * **`UE(a)`** es la utilidad esperada de la acción `a`. * **`P(resultado | a)`** es la probabilidad de obtener un resultado específico dada la acción `a`. * **`U(resultado)`** es la utilidad que el agente le asigna a ese resultado.

En la Inteligencia Artificial, esto se aplica en herramientas como las **redes de decisión** (diagramas de influencias), que extienden las redes bayesianas al incluir nodos de decisión (rectángulos) y nodos de utilidad (rombos). Estos diagramas permiten al agente modelar la incertidumbre y las posibles consecuencias de sus acciones para tomar la mejor decisión posible.

1.2 Respondan las siguientes preguntas. Cada respuesta corresponde a un axioma de la utilidad, indique cuál se relaciona con cada pregunta y dé una breve explicación de lo que dice el axioma.

Axiomas de la Utilidad (Teoría de la Utilidad Esperada)

Los axiomas son condiciones que, si se cumplen, implican que la preferencia del agente puede ser representada por una función de utilidad U tal que el agente prefiere una opción a otra si, y solo si, la utilidad esperada de la primera es mayor.

- 1. Ordenamiento Alternativo (Axioma de Completitud)

Este axioma, más formalmente conocido como **Completitud**, establece que el agente racional siempre tiene una preferencia definida entre dos resultados o loterías (opciones inciertas).

- **Principio:** Dadas dos opciones cualesquiera, A y B, el agente debe ser capaz de afirmar una de las siguientes tres cosas: 1. Prefiere A sobre B ($A \succ B$). 2. Prefiere B sobre A ($B \succ A$). 3. Es indiferente entre A y B ($A \sim B$).
- **Implicación:** Un agente racional no puede ser indeciso ni decir “No sé cuál prefiero”. Siempre debe tener una preferencia bien definida, lo que garantiza que el conjunto de sus preferencias está **completamente ordenado**.

- 2. Transitividad

La transitividad asegura que las preferencias del agente son lógicamente **consistentes**, evitando ciclos irracionales.

- **Principio:** Si el agente prefiere la opción A a B ($A \succ B$), y prefiere B a C ($B \succ C$), entonces debe preferir A a C ($A \succ C$).
- **Implicación:** Si este axioma se viola (por ejemplo, si se diera $C \succ A$), el agente caería en una **paradoja de la inconsistencia**. Por ejemplo, un agente que prefiere manzanas a plátanos, plátanos a cerezas, pero cerezas a manzanas, podría ser explotado en un ciclo de intercambios que lo empobrece. La transitividad es esencial para la coherencia.

- 3. Sustituibilidad (Axioma de Independencia)

Este axioma, frecuentemente llamado **Independencia** o **Sustituibilidad**, es el más crucial para la estructura matemática de la utilidad esperada. Afirma que la preferencia entre dos opciones no debe verse afectada por un resultado común que se agrega a ambas.

- **Principio:** Si el agente prefiere A a B ($A \succ B$), entonces una lotería compuesta por A y un resultado C con cierta probabilidad p, debe ser preferida a una lotería compuesta por B y el mismo resultado C con la misma probabilidad p. En notación formal:

$$\text{Si } A \succ B, \text{ entonces } [p : A; (1-p) : C] \succ [p : B; (1-p) : C]$$

- **Implicación:** La preferencia entre A y B es **independiente** del resultado común C. Si un agente debe elegir entre dos máquinas tragamonedas que son idénticas en todo, excepto en una ronda, su elección debe basarse únicamente en esa ronda diferente.

- 4. Monotonicidad (Preferencia por Mayor Probabilidad)

El axioma de **Monotonicidad** establece que, si un agente prefiere un resultado a otro, siempre preferirá la lotería que le dé una **mayor probabilidad/utilidad** de obtener su resultado preferido.

- **Principio:** Si el agente prefiere A a B ($A \succ B$), entonces si se le presentan dos loterías que solo ofrecen A o B, la lotería que asigna una probabilidad p mayor de obtener A siempre será preferida a una lotería que asigne una probabilidad q menor (donde $p > q$).
- **Implicación:** Un agente racional siempre prefiere más de lo bueno. Este axioma asegura que las preferencias responden de manera lógica y directa a las probabilidades.

- 5. Continuidad

El axioma de **Continuidad** garantiza que, por muy bueno que sea un resultado, el agente siempre puede ser disuadido si la probabilidad de obtener un resultado malo es lo suficientemente alta, y viceversa.

- **Principio:** Dados tres resultados A, B, y C con un orden de preferencia ($A \succ B \succ C$), debe existir una probabilidad p tal que el agente sea **indiferente** entre recibir B con certeza y participar en una lotería que ofrece A con probabilidad p y C con probabilidad $(1-p)$.
- **Implicación:** No hay resultados infinitamente buenos o infinitamente malos. Esto permite que las preferencias puedan ser mapeadas a valores numéricos continuos (la función de utilidad), ya que un resultado intermedio (B) siempre puede ser igualado al riesgo de una combinación extrema (A o C).

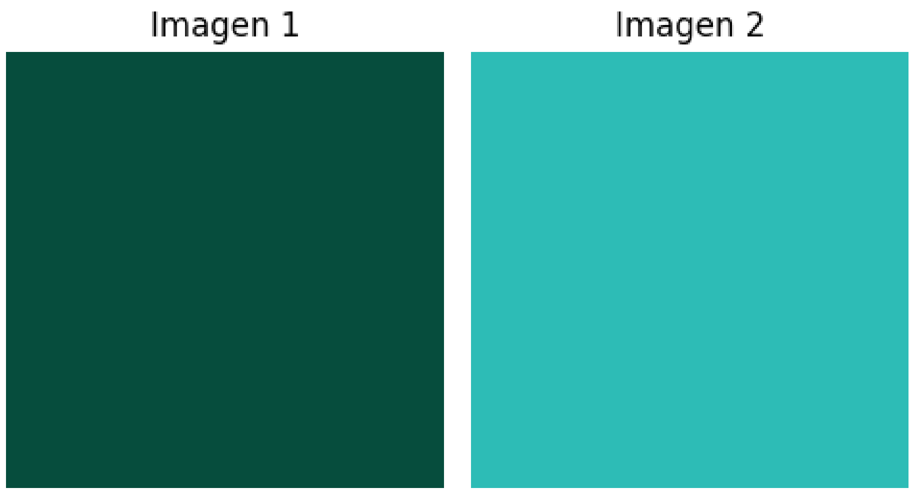
- 6. Reducción de las Loterías Compuestas (Axioma de Simplificación)

Este axioma garantiza que el agente no es irracionalmente afectado por la forma en que se presentan las probabilidades.

- **Principio:** Una lotería compuesta (una lotería donde algunos resultados son, a su vez, otras loterías) es equivalente a la lotería simple que se obtiene multiplicando las probabilidades y sumando los resultados finales. El agente debe ser indiferente entre la lotería compuesta y su equivalente simple.
- **Implicación:** El agente solo debe preocuparse por las **probabilidades finales** de los resultados finales, sin importar cuántos pasos o "rondas" haya en el proceso. Por ejemplo, una lotería con un 50% de probabilidad de ganar 100€ en el primer paso, y luego un 50% de probabilidad de ganar 200€ en el segundo paso, es simplemente equivalente a un 25% de probabilidad de ganar 200€ y el resto de probabilidades del primer paso. El agente no debe preferir la complejidad o la simplicidad por sí mismas.

1.2.1 ¿Qué color prefieren?

[ ]:
```
import requests
from PIL import Image
from io import BytesIO
import matplotlib.pyplot as plt

# URLs directas de Google Drive
url1 = "https://drive.google.com/uc?
 ↪export=view&id=1aofwvS7bUVmKS6MlTcpL2ZgVa9JuGJex"
url2 = "https://drive.google.com/uc?
 ↪export=view&id=1-htp7twx9y5-GpzMUXl9hk-X3kV-BVz6"

# Función para descargar y redimensionar imagen
def load_and_resize(url, size=(10, 10)):
    response = requests.get(url)
    img = Image.open(BytesIO(response.content))
    img = img.resize(size, Image.Resampling.LANCZOS)  # redimensionar a 150x150
    return img

# Cargar imágenes
img1 = load_and_resize(url1)
img2 = load_and_resize(url2)

# Crear figura con 2 columnas
fig, axes = plt.subplots(1, 2, figsize=(5, 5))

# Mostrar primera imagen
axes[0].imshow(img1)
axes[0].axis('off')
axes[0].set_title("Color 1")

# Mostrar segunda imagen
axes[1].imshow(img2)
axes[1].axis('off')
axes[1].set_title("Color 2")

plt.tight_layout()
plt.show()
```



Esta pregunta se relaciona con el axioma de Ordenamiento Alternativo o Completitud. Esto es debido a que un agente no puede no decidir entre una de las opciones, y por ello, siempre debe poder establecer una relación de preferencia o indiferencia entre las opciones presentadas.

Si tomo una decisión desde la subjetividad puedo elejir la imagen 2 por ejemplo

1.2.2 Entre sacar un 10 o un 7 en un parcial, ¿qué prefieren? ¿Y entre el 7 y desaprobar?

Esta pregunta se relaciona al axioma de transitividad.

- A: Sacar un 10
- B: Sacar un 7
- C: Desaprobar

Se prefiere a (A > B) y a (B > C) entonces a (A > C), es decir que son preferencias logicamente consistentes. En el caso de elegir entre un 10 y un 7, puedo querer elegir la nota mas alta, osea el 10, mientras que en el otro caso, donde debo elegir entre un 7 y un desaprobado, elijo el menos malo de ambos aunque un 7 no sea una nota tan óptima

1.2.3 Imagínense que hacen una apuesta de $100 jugando al cara o cruz, si pudieran trucar la moneda para que ustedes ganaran un porcentaje *p* de las veces, ¿en qué valor configurarían *p*?

Esta pregunta se relaciona con el axioma de Monotonicidad. En este caso puedo decidir que quiero que p sea 1 (o 100%), porque queremos asegurarnos de ganar, pero no es la única opción correcta. También es válido decir que queremos aumentar las posibilidades de ganar pero sin que sea tan obvio, entonces cambiar p a 0,65 por ejemplo tambien es una opción válida. Tambien podríamos querer perder, eb ese caso cambiamos las posibilidades para que sean menores a 0,5.

Personalmente creo que la mejor opcion es cambiar las posibilidades a mi favor sin que sea obvio, con el p=0,65 que mencioné anteriormente.

1.2.4 En una feria hay dos juegos: uno en el que el premio son $100 y otro en el que el premio es una piedra, ambos con una probabilidad p de ganar al juego. ¿A qué juego prefieren jugar?

Esta pregunta se relaciona con el axioma de Monotonicidad. Ya que la probabilidad p de ganar es la misma en ambos juegos, elijo el juego donde el premio sea el que me da mayor utilidad.

1.2.5 En un casino hay dos juegos y en ambos el premio son $1000. El primer juego se gana cuando se tira una moneda y sale cara y el segundo juego se gana cuando sale 6 en un dado. ¿A qué juego prefieren jugar?

Esta pregunta se relaciona con el axioma de Monotonicidad. Ya que el premio de ambos juegos me genera la misma utilidad, elijo en el que tenga mayor probabilidad de ganar. En el juego de la moneda tengo una probabilidad del 50% mientras que en el juego del Dado tengo un 16.67% de probabilidades de ganar. Por ende elijo el juego de la Moneda.

1.2.6 Supongamos que quiere postularse a una beca. Primero, existe un 60% de probabilidad de que le llamen para una entrevista. Si no le llaman, entonces hay un 20% de probabilidad de que le ofrezcan participar en un curso online ¿Cuál es la probabilidad total de cada evento (recibir la beca, recibir el curso o no recibir nada)?

Cálculo de Probabilidades de los Eventos Para calcular la probabilidad de cada evento, se debe considerar la dependencia secuencial de los resultados:

Datos Iniciales: * $P(\text{Entrevista}) = 60\% = 0.60$ * $P(\text{No Entrevista}) = 1 - 0.60 = 0.40$ * $P(\text{Curso} \mid \text{No Entrevista}) = 20\% = 0.20$

1. Recibir la Beca

Asumiendo que el evento más deseado (la beca) es el resultado directo de ser llamado a la entrevista:

$$\text{Probabilidad de Beca} = P(\text{Entrevista}) = \mathbf{0.60}\ (60\%)$$

2. Recibir el Curso Online

El curso solo se ofrece si **no** hay entrevista, y luego existe una probabilidad del 20%.

$$\text{Probabilidad de Curso} = P(\text{No Entrevista}) \times P(\text{Curso} \mid \text{No Entrevista})$$

$$\text{Probabilidad de Curso} = 0.40 \times 0.20 = \mathbf{0.08}\ (8\%)$$

3. Recibir Nada

Esto ocurre si no hay entrevista Y si no se ofrece el curso (que tiene un $1 - 0.20 = 0.80$ de probabilidad de ocurrir).

$$\text{Probabilidad de Nada} = P(\text{No Entrevista}) \times P(\text{No Curso} \mid \text{No Entrevista})$$

$$\text{Probabilidad de Nada} = 0.40 \times 0.80 = \mathbf{0.32}\ (32\%)$$

Evento	Probabilidad Total
Recibir la Beca	60% (0.60)
Recibir el Curso	8% (0.08)
Recibir Nada	32% (0.32)
TOTAL	100%

Axioma Relacionado: Reducción de las Loterías Compuestas

El proceso de combinar las probabilidades de las etapas secundarias (No Entrevista → Curso o Nada) en una única probabilidad de resultado final se rige por el **Axioma de Reducción de Loterías Compuestas**.

Explicación del Axioma

- **Principio:** Este axioma establece que un agente racional debe ser **indiferente** entre participar en una lotería con múltiples etapas (una lotería compuesta) y participar en una lotería simple que tenga las **mismas probabilidades finales** de los resultados definitivos.
- **Aplicación al Ejemplo:** El axioma nos dice que al agente solo le importa la probabilidad final de obtener la beca (60%), el curso (8%), o nada (32%), sin importar si la beca era el resultado de una entrevista y el curso era el resultado de una segunda tirada de dados. El agente no debe preferir la complejidad o la simpleza del proceso, solo el resultado final.

Por lo tanto, este axioma permite al agente reducir el escenario complejo y de múltiples pasos a una **lotería simple** con tres resultados finales y sus probabilidades combinadas.

1.3 Los boletos de una lotería cuestan un dólar. Hay dos juegos posibles con distintos premios: uno de 10 dólares con una probabilidad de uno entre 50 y otro de 1.000.000 dólares con una probabilidad de uno entre 2.000.000.

1.3.1 ¿Cuál es el valor monetario esperado del boleto de lotería?

1.3.2 Problema del boleto de lotería

Un boleto cuesta **1 dólar**. Hay dos juegos posibles:

1. Premio de **10 dólares** con probabilidad de 1/50.
2. Premio de 1.000.000 **dólares** con probabilidad de 1/2.000.000.

1.3.3 Cálculo del valor esperado

Juego 1 La probabilidad P_1 es:

$$P_1 = \frac{1}{50} = 0.02$$

El valor esperado monetario EM_1 es:

$$EM_1 = P_1 \times 10 = 0.20\,\text{dólares}$$

Juego 2 La probabilidad P_2 es:

$$P_2 = \frac{1}{2{,}000{,}000} = 0.0000005$$

El valor esperado monetario EM_2 es:

$$EM_2 = P_2 \times 1{,}000{,}000 = 0.50\,\text{dólares}$$

Valor esperado neto 1 El valor esperado neto EM_{neto} se calcula restando el costo del boleto:

$$EM_{\text{neto1}} = 0.20 - 1 = -0.80\,\text{dólares}$$

Valor esperado neto 2

$$EM_{\text{neto2}} = 0.50 - 1 = -0.50\,\text{dólares}$$

1.3.4 ¿Cuándo es razonable comprar un boleto?

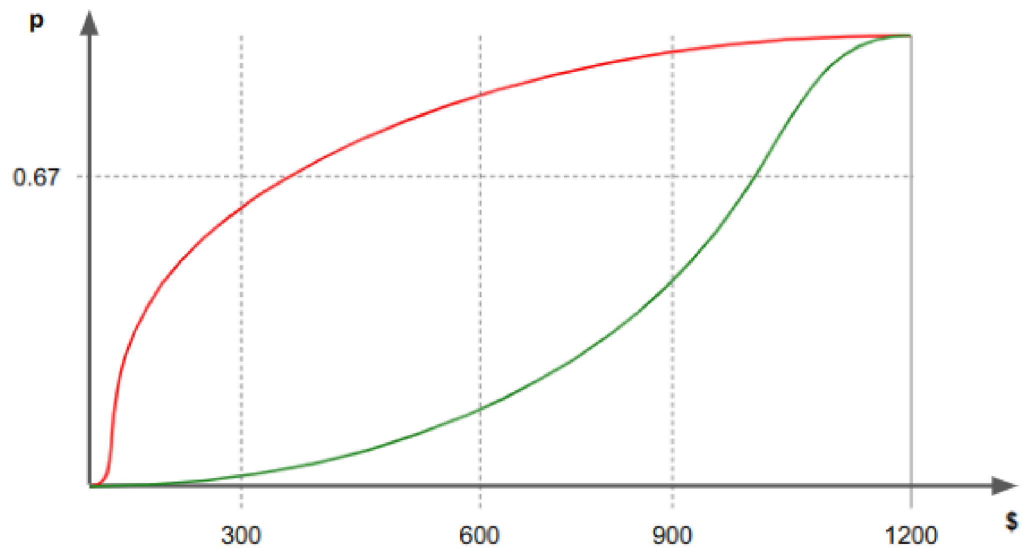
- Entre ambas 2 opciones, el segundo caso te hace perder menos dinero en sentido neto
- Por valor monetario esperado, **no es razonable comprar ninguno**.
- Sin embargo, algunas personas lo compran porque valoran la **posibilidad de ganar un premio grande** (función de utilidad $\neq$ valor esperado).

1.4 Hay que reparar una máquina averiada y el mecánico diagnostica que si la avería es leve la reparación costará \$300, pero si es grave costará \$1.200. La probabilidad de que la avería sea grave es 2/3. También se ofrece la alternativa de comprar una máquina usada por \$600. Qué decisión se tomará si:

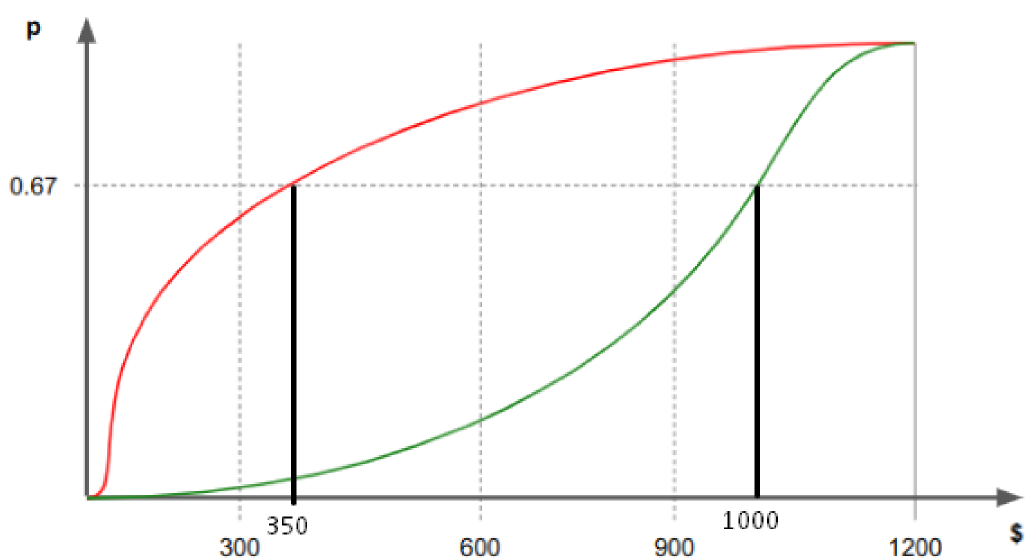
1.4.1 La función de utilidad fuese la mostrada por el agente rojo.

1.4.2 La función de utilidad fuese la mostrada por el agente verde.

[ ]:
```
url = "https://drive.google.com/uc?
 ↪export=view&id=1vNTlLArEhsxTOIvgs-NJ7NcE0B0JPJcw"
response = requests.get(url)
img = Image.open(BytesIO(response.content))

# Mostrar con figura más grande
plt.figure(figsize=(img.width / 80, img.height / 80))  # ajustá el divisor␣
 ↪según densidad deseada
plt.imshow(img)
plt.axis('off')
plt.show()
```



Del gráfico vemos



E(Reparar) = (1/3) · 300 + (2/3) · 1200 = 900

Para una probabilidad de 2/3 el agente rojo se arriesgaría hasta $350 y el agente verde se arriesgaría hasta $1000. Al comparar lo que arriesgarian con lo esperado de arreglo:

El agente rojo no se arriesgaría y compraria una maquina usada.

El agente verde si se arriesgaría y lo haría reparar.

---

2 Ejercicios de Implementación

2.1 En unos laboratorios de un hospital se está investigando sobre una sustancia para la curación de una determinada enfermedad. Dicha sustancia ha sido inyectada en varias cobayas enfermas para verificar sus efectos. Los resultados de las pruebas realizadas se sintetizan en la siguiente tabla:

Estado de la enfermedad	Concentración de la sustancia	Número de dosis	Condición física	Efecto
Incipiente	75	70	Fuerte	Curación
Incipiente	80	90	Fuerte	Defunción
Incipiente	85	85	Débil	Defunción
Incipiente	62	95	Débil	Defunción
Incipiente	79	70	Débil	Curación
Avanzado	72	90	Fuerte	Curación
Avanzado	83	78	Débil	Curación
Avanzado	64	66	Fuerte	Curación
Avanzado	81	75	Débil	Curación
Terminal	71	80	Fuerte	Defunción
Terminal	65	70	Fuerte	Defunción
Terminal	75	80	Débil	Curación
Terminal	68	80	Débil	Curación
Terminal	70	96	Débil	Curación

Determine las reglas que rigen las condiciones en las que se ha de administrar una sustancia e implemente un sistema experto en CLIPS que determine si un sujeto resultará curado o no.

$$I(p,n) = -\left(\frac{p}{p+n}\right)\log_2\left(\frac{p}{p+n}\right) - \left(\frac{n}{p+n}\right)\log_2\left(\frac{n}{p+n}\right)$$

¿Qué representa esta fórmula?

Esta es la fórmula de la **Entropía de Shannon**, o más específicamente, la **Entropía de la Información** aplicada en el contexto de **aprendizaje automático** (Machine Learning), como en los algoritmos de Árboles de Decisión (Decision Trees).

- $I(p,n)$: La entropía de un conjunto de datos.
- p: El número de ejemplos **positivos** (de una clase).
- n: El número de ejemplos **negativos** (de la otra clase).
- $\frac{p}{p+n}$: La **probabilidad** de que un ejemplo sea positivo.
- $\frac{n}{p+n}$: La **probabilidad** de que un ejemplo sea negativo.
- $\log_2$: El logaritmo en **base 2**, que indica que la entropía se mide en **bits**.

La entropía mide la **impureza** o la **incertidumbre** en un conjunto de datos. Un valor de entropía de **0** significa que el conjunto es perfectamente puro (todos son positivos o todos son negativos), y un valor de **1** (el máximo para la base 2) significa que el conjunto está perfectamente mezclado (igual número de positivos y negativos), lo que representa la máxima incertidumbre.

Usando esta formula calculamos la entropia inicial de las prubas:

$$I(9,5) = 0.94$$

Luego la Ganancia de los Estados:

$$G(E,I) = 0.04523$$

Luego la Ganancia de las Condiciones:

$$G(C2,I) = 0.000286$$

Luego Ganancia de las Concentraciones:

$$G(C1,I) = 0.000286$$

Luego la Ganancia de las Dosis:

$$G(C1,I) = 0.000286$$

Viendo las Ganancias, partimos de los Estados.

Estado de la enfermedad	Concentración de la sustancia	Número de dosis	Condición física	Efecto
Incipiente	75	70	Fuerte	Curación
Incipiente	80	90	Fuerte	Defunción
Incipiente	85	85	Débil	Defunción
Incipiente	62	95	Débil	Defunción
Incipiente	79	70	Débil	Curación

Aqui cambiamos la condicion tomando como positivo valores mayores o iguales a 70 y por ende si la Dosis es mayor a 70 se cura.

Estado de la enfermedad	Concentración de la sustancia	Número de dosis	Condición física	Efecto
Avanzado	72	90	Fuerte	Curación
Avanzado	83	78	Débil	Curación
Avanzado	64	66	Fuerte	Curación
Avanzado	81	75	Débil	Curación

Aqui vemos que si el estado es avanzado siempre se cura independientemente de los otros factores.

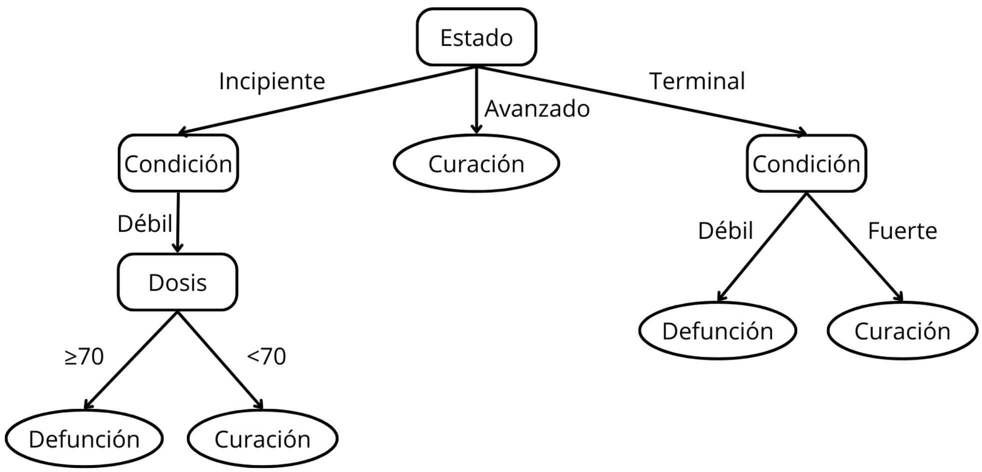
Estado de la enfermedad	Concentración de la sustancia	Número de dosis	Condición física	Efecto
Terminal	71	80	Fuerte	Defunción
Terminal	65	70	Fuerte	Defunción
Terminal	75	80	Débil	Curación
Terminal	68	80	Débil	Curación
Terminal	70	96	Débil	Curación

Aqui vemos que si la condicion es Debil, la cobaya se cura, sino no.

```
;CODIGO CLIPS

; -----------------------------------------------------------------------------
; 1. DEFINICIÓN DE ESTRUCTURAS
; -----------------------------------------------------------------------------

; Plantilla para almacenar los datos de la cobaya
(deftemplate cobaya
   (slot estado)
   (slot concentracion)
   (slot dosis)
   (slot condicion-fisica))

(deftemplate hay-resultado
   (slot resultado))

; Regla inicial para comenzar el proceso y solicitar los datos
(defrule inicio-del-programa
   =>
   (assert (resultado no))

   (printout t "Ingrese el estado de la enfermedad (Incipiente, Avanzado, Terminal): ")
   (assert (estado (read)))

   (printout t "Ingrese la concentracion de la sustancia (numero): ")
   (assert (concentracion (read)))

   (printout t "Ingrese el numero de dosis (numero): ")
   (assert (dosis (read)))

   (printout t "Ingrese la condicion fisica (Fuerte o Débil): ")
   (assert (condicion (read)))
)
```

```
; ------------------------------------------------------------------------------------
; 2. REGLAS DE CURACIÓN Y DEFUNCIÓN
; ------------------------------------------------------------------------------------

; estado ?e, concentracion ?c1, dosis ?d, condicion ?c2, resultado ?r

(defrule defuncion-incipiente
   (estado ?e)
   (dosis ?d)
   =>
   (if (and (eq ?e Incipiente) (>= ?d 70))
   then
   (printout t "Resultado: Defunción." crlf)
   (assert (resultado si))
   else
      (printout t "Resultado: Curación." crlf)
      (assert (resultado si))
   )
   )

(defrule curacion-avanzado
   (estado ?e)
   =>
   (if (eq ?e Avanzado)
   then
   (printout t "Resultado: Curación." crlf)
   (assert (resultado si))
   )
   )

(defrule curacion-terminal-debil
   (estado ?e)
   (condicion ?c2)
   =>
   (if (and (eq ?e Terminal) (eq ?c2 Debil))
   then
   (printout t "Resultado: Curación." crlf)
   (assert (resultado si))
   )
   )

(defrule defuncion-terminal-fuerte
   (estado ?e)
   (condicion ?c2)
   =>
   (if (and (eq ?e Terminal) (eq ?c2 Fuerte))
   then
   (printout t "Resultado: Defunción." crlf)
```

```
    (assert (resultado si))
    )
    )

(defrule defuncion-por-defecto
    (resultado ?r)
    =>
    (if (eq ?r no)
    then
    (printout t "No hay Resultado: Defunción por defecto." crlf)
    )
)
```



Bibliografía

Russell, S. & Norvig, P. (2004) *Inteligencia Artificial: Un Enfoque Moderno*. Pearson Educación S.A. (2a Ed.) Madrid, España

Poole, D. & Mackworth, A. (2023) *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press (3a Ed.) Vancouver, Canada