# Análisis del problema

### Características del Environment

El entorno a resolver es *MountainCar-v0*, un entorno clásico de control de refuerzo donde un carro debe escalar una colina. Las características del environment son las siguientes:

* **Tipos de Estados**:
  + El espacio de estados es continuo y bidimensional, compuesto por:
    - Posición del carro (*position*): varía entre -1.2 y 0.6.
    - Velocidad del carro (*velocity*): varía entre -0.07 y 0.07.
  + Para el algoritmo Q-learning, discretizamos el espacio de estados en una cuadrícula de 20x20, resultando en 400 posibles estados discretos.
* **Tipos de Acciones**:
  + El espacio de acciones es discreto, con tres posibles acciones:
    - *0*: Empujar el carro hacia la izquierda.
    - *1*: No empujar el carro (acción neutral).
    - *2*: Empujar el carro hacia la derecha.
* **Recompensas**:
  + El agente recibe una recompensa de *-1* por cada paso que no alcanza la meta (la cima de la colina con posición >= 0.5).
  + No hay recompensas positivas o adicionales por alcanzar la meta, pero al llegar, el episodio termina.

### Recompensa Promedio Esperada

En *MountainCar-v0*, el objetivo es que el agente aprenda a alcanzar la cima lo más rápido posible, minimizando el número de pasos y, por lo tanto, maximizando la recompensa acumulada (que es negativa). No hay una recompensa promedio estándar esperada, ya que depende del rendimiento del agente, pero un agente bien entrenado debe aprender a alcanzar la meta en un número mínimo de pasos, logrando así una recompensa promedio por episodio que tiende a ser menos negativa con el tiempo.

# Argumentación de la elección del algoritmo y pasos realizados durante el entrenamiento

### Elección del algoritmo

Se ha elegido el algoritmo **Q-learning**, un método de aprendizaje por refuerzo fuera de línea, debido a su capacidad para aprender políticas óptimas mediante la iteración de valores Q (acción-estado). Q-learning es adecuado para entornos con espacios de estado discretos, como el que se obtiene después de discretizar el continuo en *MountainCar-v0*.

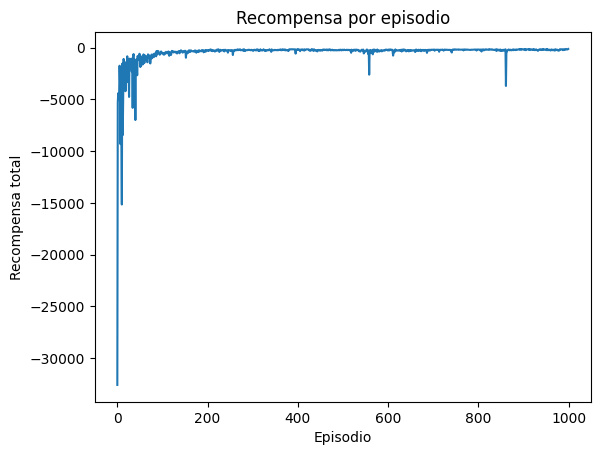
### Pasos realizados durante el entrenamiento

1. **Inicialización**:
   1. Se crea una tabla Q con ceros para todos los pares de estado-acción.
   2. Se establecen los parámetros de aprendizaje: tasa de aprendizaje (*alpha*), factor de descuento (*gamma*), *epsilon* inicial, y su tasa de decaimiento.
2. **Discretización del espacio de estados**:
   1. El espacio de estados continuo se discretiza en una cuadrícula de 20x20 para facilitar el uso de la tabla Q.
3. **Política de exploración**:
   1. Se utiliza una política *epsilon-greedy* para balancear la exploración y la explotación:
      1. Al principio del entrenamiento, se usa un valor alto de *epsilon* (0.9) para fomentar la exploración.
      2. Epsilon decae gradualmente en cada episodio (*epsilon\_decay* = 0.995) para favorecer la explotación de las acciones conocidas como buenas a medida que el agente aprende.
4. **Bucle de entrenamiento:**
   1. Para cada episodio:
      1. El entorno se reinicia y se obtiene el estado inicial discreto.
      2. En cada paso del episodio, el agente selecciona una acción basada en la política *epsilon-greedy*.
      3. Se ejecuta la acción en el entorno, obteniendo el nuevo estado y recompensa.
      4. Se actualiza la tabla Q utilizando la fórmula de Q-learning.
      5. Se acumula la recompensa total del episodio.
      6. Se ajusta *epsilon* para reducir la exploración a medida que avanza el entrenamiento.
5. **Grabación de videos**:
   1. Se graban videos de los episodios, cada 100 episodios, para analizar visualmente el comportamiento del agente.

# Presentación de resultados

### Evolución del aprendizaje del agente

Para evaluar el rendimiento del agente, se grafica la recompensa total por episodio a lo largo de los 1000 episodios de entrenamiento. La gráfica de recompensas muestra la evolución del aprendizaje del agente, indicando cómo las políticas aprendidas impactan en la capacidad del agente para alcanzar la meta.



### Análisis de resultados

* **Inicio del entrenamiento**:
  + Al comienzo del entrenamiento, las recompensas por episodio son altamente negativas debido a la exploración intensiva y la falta de conocimiento del entorno por parte del agente.
* **Medio del entrenamiento**:
  + A medida que el agente aprende de sus experiencias, la recompensa promedio comienza a mejorar, reflejando un incremento en la frecuencia con la que el agente alcanza la cima de la colina.
  + Se observan picos negativos durante el entrenamiento, que pueden deberse a, por ejemplo, exploración residual. Es decir, aunque *epsilon* disminuye a lo largo del tiempo, es posible que aún haya momentos en los que el agente elija acciones exploratorias debido a la política *epsilon-greedy*. Esto puede llevar a decisiones subóptimas que resulten en grandes penalizaciones.
* **Final del Entrenamiento:**
  + Al final del entrenamiento, la recompensa por episodio se estabiliza en valores menos negativos, indicando que el agente ha aprendido una política efectiva para alcanzar la meta de manera más eficiente.

El análisis de la evolución de las recompensas por episodio proporciona una métrica clave del rendimiento del agente, demostrando cómo el agente mejora su comportamiento con el tiempo y cuán efectiva es la política aprendida.