Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: Reinforced Learning

# Impacto de la complejidad de las observaciones en el rendimiento de algoritmos de Aprendizaje por Refuerzo en un entorno Pacman

---

Autor: Alejandro Suau Ruiz

Tutor: Marc Borras Camarasa

Profesor: David Masip Rodó

---

Palma de Mallorca, 16 de noviembre de 2025

# Créditos/Copyright

Una página con la especificación de créditos/copyright para el proyecto (ya sea aplicación por un lado y documentación por el otro, o unificadamente), así como la del uso de marcas, productos o servicios de terceros (incluidos códigos fuente). Si una persona diferente al autor colaboró en el proyecto, tiene que quedar explicitada su identidad y qué hizo.

A continuación se ejemplifica el caso más habitual, aunque se puede modificar por cualquier otra alternativa:

# FICHA DEL TRABAJO FINAL

| | |
|---|---|
| Título del trabajo: | Impacto de la complejidad de las observaciones en el rendimien |
| Nombre del autor: | Alejandro Suau Ruiz |
| Nombre del colaborador/a docente: | Marc Borras Camarasa |
| Nombre del PRA: | David Masip Rodó |
| Fecha de entrega (mm/aaaa): | 01/2026 |
| Titulación o programa: | Máster Universitario de Data Science |
| Área del Trabajo Final: | Reinforced Learning |
| Idioma del trabajo: | Español |
| Palabras clave | Aprendizaje por refuerzo, Complejidad del estado, Pacman |

# Dedicatoria/Cita

Breves palabras de dedicatoria y/o una cita.

# Agradecimientos

Si se considera oportuno, mencionar a las personas, empresas o instituciones que hayan contribuido en la realización de este proyecto.

# Abstract

This work investigates how the complexity of state representation affects the performance of reinforcement learning algorithms in a simplified Pacman environment. The central idea is that an agent's success depends not only on the algorithm itself but also on the quality and richness of the information it receives from the environment. To analyze this, we designed a set of progressively more complex observation spaces, ranging from the basic positions of the player and the ghost to configurations that include the presence and duration of power-ups and the distribution of coins across quadrants. Experiments were carried out using widely adopted algorithms such as PPO, A2C, and DQN, implemented with the Stable-Baselines3 library. Controlled training with different random seeds and consistent performance metrics allowed us to compare both learning speed and the stability of the learned policies. The results show that increasing state complexity does not always lead to better performance: in some cases, the additional information introduces noise and hinders convergence. However, for specific configurations, the agent was able to exploit the extra information to achieve more robust behavior. In conclusion, the project highlights the importance of observation design in the success of reinforcement learning agents, stressing the need to balance simplicity and expressiveness depending on the task and the chosen algorithm.

**Keywords**: Reinforcement learning, State complexity, Pacman, Stable-Baselines3, Gymnasium

# Resumen

Este trabajo explora cómo la complejidad de la representación del estado influye en el rendimiento de los algoritmos de aprendizaje por refuerzo en un entorno simplificado de Pacman. Partimos de la premisa de que un agente no aprende únicamente por el algoritmo empleado, sino también por la calidad y la riqueza de la información que percibe del entorno. Para analizarlo, hemos diseñado un conjunto de observaciones progresivamente más complejas, desde la posición básica de jugador y fantasma, hasta configuraciones que incluyen la presencia y duración de comodines o la distribución de monedas por cuadrantes. La experimentación se ha llevado a cabo con algoritmos ampliamente utilizados, como PPO, A2C y DQN, implementados con la librería Stable-Baselines3. Se han realizado entrenamientos controlados con semillas distintas y métricas de rendimiento homogéneas, lo que ha permitido comparar tanto la velocidad de aprendizaje como la estabilidad de las políticas aprendidas. Los resultados muestran que un aumento en la complejidad del estado no garantiza siempre un mejor desempeño: en ciertos casos, la mayor riqueza de información introduce ruido y dificulta la convergencia. Sin embargo, para determinadas configuraciones el agente logra aprovechar la información adicional y alcanzar un comportamiento más robusto. En conclusión, el proyecto confirma la relevancia del diseño de observaciones en el éxito de un agente de aprendizaje por refuerzo, subrayando la necesidad de equilibrar simplicidad y expresividad en función del objetivo y del algoritmo empleado.

**Palabras clave**: Aprendizaje por refuerzo, Complejidad del estado, Pacman, Stable-Baselines3, Gymnasium

# Índice general

# Índice de figuras

# Índice de cuadros

# Capítulo 1

# Introducción

El *Aprendizaje por Refuerzo* (*Reinforcement Learning*, RL) se ha consolidado como una de las ramas más activas y prometedoras del aprendizaje automático. A diferencia del aprendizaje supervisado, en el que el modelo aprende a partir de ejemplos etiquetados, el RL se fundamenta en la interacción continua de un agente con un entorno, buscando maximizar una recompensa acumulada. Este paradigma ha permitido resolver problemas complejos de toma de decisiones secuenciales en campos tan diversos como la robótica, los videojuegos, la gestión de recursos o la conducción autónoma.

En los últimos años, la combinación del RL con redes neuronales profundas —conocida como *Deep Reinforcement Learning* (DRL)— ha supuesto un gran avance, al permitir que los agentes aprendan directamente a partir de observaciones de alta dimensión. Algoritmos como *Deep Q-Network* (DQN) o *Proximal Policy Optimization* (PPO) han demostrado su eficacia en entornos complejos como Atari o MuJoCo, donde el espacio de estados y de acciones puede ser extremadamente amplio.

Sin embargo, el rendimiento de un agente no depende únicamente del algoritmo de aprendizaje empleado, sino también de la representación del estado con la que percibe el entorno. En otras palabras, la complejidad de la observación influye directamente en la capacidad del agente para generalizar, explorar y aprender políticas efectivas. Representaciones demasiado simples pueden limitar el aprendizaje al ocultar información relevante, mientras que representaciones demasiado complejas pueden ralentizar la convergencia y requerir una mayor capacidad de red o una exploración más prolongada.

En este Trabajo Fin de Máster se analiza precisamente cómo afecta la complejidad del estado observado al rendimiento del agente en un entorno controlado. Para ello, se ha desarrollado un entorno propio denominado *SimplePacmanEnv*, inspirado en el clásico juego *Pac-Man*, pero adaptado para experimentos de RL. Este entorno, implementado con *Gymnasium* y *Stable-Baselines3*, permite configurar diferentes niveles de complejidad en la observación —desde

representaciones mínimas (solo posiciones del jugador y el fantasma) hasta observaciones enriquecidas con información global sobre las monedas o incluso representaciones tipo imagen—, manteniendo constantes la dinámica del entorno y la estructura de recompensas.

Sobre esta base se han entrenado y evaluado tres algoritmos representativos de distintas familias del RL profundo: *Advantage Actor-Critic* (A2C), *Proximal Policy Optimization* (PPO) y *Deep Q-Network* (DQN). El objetivo es cuantificar cómo evoluciona el aprendizaje y el rendimiento final de cada agente en función de la información proporcionada por el entorno, y determinar si existe una relación clara entre la complejidad de la representación del estado y la eficiencia del aprendizaje.

Además, el proyecto incorpora herramientas complementarias como *TensorBoard* para la monitorización en tiempo real del entrenamiento, y el uso de *EvalCallback* para realizar evaluaciones periódicas y registrar automáticamente el mejor modelo alcanzado. Estos mecanismos no solo facilitan el análisis experimental, sino que aportan rigor y reproducibilidad a los resultados obtenidos.

En conjunto, este trabajo busca aportar una visión práctica y experimental sobre la influencia de la observación en entornos de aprendizaje por refuerzo, conectando los fundamentos teóricos con un estudio aplicado y reproducible, y poniendo de manifiesto la importancia del diseño de los estados en la efectividad de los algoritmos de RL.

# 1. Contexto y motivación

## 1.1. Justificación e interés

El estudio de la relación entre la complejidad del estado y el rendimiento del agente resulta relevante tanto desde el punto de vista teórico como práctico:

- Permite entender la eficiencia y capacidad de generalización de distintos algoritmos (A2C, PPO, DQN) en entornos de distinta dimensionalidad.
- Aporta evidencia empírica sobre el compromiso entre simplicidad del modelo y capacidad de representación, tema central en el diseño de entornos y arquitecturas RL.
- Facilita la comparación controlada entre configuraciones de observación, aislando el impacto del estado sin alterar dinámicas, recompensas ni mecánicas de juego.
- Además, el uso de un entorno propio y reproducible contribuye al valor educativo e investigativo, permitiendo futuras extensiones y análisis.

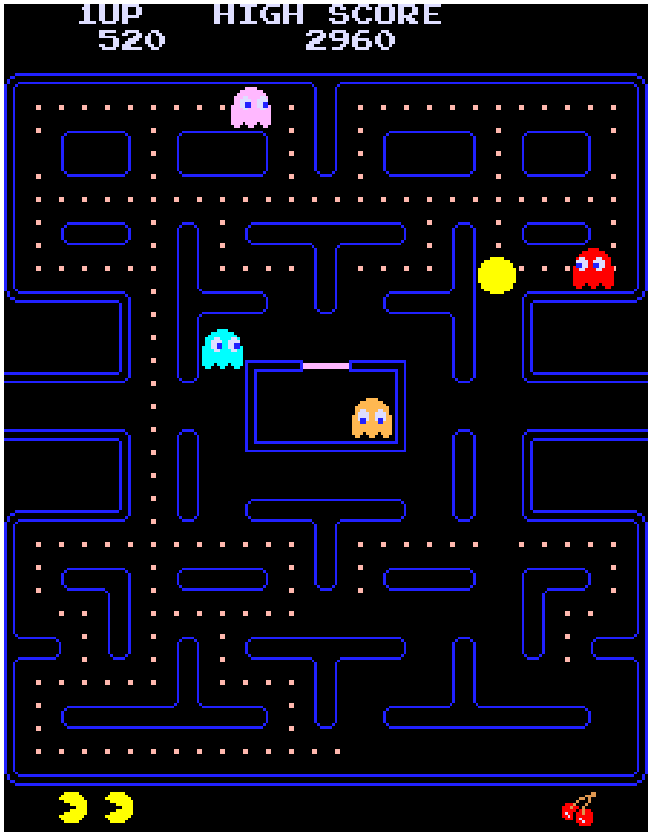


Figura 1.1: Imagen del juego original de Pac-man de 1980

## 1.2. Motivación personal

Mi motivación para realizar este proyecto es principalmente educativa e investigativa. Busco profundizar en el campo del Aprendizaje por Refuerzo, comprendiendo de forma práctica cómo los agentes aprenden a través de la interacción con su entorno y cómo la cantidad de información observada afecta dicho proceso.

El trabajo ha supuesto un ejercicio de integración de conocimientos adquiridos durante el Máster (aprendizaje automático, programación en Python, redes neuronales, y metodologías experimentales) con el objetivo de desarrollar una visión más completa y aplicada del RL. En última instancia, la meta ha sido explorar, aprender y comprender los fundamentos de la toma de decisiones secuenciales bajo incertidumbre.

# 2. Objetivos

## 2.1. Objetivo principal

Evaluar cómo el nivel de complejidad del espacio de observación afecta el rendimiento de distintos algoritmos de Aprendizaje por Refuerzo, manteniendo constantes el entorno, las recompensas y la dinámica del juego.

## 2.2. Objetivos específicos

1. Diseñar e implementar un entorno controlado y parametrizable (basado en Pac-Man) que permita modificar el tipo de observación percibida por el agente.

2. Entrenar y comparar varios algoritmos de RL (A2C, PPO, DQN) bajo distintos modos de observación.

3. Analizar métricas de rendimiento (recompensa media, estabilidad, convergencia, tiempo de entrenamiento).

4. Evaluar la relación entre dimensionalidad del estado y eficiencia del aprendizaje.

5. Documentar la metodología y resultados, destacando las implicaciones para futuros desarrollos o estudios.

# 3. Sostenibilidad, diversidad y desafíos ético/sociales

A continuación se valoran los posibles impactos en materia de sostenibilidad, ética y diversidad.

**Sostenibilidad.** El trabajo se ha realizado íntegramente en un entorno digital y no requiere recursos materiales adicionales, por lo que su impacto medioambiental directo es muy reducido. El único consumo relevante proviene del uso de recursos computacionales durante el entrenamiento de los agentes. Se ha procurado minimizar este impacto mediante el uso de modelos ligeros y tiempos de entrenamiento moderados, evitando configuraciones innecesariamente costosas. Desde una perspectiva positiva, la metodología promueve la eficiencia en la experimentación mediante la reutilización de entornos y scripts reproducibles, contribuyendo así a un uso más sostenible de los recursos de investigación. El proyecto no se ve afectado por ninguna normativa específica en materia medioambiental ni tiene relación directa con los Objetivos de Desarrollo Sostenible (ODS), más allá de fomentar el desarrollo de tecnologías digitales eficientes y abiertas.

**Comportamiento ético y responsabilidad social.** El proyecto no involucra datos personales, información sensible ni toma de decisiones sobre personas o colectivos. Su propósito es exclusivamente académico y experimental. Se han seguido principios de transparencia y reproducibilidad, haciendo uso de bibliotecas abiertas como *Stable-Baselines3* y *Gymnasium*, lo que garantiza la trazabilidad del código y los resultados. En este sentido, el trabajo se alinea con los principios deontológicos de la ingeniería informática, promoviendo la investigación responsable y el acceso abierto al conocimiento. No se identifican impactos negativos sobre el empleo ni sobre aspectos legales o de seguridad.

**Diversidad, género y derechos humanos.** El contenido técnico del trabajo no tiene incidencia directa en cuestiones de género, diversidad o derechos humanos. Sin embargo, se reconoce la importancia de la diversidad en la investigación en inteligencia artificial y el compromiso con un lenguaje inclusivo y accesible en la documentación y la memoria del proyecto. Asimismo, el entorno desarrollado puede emplearse como herramienta docente o de experimentación, favoreciendo la accesibilidad y la igualdad de oportunidades en la formación en técnicas de aprendizaje por refuerzo.

En conjunto, el proyecto presenta un impacto ético, social y medioambiental neutro o ligeramente positivo, destacando por su orientación a la investigación abierta, la eficiencia computacional y el cumplimiento de buenas prácticas profesionales en el ámbito de la inteligencia artificial.

# 4. Enfoque y metodología

El presente trabajo adopta un enfoque experimental y comparativo orientado a analizar la influencia de la complejidad del estado sobre el rendimiento de los algoritmos de Aprendizaje por Refuerzo (RL).

## 4.1. Estrategias consideradas

Durante la fase inicial del proyecto se evaluaron tres posibles enfoques:

1. **Uso de entornos estándar (Gym / Atari):** Permitiría aprovechar entornos consolidados como CartPole o Breakout. Sin embargo, estos no ofrecen un control granular sobre la estructura del estado, lo que impide aislar experimentalmente el efecto de la observación.

2. **Simulación matemática abstracta (MDP sintético):** Un enfoque más teórico, usando representaciones vectoriales arbitrarias para estudiar el tamaño del espacio de estado.

Aunque más simple de implementar, carece de un contexto visual y de dinámica tangible que facilite la interpretación y comparación intuitiva de los resultados.

3. **Diseño de un entorno personalizado (Pacman simplificado):** Este enfoque combina realismo y control. Permite modificar el nivel de información observable manteniendo constantes las reglas y recompensas. Además, el dominio es familiar y visualmente interpretable, lo que facilita tanto la depuración como la comunicación de los resultados.

## 4.2. Estrategia seleccionada

Se eligió la **tercera estrategia**, basada en el desarrollo de un entorno propio —SimplePacmanEnv— compatible con Gymnasium y Stable-Baselines3. Esta elección resulta la más adecuada por las siguientes razones:

- Permite **aislar experimentalmente** la variable de interés (la complejidad del estado), manteniendo el resto de factores constantes.
- Facilita la **comparación justa** entre distintos algoritmos de RL bajo las mismas condiciones.
- Favorece la **reproducibilidad y extensibilidad** del experimento (se pueden añadir observaciones o cambiar reglas sin alterar la estructura general).
- Proporciona una **base educativa sólida**, al exigir una comprensión integral del pipeline de RL: diseño de entorno, configuración del agente y análisis de resultados.

# Capítulo 2

# Estado del arte

## 1. Fundamentos del Aprendizaje por Refuerzo

El *Aprendizaje por Refuerzo* (Reinforcement Learning, RL) constituye una de las ramas más relevantes del aprendizaje automático orientadas a la toma de decisiones secuenciales bajo incertidumbre. En este paradigma, un agente interactúa con un entorno mediante un proceso de prueba y error, buscando maximizar una recompensa acumulada a largo plazo. Formalmente, este proceso se modela como un *Proceso de Decisión de Markov* (MDP), definido por el conjunto $(S, A, P, R, \gamma)$, donde $S$ representa el espacio de estados, $A$ el conjunto de acciones posibles, $P(s'|s, a)$ la función de transición de estados, $R(s, a)$ la función de recompensa y $\gamma \in [0, 1]$ el factor de descuento que pondera la importancia de las recompensas futuras [Sutton and Barto, 2018].

El objetivo de un agente es encontrar una política óptima $\pi^*(a|s)$ que maximice el retorno esperado:

$$G_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}\right], \tag{2.1}$$

donde $r_{t+k+1}$ representa la recompensa obtenida $k$ pasos después del tiempo $t$. A diferencia del aprendizaje supervisado, el RL no dispone de un conjunto fijo de ejemplos etiquetados: el conocimiento se adquiere mediante la experiencia directa con el entorno.

Los algoritmos de RL pueden clasificarse en dos grandes familias. Los **métodos basados en valor** estiman la utilidad esperada de los estados o pares estado–acción a través de funciones $V(s)$ o $Q(s, a)$, como en los algoritmos Q-Learning o SARSA. Por otro lado, los **métodos basados en política** parametrizan directamente la política $\pi_\theta(a|s)$ mediante un modelo (típicamente una red neuronal) y optimizan sus parámetros $\theta$ a partir del gradiente de la recompensa esperada. La combinación de ambos enfoques da lugar a los métodos **actor-crítico**, donde un componente

(*crítico*) estima el valor de los estados y otro (*actor*) actualiza la política en función de dicha estimación.

En este marco, la **definición del estado** $s_t$ es un factor determinante. La información que el agente percibe condiciona su capacidad para generalizar, explorar y aprender políticas efectivas. Una representación excesivamente simple puede ocultar detalles relevantes del entorno, mientras que una observación demasiado compleja puede introducir ruido, ralentizar la convergencia y requerir redes de mayor capacidad. Por ello, la **complejidad del estado observado** constituye una variable crítica para la eficiencia del aprendizaje, y su análisis es el eje central del presente trabajo.

# 2. Avances en Deep Reinforcement Learning

La integración del aprendizaje por refuerzo con redes neuronales profundas dio lugar al *Aprendizaje por Refuerzo Profundo* (Deep Reinforcement Learning, DRL), que permite aproximar funciones de valor o políticas directamente a partir de observaciones de alta dimensión, como imágenes o secuencias temporales. Este paradigma ha sido fundamental para extender el RL a entornos complejos donde las representaciones manuales del estado resultan inviables [Mnih et al., 2015].

## Deep Q-Network (DQN)

El punto de inflexión en la historia del DRL se produjo con el algoritmo **Deep Q-Network (DQN)** desarrollado por Mnih et al. [2015], que combinó *Q-Learning* con redes neuronales convolucionales. DQN aprende una función de acción-valor $Q(s, a; \theta)$ que estima la recompensa esperada al ejecutar una acción $a$ en un estado $s$, siguiendo una política $\pi$ derivada de dicha estimación (normalmente, una política $\epsilon$-greedy). El objetivo de la red es minimizar la diferencia entre la predicción actual y el valor objetivo definido por la ecuación de Bellman:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right], \tag{2.2}$$

donde $\theta^-$ son los parámetros de una red objetivo (*target network*) que se actualiza de forma retardada para estabilizar el aprendizaje. El uso de una memoria de experiencias (*replay buffer*) permite muestrear de manera aleatoria transiciones $(s, a, r, s')$, reduciendo la correlación temporal entre los datos.

Gracias a estos mecanismos, DQN consiguió por primera vez que un agente alcanzara rendimientos comparables a los humanos en múltiples videojuegos de Atari, aprendiendo directamente a partir de píxeles sin información previa sobre las reglas del entorno.

A partir de este modelo original surgieron numerosas variantes orientadas a mejorar la estabilidad y eficiencia, como *Double DQN*, que corrige el sesgo optimista de la estimación del valor máximo, o *Dueling DQN*, que separa la estimación del valor de estado y la ventaja de cada acción.

## Advantage Actor-Critic (A2C)

A diferencia de DQN, los métodos basados exclusivamente en valor no se adaptan bien a entornos con espacios de acción continuos. Para superar esta limitación surgieron los **métodos actor-crítico**, que combinan una *red actor*, encargada de representar la política $\pi_\theta(a|s)$, y una *red crítico*, que estima la función de valor $V_\phi(s)$ y evalúa la calidad de las acciones seleccionadas. El algoritmo **Advantage Actor-Critic (A2C)**, versión síncrona del A3C (*Asynchronous Advantage Actor-Critic*) [Mnih et al., 2016], ejecuta múltiples agentes en paralelo que comparten la misma política y sincronizan sus actualizaciones periódicamente.

El crítico proporciona una señal de error (*advantage*) definida como:

$$A(s,a) = Q(s,a) - V(s), \tag{2.3}$$

que mide cuánto mejor resulta ejecutar la acción $a$ frente al valor medio esperado del estado. El actor ajusta su política en la dirección del gradiente del rendimiento esperado:

$$\nabla_\theta J(\theta) = \mathbb{E}[A(s,a)\nabla_\theta \log \pi_\theta(a|s)]. \tag{2.4}$$

Este esquema reduce la varianza del gradiente respecto a los métodos puramente basados en política, logrando un equilibrio entre estabilidad y eficiencia de aprendizaje.

## Proximal Policy Optimization (PPO)

Posteriormente, Schulman et al. [2017] introdujeron el algoritmo **Proximal Policy Optimization (PPO)**, que mejoró la estabilidad del entrenamiento mediante una actualización de política acotada. En lugar de aplicar grandes cambios en la política tras cada actualización, PPO optimiza una función de pérdida que penaliza desviaciones excesivas entre la política nueva $\pi_\theta$ y la antigua $\pi_{\theta_{\text{old}}}$:

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right], \tag{2.5}$$

donde $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ representa la proporción entre políticas. El operador *clip* limita la magnitud del cambio a un rango controlado $(1 \pm \epsilon)$, asegurando una mejora progresiva

sin sobreajustes bruscos. Este mecanismo, junto con el uso de minibatches y actualizaciones múltiples por lote, convierte a PPO en un algoritmo especialmente robusto y fácil de entrenar, siendo actualmente uno de los más utilizados en investigación y aplicaciones prácticas.

### Síntesis comparativa

En síntesis, DQN, A2C y PPO representan tres paradigmas complementarios del Aprendizaje por Refuerzo Profundo:

- **DQN:** aprendizaje basado en valor, idóneo para entornos con acciones discretas; su estabilidad depende fuertemente de la correlación entre muestras y de la calidad de la observación.
- **A2C:** enfoque actor-crítico que combina estimación de valor y política, reduciendo la varianza del gradiente y favoreciendo el aprendizaje paralelo.
- **PPO:** optimización de políticas proximales que introduce una restricción explícita sobre el cambio de política, logrando robustez y generalización en entornos complejos.

Estas características los convierten en candidatos idóneos para analizar de forma comparativa cómo la **complejidad de la observación** afecta el rendimiento del aprendizaje, manteniendo constante la dinámica y la recompensa del entorno.

## 3. Síntesis y vacíos detectados

La revisión de la literatura evidencia que, aunque el Aprendizaje por Refuerzo Profundo ha experimentado avances notables en la última década, gran parte de los esfuerzos se ha centrado en la **optimización de los algoritmos** —ya sea mediante arquitecturas, estrategias de exploración o funciones de pérdida—, mientras que el impacto de la representación del estado ha recibido menor atención. En la mayoría de los estudios, el espacio de observación se trata como una variable fija del entorno, sin evaluar de manera sistemática cómo su complejidad o riqueza informacional influye en la eficacia del aprendizaje.

En respuesta a esta limitación, han surgido líneas de investigación como el *State Representation Learning* (SRL), que busca aprender representaciones latentes más compactas y relevantes para la toma de decisiones [Lesort et al., 2018, Ha and Schmidhuber, 2018, Hafner et al., 2020]. Estos trabajos utilizan modelos generativos o redes autoencoder para capturar información útil en espacios de menor dimensión, lo que facilita la generalización y reduce el coste computacional. No obstante, la mayoría de enfoques SRL se orientan a la **construcción automática de**

**representaciones** y no al estudio explícito del efecto de distintos niveles de complejidad observacional sobre el proceso de aprendizaje.

De forma complementaria, los estudios sobre entornos parcialmente observables [Hausknecht and Stone, 2015] han puesto de manifiesto cómo la falta de información puede limitar la convergencia y provocar políticas subóptimas. Sin embargo, apenas existen análisis controlados sobre el efecto contrario: cómo el exceso de información o la redundancia en el estado pueden degradar el rendimiento o ralentizar la convergencia.

Los entornos más utilizados como *benchmarks*, tales como Atari o MuJoCo, ofrecen espacios de observación fijos, lo que impide aislar experimentalmente esta variable. En años recientes, entornos modulares como *MiniGrid* [Chevalier-Boisvert et al., 2018] o *ProcGen* [Cobbe et al., 2020] han permitido cierta variabilidad estructural, aunque su propósito principal ha sido evaluar la capacidad de generalización entre tareas, no la complejidad del estado como tal.

En este contexto, se identifica un vacío experimental relevante: la ausencia de estudios que analicen de forma controlada la relación entre la **complejidad del espacio de observación** y el rendimiento de distintos algoritmos de RL manteniendo constantes las dinámicas, recompensas y reglas del entorno.

El presente trabajo busca contribuir a llenar ese vacío mediante el desarrollo de un entorno propio, *SimplePacmanEnv*, inspirado en el clásico juego Pac-Man. Dicho entorno permite parametrizar el nivel de información observable, desde representaciones mínimas (posición del agente y del fantasma) hasta configuraciones enriquecidas con comodines, distribución de monedas o vistas tipo imagen. Esta configuración experimental posibilita evaluar empíricamente cómo la cantidad y naturaleza de la información percibida influyen en la estabilidad del entrenamiento, la velocidad de convergencia y la calidad final de la política aprendida.

En conjunto, este análisis pretende aportar una visión empírica y reproducible sobre el equilibrio entre **simplicidad y expresividad** en el diseño de observaciones, contribuyendo así a una comprensión más profunda del papel de la percepción en el rendimiento de los agentes de aprendizaje por refuerzo.

# Capítulo 3

# Planificación del proyecto

## 1. Recursos necesarios

A continuación se detallan los recursos técnicos, de software y humanos requeridos para el desarrollo del proyecto. La selección de estos recursos se ha realizado considerando la naturaleza computacional del trabajo y la necesidad de garantizar reproducibilidad y eficiencia durante el proceso experimental.

| **Tipo de recurso** | Descripción |
|---|---|
| **Hardware** | Ordenador personal con CPU multinúcleo (Intel i7 o equivalente), 16 GB de RAM y GPU NVIDIA opcional para acelerar el entrenamiento de redes neuronales. |
| **Software** | Sistema operativo Linux/Windows, Python 3.10+, librerías *Gymnasium*, *Stable-Baselines3*, *NumPy*, *Matplotlib* y *TensorBoard* para la monitorización de entrenamientos. |
| **Control de versiones** | Repositorio privado en GitHub para gestionar código, scripts y documentación. |
| **Datos y experimentos** | No se requieren datasets externos: los datos se generan de forma simulada a través del entorno `SimplePacmanEnv`. |
| **Apoyo académico** | Tutorías con el tutor del TFM y revisión periódica según las fases definidas en las PEC. |

## 2. Planificación temporal y tareas

El proyecto se organiza siguiendo las fases de los módulos del TFM (de M1 a M5), que marcan los hitos principales de evaluación continua. Cada fase combina tareas de análisis,

desarrollo y documentación.

| **Fase / Módulo** | Periodo aproximado | Descripción de tareas principales | Hito asociado |
|---|---|---|---|
| **M1 – Definición y planificación del TFM** | 25 sep – 12 oct 2025 | ▪ Definir objetivos y alcance del proyecto.<br>▪ Revisar la viabilidad técnica y ética.<br>▪ Elaborar propuesta inicial y planificación. | Entrega M1 (12 oct) |
| **M2 – Estado del arte y fundamentos teóricos** | 13 oct – 2 nov 2025 | ▪ Revisión bibliográfica sobre aprendizaje por refuerzo, entornos Gym y algoritmos A2C, PPO y DQN.<br>▪ Identificación de trabajos similares y justificación del enfoque experimental. | Entrega M2 (2 nov) |
| **M3 – Diseño e implementación del sistema** | 3 nov – 14 dic 2025 | ▪ Implementar el entorno `SimplePacmanEnv`.<br>▪ Desarrollar scripts de entrenamiento (`train_a2c.py`, `train_ppo.py`, `train_dqn.py`).<br>▪ Validar funcionalidad y reproducibilidad.<br>▪ Documentar el código. | Entrega M3 (14 dic) |

| | | | |
|---|---|---|---|
| **M4 – Redacción y análisis de resultados** | 22 dic – 28 dic 2025 | ▪ Entrenar los modelos con distintas configuraciones de observación.<br>▪ Analizar resultados y elaborar gráficas comparativas.<br>▪ Redactar la memoria y preparar la presentación audiovisual. | Entrega M4 (21–28 dic); vídeo: 6 ene 2026 |
| **M5 – Defensa y cierre del proyecto** | 9 ene – 30 ene 2026 | ▪ Entrega final de la documentación al tribunal.<br>▪ Presentación y defensa pública del trabajo.<br>▪ Revisión y cierre de la memoria. | Entrega y defensa (9–30 ene) |

# 3. Diagrama de Gantt simplificado

El diagrama de Gantt que se presenta a continuación resume de forma visual la distribución temporal de las actividades y su solapamiento a lo largo del semestre académico. Cada marca indica aproximadamente una semana de dedicación dentro del mes correspondiente.

| **Actividad / Mes** | Sep | Oct | Nov | Dic | Ene | Feb |
|---|---|---|---|---|---|---|
| **Definición y planificación (M1)** | XX | X | | | | |
| **Estado del arte (M2)** | | XXX | | | | |
| **Diseño e implementación (M3)** | | X | XXXX | X | | |
| **Análisis y redacción (M4)** | | | | XXX | X | |
| **Defensa y cierre (M5)** | | | | | XXX | X |

*Nota:* cada “X” representa aproximadamente una semana de trabajo dentro del mes correspondiente.

# 4. Hitos principales del proyecto

Finalmente, se resumen los hitos más relevantes del proyecto, vinculados a las Pruebas de Evaluación Continua (PEC) y a las entregas oficiales del calendario académico. Estos hitos marcan los puntos de control que estructuran el avance del trabajo desde su definición inicial hasta la defensa final.

| **Hito** | Fecha aproximada | Descripción |
|---|---|---|
| **H1 – Definición del TFM (PEC1)** | 12 oct 2025 | Entrega del documento de definición y planificación del TFM. |
| **H2 – Estado del arte (PEC2)** | 2 nov 2025 | Entrega del marco teórico y análisis del contexto del trabajo. |
| **H3 – Implementación (PEC3)** | 14 dic 2025 | Finalización del entorno funcional y de los scripts de entrenamiento. |
| **H4 – Redacción preliminar y final (PEC4)** | 21–28 dic 2025 | Entrega del documento completo del TFM y del vídeo de presentación. |
| **H5 – Defensa final (PEC5)** | 30 ene 2026 | Presentación pública y defensa del trabajo ante el tribunal. |

# 5. Resumen de los productos del proyecto

No es necesario describir cada producto en detalle: esto se hará en los capítulos restantes del proyecto.

# 6. Breve descripción de los demás capítulos del informe

Breve descripción de los contenidos de cada capítulo y su relación con el resto del proyecto.

# 7. Métodos y recursos

En estas secciones, es necesario describir:

- Los aspectos más relevantes del diseño y desarrollo del proyecto.
- La metodología utilizada en el proceso de desarrollo, describiendo las alternativas posibles, las decisiones que se han tomado y los criterios utilizados para tomar estas decisiones.
- Una descripción de los productos que se han creado.

La estructura de estas secciones puede cambiar según el tipo de proyecto que se esté desarrollando.

# 8. Resultados

Describa los resultados obtenidos utilizando la metodología descrita anteriormente.

# 9. Conclusiones y trabajo futuro

Esta sección debe incluir lo siguiente:

- Una descripción de las conclusiones del trabajo.
- Una evaluación crítica del grado de logro de los objetivos iniciales.
- Una evaluación crítica de la planificación y metodología utilizadas en el proyecto.
- Considerando los desafíos de sostenibilidad, diversidad y ético-sociales vinculados al proyecto.
- Una discusión de temas para trabajo futuro potencial que no se hayan explorado en este proyecto.

# 10. Glosario

Definición de los términos y acrónimos más relevantes utilizados en este informe.

# Bibliografía


Maxime Chevalier-Boisvert, David Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. In *GitHub repository: https://github.com/maximecb/gym-minigrid*, 2018.

Karl Cobbe, Oleg Klimov, Chris Hesse, Tae Kim, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2048–2056, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2020.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *Proceedings of AAAI Fall Symposium Series*, 2015.

Timothée Lesort, Mathieu Seurin, Natalia Díaz-Rodríguez, and David Filliat. State representation learning for control: An overview. *arXiv preprint arXiv:1802.04181*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *Proceedings of the 33rd International Conference on Machine Learning*, pages 1928–1937, 2016.

OpenAI. Gymnasium documentation, 2023. URL https://gymnasium.farama.org/.

Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Noah Dormann, et al. Stable-baselines3: Reliable reinforcement learning implementations. *Journal*

*of Machine Learning Research*, 22(268):1–8, 2021. URL https://www.jmlr.org/papers/v22/20-1364.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

# 11. Anexos

1. -