

# Hoja de Trabajo 1: Agentes y Entornos

## Inteligencia Artificial - CC3085

José Antonio Mérida Castejón

18 de enero de 2026

### Task 1: Análisis de Racionalidad y Métricas

*Imaginen que están diseñando un agente de IA para el sistema de semáforos de la Ciudad de Guatemala (zona alta de tráfico).*

1. *Propongan dos (2) métricas de desempeño distintas para este agente:*

- **Métrica A (Flujo vehicular):** Podemos empezar con dos opciones bastante *naive*, tales como:
  - **Velocidad promedio de carros cruzando:** Aquí seguimos una línea lógica dónde mientras más tráfico hay en la ciudad, menor es la velocidad de los carros. Sin embargo, encontramos algunos problemas cuando nos ponemos a pensar sobre la prioridad de las vías principales. Si el agente busca maximizar la velocidad promedio, únicamente deben de no frenar (o desacelerar) los carros en las vías principales para aumentar la métrica. Esto mientras los carros en vías auxiliares no tienen prioridad alguna, ya que si se formó alguna cola estos carros irán despacio y no maximizan la métrica.
  - **Cantidad de carros cruzando:** Aquí seguimos una línea de pensamiento bastante lógica, similar al inciso anterior. Mientras más carros cruzan intersecciones, significa que tenemos menos tráfico (o un mayor flujo vehicular). El problema con esta métrica también es parecido, el modelo no tiene incentiva alguna para otorgar paso a una calle auxiliar con  $n$  carros estancados si tenemos  $2n$  carros transitando por una vía principal.

En resumen, las dos métricas inicialmente propuestas favorecen el volumen bruto e ignoran completamente la equidad dentro del sistema. Esto nos lleva a buscar una métrica más justa para las calles con un mejor flujo de tráfico, por lo cual debemos buscar *normalizar* estas medidas de cierta manera. Dando como resultado:

- **Ocupación de carril:** En lugar de contar la cantidad de autos cruzando, busquemos medir que porcentaje del tramo regulado por el semáforo se encuentra ocupado. Aquí logramos evitar el problema dónde tener 25 carros en una pequeña vía auxiliar es completamente diferente a tener 25 carros en un bulevar. Para implementar la métrica dentro de nuestro modelo, debemos buscar un punto crítico (por ejemplo, 80 % o 90 %) basándonos en observaciones de la vida real para determinar cuando se

forma un *gridlock*. Buscando penalizar de manera exponencial los estancamientos, podemos proponer la siguiente ecuación:

$$C(O) = \frac{1}{(O_{crit} - O)^k} \quad (1)$$

Dónde:

- $O$ : Representa la *ocupación actual* del carril
- $O_{crit}$ : Es el *punto crítico de saturación*
- $k$ : Actúa como un *parámetro de sensibilidad*
- $C(O)$ : Es la *función de costo* resultante.

Adicionalmente, debemos tomar en cuenta que los parámetros de esta función costo siempre deben basarse en datos de la vida real. Por ejemplo, utilizar un  $k$  alto o  $O_{crit}$  bajo sería totalmente contraproducente en una zona de alto tráfico ya que los deadlocks son inevitables. Sin embargo, la mayor crítica es que esta métrica no se podría utilizar por si sola. Podemos darnos cuenta que la manera más eficiente de minimizar el costo es simplemente no dejar ingresar ningún vehículo al sistema. Si no tenemos ningún carro dentro de la zona, el costo es 0 y el agente ha cumplido exitosamente su misión.

- **Throughput:** Aquí buscamos darle una recompensa al modelo, de manera que no se encuentre en estado de parálisis con la finalidad de evitar penalizaciones. Podemos iniciar con la métrica de número de carros cruzando, pero *normalizándola* respecto al tiempo de luz verde. Definiendo esta métrica por medio de la siguiente ecuación:

$$T = \frac{N}{t_g} \quad (2)$$

Dónde:

- $T$ : Representa el *throughput*
- $N$ : Representa la *cantidad de carros que cruzaron en el intervalo de luz verde*
- $t_g$ : Representa la *longitud del intervalo de luz verde*

Adicionalmente, podríamos asignar pesos diferentes a áreas con mayor transito para evitar sesgos dentro del sistema. Sin embargo, el *fine tuning* de los parámetros queda fuera del alcance de esta hoja de trabajo. También debemos tomar en cuenta que buscamos un balance entre ambas métricas descritas, evitando los estancamientos y priorizando una cantidad alta de autos fluyendo. Esto nos lleva a la métrica final.

- **Índice de Fluidez:** En esta métrica que nombramos Índice de Fluidez (*IDF*) buscamos tomar en cuenta el *Throughput* y *Ocupación de Carril* y para medir el desempeño de un sistema de semáforos inteligentes. Tomando las ecuaciones (1) y (2), podemos construir la siguiente ecuación:

$$IDF = w_1 T - w_2 C(O) \quad (3)$$

Dónde:

- *IDF*: Representa el *índice de fluidez* del sistema en una intersección específica.
- $w_1$ : Representa el *peso de Throughput* del sistema.
- $T$ : Representa el *Throughput* del sistema en una intersección específica.
- $w_2$ : Representa el *peso de la Ocupación* del sistema.
- $C(O)$ : Representa la *Ocupación* de las vías en una intersección específica.

Esta métrica nos permite medir el desempeño en intersecciones específicas, al igual que a lo largo de toda la red en una zona específica. En un mundo ideal, el agente es capaz de calcular esta métrica de rendimiento y tomar las decisiones que la minimicen en tiempo real (o en cada ciclo). Sin embargo, también es importante tomar en cuenta que cuenta con diferentes parámetros que deben ser escogidos por medio de decisiones informadas. Antes de querer implementar un sistema, se debería de obtener información sobre el tráfico en Guatemala (más específicamente, en la zona de implementación) y diferentes *edge cases* para asegurarnos que el agente tome decisiones beneficiosas para el tráfico.

- **Métrica B (Seguridad peatonal):** Similar al inciso anterior, primero vamos a indagar sobre algunas métricas *naive* e ir iterando para obtener una métrica más adecuada.
  - **Conteo de Colisiones:** Esta primera idea parece ser un buen acercamiento, sin embargo podemos darnos cuenta rápidamente que no es apta para un sistema como este. Realmente, esta es una métrica reactiva y un agente necesita que ocurra un accidente para recibir una penalización. Adicionalmente, estos eventos deberían ser pocos y las condiciones que los dan pueden no ser consistentes como para que el agente realmente *aprenda*.
  - **Tiempo de Respuesta Botón Pevalonal:** Dentro del sistema de tráfico, podemos asumir que una persona es más propensa a realizar alguna acción peligrosa (por ejemplo, cruzar la calle sin tener el paso) mientras más tiempo pasa esperando. Esto implica que un botón de paso peatonal con un tiempo de respuesta menor resulta en una mayor seguridad peatonal. El problema más evidente aquí, es que es completamente incompatible el manejo de fluidez vehicular con un botón peatonal *óptimo* que otorgue paso cuando sea posible. Adicionalmente, el cambio constante de estados de luz verde / roja puede generar inseguridad para tanto peatones como vehículos.

Adicionalmente, también debemos de tomar en cuenta que un sistema de inteligencia artificial por defecto no toma decisiones similares a las de un ser humano. Es decir, todo lo que *sabe* se le debe enseñar. Esto incluye razonamientos triviales, cómo lo puede ser “No coloquemos el semáforo en verde si acabamos de colocarlo en rojo para el cruce de este peatón”. Esto nos lleva a definir las siguientes dos métricas auxiliares:

- **Factor de Suficiencia:** Esta métrica mide si el tiempo de verde es físicamente suficiente para que un grupo de peatones termine de cruzar. Aquí la métrica es bastante simple, dónde podemos definir la métrica por medio de la siguiente ecuación:

$$S = t_{verde} - t_{despeje} \quad (4)$$

Dónde

- $S$ : Es el *factor de suficiencia*

- $t_{verde}$ : Es el *tiempo que la luz peatonal estuvo en verde*
- $t_{despeje}$ : Es el *tiempo para que se despejara la calle*

Aquí se podría argumentar que lo mejor sería un sistema dónde no se permitiera el cambio mientras un peatón este cruzando, sin embargo una persona a media calle podría provocar un estancamiento total del tráfico al siempre tener el paso.

- **Factor de Espera Crítica:** De manera análoga a la métrica de *Ocupación de carril*, aquí vamos a modelar el *Factor de Desesperación* como una función que crece exponencialmente conforme el tiempo de espera se acerca a un punto crítico de paciencia del peatón. Definimos la ecuación de la siguiente manera:

$$E(t_w) = \frac{1}{(t_{crit} - t_w)^k} \quad (5)$$

Dónde:

- $t_w$ : Representa el *tiempo de espera actual* del peatón (desde el accionamiento del botón).
- $t_{crit}$ : Es el *tiempo crítico de paciencia*, donde el peatón decide cruzar ilegalmente.
- $k$ : Actúa como un *parámetro de sensibilidad* al tiempo.
- $E(t_w)$ : Es la *función de costo por desesperación* resultante.

Aquí debemos de ser cuidadosos al elegir los parámetros, ya que deberían de reflejar la vida real de la manera más cercana posible. De hecho, antes de implementar esta función la primera recomendación sería verificar los supuestos sobre los cuales trabaja el modelo con información del mundo real. Por último, tenemos una métrica que busca combinar los factores anteriores.

- **Índice de Seguridad Pevalonal:** El *Índice de Seguridad Pevalonal* o *ISP*, busca minimizar los tiempos de espera peatonales al igual que asegurar tiempos correctos de despeje de vía. Esta métrica penaliza al modelo si el tiempo de verde otorgado es suficiente para que los peatones completen el cruce, y busca penalizar tiempos de espera cercanos al punto crítico que incentiva al peatón a cruzar sin paso. Tomando las ecuaciones (4) y (5), lo modelamos por medio de la siguiente ecuación:

$$ISP = w_3S - w_4E(t_w) \quad (6)$$

Dónde:

- $ISP$ : Representa el *índice de seguridad peatonal* en el cruce.
- $w_3$ : Es el peso asignado al *Factor de Suficiencia*.
- $S$ : Representa la *Suficiencia de Despeje* ( $t_{verde} - t_{despeje}$ ).
- $w_4$ : Es el peso asignado al *Factor de Espera Crítica*.
- $E(t_w)$ : Es la función de costo por *Desesperación* definida anteriormente.

Como reflexión o conclusión de este inciso, implementar un sistema inteligente de control de tráfico en Guatemala sería un proyecto de complejidad alta. Algo básico como definir métricas de desempeño puede volverse complicado, teniendo que tomar en cuenta diferentes factores como:

- Situaciones de tráfico específicas a la zona.
- Comportamiento de las métricas en *edge cases*.
- Viabilidad de implementación de sensores.
- Balance en priorización de diferentes factores, cómo seguridad peatonal y flujo vehicular.

Adicionalmente, los supuestos detallados en cada métrica propuesta definitivamente deben ser validados.

2. *Describe un escenario específico dónde en agente actuaría de manera “racional” bajo la métrica A, pero esta acción sería considerada un desastre bajo la Métrica B.*

A pesar de haber iterado y analizado extensivamente las métricas, considero que el sistema debe de ser probado exhaustivamente para poder llegar a un *punto de equilibrio* dónde no le de prioridad a una única métrica sobre la otra. Ya que en la mayoría de los casos, un desbalance entre peatones y vehículos puede resultar en desastres absolutos.

Por ejemplo: Bajo la métrica A, se llega a un punto crítico  $O_{crit}$  de atascamiento que aumenta exponencialmente. Siendo un día lluvioso, fin de semana y quincena en una intersección con una vía principal sumamente transcurrida cercana a  $O_{crit}$ . La acción más racional del agente, es intentar minimizar la penalización dada por  $C(O)$  y dar paso a la mayor cantidad de carros posible por esta vía principal y maximizar si Índice de Fluidez ( $IDF$ ). Esto también aumentaría el factor *Throughput* ( $T$ ), creando una situación dónde el comportamiento obvio es quitarle prioridad a los peatones. Mientras tanto, los peatones se encuentran esperando con un *Factor de Espera* ( $E(t_w)$ ) crítico, aumentando exponencialmente mientras el tiempo sigue transcurriendo. Adicionalmente, digamos que el agente decide por fin que los tiempos de espera peatonales han sido suficientes. Se ha dado paso peatonal, pero mientras los peatones están cruzando la calle, la vía principal se atascó nuevamente y se ha llegado a  $O_{crit}$ . Aquí se puede llegar a cortar la vía peatonal, generando aún otra penalización, esta vez del *Factor de Suficiencia* ( $S$ ). En este caso, se tomó la acción óptima en cuanto a la métrica A, pero la métrica B se vio penalizada fuertemente tomando en cuenta ambos factores.

3. *Basándose en la diapositiva de Incertidumbre, explique por qué este entorno de tráfico nunca podría ser Completamente Observable y cómo la “Limitante en sensores” afecta la racionalidad de su agente*

El entorno del tráfico en Guatemala siempre será *Parcialmente Observable*, ya que un agente jamás tendrá acceso total a un estado global del sistema en tiempo real. En el caso de este modelo, buscamos tener acceso a la mayor cantidad de información posible pero siempre nos veremos limitados. Por ejemplo, podemos identificar los siguientes sensores y sus limitantes:

- Modelos de Visión por Computadora: Si un peatón o moto es tapado por un vehículo pesado, somos incapaces de percibir el estado del sistema.
- Sensores de Ocupación: Los sensores están físicamente limitados a su posición, la ocupación de carril por ejemplo, jamás podrá ser medida en un intervalo continuo de manera perfecta.

Tomando también en cuenta, que tenemos algunos otros tipos de sensores que simplemente no serían viables en esta situación. Por ejemplo, si tomáramos modelos más pesados o complejos como de mapeo 3D del entorno, el procesamiento de estos datos tendría un costo computacional demasiado alto para el modelo.

Esta limitante en los sensores básicamente redefine la racionalidad del agente. Este no es juzgado en función del éxito absoluto (o verdadero) en el mundo real, si no en función de las percepciones disponibles. Gracias a estas limitaciones, el agente opera bajo una racionalidad limitada. Dónde, por ejemplo, si sus sensores no logran detectar a un peatón cruzando la calle, este peatón realmente jamás existió y no pudo existir alguna decisión o penalización asociada. El modelo sigue tomando decisiones óptimas y racionales, pero al hacerlo sobre datos incompletos no paran reflejando la decisión óptima real.

## Task 2: PEAS y Entornos

*Consideren un Robot Autónomo de Limpieza de Páneles Solares instalado en una granja solar en el desierto de Zacapa. El robot se mueve sobre rieles instalados en los páneles, tiene una cámara para detectar suciedad/daños, un brazo con cepillo y agua, y una conexión a internet para recibir reportes del clima. En base a esto, responda lo siguiente:*

1. *Complete la siguiente descripción del agente:*

**P (Performance/Desempeño):** *(Mencione al menos 3 indicadores cuantificables)*

- a) Calidad / Eficiencia de Limpieza:** Esta métrica busca medir que tan bien realiza la limpieza de cada panel individual el agente. Aquí prácticamente contamos con dos opciones para realizar las mediciones. Primero, podemos utilizar la cámara para identificar el % de área con suciedad y ver cuanto se logró remover. O segundo, podemos comparar la cantidad de energía generada por el panel antes y después de la limpieza. Personalmente, me gusta más la segunda opción ya que movemos la capa de abstracción un poco más arriba, dónde permitimos que el agente tome decisiones sobre las capas que nos interesan un poco menos (*optimicidad de limpieza*) y se enfoque en la generación energética real. Cómo formula de medición, podemos tomar la diferencia en producción de energía:

$$\Delta P = P_{post} - P_{pre}$$

- b) Tiempo de Limpieza:** Esta métrica busca medir la eficiencia en cuánto a tiempo de limpieza por panel individual. Se puede medir de dos maneras, primero el tiempo promedio de limpieza en cada panel dentro de un periodo de tiempo. O segundo, la cantidad de páneles limpiados dentro de cierto periodo de tiempo. La diferencia yace en que en uno se está tomando en cuenta el periodo de desplazamiento, mientras que en otro no. Por lo cual, decidiría ir por la primera en caso que tuviera que implementar el sistema:

$$\bar{t} = \frac{1}{n} \sum_{i=1}^n (t_{final,i} - t_{start,i})$$

- c) Consumo de Agua:** Esta métrica busca optimizar el uso de agua parte del robot. Esta métrica es especialmente clave, considerando que nos encontramos en el desierto de Zacapa y el agua puede ser un recurso relativamente escaso. Buscamos optimizar el recurso utilizado por panel para no penalizar al robot por limpiar más páneles.

$$\bar{v} = \frac{V_{total}}{n}$$

- d) **Numero de Paneles:** Esta métrica busca aumentar (o por lo menos, generar una incentivo) la cantidad de paneles limpiados por el robot en un determinado periodo de tiempo. Si nos regimos únicamente por las métricas anteriores, el robot ideal realiza una limpieza súmamente eficiente y deja de operar. Ya que se continua realizando operaciones, puede que las métricas bajen.

**E (Environment/Entorno):** (*¿Qué rodea al agente y con qué interactúa?*)

El agente opera dentro de una granja solar en el desierto de Zacapa. Físicamente, lo rodean:

- **Infraestructura:** Arreglos de paneles solares, posiblemente arreglados dentro de un grid.
- **Factores Climáticos:** Condiciones climáticas extremas, como calor intenso o radiación UV.
- **Obstáculos Físicos:** Posiblemente arena o piedras pequeñas dentro de los rieles.
- **Entorno Digital:** Una conexión a la red la consulta de APIs meteorológicas.
- **Suciedad y Daños:** Presencia de polvo / tierra en los paneles solares, al igual que daños físicos que no necesariamente se deben reparar.

Mientras que principalmente interactúa con los paneles y suciedad en si, también es importante tomar en cuenta que la *percepción* del robot cubre cualquier información que recaude por medio de sensores o esté presente en el ambiente.

**A (Actuators/Actuadores):** (*Mecanismos para afectar el entorno*)

- **Motores de Movimiento:** Permiten el desplazamiento del robot sobre el sistema de rieles.
- **Motores / Controles del Brazo:** Controlan el brazo robótico, al igual que sus movimientos sobre la superficie del panel.
- **Bomba de Agua:** Se encargan de la dosificación y esparsión de agua.

**S (Sensors/Sensores):** (*Asumiendo que no hace ningún preprocesamiento, ¿qué métrica usaría para medir el desempeño de su modelo?*)

Realmente, realizar mediciones sin algún tipo de preprocesamiento se me hace algo bastante difícil. Algunas de mis ideas iniciales principalmente se basan en modelos que ya cuentan con cierto entrenamiento, o tienen capacidades básicas. Si no contamos con preprocesamiento, voy a asumir que estamos entrenando a un modelo súmamente básico bajo la condición que su *task* es dejar un panel lo más limpio posible.

El *approach* inicial podría ser comparar la imagen con la de un panel limpio, sin embargo los ángulos o iluminación diferente hacen que esta métrica no sea muy exitosa. Otra idea también podría ser calcular el promedio de intensidad de los píxeles (si es más oscuro, está más sucio), pero esto fallaría si encontramos sombras de nubes por ejemplo.

Tomando en cuenta que los paneles solares tienen un *grids* bastante definidos (y pequeños) en su diseño, podemos calcular la diferencia de intensidad entre píxeles adyacentes. Esto nos ayuda a definir si el panel está limpio, y por ende si el modelo ha tenido un buen desempeño en su tarea de limpiar un único panel solar.

- **Panel Limpio:** Aquí tenemos bordes bastante definidos entre los grids, lo que hace que los cambios de intensidad entre píxeles adyacentes sean más bruscos.
- **Panel Sucio:** El polvo y otros sedimentos suavizan las transiciones, esto hace que los píxeles vecinos se parezcan más entre si. Resultando en una medida más baja.

En conclusión, el no realizar preprocesamiento de datos es una limitante bastante significativa. Sin embargo, utilizando los cambios de intensidad in píxeles adyacentes podemos obtener una métrica para estimar el desempeño del modelo.

2. *Clasifique el entorno de este robot según las 4 dimensiones vistas en clase. Debe justificar cada elección (una elección sin justificación tiene valor de 0pts).*

a) *¿Completamente o Parcialmente observable?*

**Parcialmente Observable.** La percepción del entorno se ve limitada por la posición de robot, al igual que la dirección a la que está apuntado la cámara. No se cuenta con manera alguna de conocer el estado del panel solar 30m hacia la izquierda, por lo tanto el estado global no es accesible.

b) *¿Determinístico o Estocástico?*

**Estocástico.** El estado siguiente del entorno no se determina únicamente por las acciones del robot, al operar en un ambiente abierto existen variables fuera del control del agente. Si fuera determinista, podríamos saber el estado en el que se encontrarán los paneles solares durante los siguientes 6 meses basados en las acciones que toma el robot. Claramente, esto no es posible dados factores como clima, fallas mecánicas en el entorno, etc.

c) *¿Discreto o Continuo?*

**Continuo.** El sistema no cuenta con un conjunto finito de representaciones. De hecho, hasta el estado de un único panel solar tomando en cuenta suciedad y daños sigue sin tener un conjunto finito de representaciones. Agregando las diferentes variables que encontramos en el entorno, también se deben de tomar en cuenta obstáculos o factores climáticos que evidentemente tampoco tienen un conjunto finito de representaciones. Por lo tanto, en entorno es continuo.

d) *¿Benigno o Adverso?*

**Benigno.** En este caso, no tenemos un oponente o agente competidor que esté buscando jugar en nuestra contra. A pesar de estar en un ambiente hostil, tenemos factores físicos naturales y no *planes* de un enemigo. Dado que el ambiente no está buscando minimizar nuestra métrica de desempeño, el entorno se clasifica como benigno.



## Task 3: Modelado

- a) *Describa abstractamente cómo representaría el estado de un panel solar en una estructura de datos*

**Una matriz.** Podemos tomar el panel solar, y dividirlo dentro de diferentes celdas de un grid. Donde cada celda del grid tiene una densidad de suciedad que debe ser removida, en un intervalo de  $[0, 1]$  donde 0 es perfectamente limpio y 1 totalmente sucio. Esta estructura es ideal, ya que mapea la superficie física a una tabla de datos que el robot puede procesar por coordenadas.

- b) *Si el robot tiene un modelo del panel, ¿qué significa “inferir” la siguiente acción? Relacionarlo con:  $f(\text{estado}) \rightarrow \text{accion}$*

Inferir es el cálculo u operación que realiza el robot para transformar la información percibida (entrada) a una decisión concreta.

En la función  $f(s) = a$ , el término “inferir” se refiere a la evaluación lógica donde:

- **Input ( $s$ ):** Es el estado actual representado por la matriz de suciedad.
- **Función ( $f$ ):** Es el modelo o “lógica” que el robot ya tiene cargada.
- **Output ( $a$ ):** Es la acción resultante

En términos simples, inferir es pasar del “qué está pasando” al “qué voy a hacer” mediante el resultado de esa función.

- c) *¿Qué parámetro del modelo debería ajustarse si el robot nota que, tras limpiar un panel muy sucio con poca agua, el panel sigue sucio?*

Se deben ajustar los parámetros del modelo que vinculan la densidad de suciedad con el volumen de agua necesario (como lo podrían ser los de control de bomba de agua). Tomando como ejemplo nuestras métricas de desempeño, podemos decir que se ha dado demasiada prioridad a una reducción en utilización de agua. En este caso, podríamos tomar otra de nuestras métricas (calidad de limpieza) y darle una prioridad más alta, asegurándonos que el bot logre ajustar los parámetros de flujo de agua correctamente. Otra manera más temporal de arreglarlo, podría ser cambiando las entradas para normalizar ligeramente más altos los niveles de la matriz a valores ligeramente más altos. De esta manera estaríamos utilizando más agua al *engañar* al bot y hacerlo pensar que hay más suciedad. Sin embargo, esto último debería de tomarse como último recurso.

## Task 4: Práctica - Agente Reflejo Simple

El código fue entregado dentro de Canvas como `main.py`, adicionalmente se encuentra disponible en este [repositorio de GitHub](#).