Trabajo de Fin de Máster

Del Potencial al Grafo: Estudio de Estrategias de Aprendizaje por Refuerzo para Navegación en Entornos Parcialmente Observables

Antón Carlos Vázquez Martínez, Universidad Alfonso X el Sabio (UAX)

Universidad Alfonso X el Sabio

Agosto 2025

Resumen

Este Trabajo de Fin de Máster estudia la navegación de un agente único en un entorno de habitaciones interconectadas por pasillos estrechos, un escenario representativo de problemas de exploración y recompensas escasas. Se implementaron tres enfoques progresivos: (1) observaciones globales absolutas, (2) observaciones globales con shaping de recompensas, y (3) observaciones locales apoyadas en un grafo de propagación de señales y shaping por récords de aproximación.

El análisis muestra que las observaciones globales sin estructura no inducen políticas útiles, mientras que el shaping basado en potenciales resulta frágil ante dinámicas de aparición y desaparición de objetivos. En contraste, el enfoque local con soporte topológico permitió un aprendizaje rápido, estable y robusto, guiando la atención del agente y evitando los exploits detectados previamente.

Los resultados confirman que la calidad de la navegación depende más de la estructura de las observaciones y de la definición del shaping que de la cantidad de información disponible. Además, la formulación propuesta es escalable a escenarios multiagente, donde aparecen fenómenos de coordinación y competencia de interés académico e industrial.

Índice

Introducción

Contexto y motivación

La navegación autónoma en entornos complejos constituye un reto central dentro del aprendizaje por refuerzo (RL) y la robótica móvil. A diferencia de tareas con recompensas densas y observaciones simplificadas, los entornos de **habitaciones conectadas por pasillos estrechos** presentan **recompensas escasas** y **cuellos de botella** que dificultan tanto la exploración como la generalización. Estos escenarios, a pesar de su simplicidad relativa, son análogos a problemas reales de logística o de robots de servicio en instalaciones industriales: un agente que debe **desplazarse por una zona fija de trabajo** para acudir a distintos objetivos o tareas.

En la industria, esta necesidad aparece en casos como la reposición en almacenes, el transporte de materiales o la limpieza en instalaciones, donde los robots deben desenvolverse en layouts fijos pero complejos. Los sistemas comerciales (ROS2 Navigation2, Nav2) recurren a arquitecturas modulares con planificación topológica y control local. Sin embargo, los enfoques basados en RL ofrecen la posibilidad de **aprender políticas adaptativas** que se ajusten a dinámicas cambiantes, obstáculos inesperados o reconfiguraciones del entorno.

Estado del arte

Diversas líneas de investigación han intentado superar los retos de exploración y navegación:

- **Shaping de recompensas.** Ng et al. [1] establecieron las bases del *potential-based reward shaping*, garantizando la invariancia de política al añadir funciones potenciales estacionarias. No obstante, se ha observado que en entornos no estacionarios (con spawn/despawn de objetivos) o con geometrías restrictivas, el shaping puede inducir **exploits** o **mínimos locales** (Khatib [2]).
- **Navegación topológica.** En lugar de depender de coordenadas globales, métodos recientes han mostrado la eficacia de representar el entorno como un **grafo de nodos y conexiones**, sobre el cual una política global selecciona metas intermedias mientras un controlador local ejecuta los movimientos [3, 4]. Además, la memoria topológica en grafo permite persistir landmarks y relaciones espaciales a largo plazo [5].
- **Exploración activa.** Extensiones como Active Neural SLAM [6] integran percepción y planificación en un marco jerárquico, facilitando que el agente aprenda no solo a seguir objetivos, sino también a descubrir nuevas regiones del entorno de forma eficiente.

- **Tendencias recientes.** Benchmarks como MiniGrid MultiRoom [7] confirman que entornos de habitaciones y pasillos son un marco adecuado para estudiar estos retos de manera controlada. Revisiones como la de Sun et al. [8] en **Object Goal Navigation** clasifica los trabajos existentes sobre la Navegación por Objetivos (ObjectNav) en tres categorías principales:
  - **Métodos "end-to-end"**: Estos métodos mapean directamente las observaciones del entorno a las acciones del agente. El artículo subdivide esta categoría en dos enfoques principales:
    - **Representación Visual**: Se centra en extraer información útil de las observaciones para mejorar la comprensión del entorno por parte del agente.
    - **Aprendizaje de Políticas**: Aborda los problemas de generalización deficiente, recompensas escasas y la ineficiencia de las muestras en el aprendizaje.
  - **Métodos Modulares**: Estos métodos se componen de varios módulos, incluyendo uno de mapeo, uno de políticas y uno de planificación de rutas. El artículo también los divide en subcategorías :
    - **Mapa de cuadrícula sin predicción**
    - **Mapa de cuadrícula con predicción**
    - **Representación de mapa basada en gráficos**
  - **Métodos "Zero-shot"**: Utilizan el aprendizaje "zero-shot" para la navegación, lo que permite al agente encontrar objetos que no ha visto previamente durante el entrenamiento. Dentro de esta categoría, el artículo distingue entre:
    - **Configuración "Zero-shot"**
    - **Configuración de vocabulario abierto**

Aportación de este trabajo

En este proyecto se estudia la navegación de un **agente único** en un entorno de varias habitaciones interconectadas por pasillos, con recolectables que aparecen y expiran de forma autónoma. El trabajo explora tres enfoques progresivos:

1. **Observaciones globales** (coordenadas globales normalizadas) con recompensas básicas.
2. **Observaciones globales + shaping**: bonus de exploración espacial y función potencial basada en la distancia al objetivo.
3. **Observaciones locales + grafo de señales**: raycast en y propagación de señales desde los recolectables a través de un grafo (coordenadas locales polares), con shaping por récords de aproximación. Capa extra de dificultad añadida con **obstáculos dinámicos**.

El objetivo es analizar las limitaciones de las aproximaciones ingenuas (1 y 2) y demostrar que un enfoque local con soporte topológico (3) permite superar dificultades de navegación y guiar la atención de manera más eficaz.

Proyección a entornos multiagente

Aunque este trabajo se centra en el entrenamiento de un **agente único**, el diseño del entorno y de las señales de recompensa está concebido para ser **escalable al caso multiagente**. En escenarios con varios robots que comparten un mismo espacio con pasillos estrechos, aparecen de forma natural fenómenos como:

- **Conflictos en cuellos de botella**, donde los agentes deben ceder el paso o coordinarse para evitar bloqueos.
- **Asignación de tareas distribuidas**, en la que varios agentes deben decidir cómo repartirse los recolectables o zonas a explorar.
- **Gestión de información parcial**, ya que cada agente dispone únicamente de observaciones locales y puede necesitar compartir o inferir señales globales a través de comunicación explícita o implícita.

La formulación mediante **grafo de señales** y **shaping por récords** ofrece un marco flexible para este salto: los nodos del grafo pueden actuar como puntos de coordinación o encuentro, mientras que el shaping por récords evita castigos locales que podrían complicarse aún más en presencia de múltiples agentes.

De este modo, el trabajo no solo aporta un análisis sobre cómo superar los retos de navegación en entornos de habitaciones y pasillos para un agente único, sino que también sienta las bases para estudiar fenómenos de cooperación, competición y resolución de dilemas sociales en entornos multiagente, de interés tanto académico como industrial.

Material disponible

El material completo de este Trabajo de Fin de Máster (TFM), incluyendo código, documentación y experimentos, está disponible públicamente en el siguiente repositorio de GitHub:

https://github.com/AntonCVM/TFMGitRepo

Para probar el entorno tanto manualmente como con el modelo entrenado según el escenario tercero ejecutar UnityEnvironment.exe de la carpeta build. Indicaciones:

- AWSD: Mover el agente manualmente.
- Q: Alterna la visualización del grafo de señales.
- E: Alterna la visualización de observaciones del agente en el grafo de señales.
- F: Cambia entre control manual y control por modelo de IA preentrenado.
- Tab: Cambia el ángulo de la cámara.
- º (BackQuote): Vista general del área.
- 1: Vista sobre el agente principal.
- 2, 3, 4: Vista sobre otros agentes (si están activos).

Metodología

Descripción del entorno

El entorno experimental se diseñó con el objetivo de reproducir un escenario simple pero representativo de los problemas de navegación con recompensas escasas. Consta de **cuatro habitaciones principales**, dispuestas en torno a un eje central y conectadas mediante **pasillos estrechos en forma de cruz (+)**. Esta configuración obliga al agente a **atravesar cuellos de botella** para desplazarse entre habitaciones, lo que dificulta la exploración aleatoria y resalta la necesidad de guías estructuradas.

Implementación de los objetivos (recolectables)

Los objetos recolectables actúan como fuente principal de recompensa. Se implementaron con un **tiempo de vida limitado**, decreciente hasta desaparecer, de modo que el agente debe aprender a priorizar su búsqueda. La visualización incorpora un indicador de tiempo (altura), lo que permite un feedback intuitivo en simulación.

Además, los recolectables pueden estar **asociados a un agente específico** o ser genéricos, lo que en trabajos futuros permitirá extender el análisis a escenarios competitivos multiagente. En este trabajo se emplearon recolectables genéricos (con menor valor) y específicos por agente (con mayor valor). El **respawn controlado** asegura un flujo dinámico de recompensas, evitando memorizar posiciones fijas.

Dinámica temporal del entorno

El entorno está gobernado por **ciclos y subciclos** que determinan qué tipo de recolectables aparecen en cada fase. Esto introduce una variación temporal que simula cambios de disponibilidad de recursos, aumentando la complejidad del problema. En la configuración final se emplearon **cuatro subciclos**, alternando fases con abundancia y escasez de determinados objetos: - En uno de ellos abundan los recolectables específicos del agente 1 (el agente estudiado). - En el subciclo previo y en el posterior, estos recolectables aparecen ocasionalmente. - En el subciclo opuesto, no aparece ninguno de los específicos del agente 1. Por otro lado, los recolectables genéricos aparecen de forma homogénea durante todo el episodio, sin depender del subciclo.

Acciones del agente

El agente dispone de un conjunto **discreto de acciones básicas**:

- **Rotación de 45º** (izquierda/derecha con limitación por cooldown).
- **Movimiento adelante/atrás** con velocidad fija.

Estas acciones simples se eligieron para que la dificultad emergiera del **entorno y las observaciones**, y no de la dinámica motriz.

Observaciones y evolución de los tres enfoques

Uno de los principales objetivos de los experimentos fue **probar distintos diseños de observaciones** y comprobar cómo afectaban al aprendizaje. El proceso siguió un enfoque iterativo, simplificando progresivamente las entradas para reducir la carga cognitiva del agente.

Nótese que la falta de un estándar entre los diferentes enfoques dificultó la comparación precisa de resultados. Esta falta de estandarización responde a un diseño exploratorio que priorizaba encontrar una configuración funcional antes que contrastar enfoques.

Enfoque 1 – Global absoluto

- **Observaciones:** coordenadas absolutas normalizadas del agente y de todos los objetivos, junto con su tiempo de expiración más información del subciclo actual:

    1. **Datos propios del agente:**
        - Cooldown de rotación (normalizado entre 0 y 1)
        - ID propio

2. **Para cada agente en el área (incluyéndose a sí mismo):**
    - Posición X relativa al centro del área, normalizada
    - Posición Z relativa al centro del área, normalizada
    - Velocidad X, normalizada
    - Velocidad Z, normalizada
    - Rotación Y discretizada y normalizada
    - ID del agente
3. **Para cada recolectable en el área:**
    - Posición X relativa al centro del área, normalizada
    - Posición Z relativa al centro del área, normalizada
    - ID permitido para recoger
    - ¿Está activo? (1 o 0)
    - Tiempo restante de vida, normalizado
4. **Datos del área:**
    - Índice de subciclo actual
    - Índice de subciclo normalizado

- **Recompensa:** únicamente por recoger un objetivo. 4 puntos por recolectable asociado y 1 punto por recolectable genérico.
- **Motivación:** establecer una línea base simple.

Enfoque 2 – Global con shaping

- **Observaciones:** similares que en el enfoque 1 pero un poco simplificadas. Se eliminó información de los recolectables que no se podían recoger y la información del subciclo actual:

1. **Datos propios del agente:**
    - Cooldown de rotación (normalizado entre 0 y 1)
    - Posición X relativa al centro del área, normalizada
    - Posición Z relativa al centro del área, normalizada
    - Velocidad X, normalizada
    - Velocidad Z, normalizada
    - Orientación Y discretizada y normalizada
2. **Agente activo más cercano:**
    - Posición X relativa al centro del área, normalizada
    - Posición Z relativa al centro del área, normalizada
3. **Para cada recolectable:**

  - Posición X relativa al centro del área, normalizada
  - Posición Z relativa al centro del área, normalizada
  - Tiempo de vida restante normalizado (remainingSeconds / maxLifetime)

- **Recompensas adicionales:**
  - Bonus por **pisar nuevas baldosas** dentro del episodio (incentivar exploración). A lo largo de un episodio el agente obtenía al menos de 1 punto por esta fuente.
  - Variación de un **potencial** definido como la distancia al objetivo más cercano (premiar al acercarse, castigar al alejarse). lo largo de un episodio el agente obtenía cerca de 5 puntos por esta fuente.
- **Motivación:** guiar al agente hacia objetivos y fomentar la exploración estructurada de pasillos.

Enfoque 3 – Local con grafo de señales

- **Observaciones:**
  - Datos locales en **coordenadas polares** apoyados por un sistema de **raycast**. Todas las observaciones se proporcionaron con histórico de profundidad 3 para facilitar la navegación.
  - Señales propagadas a través de un **grafo topológico**: cada recolectable emite una señal que viaja por el grafo, y el agente recibe únicamente la señal más intensa visible emitida por cada recolectable.
    1. **Observaciones propias compactas:**
       - Cooldown de rotación normalizado (0..1)
       - Orientación lateral (forward.x)
       - Orientación longitudinal (forward.z)
       - Velocidad relativa (módulo normalizado, seno y coseno respecto a la orientación)
       - 5 raycast en 45º que detectan tags y distancias
    2. **Observaciones de broadcasters externos:** Para cada broadcaster (recolectable en el grafo) observado, se añaden 4 observaciones:
       - Distancia normalizada al último propagador de la señal propia más cercana
       - Seno del ángulo relativo
       - Coseno del ángulo relativo
       - Tiempo restante normalizado de la señal (expiryNorm)
- **Recompensas:**
  - Premio por **batir récords de aproximación** hacia cada objetivo (mejor mínima distancia alcanzada), eliminando penalizaciones por retroceder.

- En suposición (acertada) de que este enfoque facilitaría la exploración se planteó un reto adicional generando obstáculos aleatorios en la forma de recolectables con reward negativo, mucho más grandes y abundantes que el resto. Cada obstáculo resta 2 puntos.
- Para compensar que los recolectables asociados al agente incluían recompensa extra por acercarse, se incrementó la recompensa de los genéricos a 5 puntos.

- **Motivación:** simplificar las observaciones a información **local y estructurada**, reducir exploits del shaping y guiar de forma natural la exploración de pasillos.

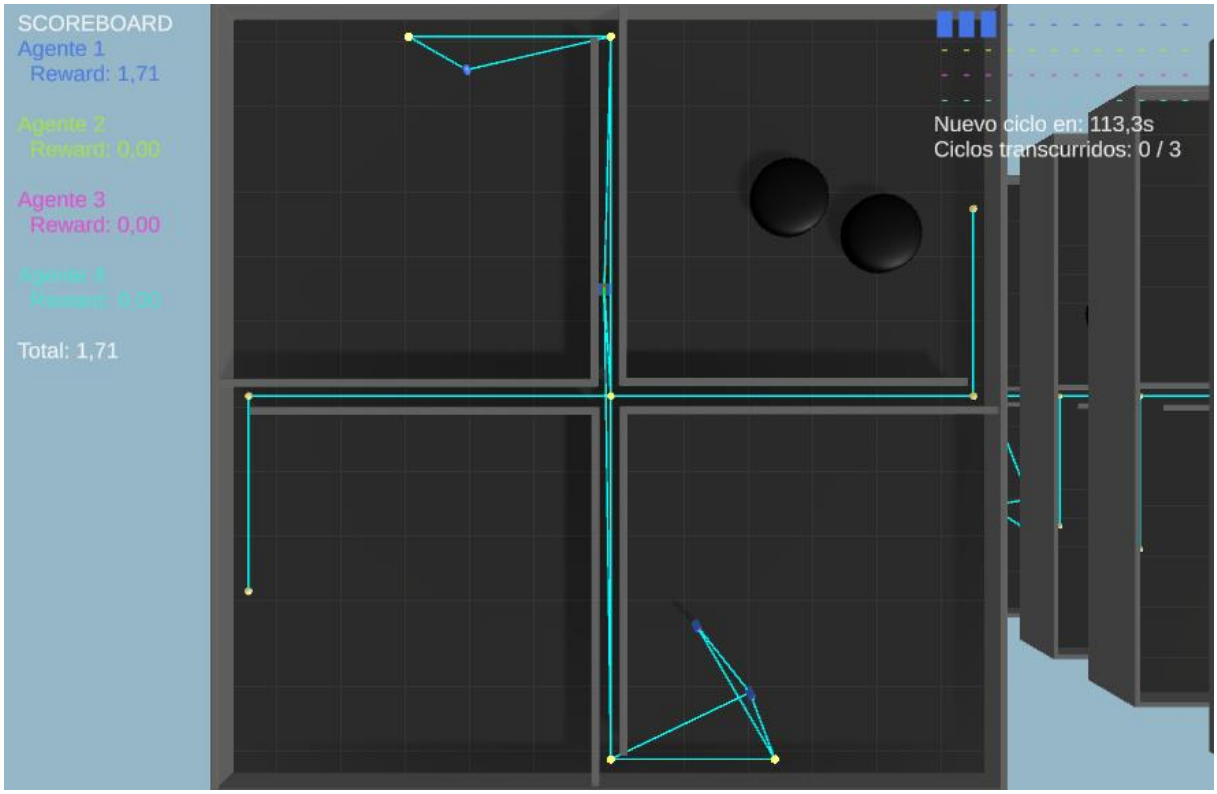

Figura 1: Captura del entorno

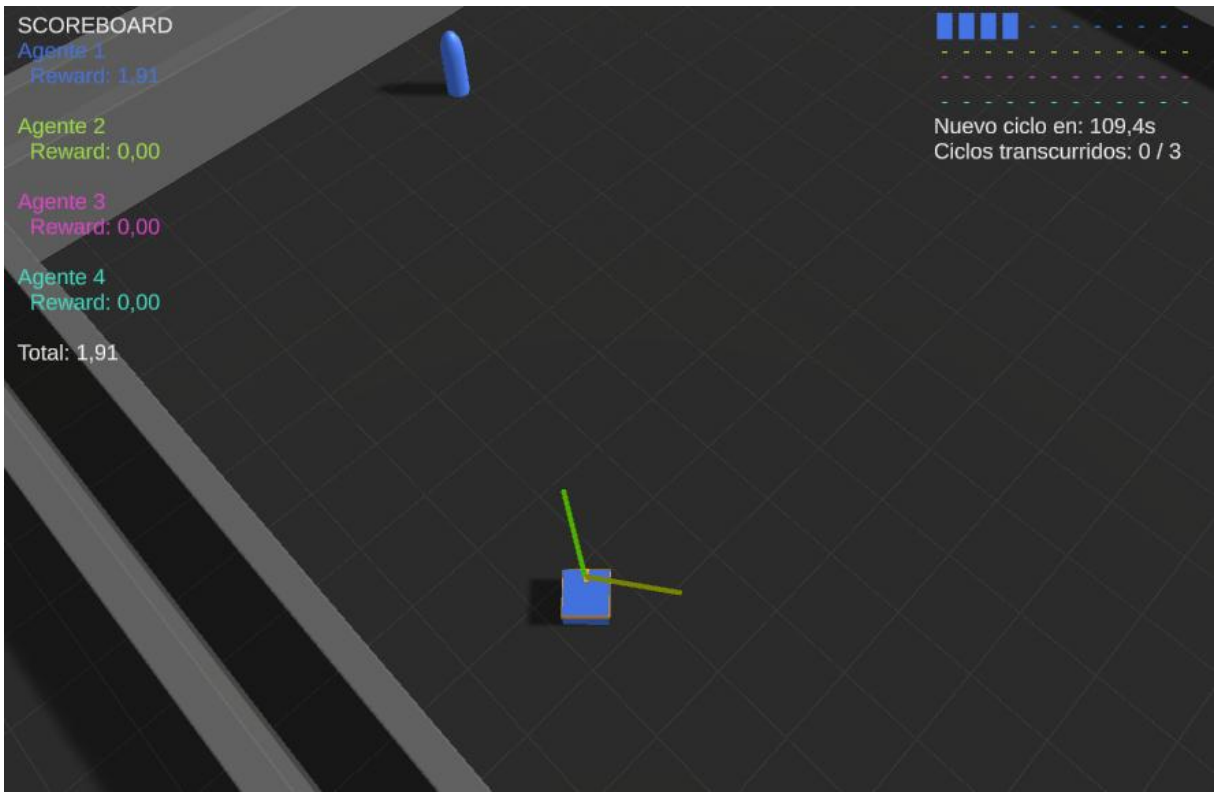

Figura 2: Captura del entorno

Algoritmo de entrenamiento (PPO con Unity ML-Agents)

Configuración común

Se entrenó un **único agente** con **PPO** (ML-Agents) en los tres experimentos, manteniendo una base homogénea para que las diferencias de rendimiento provinieran principalmente del **diseño de observaciones** y del **shaping**:

- **Algoritmo:** Proximal Policy Optimization (PPO) con *clipping* (ϵ =0.2) y **GAE** (λ =0.95).
  - *Motivo:* estabilidad y buen desempeño con espacios de observación medianos, además de ser el estándar recomendado en ML-Agents para navegación.
- **Descuento y horizonte:** γ =0.99, **time_horizon**=128.
  - *Motivo:* balance entre crédito a medio plazo (γ alto) y eficiencia de ventaja (horizonte suficiente para capturar transiciones pasillo $\leftrightarrow$ habitación sin degradar la varianza).
- **Batching de PPO:** *buffer_size* » *batch_size* con **num_epoch**=3.

  - *Motivo:* cada actualización ve múltiples *minibatches* de una gran reserva de experiencias, reduciendo sobreajuste y oscilaciones.

- **Entropía (β)** con *schedule* lineal.
  - *Motivo:* fomentar exploración al inicio y reducirla gradualmente para consolidar la política.
- **Red actor-crítico MLP**, **2 capas**, *hidden units* (128–256) según experimento.
  - *Motivo:* capacidad suficiente sin sobreparametrizar, evitando inestabilidades con datos ruidosos.
- **Normalización de entradas**: activada en los enfoques con observaciones locales/mixtas.
  - *Motivo:* estabilizar la escala cuando se mezclan señales heterogéneas (polares, raycast, intensidades).
- **Memoria (LSTM)**: activada cuando el agente usa observaciones locales.
  - *Motivo:* compensar **parcial observabilidad** (oclusiones/pasillos) acumulando contexto.
- **Señal de recompensa principal: extrínseca** (recogida de objetivos) en todos los experimentos.
- **Infraestructura de entrenamiento:** *time_scale* $\approx$ 20 (aceleración de simulación), *no_graphics* desactivado salvo en el Exp. 1, *summary_freq* entre 50k–60k pasos para registro periódico.

Configuración específica por experimento

Experimento 1 — Global absoluto

- **YAML clave:**
  - `batch_size: 2048`, `buffer_size: 40960`, `num_epoch: 3`
  - `learning_rate: 3e-4`, `beta: 0.01`, `epsilon: 0.2`, `lambd: 0.95`
  - `network_settings`: `hidden_units: 128`, `num_layers: 2`, **sin memoria**, **sin normalización**
  - `reward_signals`: **extrinsic** (γ =0.99, strength=1.0) **+ curiosity** (*ICM/RND de ML-Agents*, γ =0.99, strength=0.01, *lr* 3e-4, *encoder* 256)
  - `max_steps: 28.8M`, `summary_freq: 60k`, `checkpoint_interval: 5.76M`
  - Motor: `time_scale: 20`, `no_graphics: true`.

- **Racional de diseño:**
  - Se añadió **curiosity** de baja intensidad (0.01) para mitigar la **escasez de recompensas** con observaciones globales.
  - Arquitectura **compacta (128)** y **sin normalización** para no introducir transformaciones adicionales al vector global.
- **Resultado observado (metodológico):** pese a la ayuda de curiosity, el agente no logró políticas útiles de salida de habitación; este punto justifica los cambios del Exp. 2 (shaping guiado) y del Exp. 3 (observaciones locales + estructura topológica).

Experimento 2 — Global + shaping

- **YAML clave:**
  - `batch_size: 1024`, `buffer_size: 20480`, `num_epoch: 3`
  - `learning_rate: 3e-4`, `beta: 0.02` (más exploración inicial que en Exp. 1), `epsilon: 0.2`, `lambd: 0.95`
  - `network_settings`: **normalize: true**, `hidden_units: 256`, `num_layers: 2`, **memoria activada** (`sequence_length: 64`, `memory_size: 128`)
  - `reward_signals`: **solo extrinsic** (γ =0.99, strength=1.0)
  - `max_steps: 92.16M`, `summary_freq: 50k`, `keep_checkpoints: 12`, `checkpoint_interval: 7.68M`
  - Motor: `time_scale: 20`, **gráficos activados** (útil para depuración visual).
- **Racional de diseño:**
  - Se retiró curiosity para que el **shaping** (bonus de **baldosas nuevas** + **potencial por distancia al objetivo**) fuese el motor principal de aprendizaje.
  - Se incrementó la **capacidad del MLP (256)** y se activó **memoria** para absorber la variabilidad temporal (caducidad de objetivos) y pequeñas **parcialidades de observación** debidas a embudos.
- **Resultado observado (metodológico):** el shaping impulsó algo de exploración y captación de objetivos, pero emergieron **exploits** por *spawn/despawn* y **miopías** cerca de paredes y movimientos circulares excesivos pisando baldosas. Esto motivó el rediseño del Exp. 3.

Experimento 3 — Local + grafo de señales

- **YAML clave:**
  - `batch_size: 1024`, `buffer_size: 20480`, `num_epoch: 3`

  - `learning_rate: 3e-4`, `beta: 0.02`, `epsilon: 0.2`, `lambd: 0.95`
  - `network_settings`: **normalize: true**, `hidden_units: 256`, `num_layers: 2`, **memoria activada** (`sequence_length: 64`, `memory_size: 128`)
  - `reward_signals`: **solo extrinsic** (γ =0.99, strength=1.0)
  - `max_steps: 24.0M`, `summary_freq: 50k`, `checkpoint_interval: 2.0M` (más frecuente para capturar la fase de aprendizaje acelerado)
  - Motor: `time_scale: 20`, **gráficos activados**.

- **Racional de diseño:**
  - Observaciones **locales normalizadas** (polares + raycast) y **estructura topológica** (grafo con señales por objetivo) exigen **normalización** y **memoria** para estabilizar y retener contexto de corto plazo (orientación/giros, señales cambiantes).
  - Sin *intrinsic reward*: el **shaping por récords** (definido en el entorno) reemplaza la presión temporal miopie del potencial tradicional y evita **penalizar retrocesos** necesarios para rodear obstáculos.
- **Resultado observado (metodológico):** aprendizaje rápido y estable; las *checkpoints* más frecuentes facilitaron seleccionar políticas antes de posibles sobre-ajustes.

Parámetros y su papel (lectura rápida)

- **`learning_rate=3e-4` con *schedule* lineal:** rango estándar para PPO en ML-Agents; facilita *annealing* hacia el final para estabilizar.
- **`beta` (entropía) 0.01–0.02:** 0.02 aumenta la diversidad de acciones al principio; útil cuando el shaping (Exp. 2) o la estructura local (Exp. 3) requieren **exploración dirigida**.
- **`epsilon=0.2` (clipping PPO):** evita pasos de política demasiado grandes; valor canónico.
- **`buffer_size`/`batch_size` (20–40k / 1–2k):** suficiente variedad por actualización para evitar *myopic updates* sin saturar memoria.
- **`hidden_units` 128→256 y** memoria** ON en Exp. 2–3:** más capacidad y estado para gestionar **parcial observabilidad** y dinámicas por caducidad de objetivos.
- **Normalización ON (Exp. 2–3):** esencial cuando se combinan **polares**, **raycasts** y **intensidades** de señales en magnitudes no homogéneas.

Procedimiento y reproducibilidad

- **Ejecución:** cada enfoque se entrenó con su `run_id` independiente (p. ej., `Competitive_1`, `CompetitiveByPhase`, `Graph`).
- **Registro:** resúmenes cada 50k–60k pasos y **checkpoints** (2–7.68 M pasos) para trazar curvas de aprendizaje y seleccionar la política final.
- **Motor:** `time_scale $\approx$ 20` para acelerar simulación; resolución 84×84 (config por defecto de ML-Agents para rendimiento).
- **Semillas:** valor por defecto del trainer (seed = −1) para aleatoriedad controlada por el entorno; al reportar resultados se recomienda incluir *n* ejecuciones por enfoque (si el tiempo lo permite).

Resultados

En este capítulo se presentan los resultados obtenidos a lo largo de los tres enfoques de entrenamiento implementados. Cada uno representa un aumento progresivo de complejidad en la representación del entorno y en las señales de recompensa, lo que permite evaluar las limitaciones y ventajas relativas de cada aproximación.

Enfoque 1: Observaciones globales sin shaping

El agente recibía coordenadas globales normalizadas tanto de sí mismo como de los objetivos. Este exceso de información global, sin estructura, dificultó que el agente encontrara estrategias útiles y saliera de la habitación inicial. En general resultó en una **exploración muy deficiente**:

- Rara vez abandonaba la habitación inicial.
- Apenas conseguía recolectar objetos.
- El aprendizaje se caracterizó por una **pendiente casi nula** en la evolución del reward.

Reward vs steps. Cada step es una decisión del agente, por tanto un episodio consta de muchos steps.

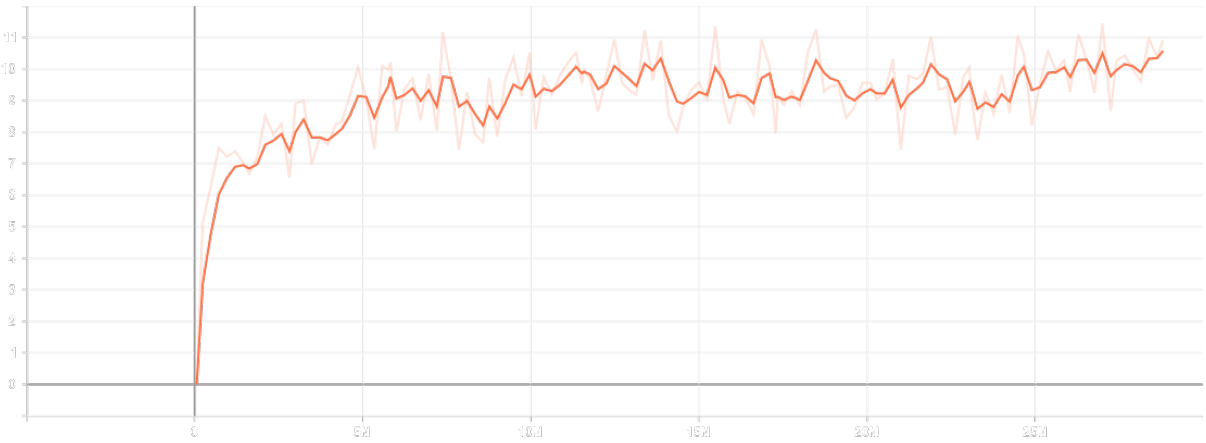

Figura 3: Evolución del reward promedio por episodio en escenario 1 y 2

Enfoque 2: Observaciones globales + shaping basado en potencial

La introducción de un shaping de recompensa no produjo **ninguna mejora** en la recogida de colectables ni tampoco incentivó la exploración de habitaciones adicionales, en verdad fue **contraproducente**. De hecho, aparecieron limitaciones y comportamientos no deseados:

- El shaping no estaba bien adecuado al experimento. Este proporcionaba recompensas al reducir el potencial y las quitaba al aumentarlo dejando por defecto un saldo neto de 0. Tan solo produciría un saldo positivo en caso de que el agente terminara el episodio cerca de un recolectable, y dado que los episodios eran largos, este posible incentivo era desdeñable.
- **Exploits del potencial** debido a cambios abruptos en el spawn y despawn de objetivos.
- Agente que permanecía inmóvil junto a las paredes, recibiendo recompensas sin progresar en la tarea.
- Problemas de granularidad espacial:
  - Con **baldosas grandes**, no era capaz de detectar los pasillos.
  - Con **baldosas pequeñas**, se movía erráticamente en círculos.
- Al final estuvo calibrado con baldosas pequeñas. Los episodios no fueron lo suficientemente largos como para que al agente le diera tiempo a visitar consistentemente todas las baldosas de la habitación inicial, lo que le desincentivaba a buscar nuevas habitaciones.

Gráficas de resultados

- **Reward vs steps**.

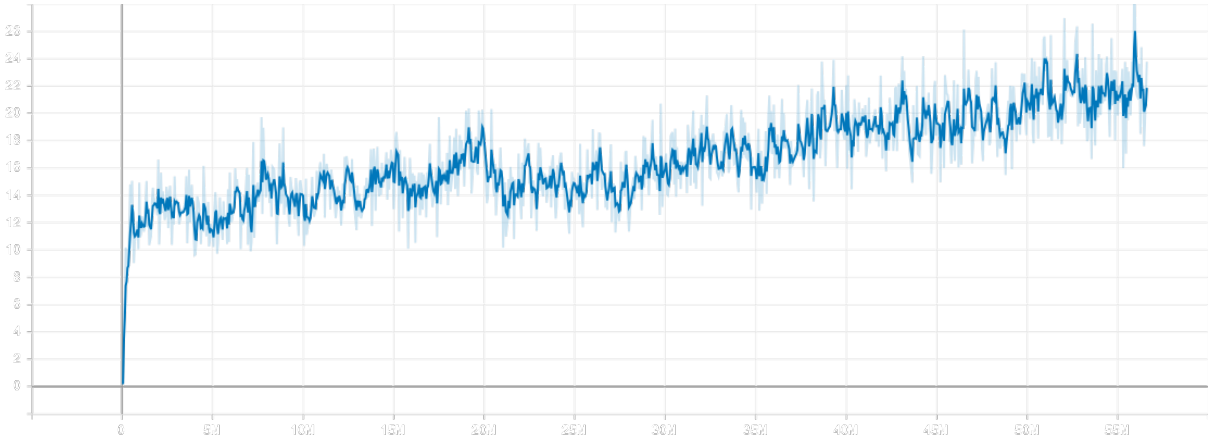

Figura 4: Evolución del reward promedio por episodio en escenario 1 y 2

- **Comparativa con resultados previos**.A continuación se muestran gráficas de la cantidad de recolectables recogidos en ambos enfoques, de los cambios de habitación y de la cantidad de habitaciones exploradas:

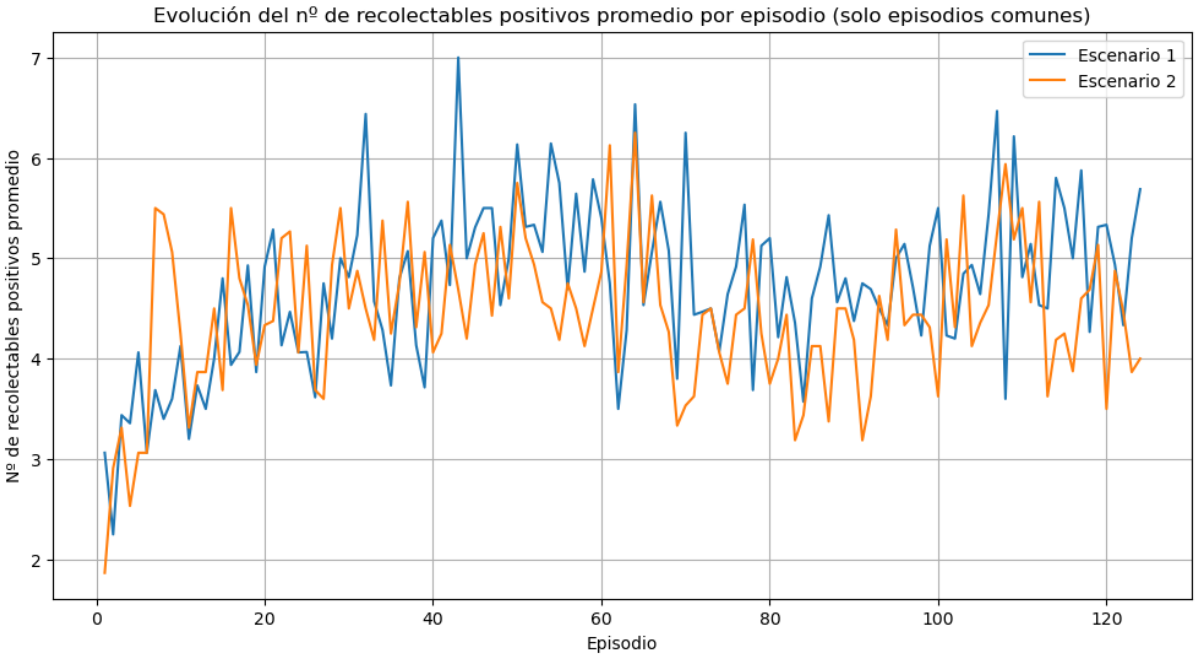

Figura 5: Evolución del nº de recolectables positivos promedio por episodio en escenario 1 y 2

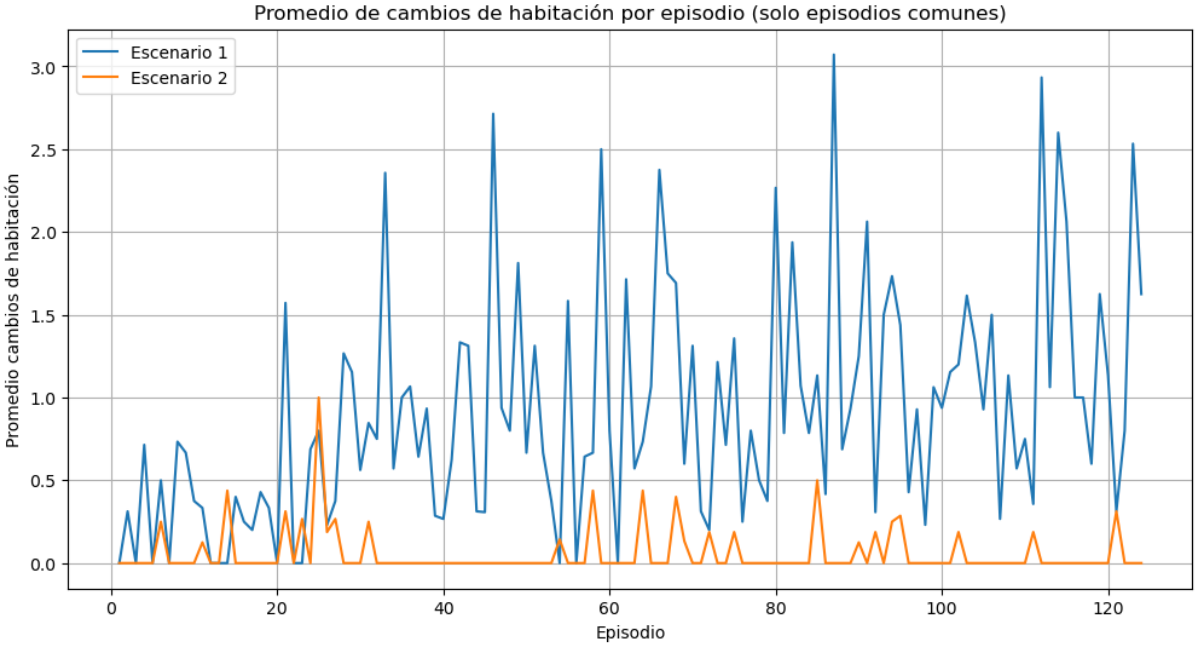

Figura 6: Promedio de cambios de habitación por episodio en escenario 1 y 2

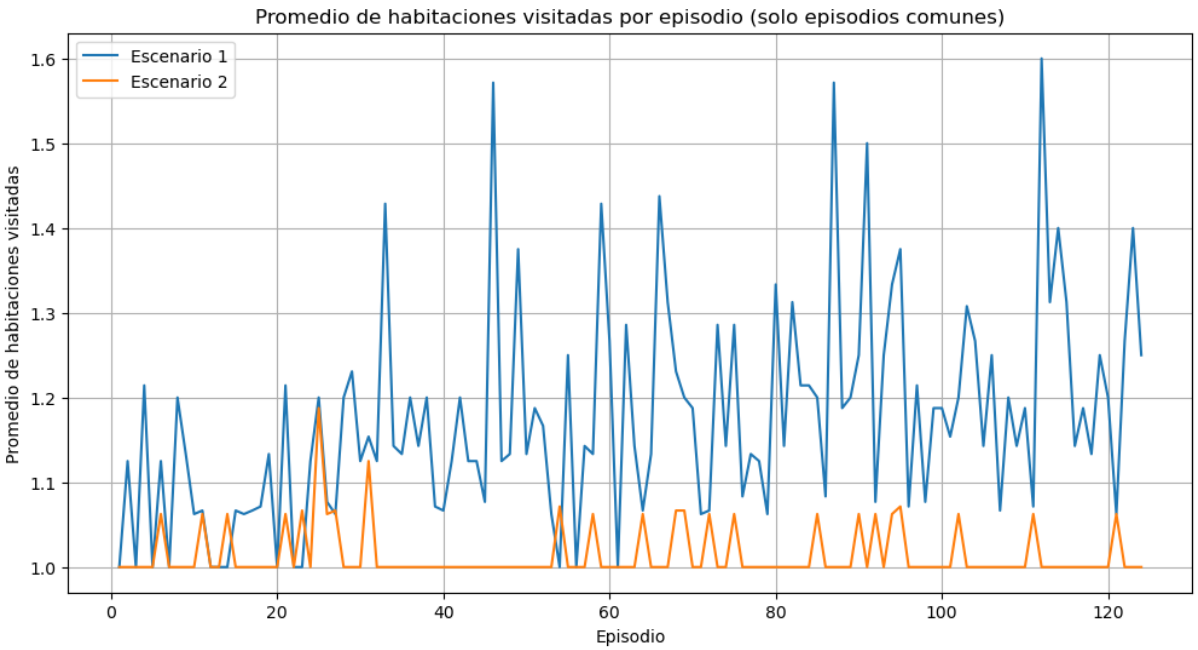

Figura 7: Promedio de habitaciones visitadas por episodio en escenario 1 y 2

Enfoque 3: Observaciones locales + grafo de señales

El tercer enfoque introdujo un **grafo de propagación de señales** desde los recolectables, junto con observaciones locales polares. Además, se aplicó shaping basado en récords de aproximación en lugar de potencial directo.

Los resultados fueron claramente superiores:

- El agente aprendió **rápidamente** a navegar habitaciones y pasillos.
- Las señales del grafo guiaron su atención de forma consistente.
- El shaping por récords **evitó los exploits** anteriores.
- Se observó un comportamiento **estable y eficiente**, con una tasa sostenida de progreso hacia los objetivos.
- También mostró la capacidad de **distinguir y esquivar obstáculos negativos** con su información local. Sin embargo, esta evitación no fue completamente fiable, lo que sugiere que la penalización aplicada a los obstáculos era demasiado baja para consolidar un rechazo perfecto.

Dado que en el mismo experimento se introdujo el grafo y las observaciones polares locales es imposible determinar el impacto individual de cada uno por separado.

Gráficas de resultados

- **Reward vs steps**.

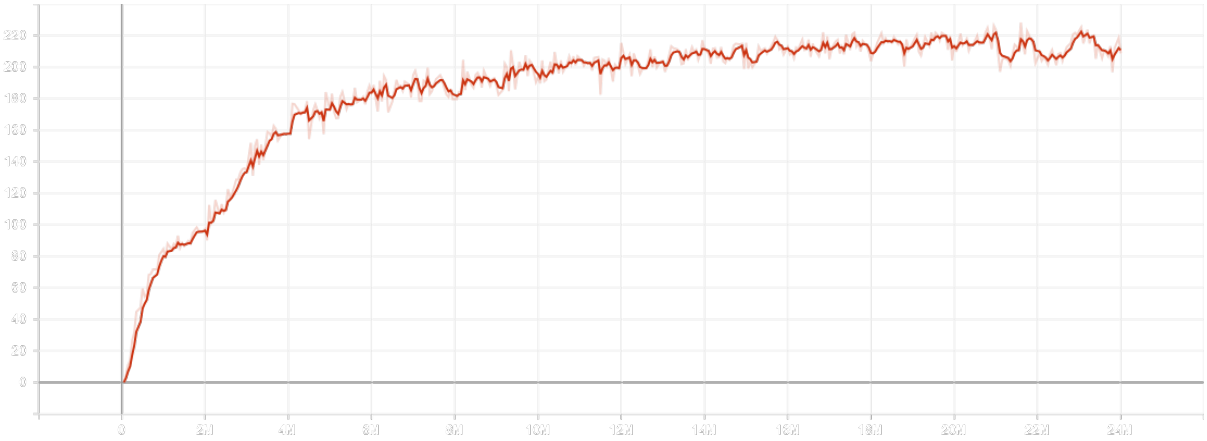

Figura 8: Evolución del reward promedio por episodio en escenario 1 y 2

- **Comparativa con resultados previos**. A continuación se muestran gráficas de la cantidad de recolectables recogidos en ambos enfoques, de los cambios de habitación y de la cantidad de habitaciones exploradas:

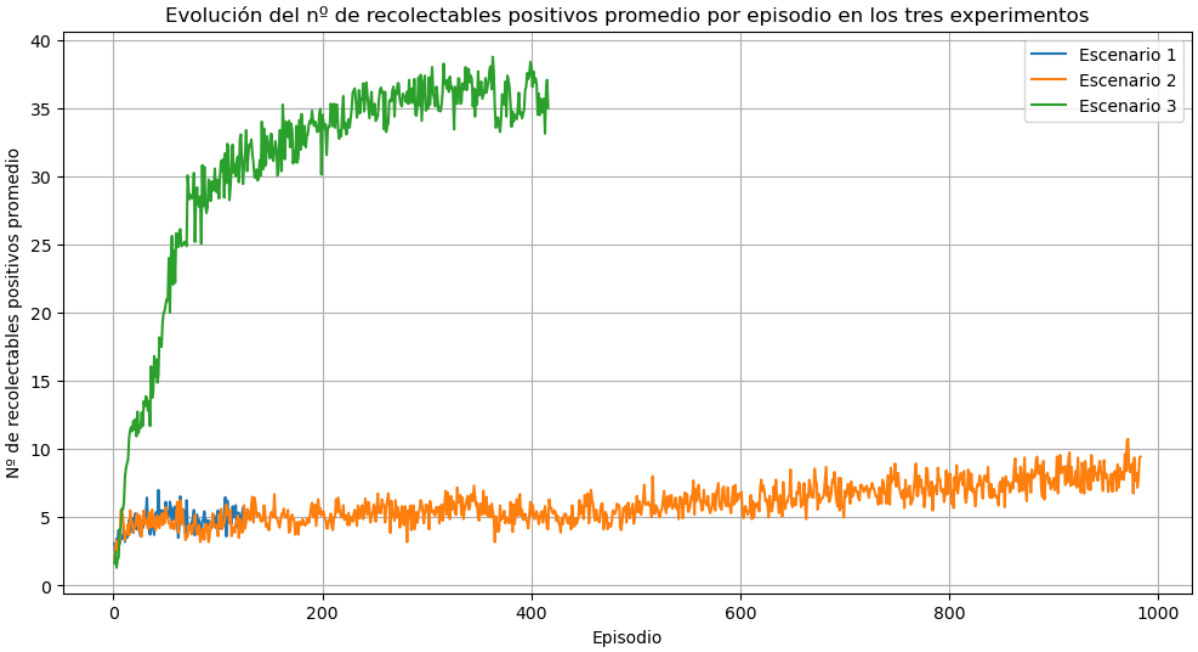

Figura 9: Evolución del nº de recolectables positivos promedio por episodio en escenario 1 y 2

Figura 10: Promedio de cambios de habitación por episodio en escenario 1 y 2

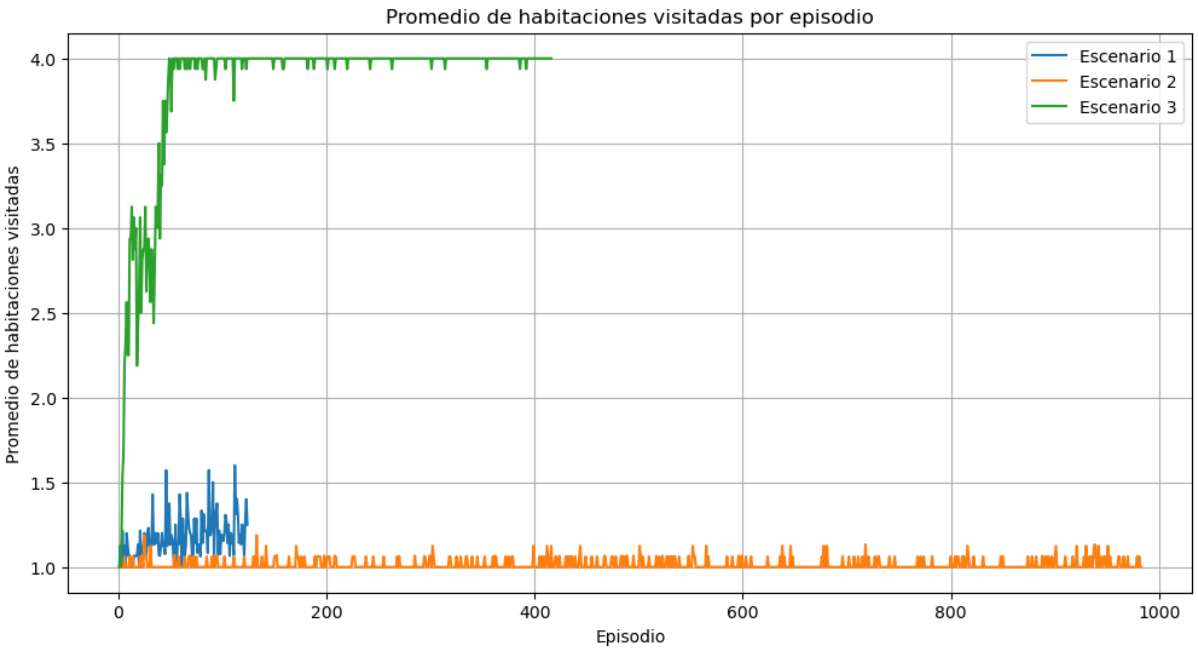

Figura 11: Promedio de habitaciones visitadas por episodio en escenario 1 y 2

- **Aciertos vs errores**. En la siguiente gráfica puede verse como los recolectables negativos se estancan en lugar de reducirse:

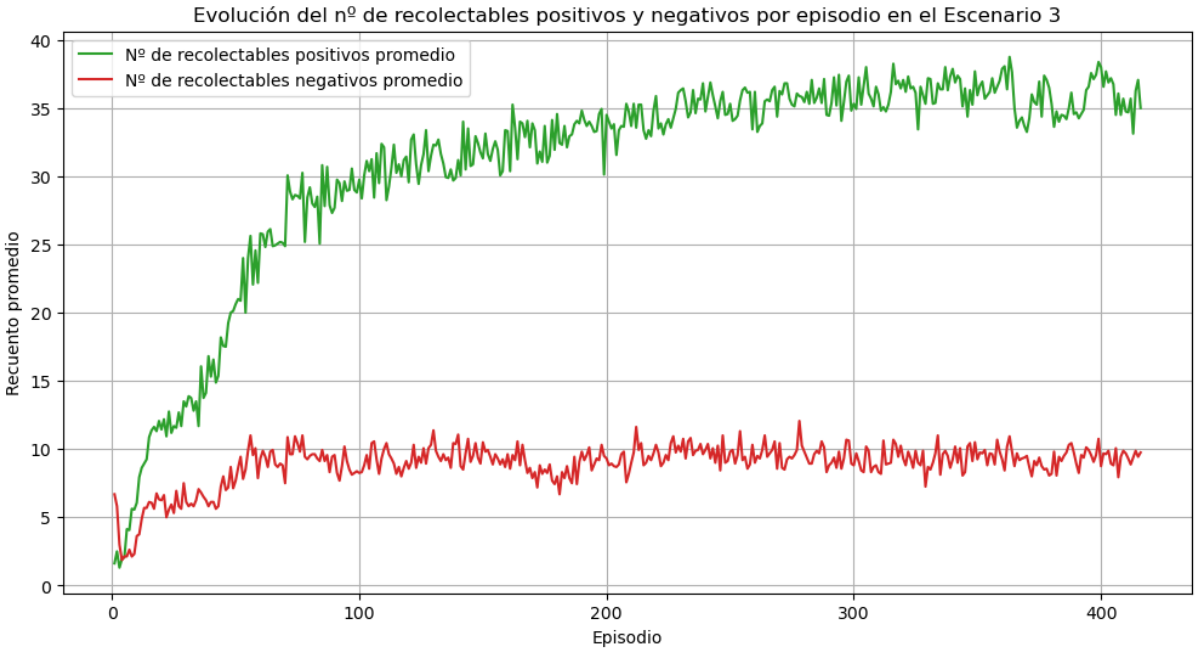

Figura 12: Evolución del nº de recolectables positivos y negativos por episodio en el Escenario 3

Comparativa general

Aunque los valores absolutos de reward no son directamente comparables entre enfoques, las gráficas de recolectables permiten observar tendencias claras:

- **Enfoque 1**: estancamiento casi completo.
- **Enfoque 2**: empeoramiento por falta de adecuación del shaping.
- **Enfoque 3**: avance sostenido y eficiente, con un patrón de aprendizaje robusto.

Conclusiones

Los resultados muestran que la calidad de la navegación del agente depende más de **cómo se estructura la información y el shaping** que de la cantidad bruta de observaciones disponibles.

El Enfoque 1 evidenció que observaciones globales ricas, sin jerarquía ni estructura, no inducen comportamientos útiles. El Enfoque 2 demostró que los *potential-based shapings* pueden ser frágiles frente a dinámicas de spawn/despawn y a geometrías discretizadas de forma irregular. Finalmente, el Enfoque 3 introdujo un **grafo de señales** y un shaping basado en récords, mostrando un aprendizaje mucho más estable y con comportamientos que emergen de forma más natural.

En conjunto, se confirma que:

- **La estructura topológica** guía la exploración de manera más fiable que la información absoluta.
- **El shaping por récords** evita artefactos y fomenta progresión consistente.
- **La calibración de las penalizaciones** es clave para inducir conductas robustas frente a obstáculos.

Limitaciones

Aunque los resultados del tercer enfoque fueron prometedores, existen amenazas a su validez y limitaciones relevantes de cara a su aplicación práctica:

- El uso de un **grafo de señales requiere infraestructura adicional** introduce un coste y una complejidad logística que limita la aplicabilidad directa del método en entornos reales. En simulación, las coordenadas de nodos y agentes están disponibles de forma directa, pero en un entorno real sería necesario:
    - O bien disponer de un sistema preciso de localización constante de los agentes.
    - O bien materializar físicamente los nodos como balizas o routers distribuidos en el espacio.
- La comparación entre enfoques no es 100 % aislada (polares+grafo introducidos juntos), lo que dificulta evaluar el impacto en aislado de cada componente.

Futuro trabajo

De cara a avanzar sobre estas bases, se proponen varias líneas:

- **Reducir la dependencia del grafo**: sustituir las observaciones precisas de coordenadas por referencias más plausibles en un entorno real, como detecciones visuales o mediciones aproximadas de distancia.
- **Priorización de tareas**: usar la distancia estimada (mediante el grafo) para que el agente decida qué objetivos atender en función de su urgencia y lejanía.
- **Penalización por expiración**: añadir una penalización explícita por dejar expirar tareas, para forzar estrategias que no ignoren objetivos cercanos.
- **Escenarios procedurales**: generar mapas de forma automática y entrenar un agente auxiliar encargado de colocar nodos del grafo al inicio (o dinámicamente durante la exploración).
- **Curriculum learning con señales progresivas**: utilizar el grafo como andamiaje temporal en fases iniciales de entrenamiento, e ir retirando sus señales progresivamente para que el agente aprenda a navegar sin depender de infraestructura extra.
- **Memoria de grafo a partir de señales visuales**: Aprender una memoria de grafo a partir de señales visuales en lugar de dar el grafo explícito [5].

Referencias

1. Ng AY, Harada D, Russell SJ (1999) Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. En: Proceedings of the 16th International Conference on Machine Learning (ICML). pp 278-287

2. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research 5:90-98. https://doi.org/10.1177/027836498600500106

3. Chaplot DS, Salakhutdinov R, Gupta A, Gupta S (2020) Neural Topological SLAM for Visual Navigation. En: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

4. Chen K, Vicente JP de, Sepulveda G, et al (2019) A Behavioral Approach to Visual Navigation with Graph Localization Networks. En: Robotics: Science and Systems (RSS)

5. Kwon O, Kim N, Choi Y, et al (2021) Visual Graph Memory with Unsupervised Representation for Visual Navigation. En: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)

6. Chaplot DS, Gandhi D, Gupta S, et al (2020) Learning to Explore using Active Neural SLAM. En: International Conference on Learning Representations (ICLR)

7. (2025) MiniGrid: MultiRoom Environment

8. Sun J, Wu J, Ji Z, Lai Y-K (2024) A Survey of Object Goal Navigation. IEEE Transactions on Automation Science and Engineering