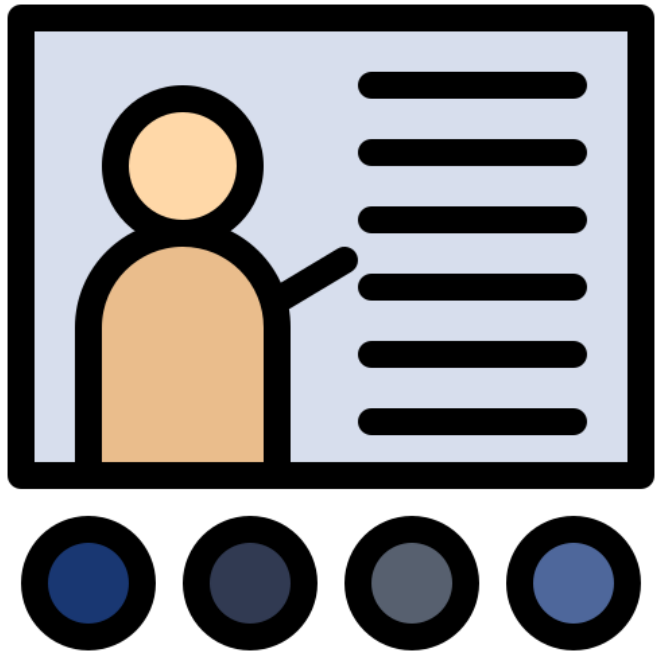
Modelado de Incertidumbre en Inteligencia Artificial

Solución del problema Cart-Pole evaluando algoritmos de Aprendizaje por Refuerzo

Ponentes:

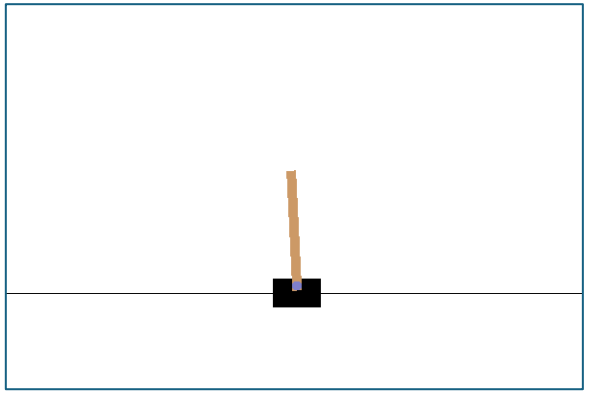
- **Juan Herencia**
- **Jeyson Lino**
- **José Zúñiga**

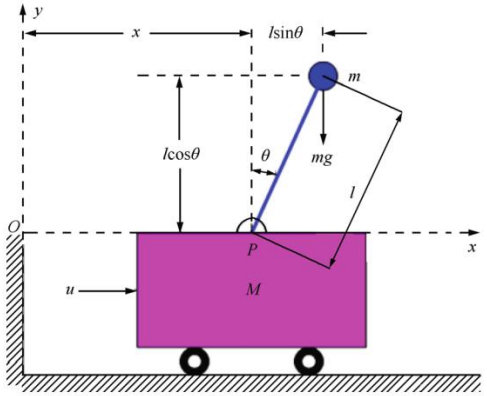
1. Introducción



Introducción

Descripción del problema





Cart-Pole es un péndulo con un centro de gravedad por encima de su punto de pivote. Es inestable, pero se puede controlar moviendo el punto de pivote por debajo del centro de masa.

El problema es la inestabilidad del poste cuando el carrito está en movimiento horizontal. La solución es mover el carrito de manera que el poste se mantenga en posición vertical.

Introducción

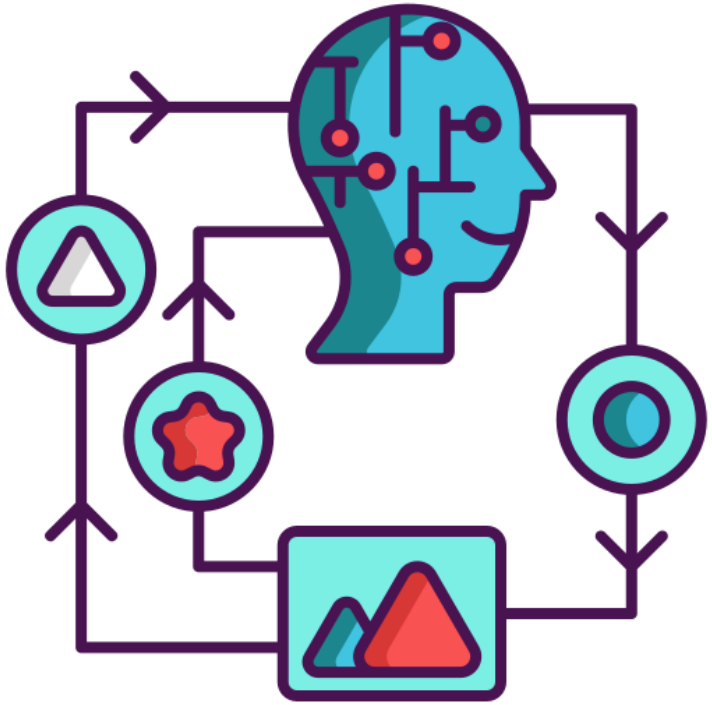
Objetivos del proyecto

Encontrar la solución al problema del Cart-Pole usando algoritmos de aprendizaje por refuerzo que puedan controlar sistemas dinámicos.

Criterio de éxito

El problema se considera resuelto si el poste se mantiene en equilibrio por una cierta cantidad de pasos de tiempo.

2. Adaptación del problema



Adaptación del problema

El objetivo es mover el carrito de manera que el poste se mantenga en posición vertical (o lo más cercano a la vertical posible) sin que se caiga.

Agente: sistema que permite el ingreso de valores

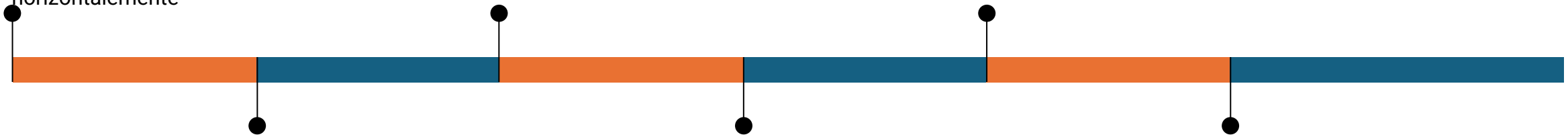
Entorno: carrito que se puede mover horizontalemente

Acción (a)

Mover el carrito hacia la izquierda o hacia la derecha, aplicando fuerzas

Política (Pi)

Estrategia que define qué acción tomar en cada estado



Estado (s)

- Posición del carrito
- Velocidad del carrito
- Ángulo del poste
- Velocidad angular del poste

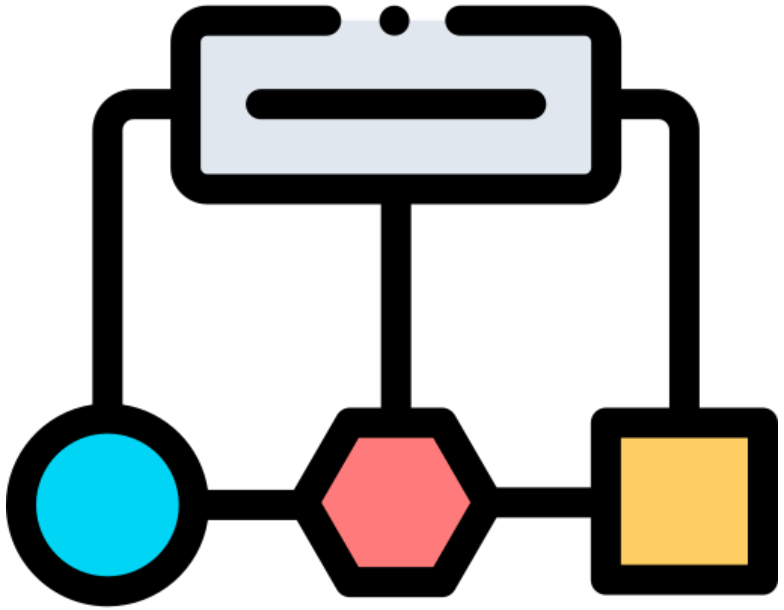
Recompensa (r)

Positiva si el poste está equilibrado y negativa si éste se cae

Valor (V): recompensa esperada desde un estado, siguiendo una política

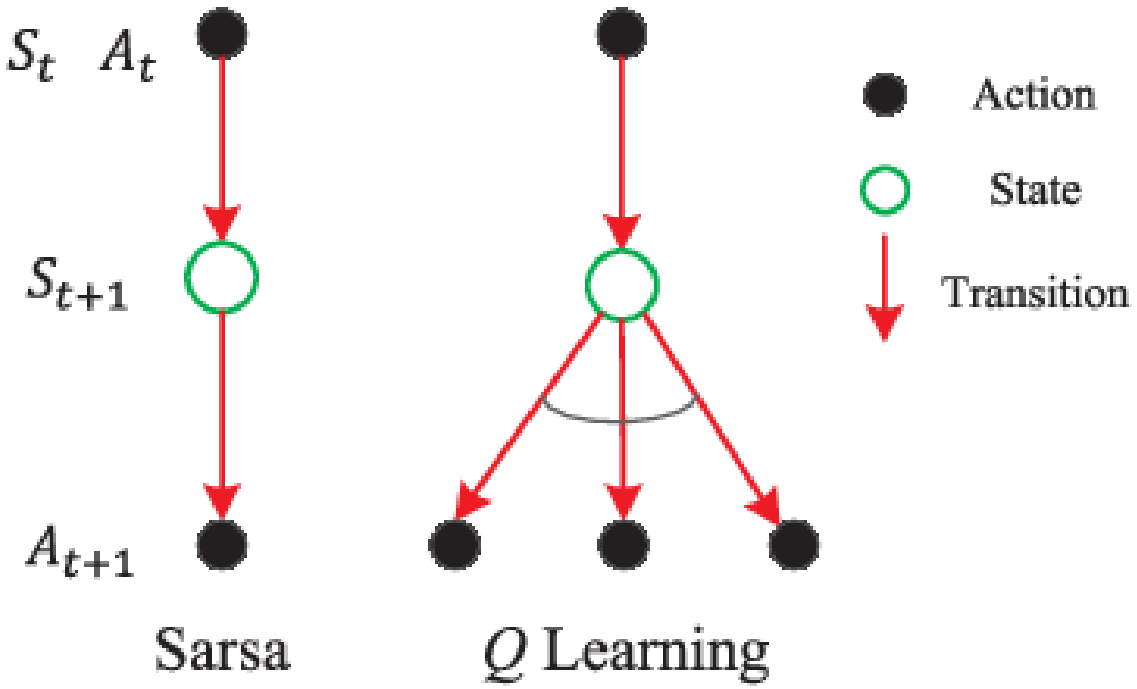
Función Q (Q): recompensa esperada de tomar una acción en un estado dado, luego de seguir una política

3. Metodología



Metodología

Selección de algoritmos



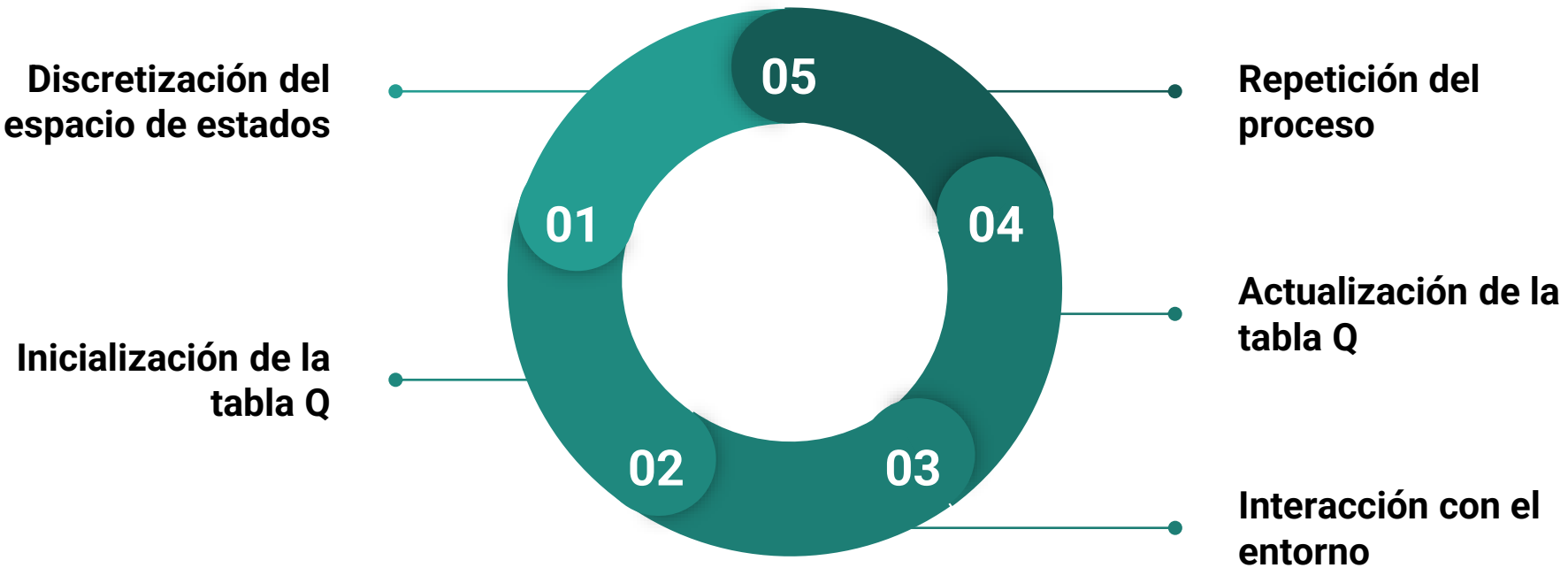
Con estos algoritmos se gestionarán las variables de estado:

- Posición del carrito
- Velocidad del carrito
- Ángulo del poste
- Velocidad angular del poste

Metodología

Algoritmo Q-Learning

$$Q(s_t, at_) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

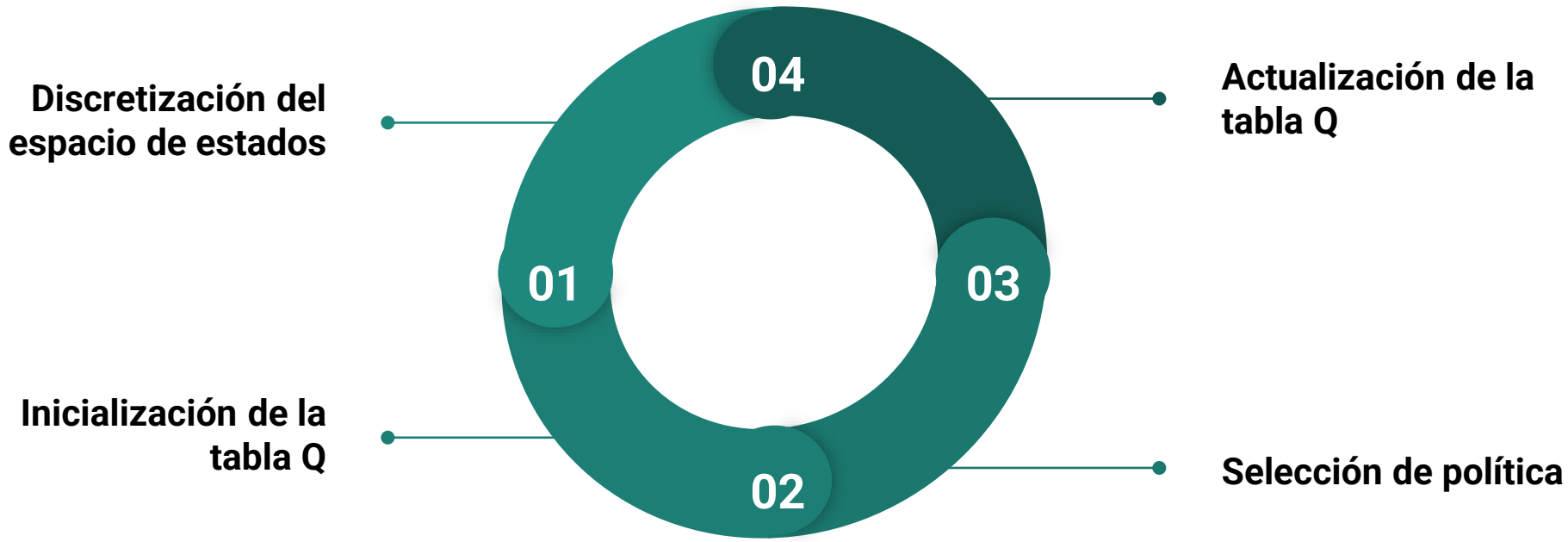


Pasos para adaptar el algoritmo Q-Learning

Metodología

Algoritmo Sarsa

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$



Pasos para adaptar el algoritmo Sarsa

4. Herramientas utilizadas

Gymnasium

Herramientas utilizadas

CartPole - v0

- Versión simple, adecuada para primeros experimentos en aprendizaje por refuerzo.
- Límite de episodios corto (200 pasos).
- El episodio termina si el poste se inclina más de 15 grados desde la vertical.
- El episodio termina si el carro se desplaza más de 2.4 unidades desde el centro.
- El episodio termina si el poste se mantiene en posición durante 200 pasos.
- La recompensa es +1 por cada paso de tiempo en el que el poste se mantiene en posición.

Herramientas utilizadas

CartPole - v1

- Límites más estrictos en ángulo del poste.
- Evaluación más exhaustiva del rendimiento del agente (500 pasos).
- Ideal para aplicaciones que han superado a CartPole - v0.
- El episodio termina si el poste se inclina más de 12 grados desde la vertical (más estricto que en v0).
- El episodio termina si el carro se desplaza más de 2.4 unidades desde el centro (igual que en v0).
- El episodio termina si el poste se mantiene en posición durante 500 pasos (más largo que en v0).
- La recompensa es +1 por cada paso de tiempo en el que el poste se mantiene en posición.

5. Implementación

Implementación

Q - Learning

Configuración inicial

```
# Configuraciones del entorno y del algoritmo
env = gym.make('CartPole-v1')
n_actions = env.action_space.n
n_bins = (10, 10, 10, 10)  # Número N limites o valores discretos para cada dimensión (car_position, car_velocity, pole_angle, pole_velocity)
alpha = 0.1  # Tasa de aprendizaje
gamma = 0.99  # Factor de descuento
epsilon = 1.0  # Tasa de exploración inicial
epsilon_decay = 0.995 # decaimiento
epsilon_min = 0.01 # 0.01
n_episodes = 10000 # ##### 500
goal_avg_reward = 500  # Recompensa media objetivo para considerar el problema resuelto
max_steps = 5000 # cant máxima de pasos por episodio
window_size = 100  # Ventana para calcular la media móvil
```

Implementación

Q - Learning

Discretización

```python
# Límites para la discretización
# Estos límites crean N - 1 segmentos, por eso se le agrega 1
state_bins = [
    np.linspace(-4.8, 4.8, n_bins[0] - 1 ),  # car_position
    np.linspace(-3.0, 3.0, n_bins[1] - 1),  # car_velocity
    np.linspace(-0.418, 0.418, n_bins[2] - 1 ),  # pole_angle (~24° en radianes)
    np.linspace(-3.0, 3.0, n_bins[3] - 1)   # pole_velocity
]

# Función para discretizar los estados
# Se obtiene los valores discretos  1,2,3, ..., N-1 segmentos
def discretize_state(state):
    car_position, car_velocity, pole_angle, pole_velocity = state
    state_discrete = [
        np.digitize(car_position, state_bins[0]),
        np.digitize(car_velocity, state_bins[1]),
        np.digitize(pole_angle, state_bins[2]),
        np.digitize(pole_velocity, state_bins[3])
    ]
    return tuple(state_discrete)
```

Implementación

Algoritmo de entrenamiento

```python
# Entrenamiento con Q-Learning
rewards_per_episode = []
mean_durations = []
mean_duration = 0 # corrección!
t_ini = time.time()
for episode in range(n_episodes):
    state, _ = env.reset()
    state = discretize_state(state)
    total_reward = 0

    for t in range(max_steps):
        action = choose_action(state)
        next_state, reward, terminated, truncated, _ = env.step(action)
        next_state = discretize_state(next_state)

        # Actualización de la tabla Q
        best_next_action = np.argmax(q_table[next_state])
        q_target = reward + gamma * q_table[next_state][best_next_action]
        q_table[state][action] += alpha * (q_target - q_table[state][action])

        state = next_state
        total_reward += reward

        if terminated or truncated:
            break

    rewards_per_episode.append(total_reward)

    # Calcular la duración media de los últimos 100 episodios
    if episode >= window_size:
        mean_duration = np.mean(rewards_per_episode[-window_size:])
        mean_durations.append(mean_duration)
    else:
        mean_durations.append(np.mean(rewards_per_episode))

    # Decaimiento de epsilon
    if epsilon > epsilon_min:
        epsilon *= epsilon_decay

    # Verificar si se ha resuelto el problema
    if episode >= window_size and mean_duration >= goal_avg_reward:
        print(f'Problema resuelto en {episode + 1} episodios con una duración media de {mean_duration:.2f}')
        file_csv.writelines(f'{episode + 1},{total_reward},{mean_duration:.2f}\n')
        break
```

Q - Learning

Implementación

SARSA

Configuración inicial

```
# Configuraciones del entorno y del algoritmo
env = gym.make('CartPole-v1')
n_actions = env.action_space.n
n_bins = (10, 10, 10, 10)  # Número N limites o valores discretos para cada dimensión
alpha = 0.1  # Tasa de aprendizaje
gamma = 0.99  # Factor de descuento
epsilon = 1.0  # Tasa de exploración inicial
epsilon_decay = 0.995  # Decaimiento
epsilon_min = 0.01  # Mínimo valor de epsilon
n_episodes = 20000  # Número de episodios
goal_avg_reward = 500  # Recompensa media objetivo para considerar el problema resuelto
max_steps = 5000  # Cantidad máxima de pasos por episodio
window_size = 100  # Ventana para calcular la media móvil
```

Implementación

SARSA

Discretización

```python
# Límites para la discretización
state_bins = [
    np.linspace(-4.8, 4.8, n_bins[0] - 1),  # car_position
    np.linspace(-3.0, 3.0, n_bins[1] - 1),  # car_velocity
    np.linspace(-0.418, 0.418, n_bins[2] - 1),  # pole_angle (~24° en radianes)
    np.linspace(-3.0, 3.0, n_bins[3] - 1)   # pole_velocity
]

# Función para discretizar los estados
def discretize_state(state):
    car_position, car_velocity, pole_angle, pole_velocity = state
    state_discrete = [
        np.digitize(car_position, state_bins[0]),
        np.digitize(car_velocity, state_bins[1]),
        np.digitize(pole_angle, state_bins[2]),
        np.digitize(pole_velocity, state_bins[3])
    ]
    return tuple(state_discrete)

# Inicialización de la tabla Q
q_table = np.zeros(n_bins + (n_actions,))
```

Implementación

Algoritmo de entrenamiento

```python
# Entrenamiento con SARSA
rewards_per_episode = []
mean_durations = []
mean_duration = 0
t_ini = time.time()

for episode in range(n_episodes):
    state, _ = env.reset()
    state = discretize_state(state)
    action = choose_action(state)
    total_reward = 0

    for t in range(max_steps):
        next_state, reward, terminated, truncated, _ = env.step(action)
        next_state = discretize_state(next_state)
        next_action = choose_action(next_state)

        # Actualización de la tabla Q con SARSA
        q_target = reward + gamma * q_table[next_state][next_action]
        q_table[state][action] += alpha * (q_target - q_table[state][action])

        state = next_state
        action = next_action
        total_reward += reward

        if terminated or truncated:
            break

    rewards_per_episode.append(total_reward)

    # Calcular la duración media de los últimos 100 episodios
    if episode >= window_size:
        mean_duration = np.mean(rewards_per_episode[-window_size:])
        mean_durations.append(mean_duration)
    else:
        mean_durations.append(np.mean(rewards_per_episode))

    # Decaimiento de epsilon
    if epsilon > epsilon_min:
        epsilon *= epsilon_decay

    # Verificar si se ha resuelto el problema
    if episode >= window_size and mean_duration >= goal_avg_reward:
        print(f'Problema resuelto en {episode + 1} episodios con una duración media de {mean_duration:.2f}')
        file_csv.writelines(f'{episode + 1},{total_reward},{mean_duration:.2f}\n')
        break
```

SARSA

6. Experimentación y resultados

Experimentación 1

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 10000
Pasos x episodio = 5000

Problema resuelto en 8491 episodios con una duración media de 500.00

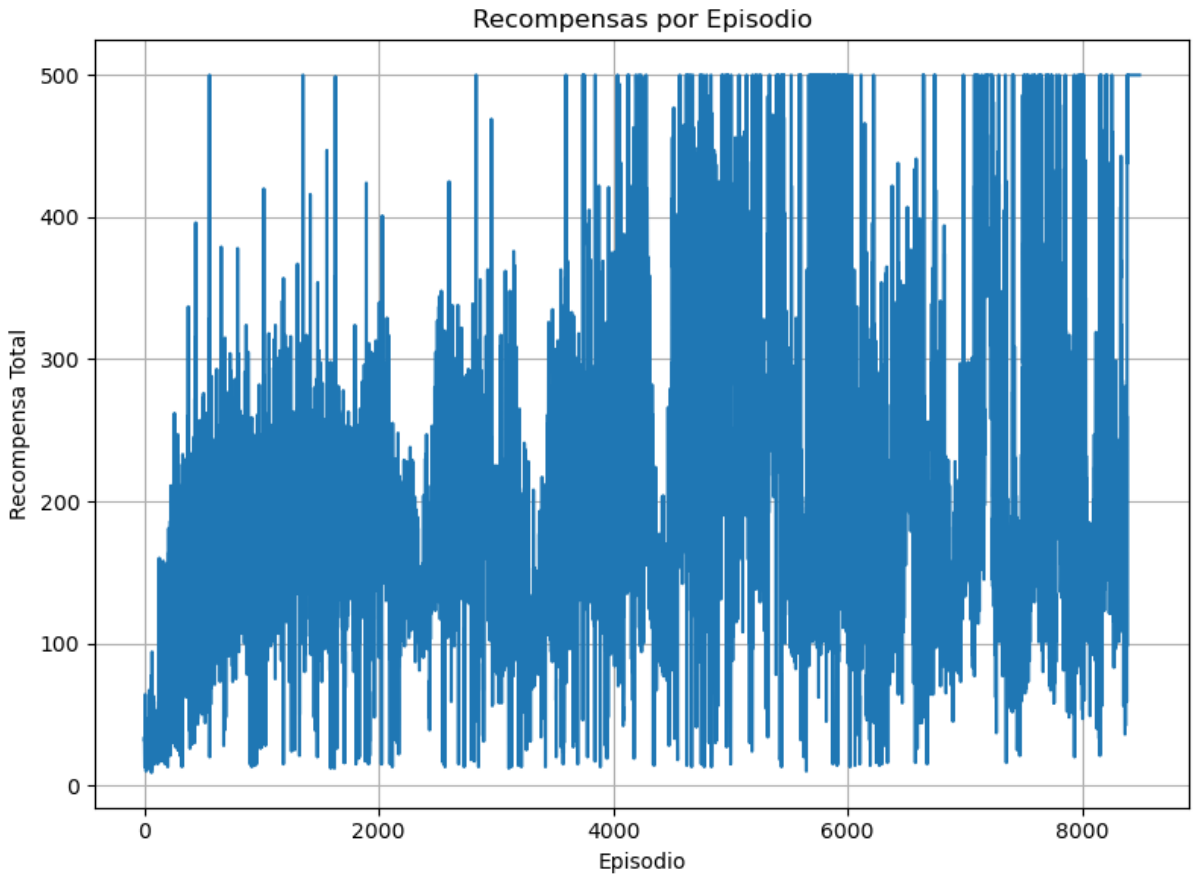
Cálculo de datos (salida_QL_1.csv)

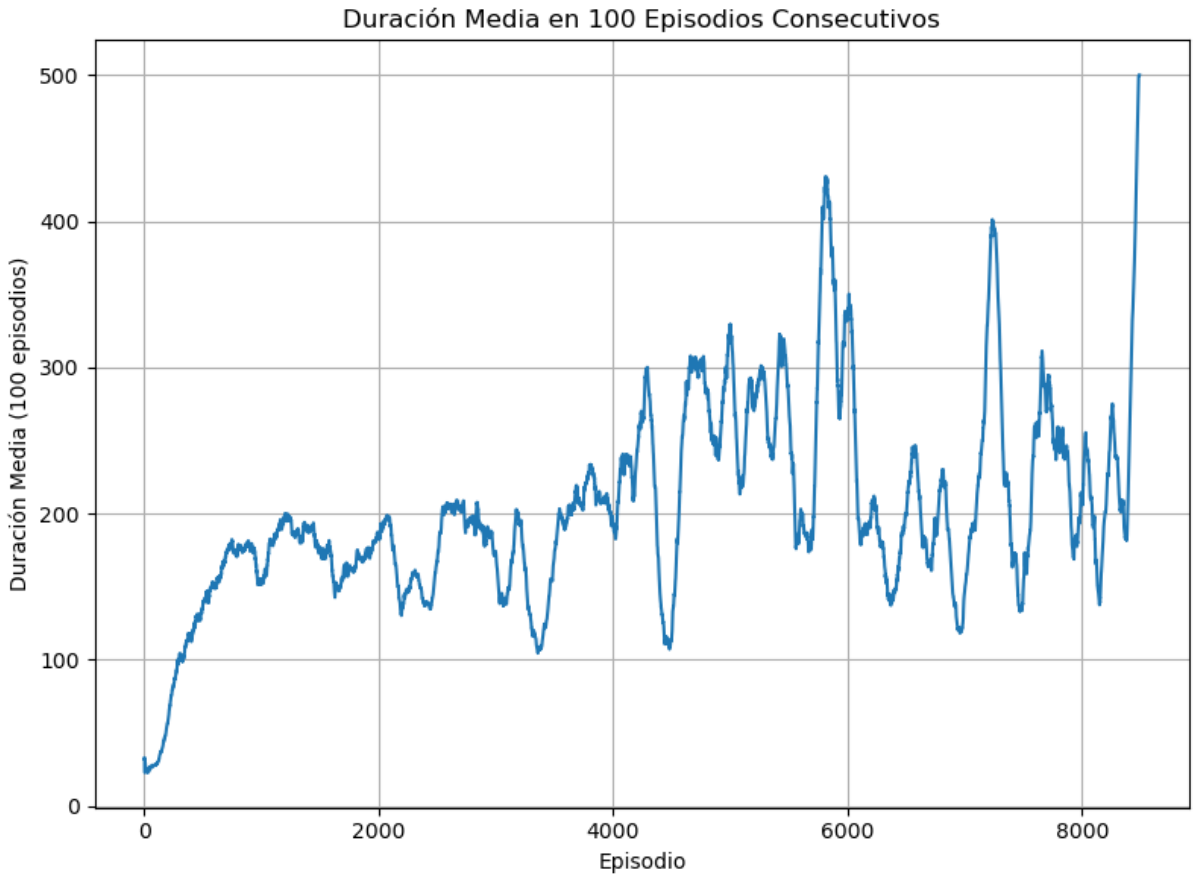
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,32.0,0.00
2,34.0,0.00
3,29.0,0.00
4,14.0,0.00
5,13.0,0.00
6,14.0,0.00
7,34.0,0.00
8,64.0,0.00
9,28.0,0.00
10,16.0,0.00
    ...
8464,500.0,422.21
8465,500.0,426.54
8466,500.0,430.47
8467,500.0,434.38
8468,500.0,438.72
8469,500.0,443.36
8470,500.0,445.93
8471,500.0,449.68
8472,500.0,454.26
8473,500.0,456.45
8474,500.0,460.64
8475,500.0,464.25
8476,500.0,467.19
8477,500.0,471.58
8478,500.0,475.39
8479,500.0,478.09
8480,500.0,481.75
8481,500.0,484.66
8482,500.0,488.55
8483,500.0,492.96
8484,500.0,495.36
8485,500.0,499.38
8486,500.0,499.38
8487,500.0,499.38
8488,500.0,499.38
8489,500.0,499.38
8490,500.0,499.38
8491,500.0,500.00
```

Experimentación 1

1. Algoritmo Q-Learning

Gráfico





Problema resuelto en 8491 episodios con una duración media de 500.00

Experimentación 2

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 10000
Pasos x episodio = 5000

Tiempo realizado : 90.44962549209595 segundos

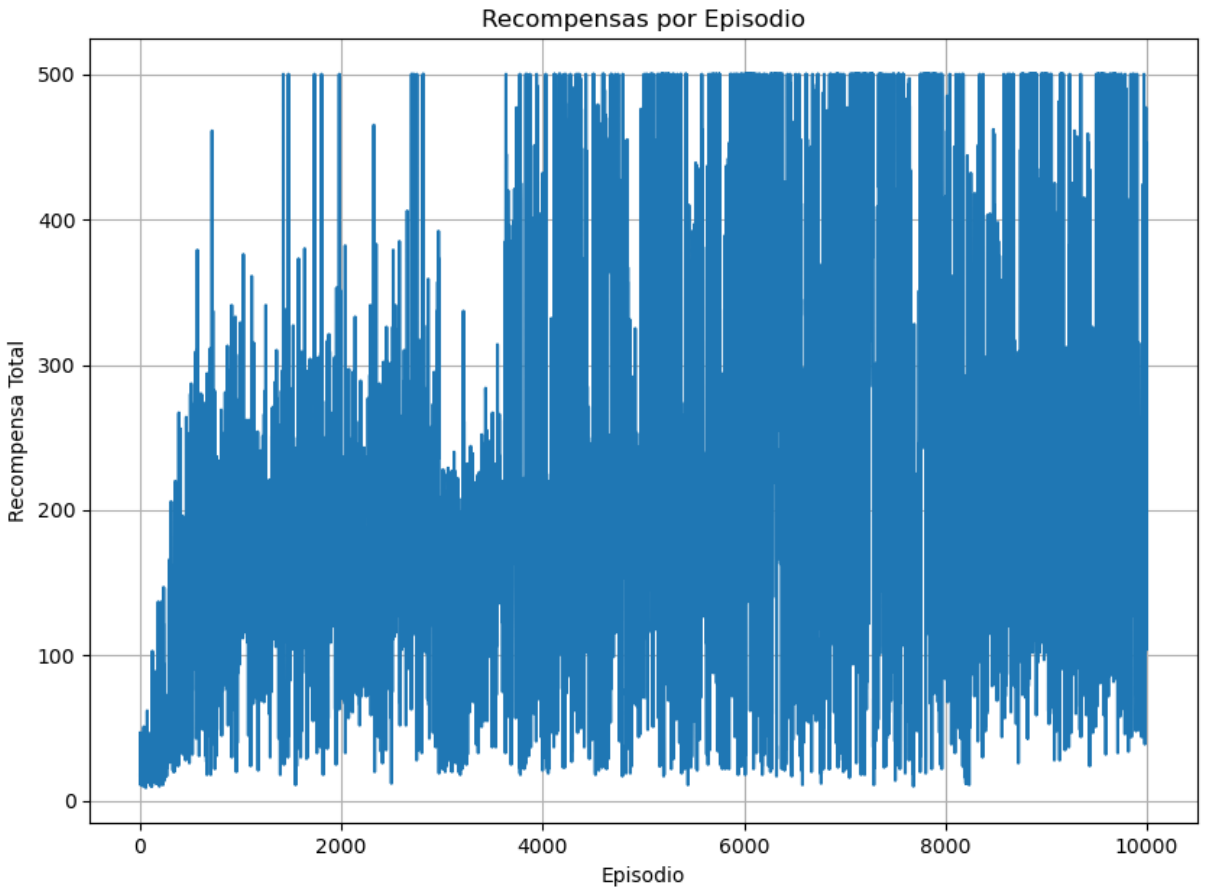
Con los parámetros indicados no se ha podido resolver

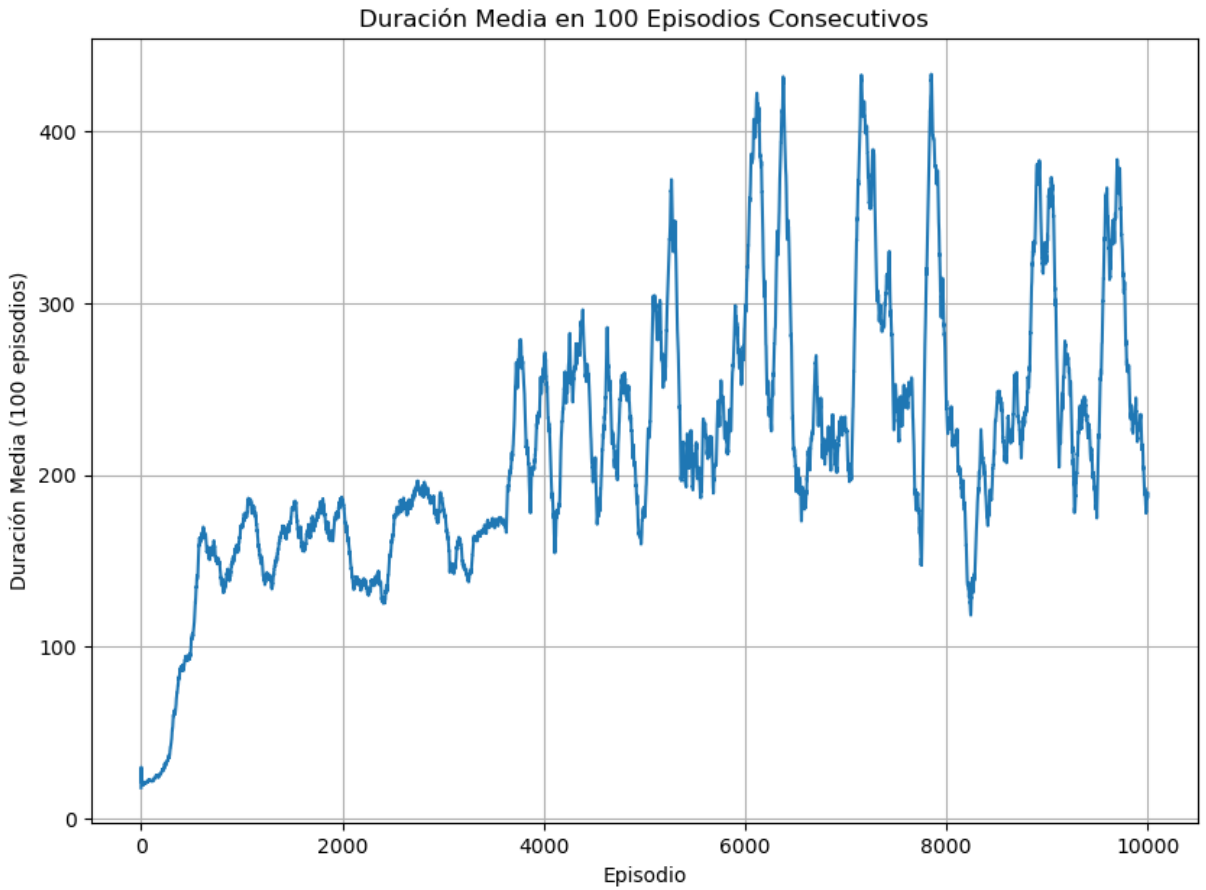
Cálculo de datos (salida_QL_2.csv)

```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,18.0,0.00
2,25.0,0.00
3,47.0,0.00
4,13.0,0.00
5,16.0,0.00
6,11.0,0.00
7,19.0,0.00
8,21.0,0.00
9,14.0,0.00
10,23.0,0.00
...
9979,130.0,184.12
9980,121.0,183.48
9981,39.0,182.51
9982,91.0,181.66
9983,112.0,177.78
9984,211.0,178.62
9985,222.0,179.03
9986,146.0,178.73
9987,261.0,179.49
9988,132.0,179.21
9989,341.0,181.05
9990,383.0,182.96
9991,477.0,185.45
9992,159.0,184.95
9993,199.0,186.46
9994,338.0,189.11
9995,181.0,188.74
9996,177.0,189.78
9997,174.0,189.34
9998,104.0,189.44
9999,106.0,186.75
10000,452.0,189.46
```

Experimentación 2

1. Algoritmo Q-Learning





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 3

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 97.7299702167511 segundos

Problema resuelto en 11455 episodios con una duración media de 500.00

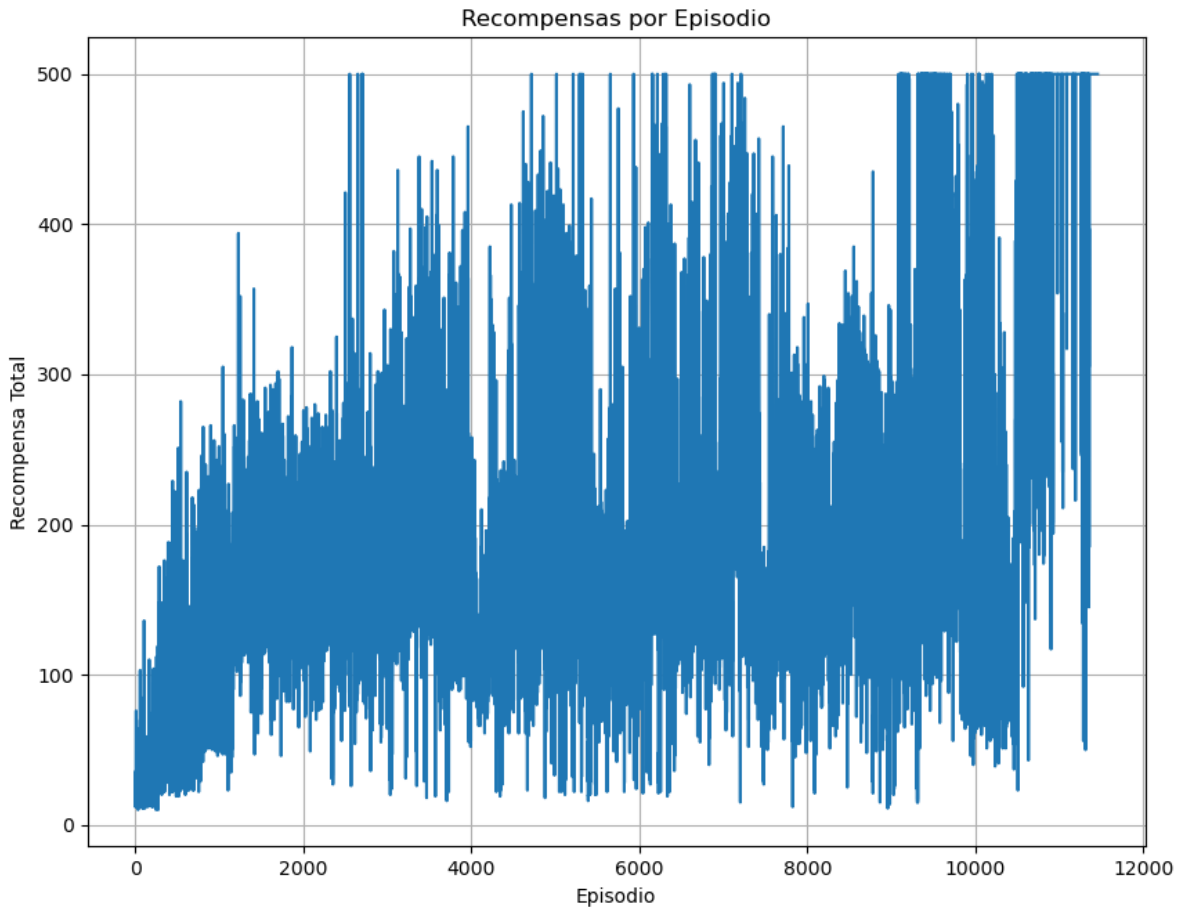
Cálculo de datos (salida_QL_3.csv)

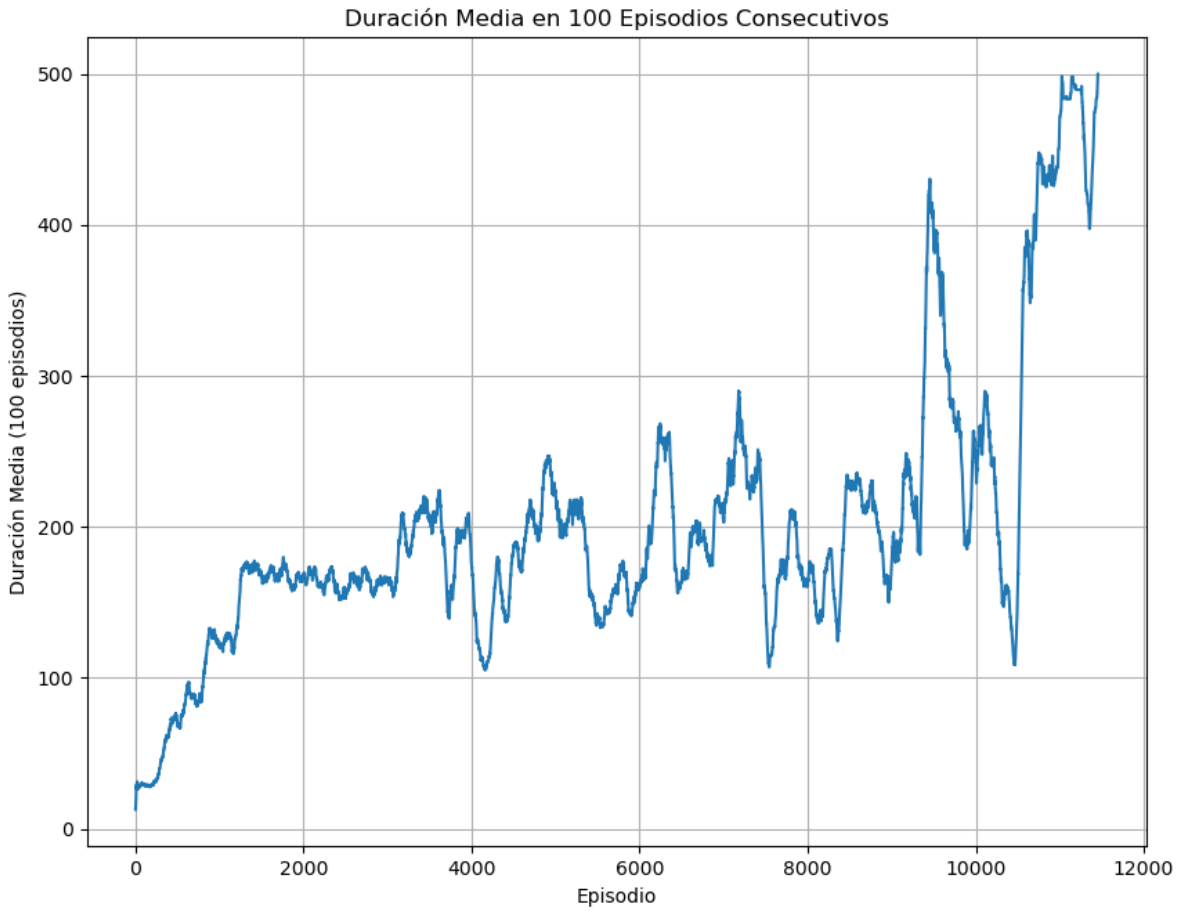
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,13.0,0.00
2,12.0,0.00
3,15.0,0.00
4,20.0,0.00
5,35.0,0.00
6,16.0,0.00
7,29.0,0.00
8,36.0,0.00
9,14.0,0.00
10,46.0,0.00
...
11437,500.0,483.53
11438,500.0,483.53
11439,500.0,483.53
11440,500.0,484.53
11441,500.0,484.53
11442,500.0,484.53
11443,500.0,484.53
11444,500.0,484.53
11445,500.0,484.90
11446,500.0,487.87
11447,500.0,487.92
11448,500.0,487.92
11449,500.0,489.64
11450,500.0,490.32
11451,500.0,493.87
11452,500.0,493.87
11453,500.0,495.82
11454,500.0,496.85
11455,500.0,500.00
```

Experimentación 3

1. Algoritmo Q-Learning

Gráfico





Problema resuelto en 11455 episodios con una duración media de 500.00

Experimentación 4

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.2
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 83.83568835258484 segundos

Problema resuelto en 8427 episodios con una duración media de 500.00

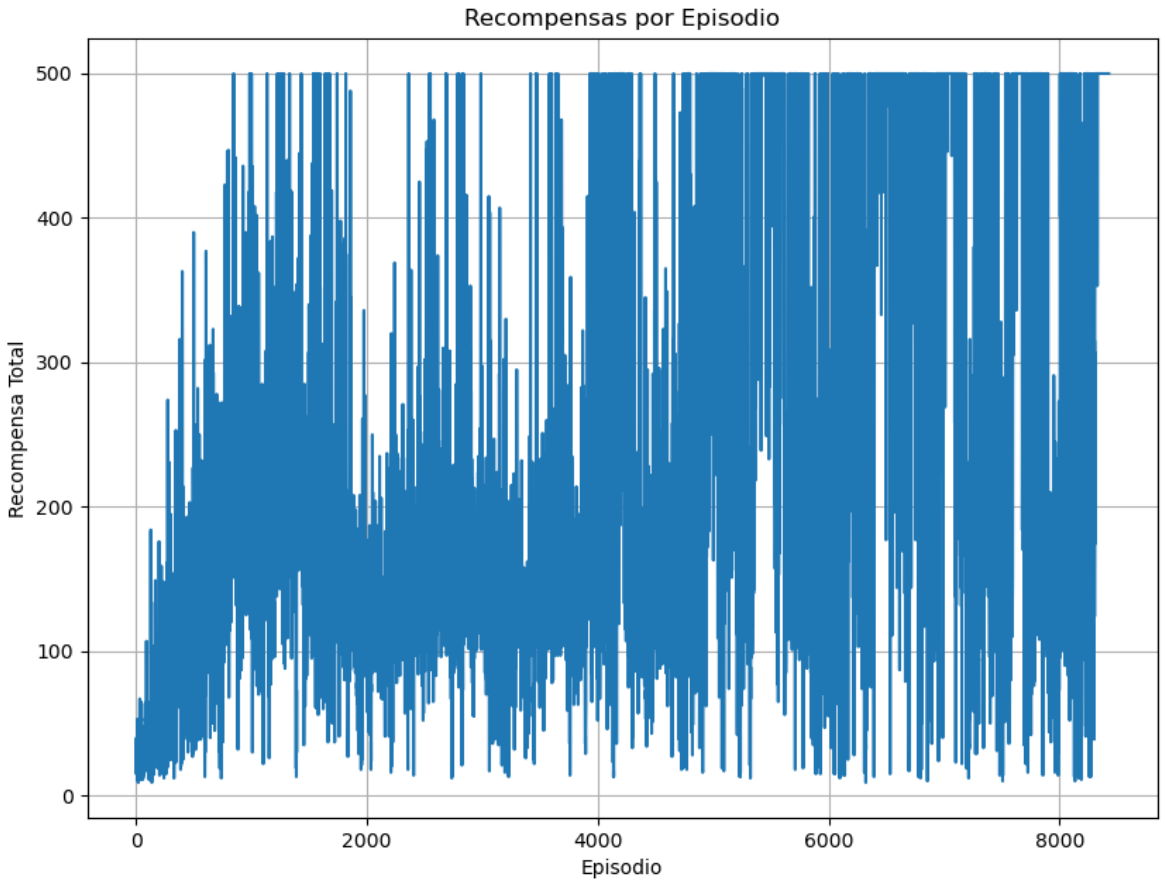
Cálculo de datos (salida_QL_4.csv)

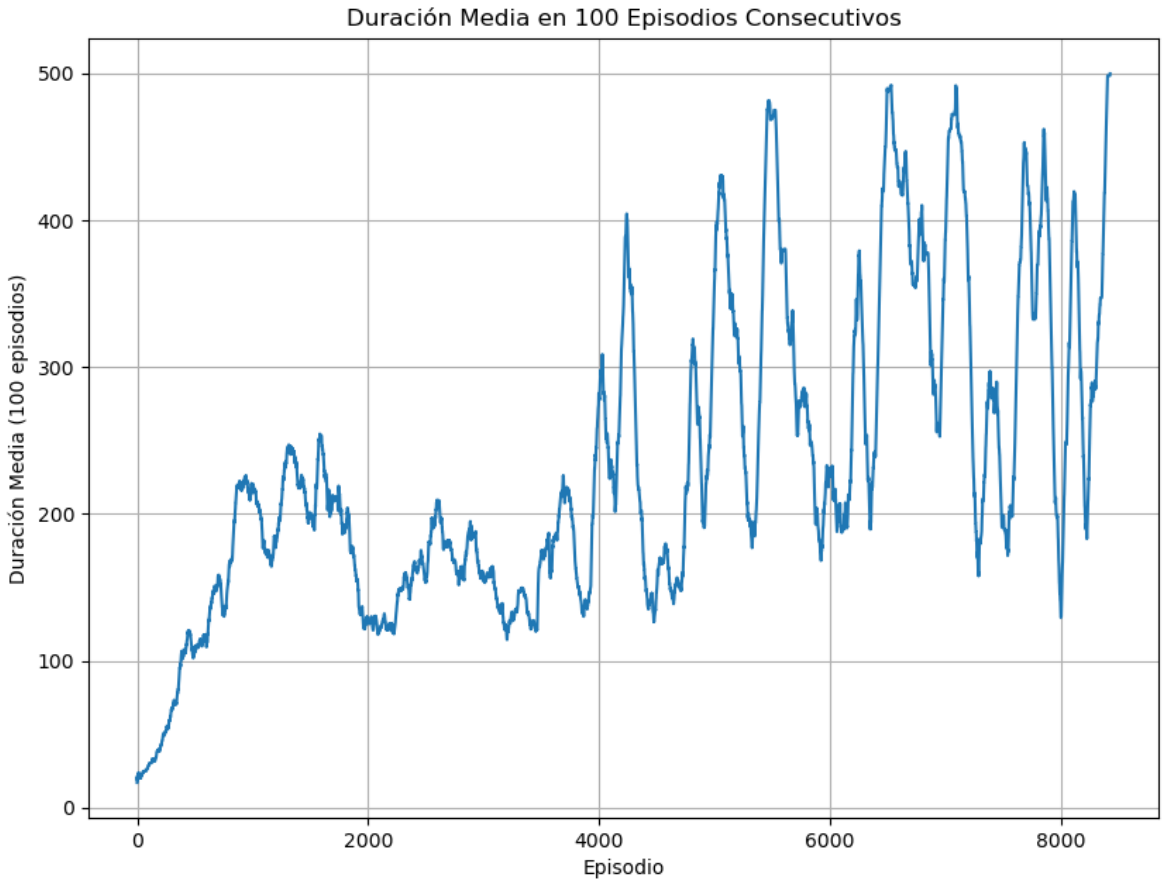
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,17.0,0.00
2,25.0,0.00
3,18.0,0.00
4,17.0,0.00
5,15.0,0.00
6,40.0,0.00
7,20.0,0.00
8,17.0,0.00
9,15.0,0.00
10,15.0,0.00
...
8406,500.0,498.53
8407,500.0,498.53
8408,500.0,498.53
8409,500.0,498.53
8410,500.0,498.53
8411,500.0,498.53
8412,500.0,498.53
8413,500.0,498.53
8414,500.0,498.53
8415,500.0,498.53
8416,500.0,498.53
8417,500.0,498.53
8418,500.0,498.53
8419,500.0,498.53
8420,500.0,498.53
8421,500.0,498.53
8422,500.0,498.53
8423,500.0,498.53
8424,500.0,498.53
8425,500.0,498.53
8426,500.0,498.53
8427,500.0,500.00
```

Experimentación 4

1. Algoritmo Q-Learning

Gráfico





Problema resuelto en 8427 episodios con una duración media de 500.00

Experimentación 5

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.3
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 201.11599922180176 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

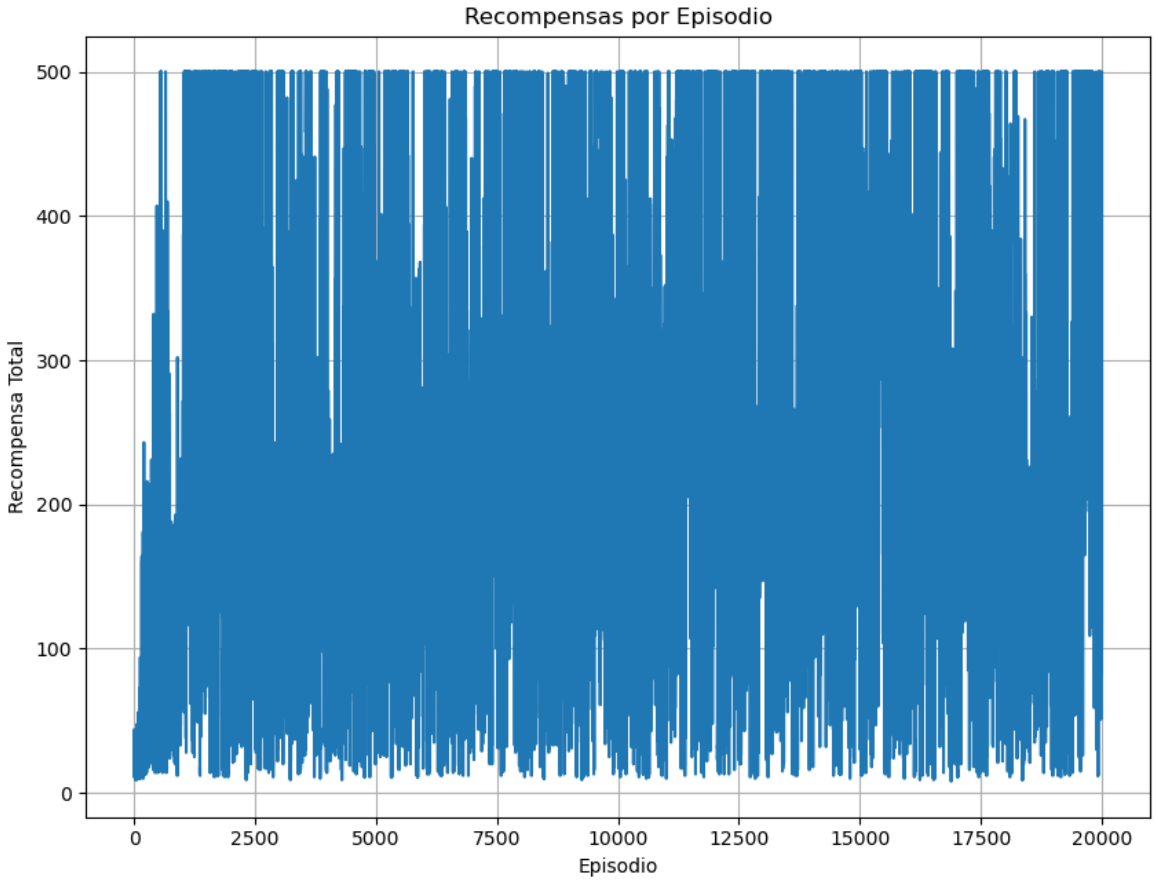
Cálculo de datos (salida_QL_5.csv)

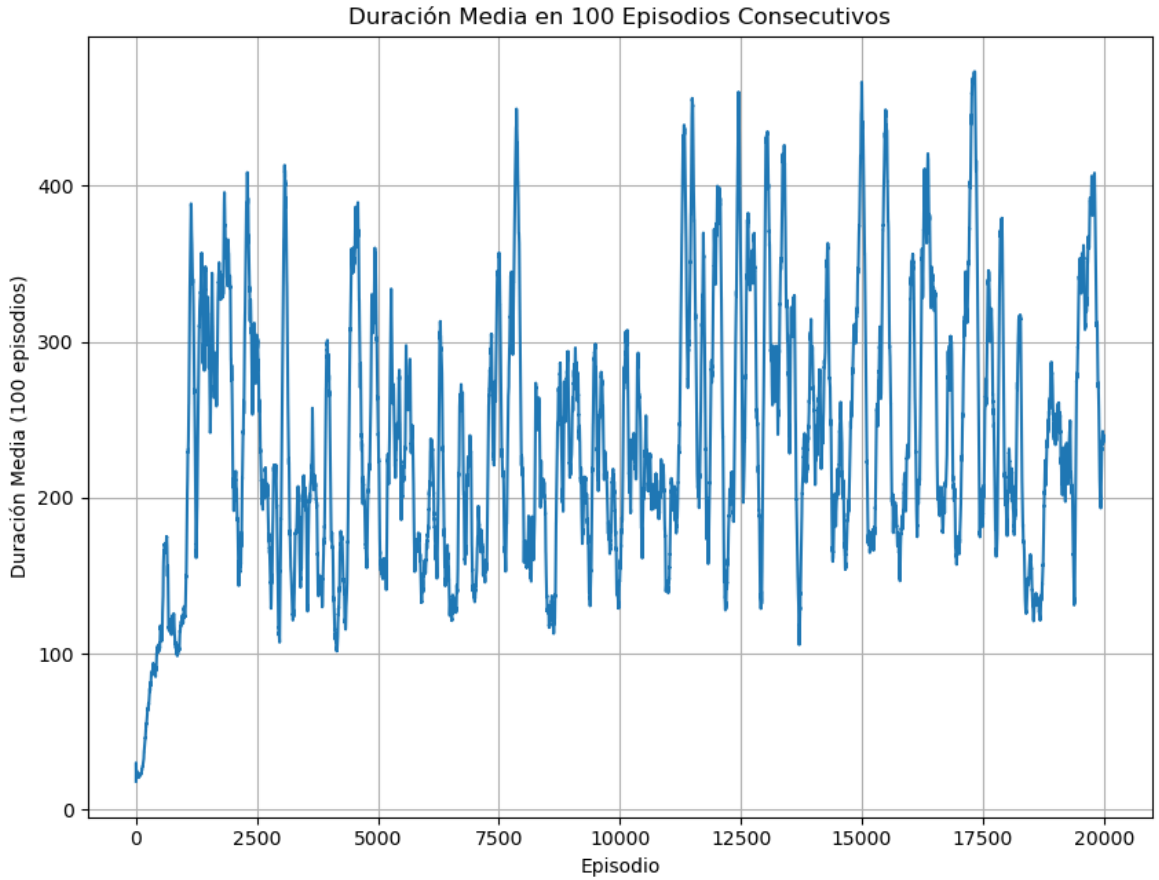
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,18.0,0.00
2,28.0,0.00
3,44.0,0.00
4,11.0,0.00
5,26.0,0.00
6,20.0,0.00
7,29.0,0.00
8,16.0,0.00
9,35.0,0.00
10,20.0,0.00
...
19981,51.0,237.34
19982,371.0,239.38
19983,285.0,239.23
19984,194.0,236.17
19985,201.0,234.72
19986,388.0,236.66
19987,127.0,236.62
19988,129.0,236.77
19989,210.0,238.57
19990,82.0,238.14
19991,234.0,239.18
19992,234.0,240.32
19993,144.0,239.35
19994,255.0,238.11
19995,147.0,236.89
19996,265.0,237.53
19997,222.0,237.30
19998,120.0,236.29
19999,500.0,239.04
20000,102.0,238.03
```

Experimentación 5

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 6

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.4
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 186.48400712013245 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

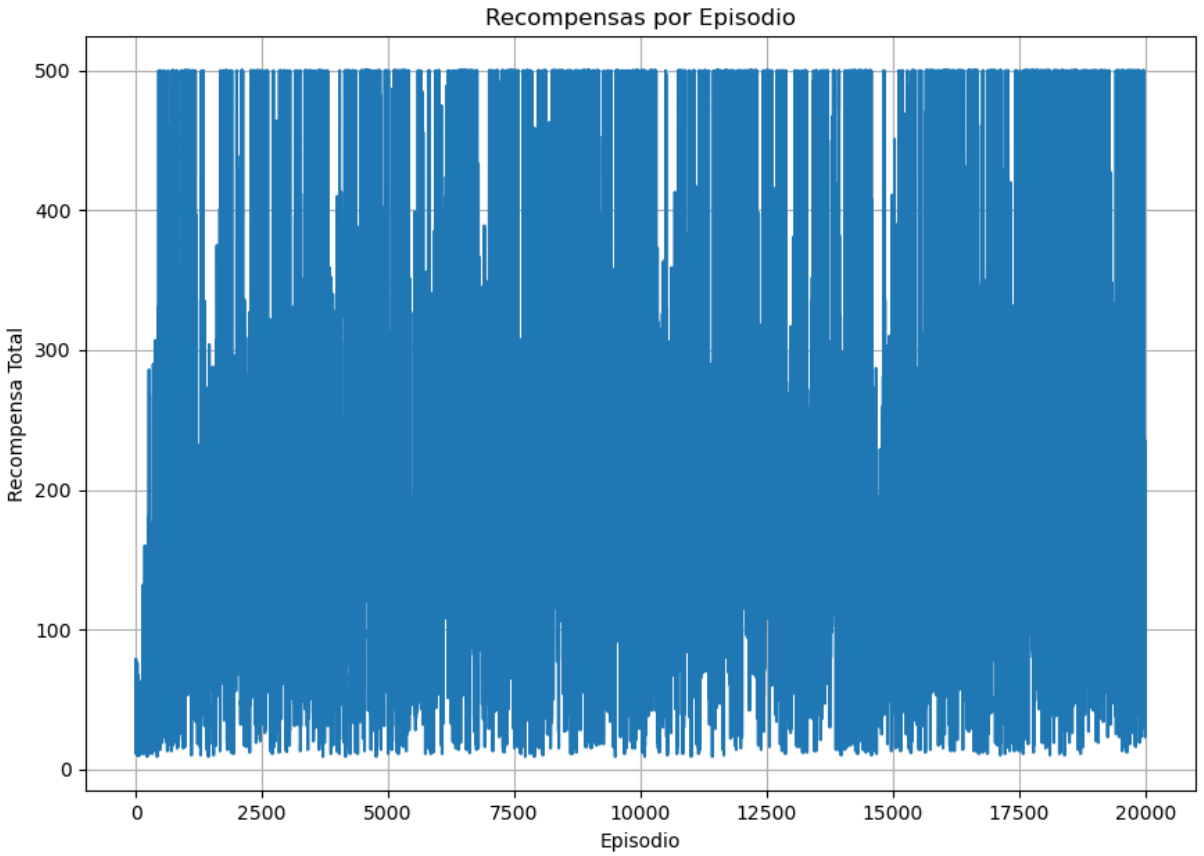
Cálculo de datos (salida_QL_6.csv)

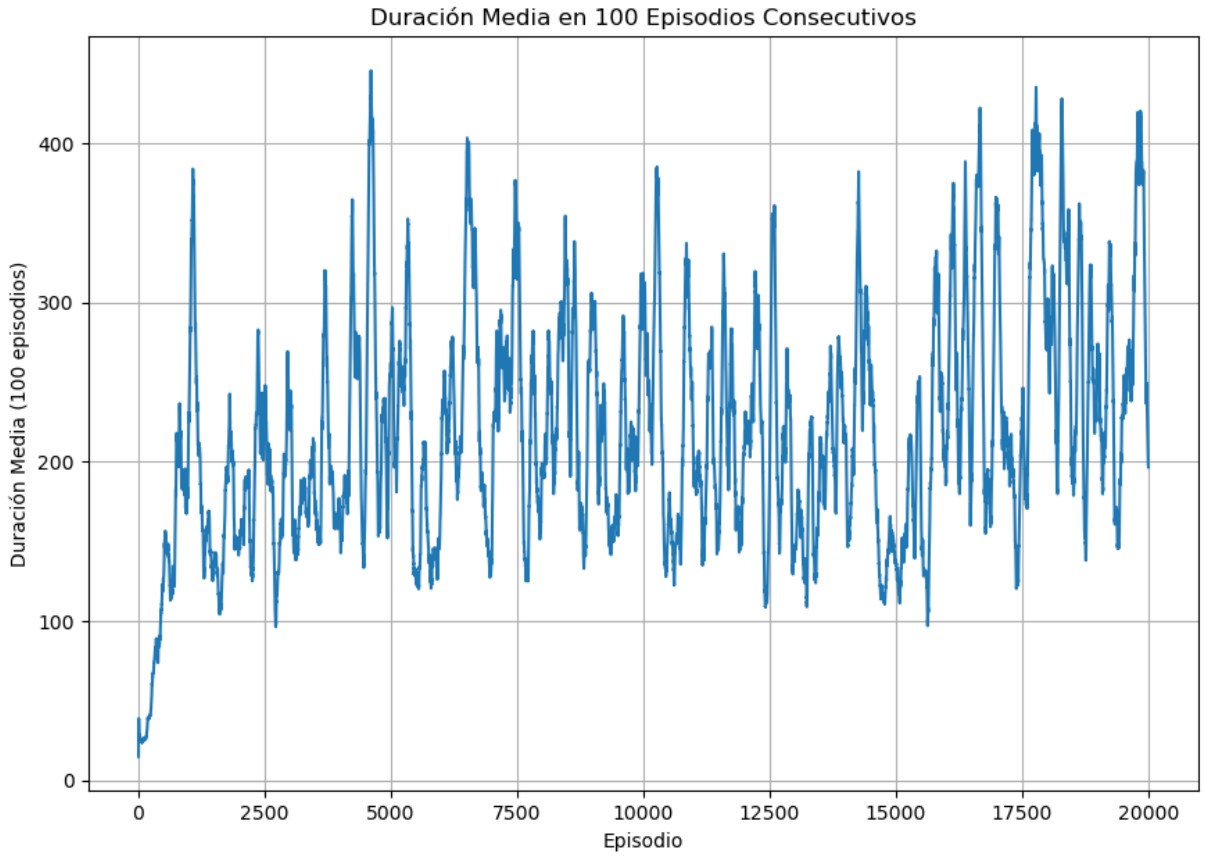
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,15.0,0.00
2,28.0,0.00
3,47.0,0.00
4,25.0,0.00
5,79.0,0.00
6,40.0,0.00
7,27.0,0.00
8,18.0,0.00
9,13.0,0.00
10,17.0,0.00
...
19976,124.0,225.02
19977,110.0,224.92
19978,123.0,225.85
19979,134.0,227.00
19980,120.0,224.78
19981,123.0,221.01
19982,193.0,217.94
19983,134.0,214.28
19984,115.0,210.43
19985,112.0,210.23
19986,153.0,208.73
19987,107.0,208.96
19988,51.0,205.09
19989,113.0,202.28
19990,112.0,201.01
19991,164.0,201.03
19992,23.0,199.43
19993,150.0,200.17
19994,74.0,200.32
19995,53.0,199.52
19996,127.0,198.23
19997,143.0,197.51
19998,153.0,197.13
19999,235.0,197.27
20000,197.0,196.70
```

Experimentación 6

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 7

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.5
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 157.67089462280273 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

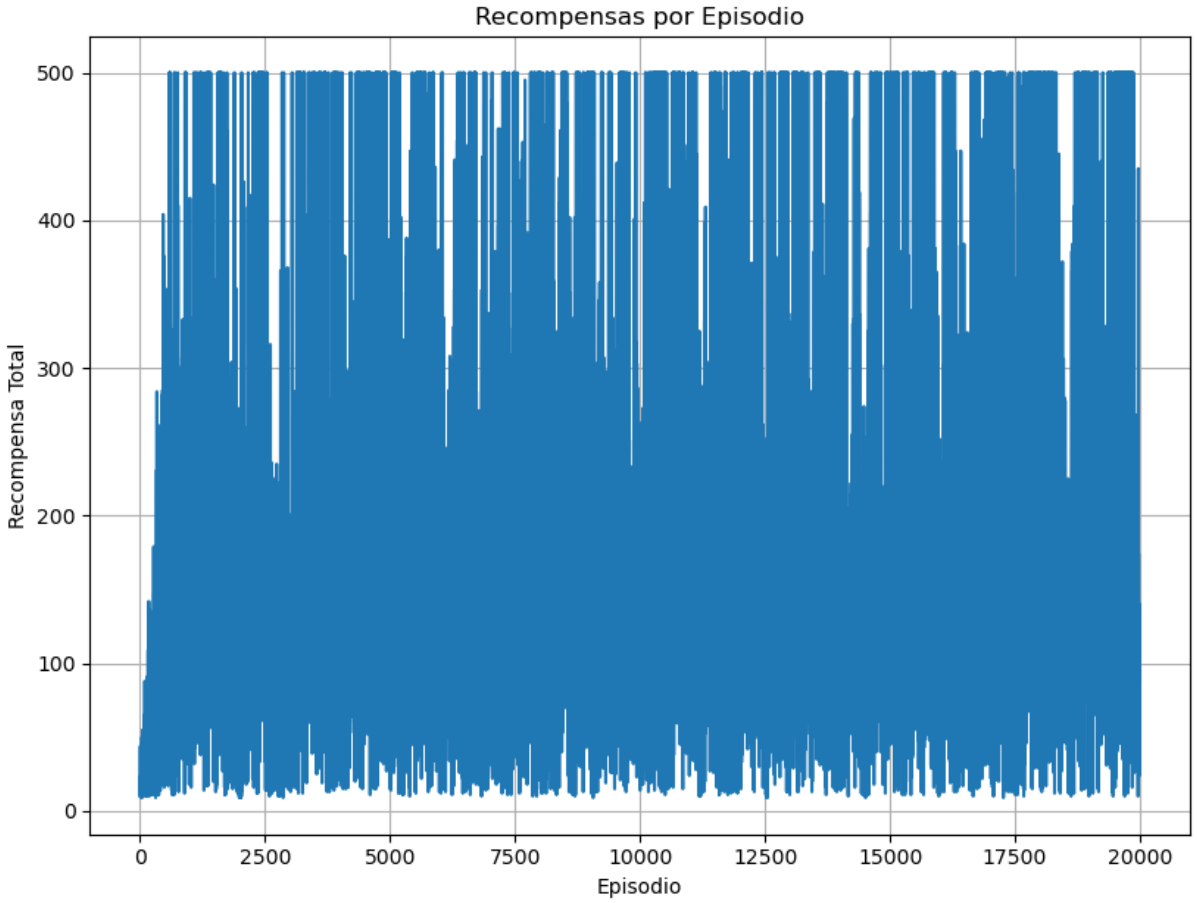
Cálculo de datos (salida_QL_7.csv)

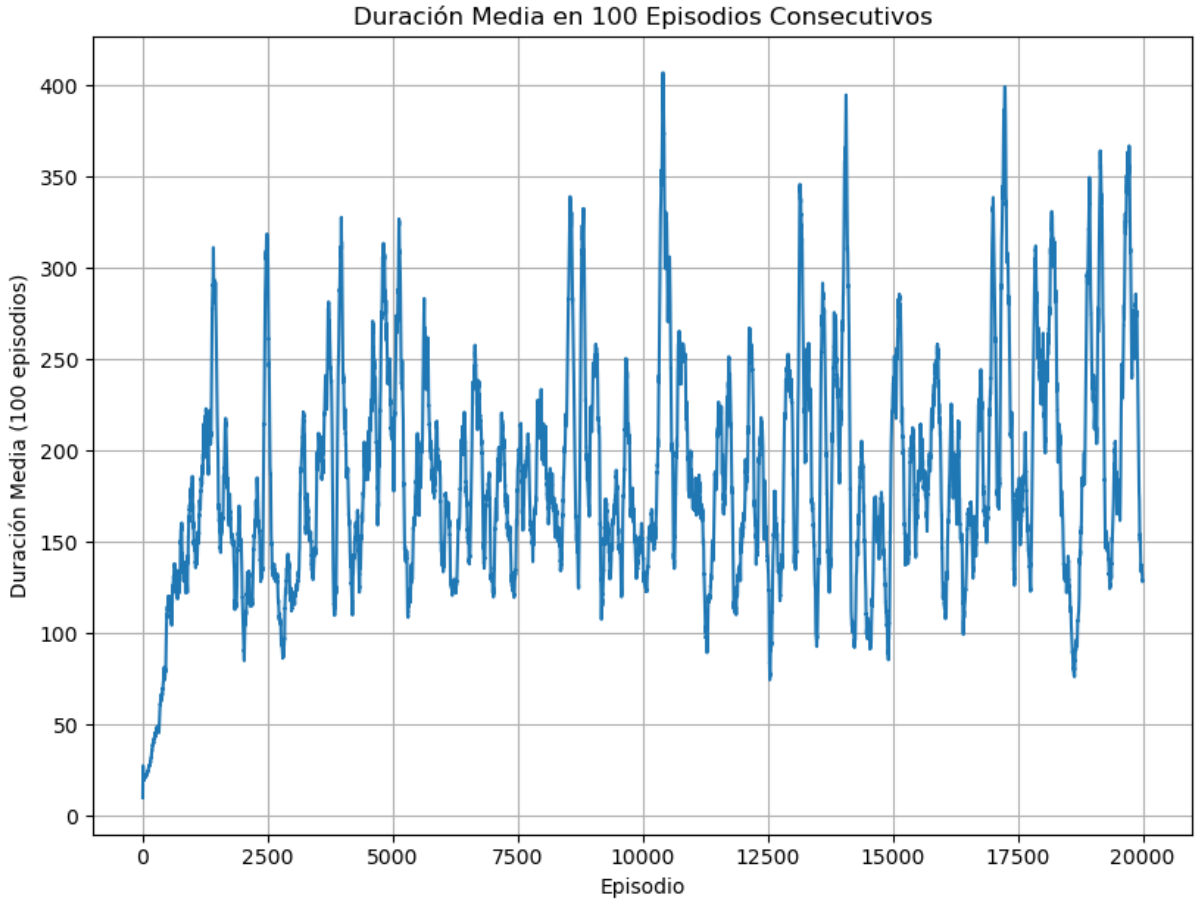
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,10.0,0.00
2,19.0,0.00
3,37.0,0.00
4,30.0,0.00
5,17.0,0.00
6,36.0,0.00
7,44.0,0.00
8,23.0,0.00
9,16.0,0.00
10,19.0,0.00
...
19977,79.0,136.42
19978,88.0,136.42
19979,112.0,136.69
19980,137.0,135.61
19981,85.0,135.69
19982,103.0,136.09
19983,130.0,136.88
19984,87.0,135.30
19985,81.0,134.50
19986,140.0,133.57
19987,132.0,134.63
19988,81.0,133.10
19989,136.0,133.67
19990,101.0,132.90
19991,118.0,132.79
19992,67.0,131.84
19993,140.0,131.77
19994,111.0,131.93
19995,96.0,131.95
19996,24.0,130.11
19997,84.0,129.32
19998,115.0,128.95
19999,84.0,129.14
20000,94.0,128.33
```

Experimentación 7

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 8

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.5
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 157.63582253456116 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

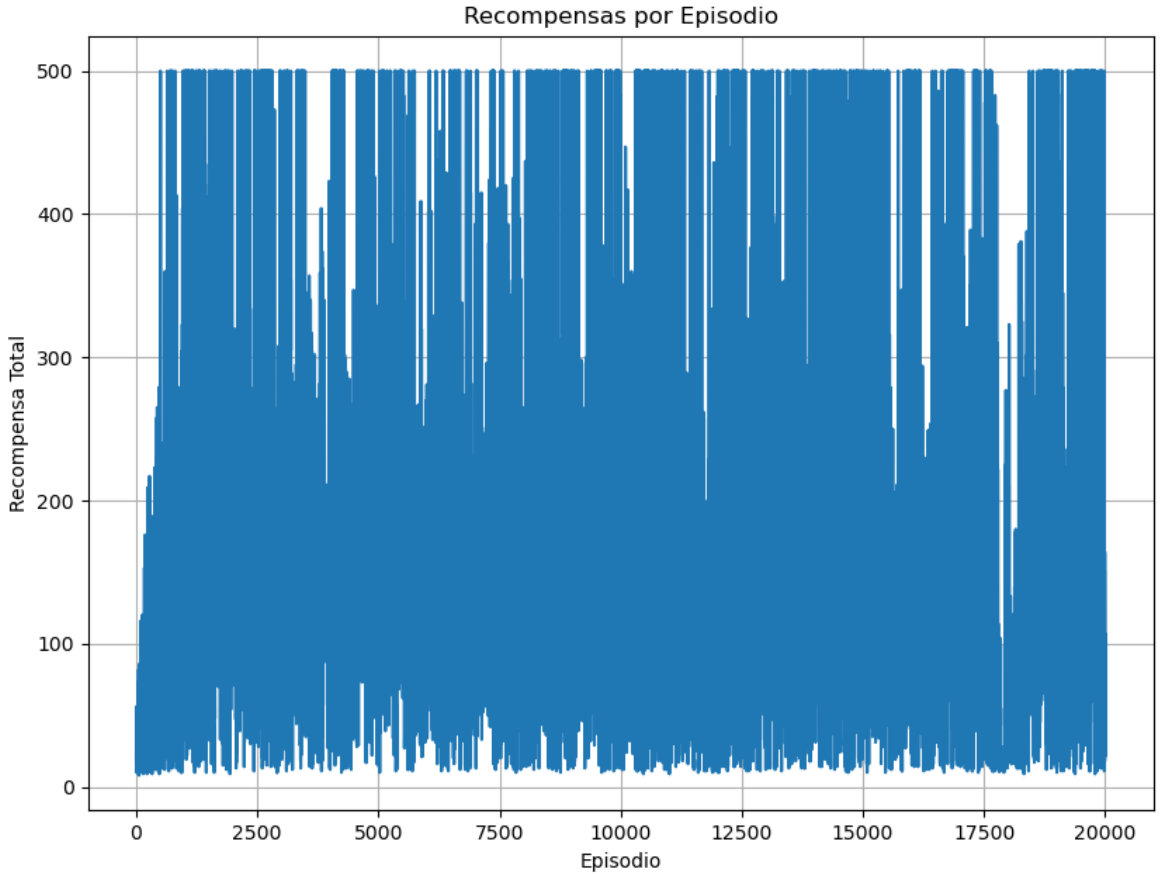
Cálculo de datos (salida_QL_8.csv)

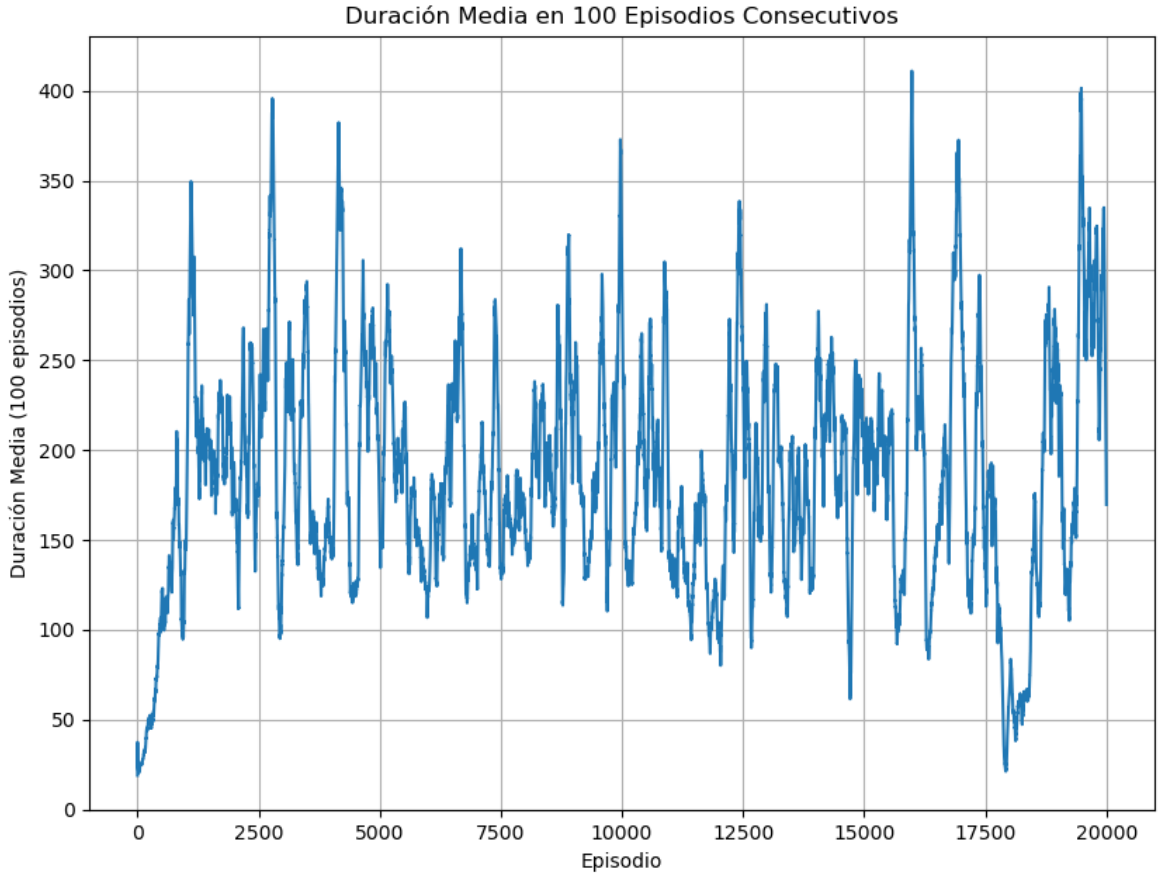
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,19.0,0.00
2,56.0,0.00
3,17.0,0.00
4,11.0,0.00
5,15.0,0.00
6,24.0,0.00
7,30.0,0.00
8,28.0,0.00
9,16.0,0.00
10,39.0,0.00
...
19977,55.0,249.33
19978,17.0,246.58
19979,55.0,244.86
19980,63.0,241.85
19981,15.0,239.89
19982,11.0,235.24
19983,52.0,232.75
19984,500.0,235.49
19985,73.0,234.34
19986,52.0,229.86
19987,64.0,225.50
19988,12.0,220.62
19989,151.0,217.13
19990,78.0,212.91
19991,20.0,208.11
19992,57.0,203.68
19993,85.0,199.53
19994,18.0,194.71
19995,164.0,191.35
19996,58.0,186.93
19997,56.0,182.49
19998,21.0,177.70
19999,107.0,173.77
20000,88.0,169.65
```

Experimentación 8

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 9

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.5
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 5000

Tiempo realizado : 218.34581780433655 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

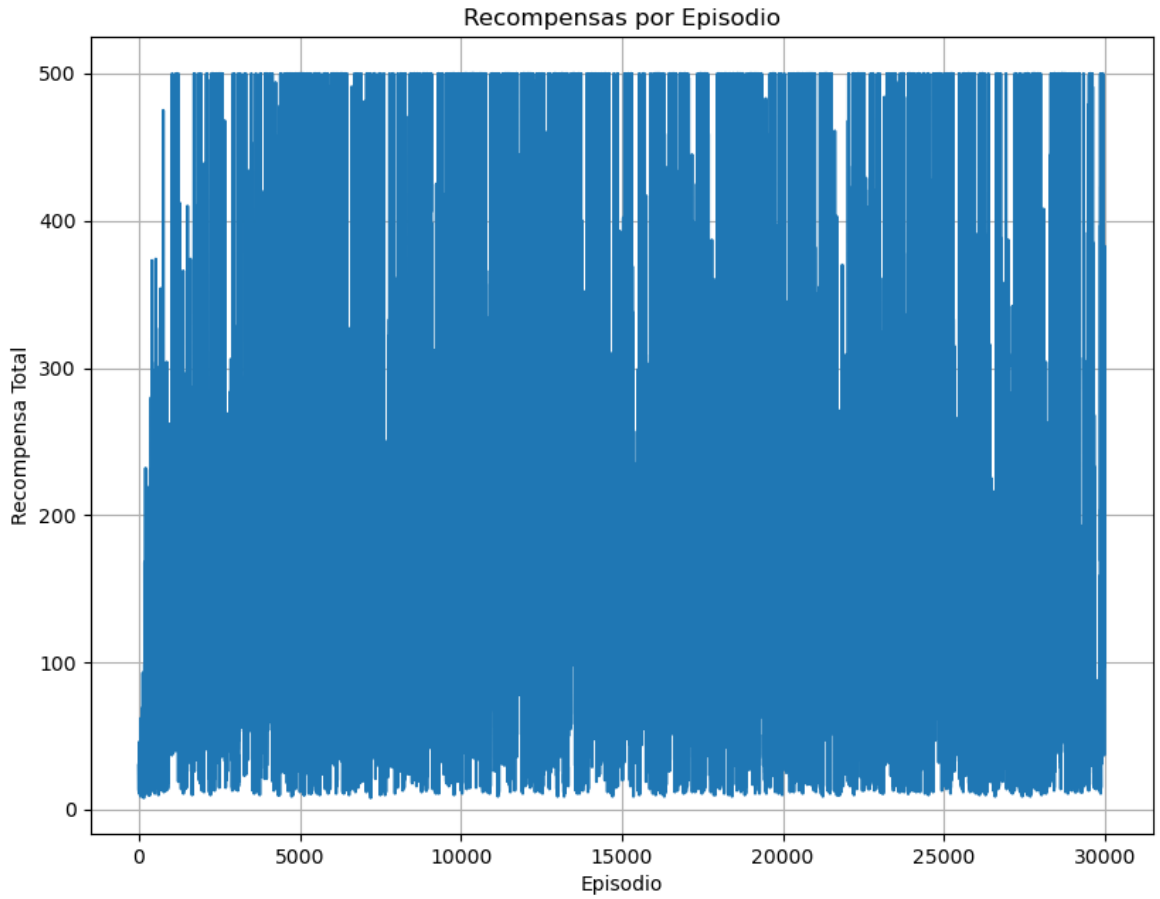
Cálculo de datos (salida_QL_9.csv)

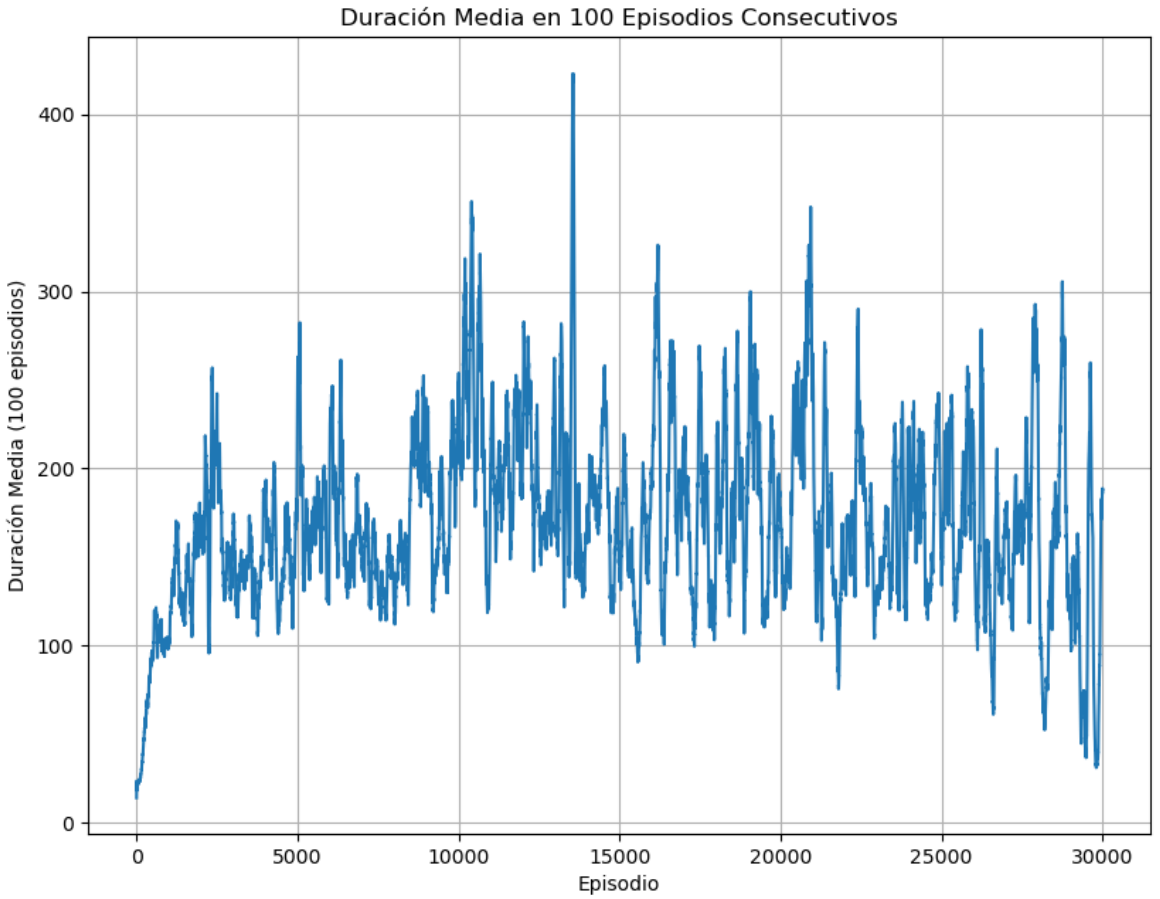
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,14.0,0.00
2,31.0,0.00
3,19.0,0.00
4,30.0,0.00
5,24.0,0.00
6,15.0,0.00
7,23.0,0.00
8,16.0,0.00
9,12.0,0.00
10,14.0,0.00
...
29976,106.0,172.59
29977,178.0,171.57
29978,212.0,173.38
29979,232.0,175.26
29980,37.0,175.48
29981,227.0,176.87
29982,80.0,177.52
29983,74.0,177.90
29984,325.0,180.68
29985,191.0,181.91
29986,383.0,184.10
29987,279.0,183.78
29988,286.0,186.26
29989,251.0,188.04
29990,147.0,187.07
29991,97.0,187.27
29992,165.0,185.85
29993,177.0,185.95
29994,103.0,186.67
29995,180.0,186.66
29996,273.0,188.59
29997,102.0,188.74
29998,315.0,188.77
29999,191.0,186.94
30000,185.0,188.45
```

Experimentación 9

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 10

1. Algoritmo Q-Learning

Parámetros iniciales

Resultados para Q-Learning
Alpha = 0.5
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 5000

Tiempo realizado : 281.175799369812 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

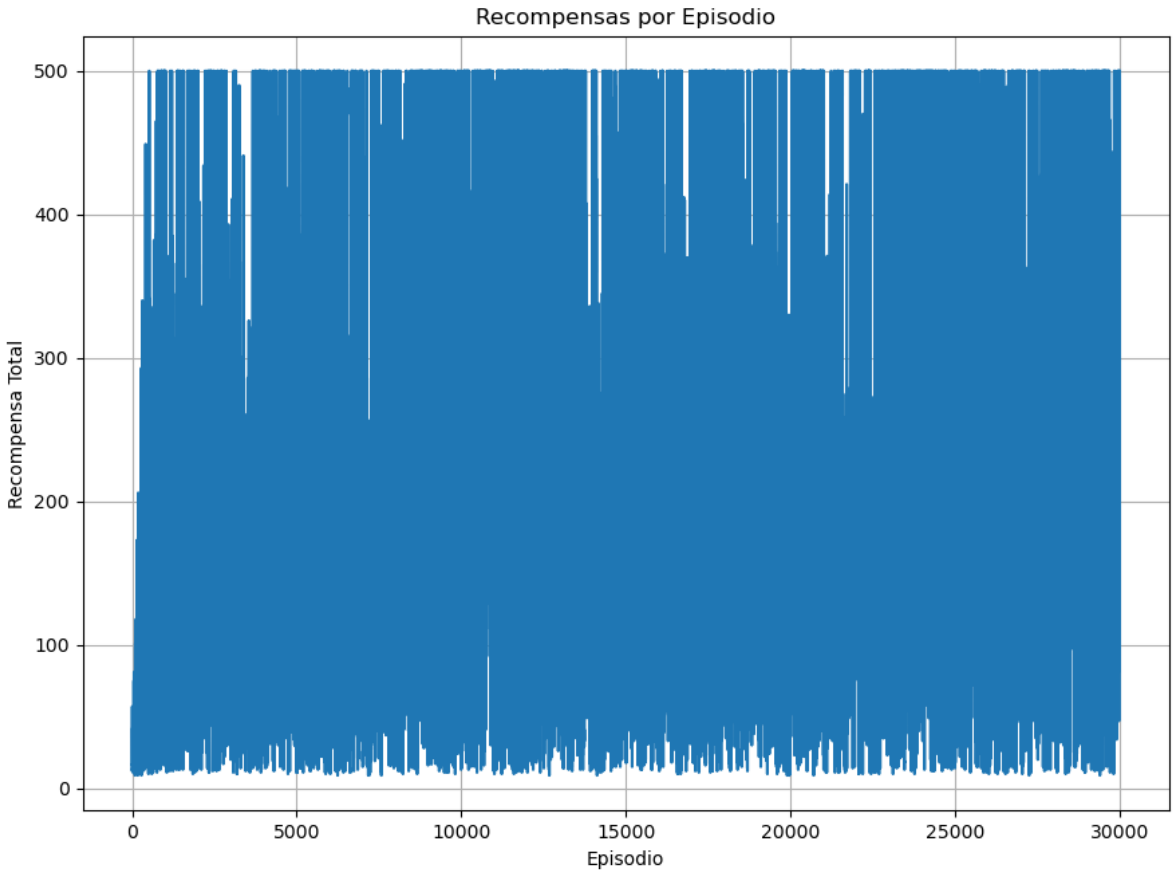
Cálculo de datos (salida_QL_10.csv)

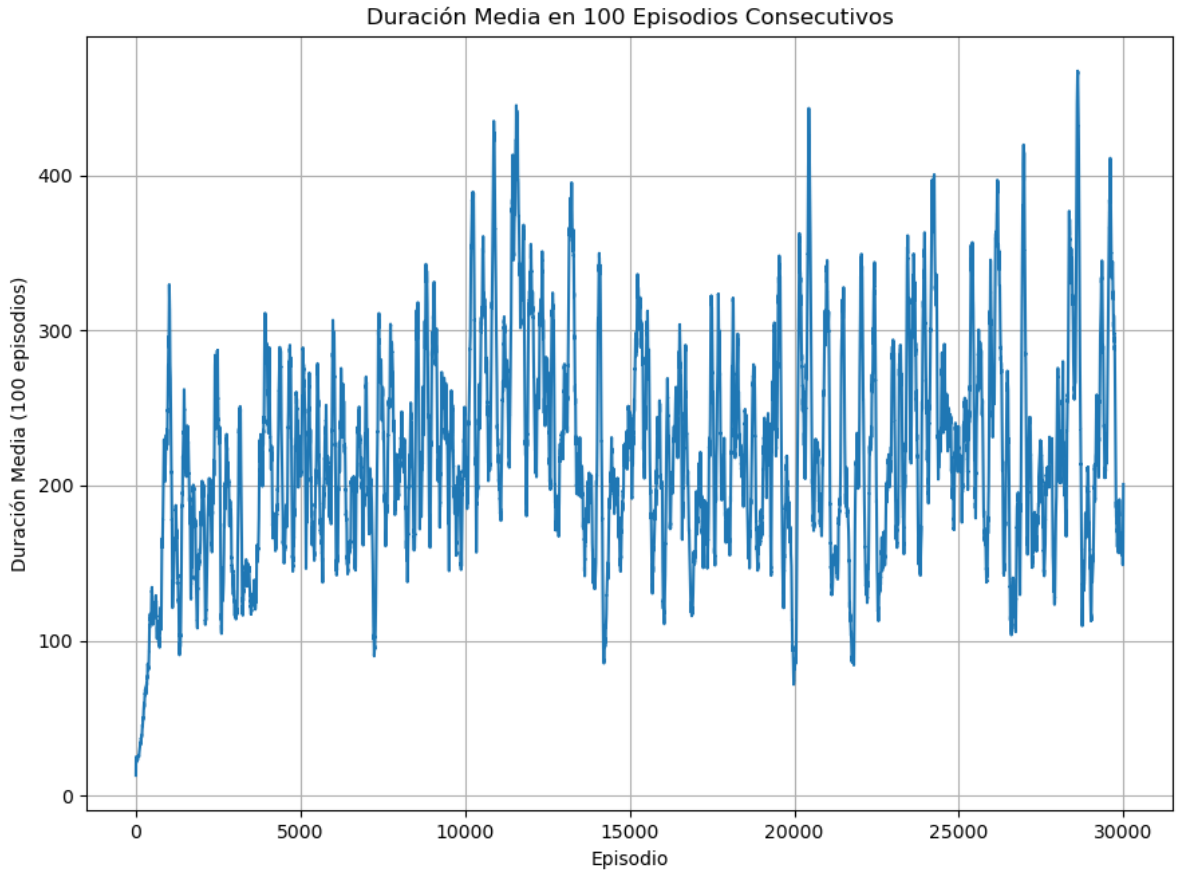
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,13.0,0.00
2,23.0,0.00
3,12.0,0.00
4,24.0,0.00
5,21.0,0.00
6,57.0,0.00
7,24.0,0.00
8,16.0,0.00
9,19.0,0.00
10,22.0,0.00
...
29978,146.0,154.10
29979,136.0,152.80
29980,47.0,150.03
29981,93.0,149.27
29982,82.0,148.78
29983,268.0,150.18
29984,189.0,150.85
29985,241.0,151.68
29986,199.0,152.66
29987,175.0,153.55
29988,197.0,154.96
29989,284.0,157.16
29990,500.0,161.09
29991,500.0,165.41
29992,500.0,169.58
29993,500.0,173.95
29994,500.0,178.53
29995,500.0,182.48
29996,500.0,186.84
29997,500.0,191.30
29998,381.0,194.12
29999,500.0,197.01
30000,500.0,200.82
```

Experimentación 10

1. Algoritmo Q-Learning

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 1

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 10000
Pasos x episodio = 5000

Tiempo realizado : 83.39774894714355 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

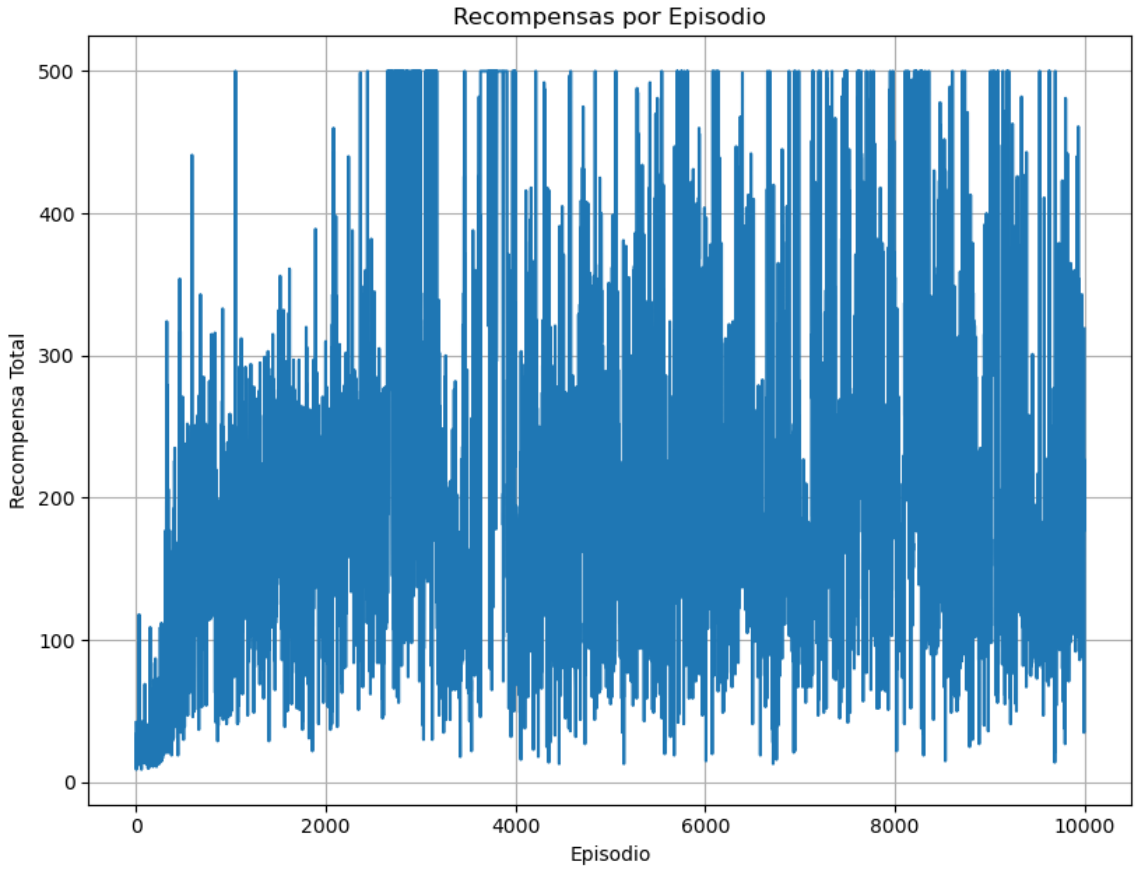
Cálculo de datos (salida_SARSA_1.csv)

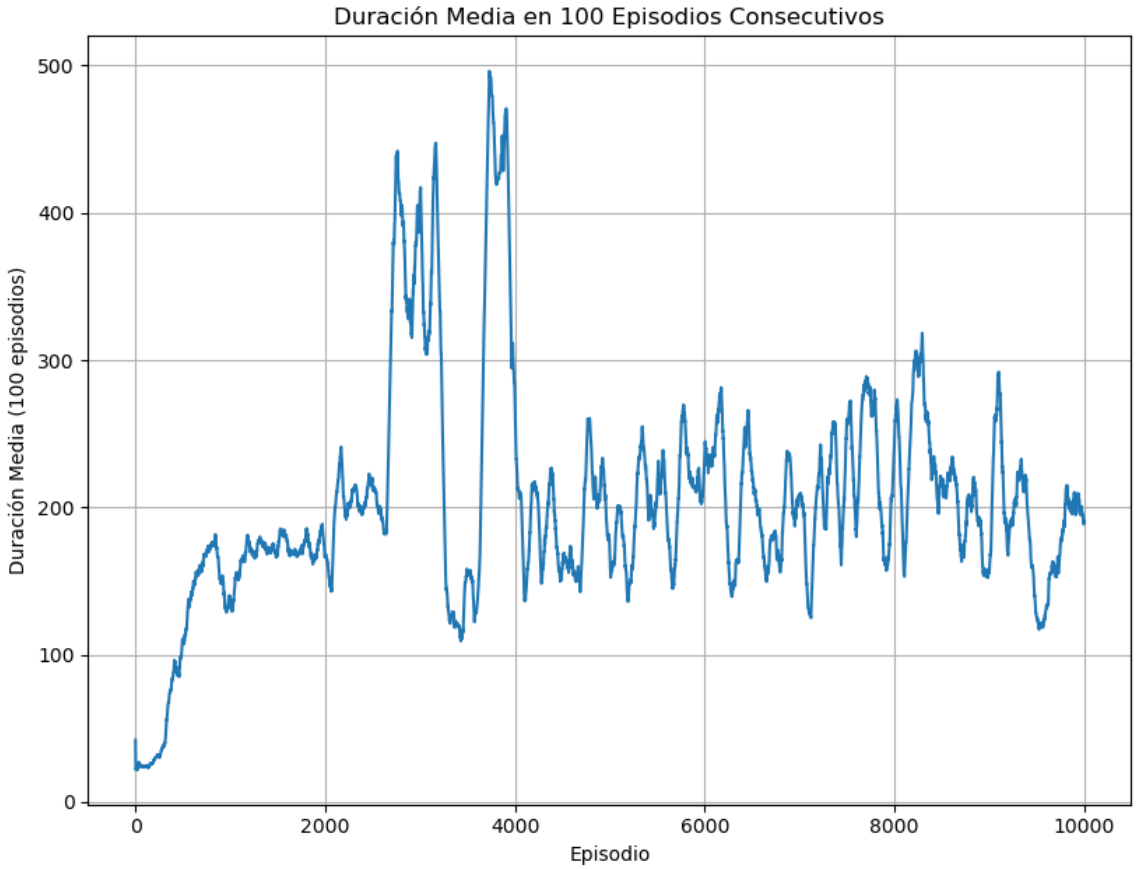
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,42.0,0.00
2,22.0,0.00
3,18.0,0.00
4,19.0,0.00
5,9.0,0.00
6,23.0,0.00
7,24.0,0.00
8,20.0,0.00
9,24.0,0.00
10,24.0,0.00
...
9977,137.0,198.68
9978,261.0,198.35
9979,120.0,197.87
9980,224.0,197.84
9981,176.0,197.03
9982,196.0,195.98
9983,168.0,196.43
9984,91.0,196.14
9985,102.0,196.09
9986,108.0,195.64
9987,151.0,194.58
9988,88.0,193.63
9989,183.0,194.32
9990,173.0,194.03
9991,207.0,192.99
9992,223.0,193.50
9993,221.0,194.62
9994,185.0,194.63
9995,200.0,194.09
9996,35.0,190.84
9997,227.0,189.85
9998,177.0,189.56
9999,193.0,188.56
10000,319.0,190.66
```

Experimentación 1

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 2

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 10000
Pasos x episodio = 5000

Tiempo realizado : 50.06770944595337 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

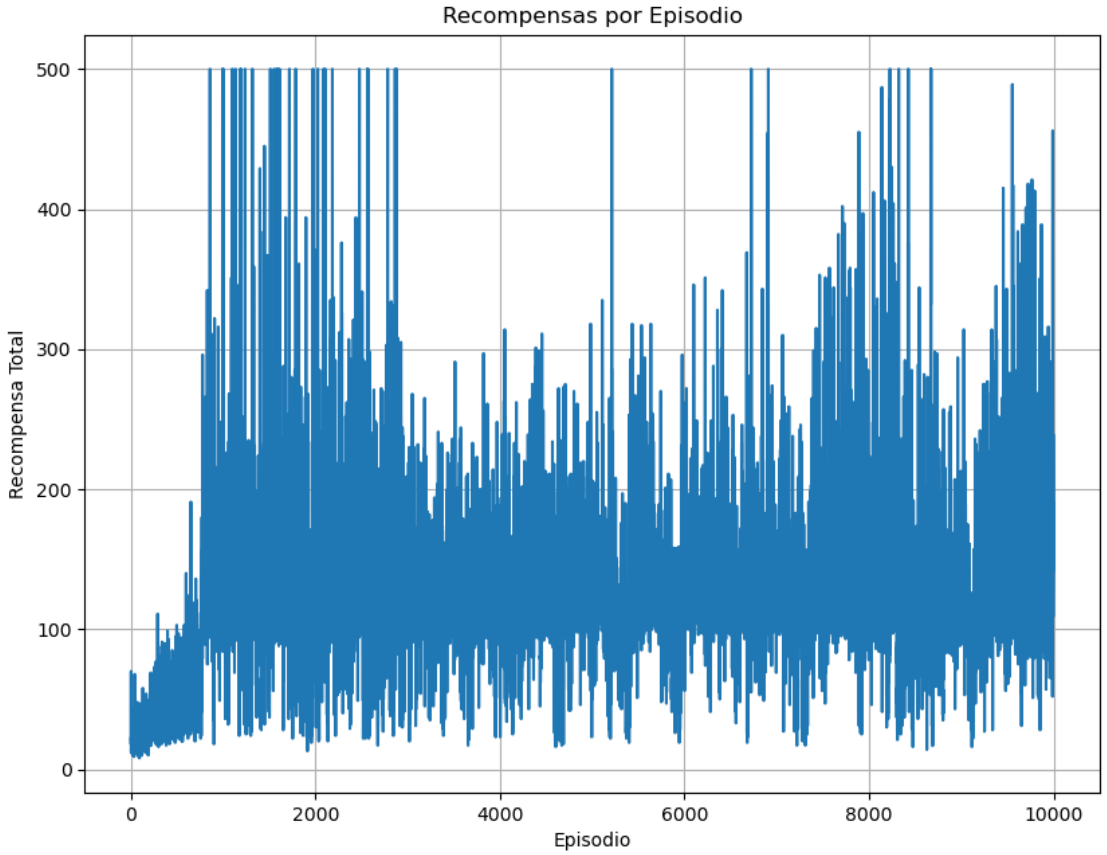
Cálculo de datos (salida_SARSA_2.csv)

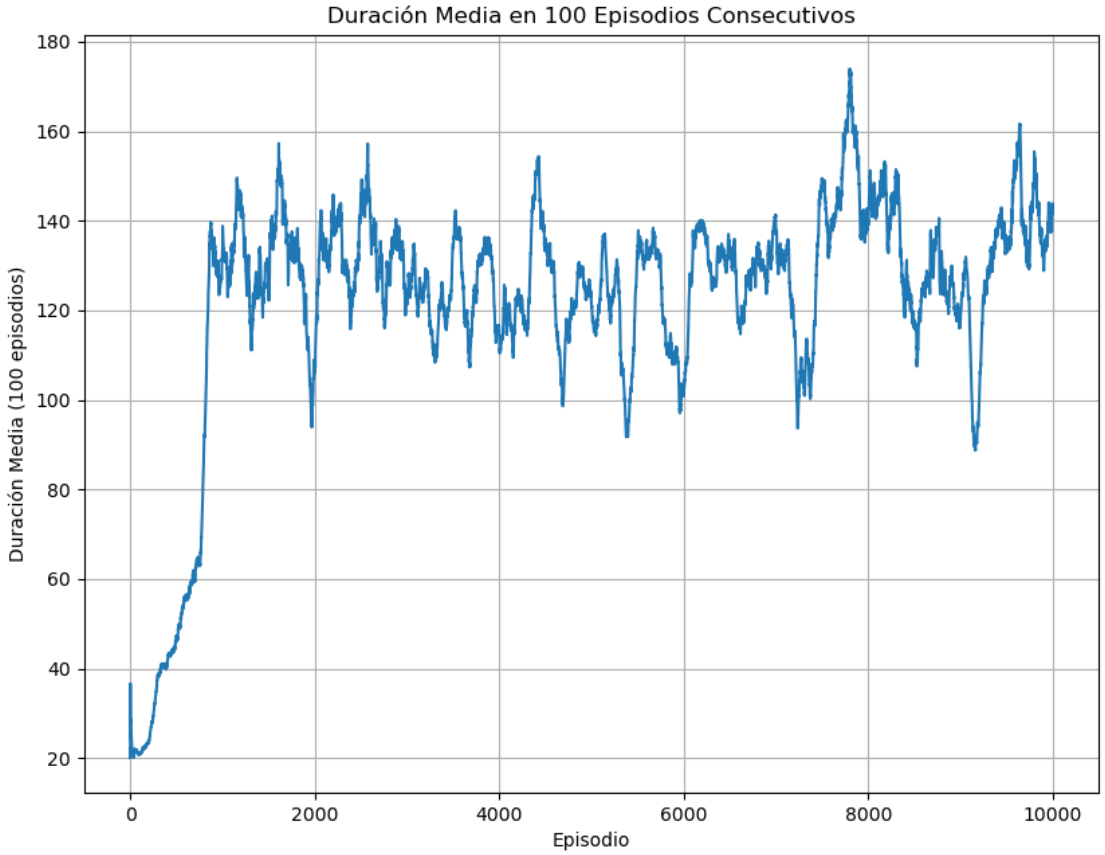
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,22.0,0.00
2,18.0,0.00
3,70.0,0.00
4,12.0,0.00
5,18.0,0.00
6,32.0,0.00
7,19.0,0.00
8,15.0,0.00
9,21.0,0.00
10,13.0,0.00
...
9980,109.0,138.21
9981,83.0,137.71
9982,201.0,137.39
9983,196.0,138.30
9984,195.0,138.78
9985,107.0,138.93
9986,80.0,138.80
9987,94.0,138.75
9988,96.0,138.43
9989,52.0,137.86
9990,193.0,138.71
9991,131.0,139.13
9992,456.0,142.63
9993,128.0,142.22
9994,190.0,143.19
9995,191.0,143.88
9996,117.0,142.21
9997,109.0,141.91
9998,239.0,141.79
9999,145.0,141.56
10000,142.0,142.19
```

Experimentación 2

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 3

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 20000
Pasos x episodio = 5000

Tiempo realizado : 113.96758198738098 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

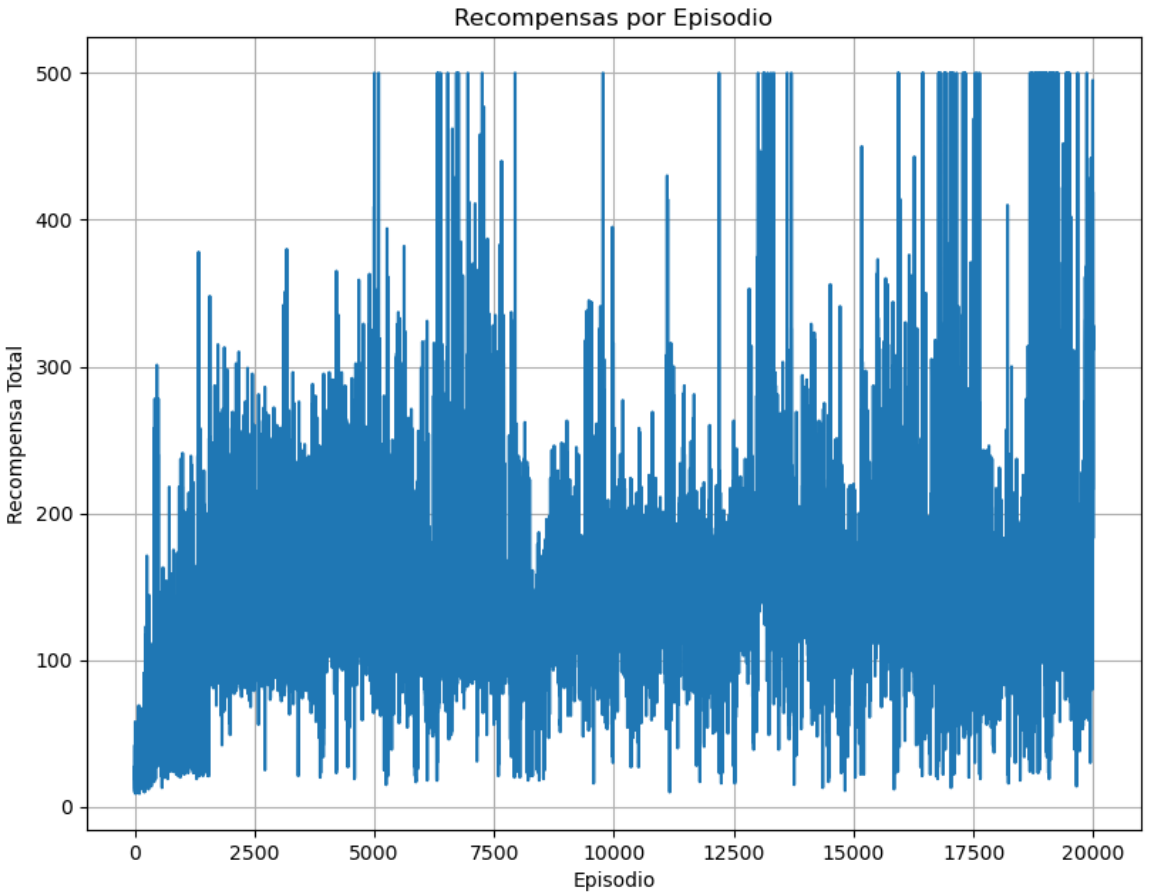
Cálculo de datos (salida_SARSA_3.csv)

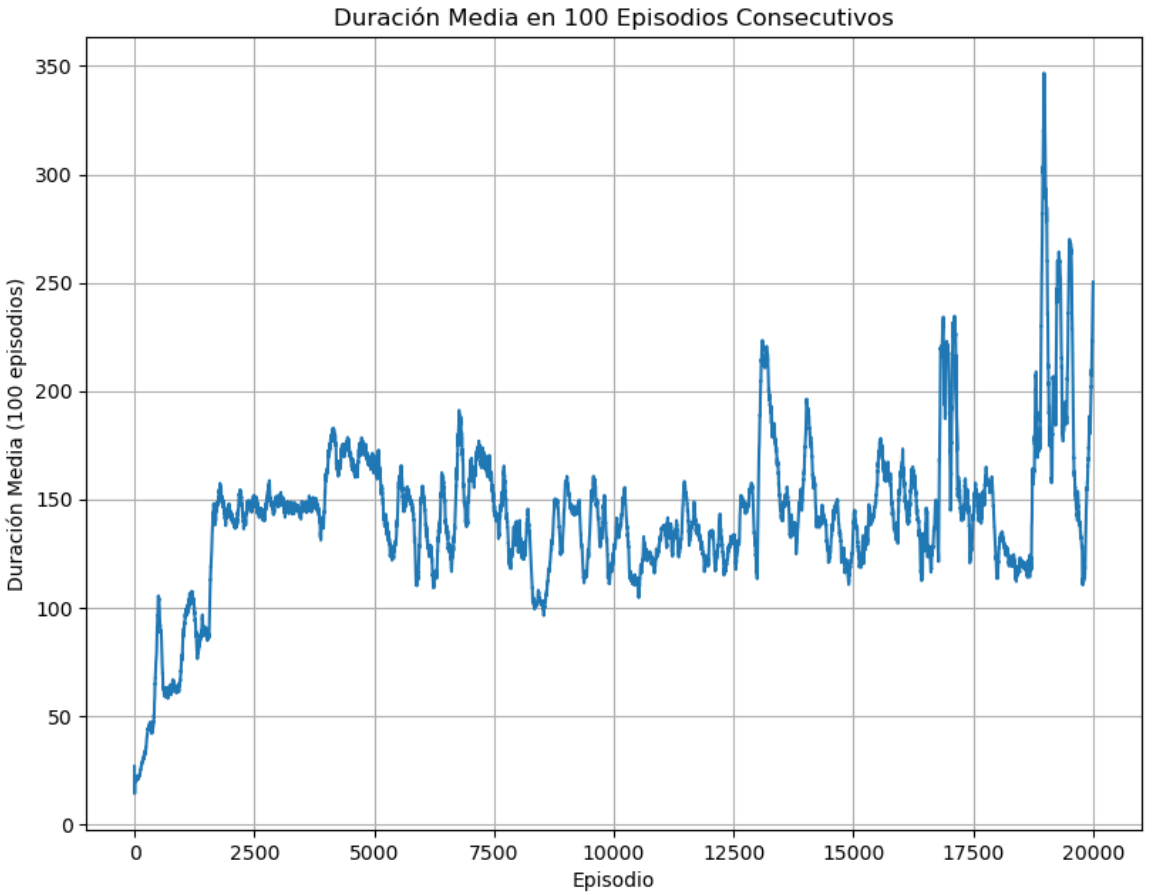
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,27.0,0.00
2,12.0,0.00
3,12.0,0.00
4,11.0,0.00
5,10.0,0.00
6,36.0,0.00
7,15.0,0.00
8,42.0,0.00
9,19.0,0.00
10,20.0,0.00
...
19982,198.0,225.80
19983,322.0,227.54
19984,279.0,228.44
19985,313.0,230.02
19986,279.0,230.44
19987,481.0,232.63
19988,285.0,234.40
19989,495.0,236.82
19990,409.0,238.46
19991,216.0,238.55
19992,419.0,241.61
19993,284.0,243.26
19994,315.0,243.98
19995,270.0,244.96
19996,291.0,246.44
19997,183.0,246.51
19998,328.0,247.62
19999,295.0,248.53
20000,312.0,250.33
```

Experimentación 3

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 5

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 5000

Tiempo realizado : 270.8059663772583 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

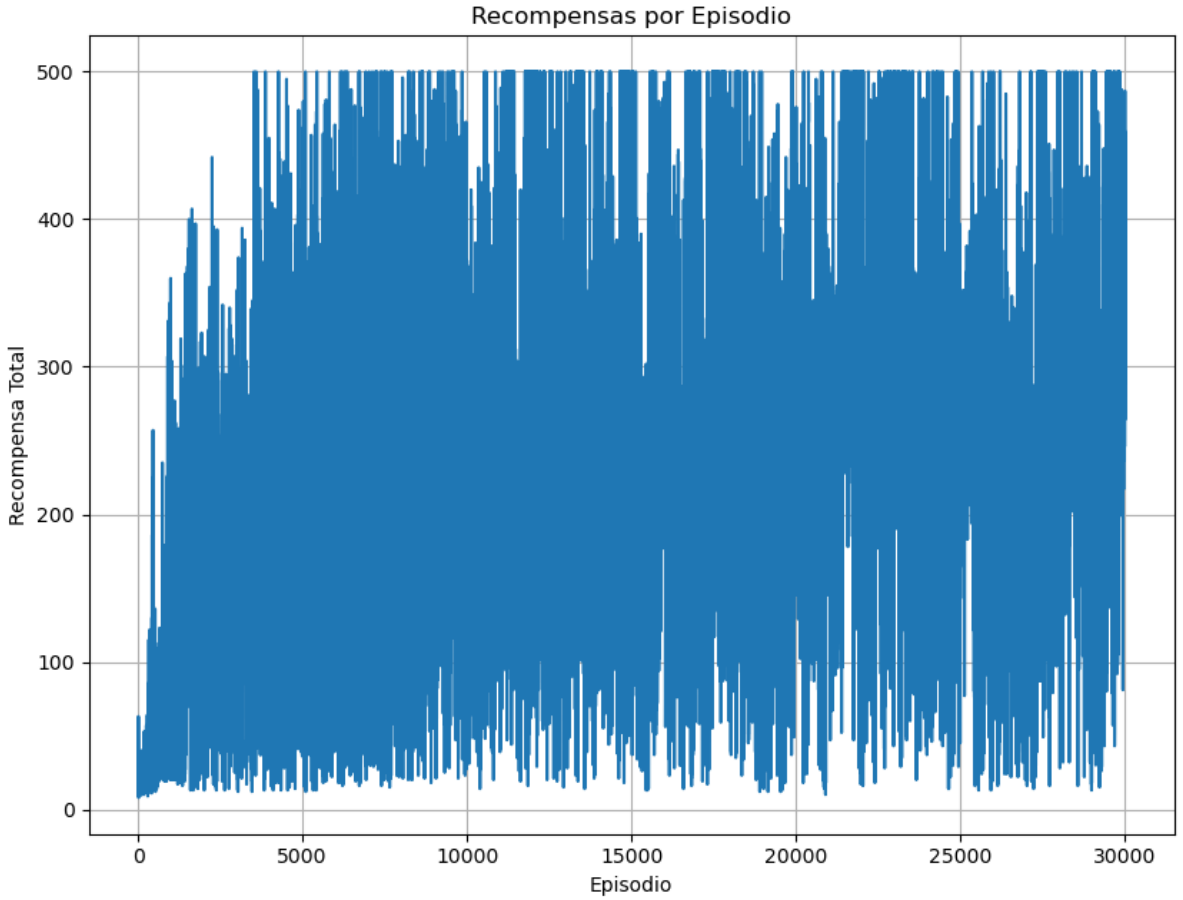
Cálculo de datos (salida_SARSA_5.csv)

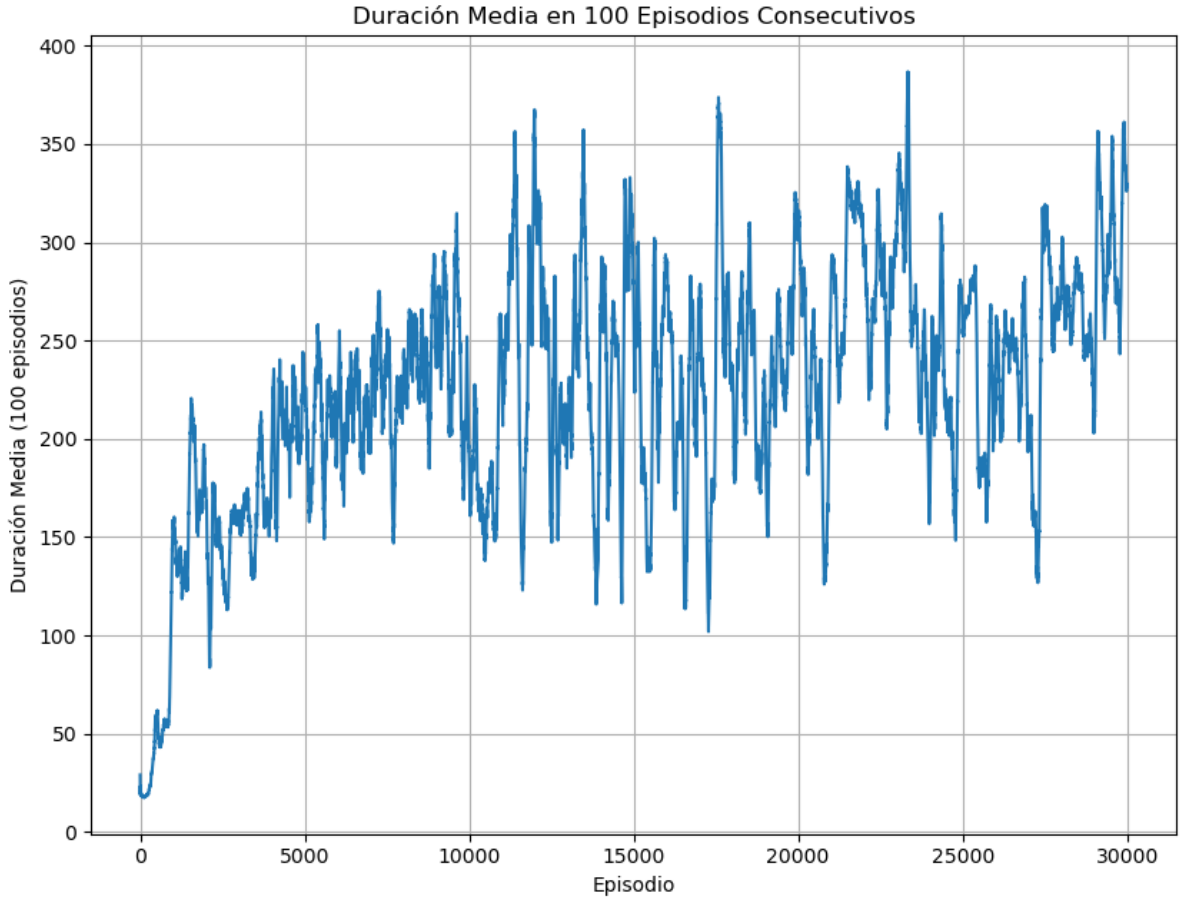
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,29.0,0.00
2,27.0,0.00
3,10.0,0.00
4,16.0,0.00
5,15.0,0.00
6,16.0,0.00
7,26.0,0.00
8,23.0,0.00
9,15.0,0.00
10,30.0,0.00
...
29978,284.0,328.34
29979,263.0,328.22
29980,276.0,326.86
29981,327.0,325.93
29982,347.0,326.43
29983,407.0,327.68
29984,237.0,326.96
29985,436.0,326.51
29986,340.0,327.08
29987,410.0,328.35
29988,375.0,328.35
29989,452.0,328.62
29990,246.0,328.42
29991,453.0,329.98
29992,315.0,328.25
29993,385.0,329.61
29994,381.0,329.19
29995,289.0,328.43
29996,411.0,328.57
29997,348.0,329.31
29998,264.0,329.05
29999,487.0,329.38
30000,459.0,329.48
```

Experimentación 5

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 7

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 10000

Tiempo realizado : 290.2330174446106 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

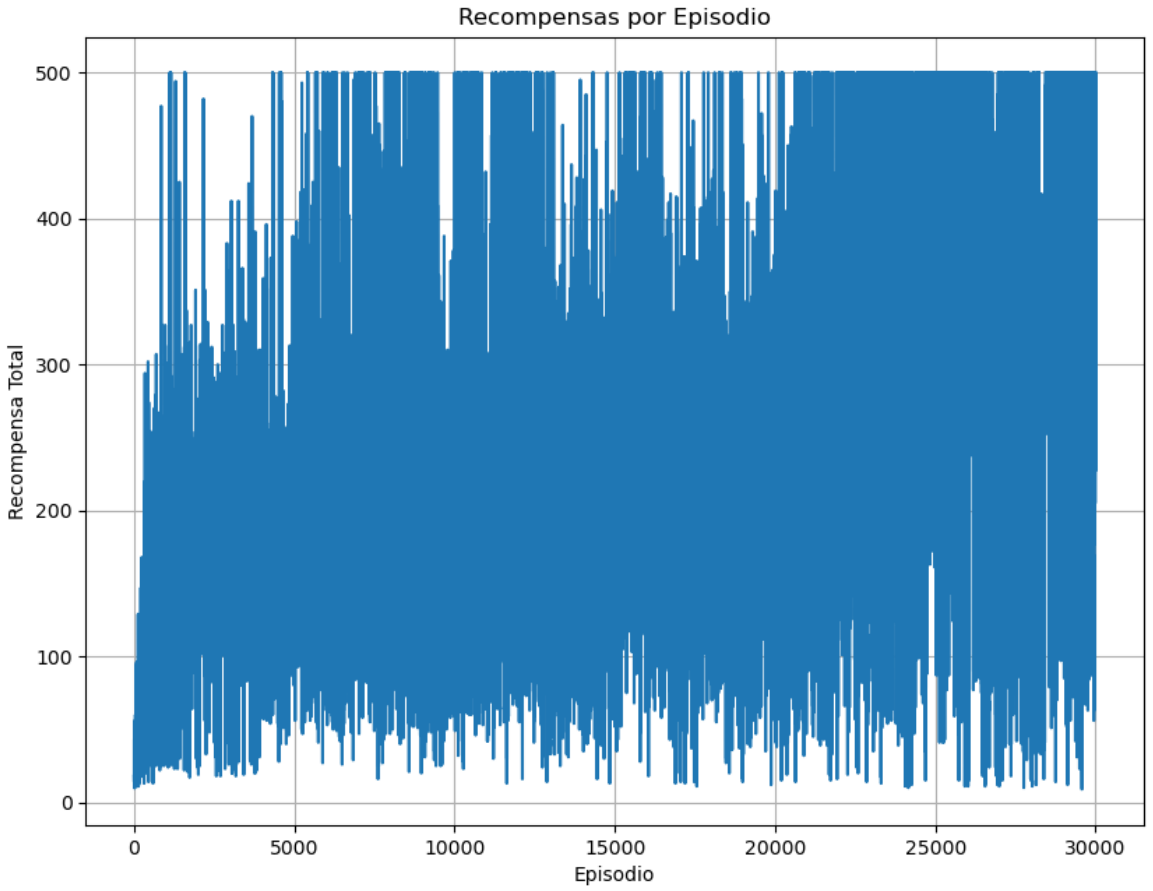
Cálculo de datos (salida_SARSA_7.csv)

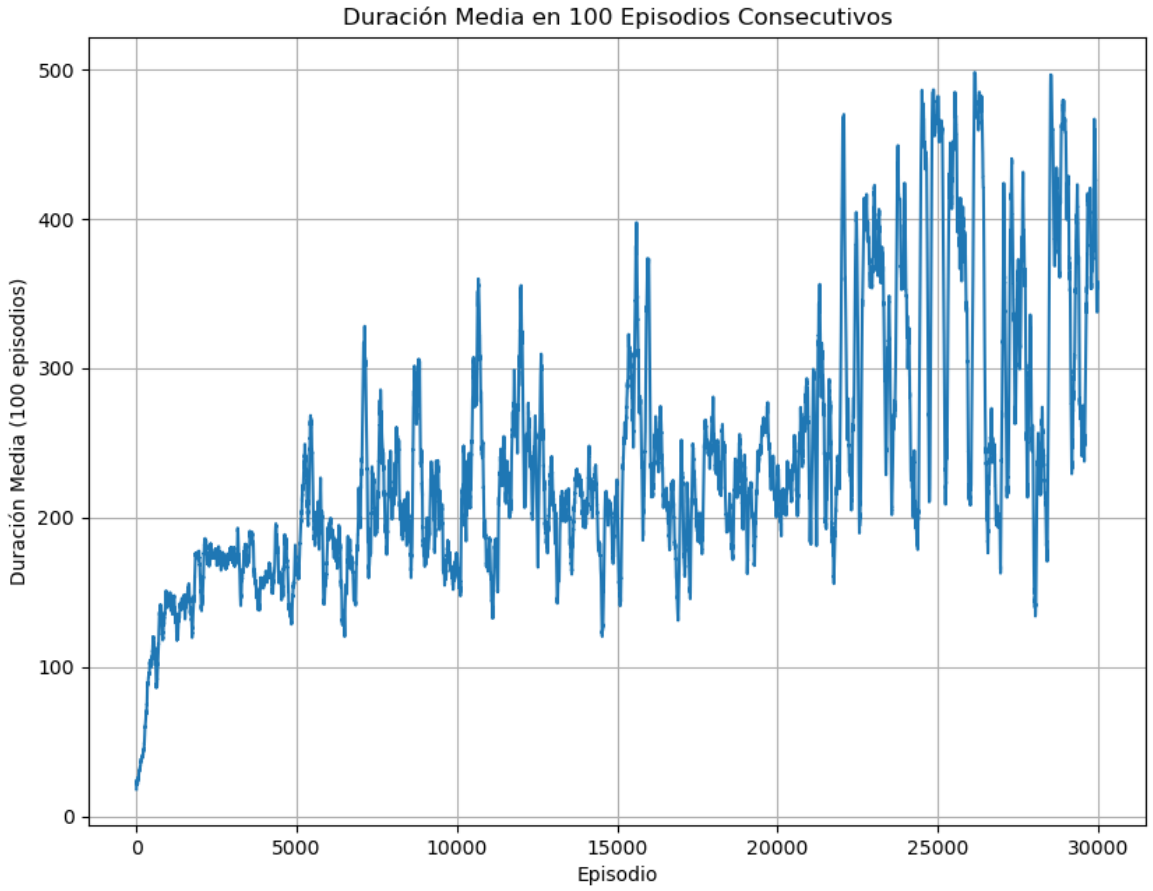
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,18.0,0.00
2,25.0,0.00
3,27.0,0.00
4,16.0,0.00
5,14.0,0.00
6,26.0,0.00
7,43.0,0.00
8,19.0,0.00
9,16.0,0.00
10,10.0,0.00
...
29975,500.0,337.68
29976,500.0,337.77
29977,500.0,337.77
29978,500.0,337.77
29979,500.0,337.77
29980,500.0,337.77
29981,500.0,337.77
29982,500.0,337.77
29983,500.0,337.77
29984,500.0,337.77
29985,500.0,337.77
29986,500.0,341.65
29987,500.0,345.78
29988,205.0,342.83
29989,350.0,341.33
29990,500.0,342.18
29991,500.0,345.56
29992,500.0,349.23
29993,500.0,353.23
29994,500.0,356.53
29995,227.0,353.80
29996,500.0,353.80
29997,500.0,353.80
29998,500.0,353.80
29999,500.0,357.87
30000,500.0,357.87
```

Experimentación 7

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 9

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.2
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 10000

Tiempo realizado : 345.307461977005 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

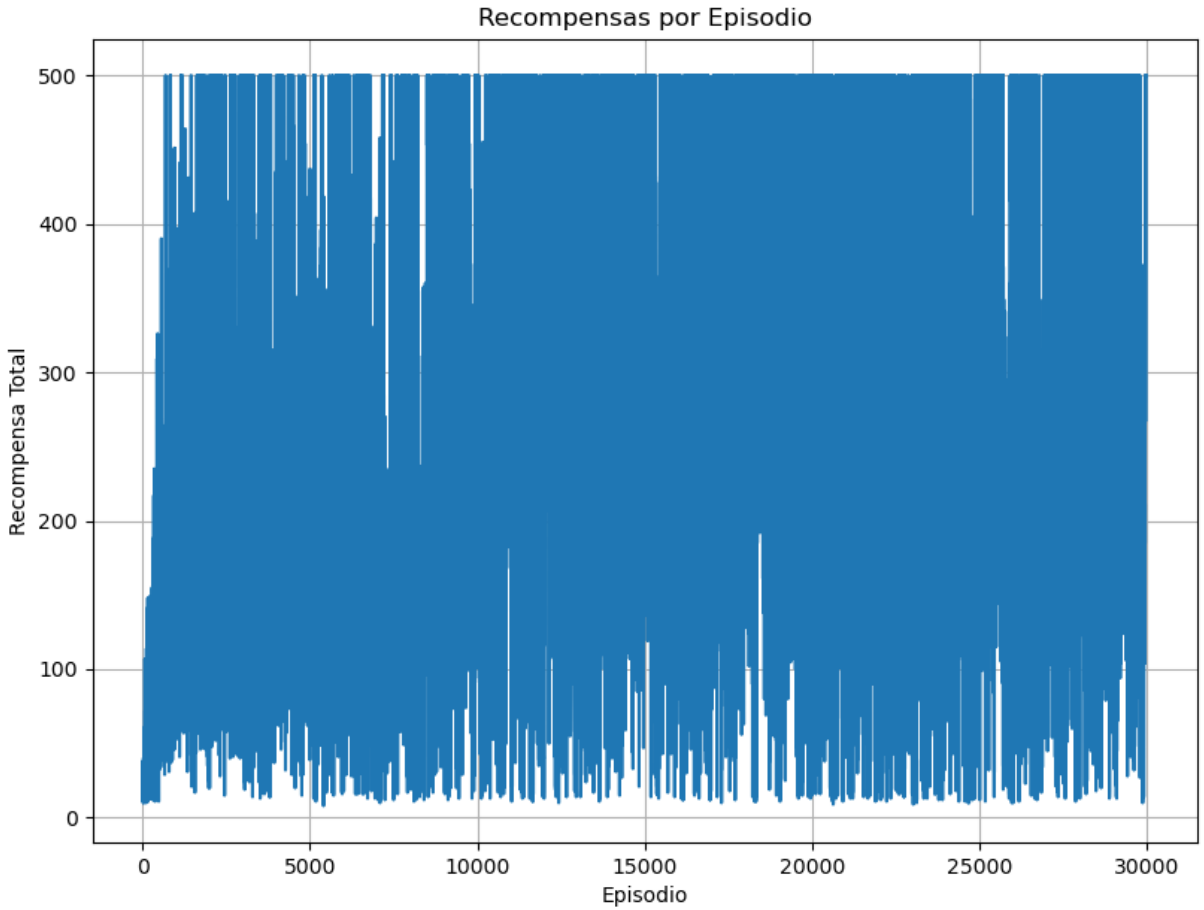
Cálculo de datos (salida_SARSA_9.csv)

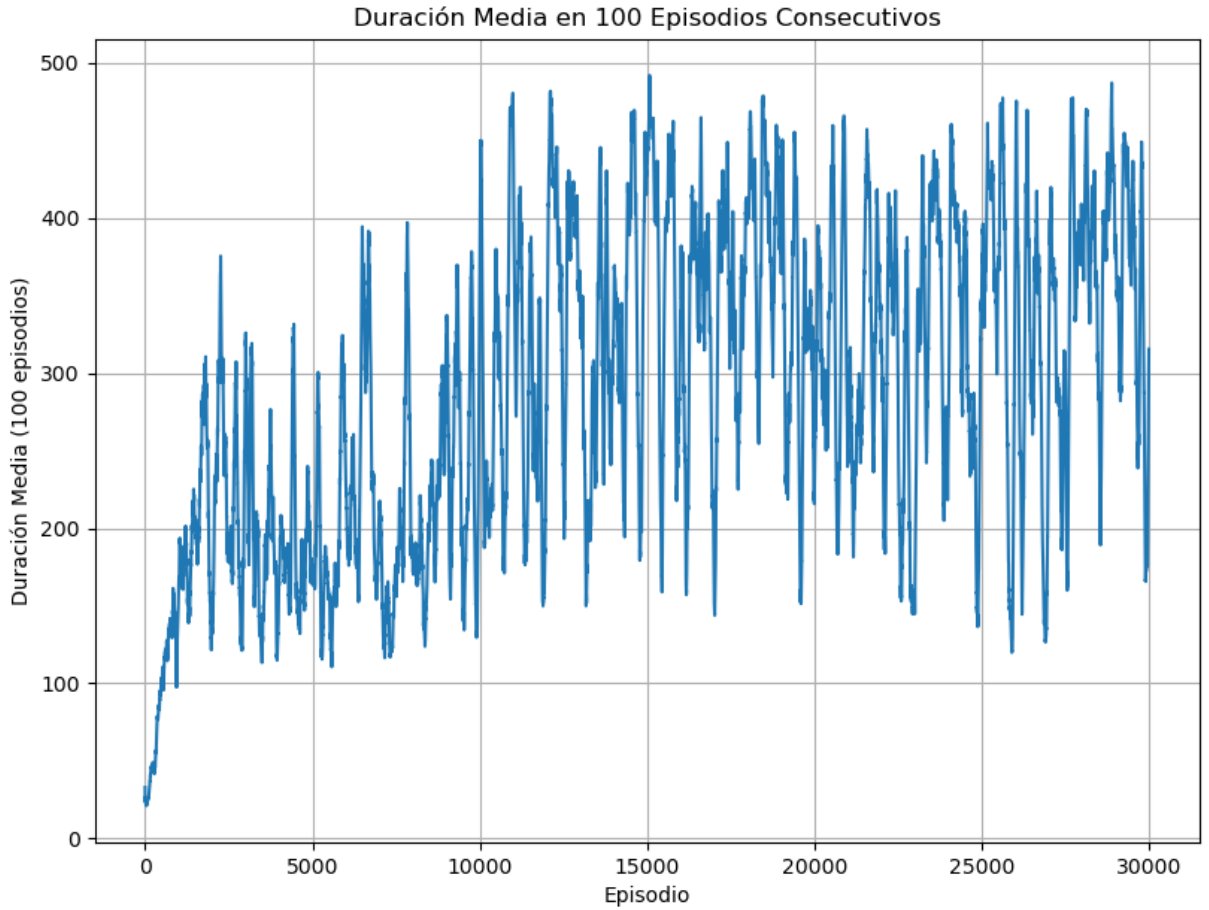
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,33.0,0.00
2,15.0,0.00
3,33.0,0.00
4,30.0,0.00
5,17.0,0.00
6,30.0,0.00
7,23.0,0.00
8,18.0,0.00
9,30.0,0.00
10,38.0,0.00
...
29979,500.0,242.98
29980,500.0,246.63
29981,500.0,250.32
29982,500.0,254.52
29983,500.0,259.23
29984,500.0,263.88
29985,500.0,268.29
29986,500.0,273.12
29987,500.0,278.02
29988,500.0,282.83
29989,500.0,285.80
29990,500.0,287.33
29991,500.0,290.00
29992,500.0,292.94
29993,500.0,296.40
29994,500.0,299.52
29995,500.0,303.34
29996,500.0,304.62
29997,500.0,307.74
29998,500.0,310.08
29999,500.0,312.94
30000,500.0,315.34
```

Experimentación 9

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 11

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.5
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 10000

Tiempo realizado : 216.54665875434875 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

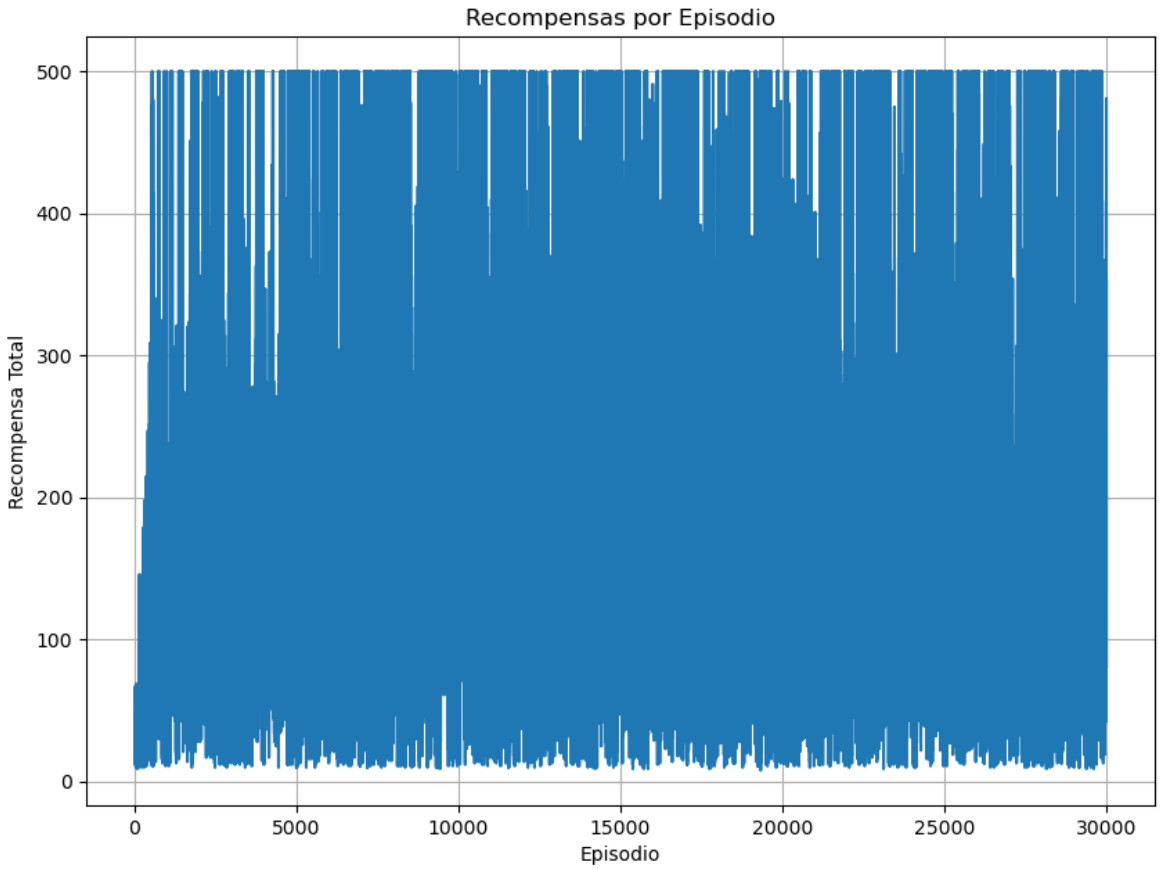
Cálculo de datos (salida_SARSA_11.csv)

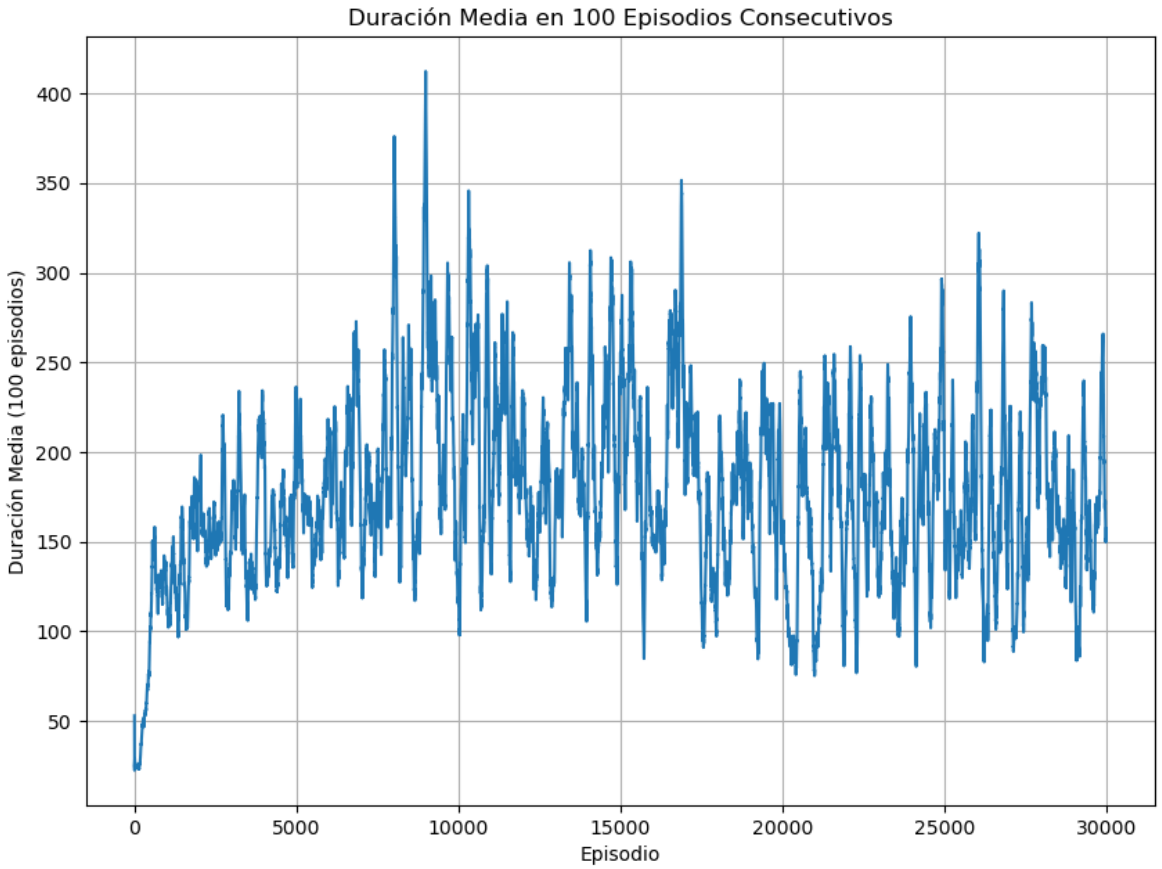
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,53.0,0.00
2,15.0,0.00
3,17.0,0.00
4,14.0,0.00
5,14.0,0.00
6,22.0,0.00
7,36.0,0.00
8,31.0,0.00
9,17.0,0.00
10,67.0,0.00
...
29979,203.0,162.41
29980,42.0,160.70
29981,48.0,159.68
29982,62.0,157.41
29983,82.0,154.95
29984,142.0,155.36
29985,160.0,153.87
29986,66.0,152.22
29987,144.0,150.89
29988,180.0,150.46
29989,201.0,151.17
29990,177.0,151.53
29991,114.0,150.61
29992,156.0,149.77
29993,200.0,150.11
29994,171.0,150.72
29995,245.0,151.77
29996,481.0,155.15
29997,107.0,154.80
29998,80.0,154.46
29999,132.0,154.81
30000,352.0,157.31
```

Experimentación 11

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 12

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.7
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 30000
Pasos x episodio = 10000

Tiempo realizado : 143.38361144065857 segundos

No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

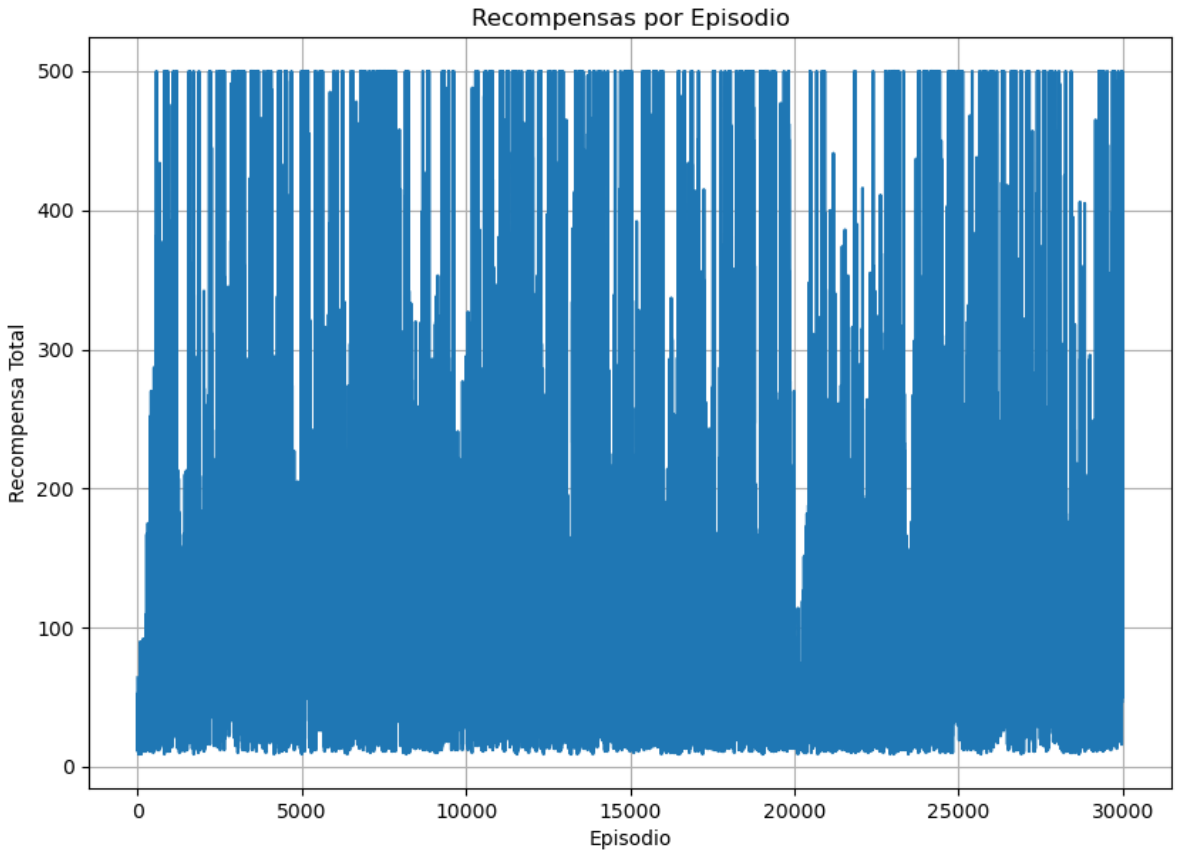
Cálculo de datos (salida_SARSA_12.csv)

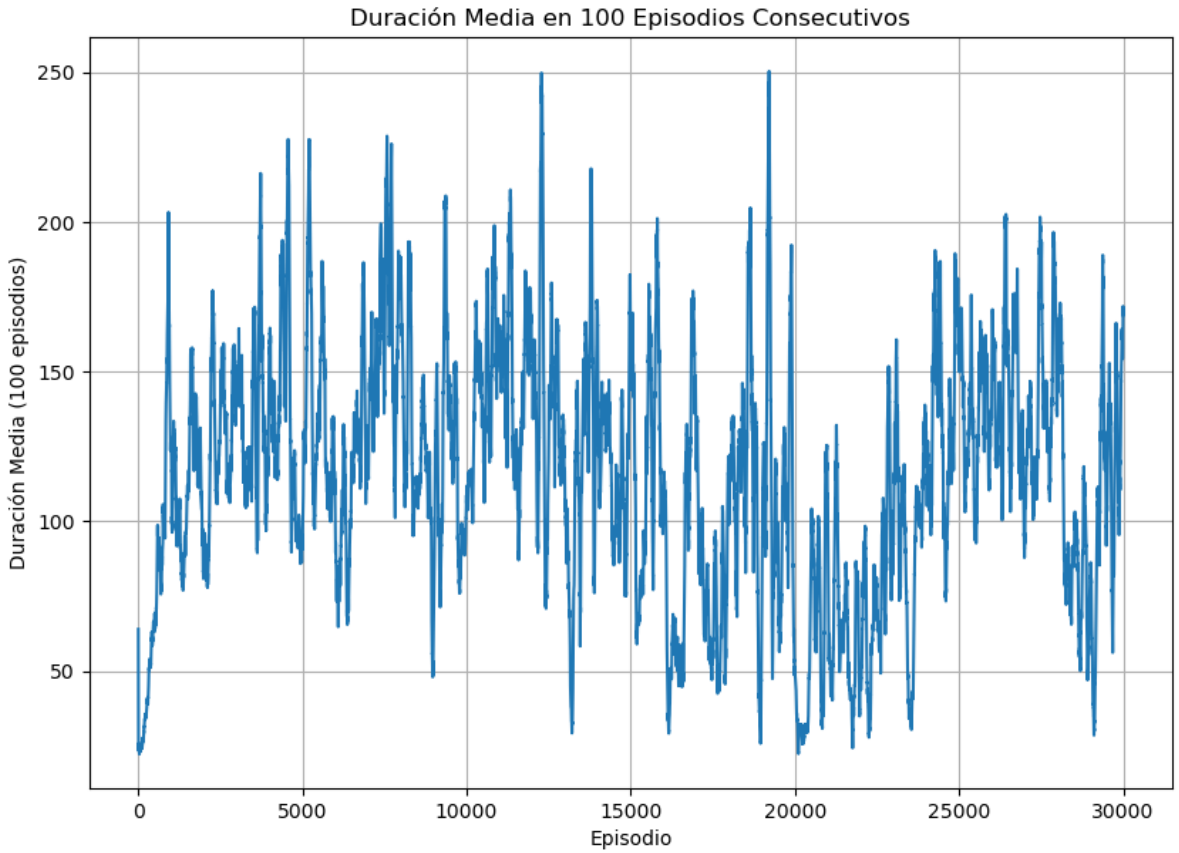
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,53.0,0.00
2,15.0,0.00
3,17.0,0.00
4,14.0,0.00
5,14.0,0.00
6,22.0,0.00
7,36.0,0.00
8,31.0,0.00
9,17.0,0.00
10,67.0,0.00
...
29979,203.0,162.41
29980,42.0,160.70
29981,48.0,159.68
29982,62.0,157.41
29983,82.0,154.95
29984,142.0,155.36
29985,160.0,153.87
29986,66.0,152.22
29987,144.0,150.89
29988,180.0,150.46
29989,201.0,151.17
29990,177.0,151.53
29991,114.0,150.61
29992,156.0,149.77
29993,200.0,150.11
29994,171.0,150.72
29995,245.0,151.77
29996,481.0,155.15
29997,107.0,154.80
29998,80.0,154.46
29999,132.0,154.81
30000,352.0,157.31
```

Experimentación 12

2. Algoritmo SARSA

Gráfico





No se ha resuelto, no se llega a la media de 500 de 100 episodios continuos

Experimentación 13

2. Algoritmo SARSA

Parámetros iniciales

Resultados para SARSA
Alpha = 0.1
Gamma = 0.99
Epsilon = 1.0
Epsilon Decay = 0.995
Episodios = 100000
Pasos x episodio = 10000

Tiempo realizado : 307.3265233039856 segundos

Problema resuelto en 39706 episodios con una duración media de 500.00

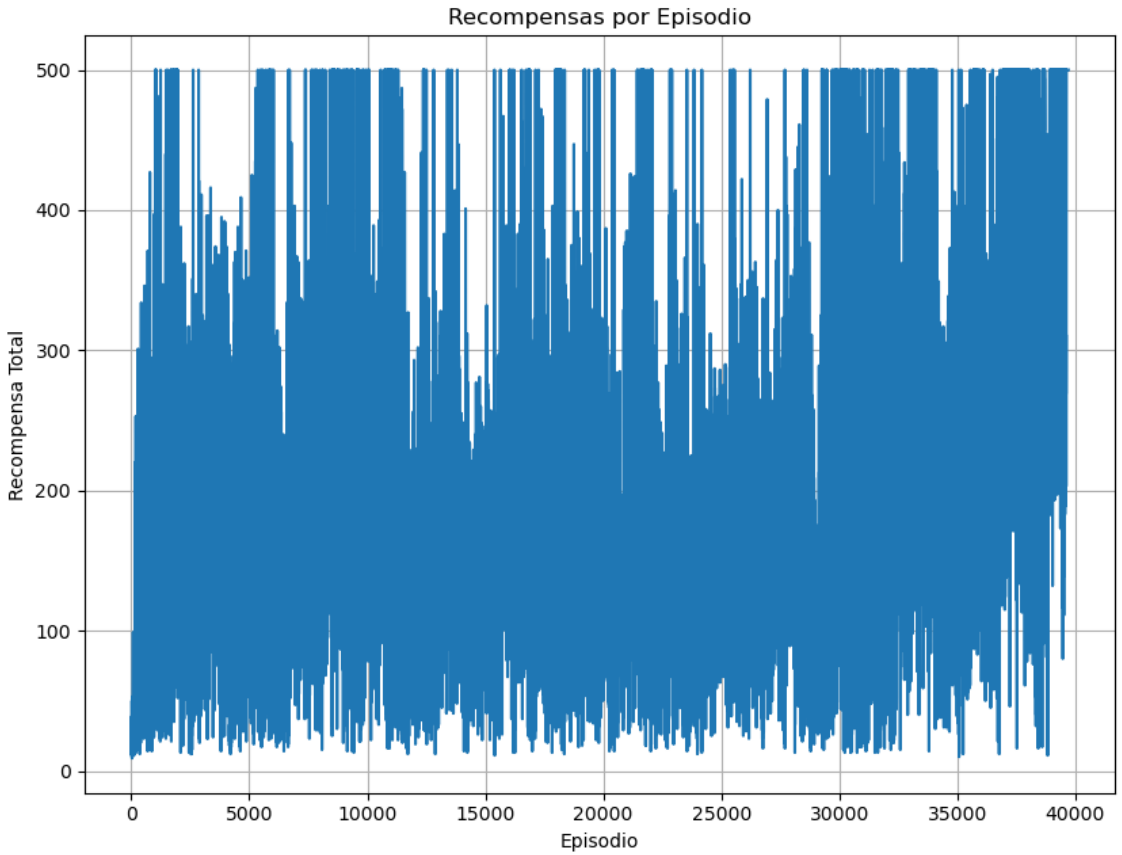
Cálculo de datos (salida_SARSA_13.csv)

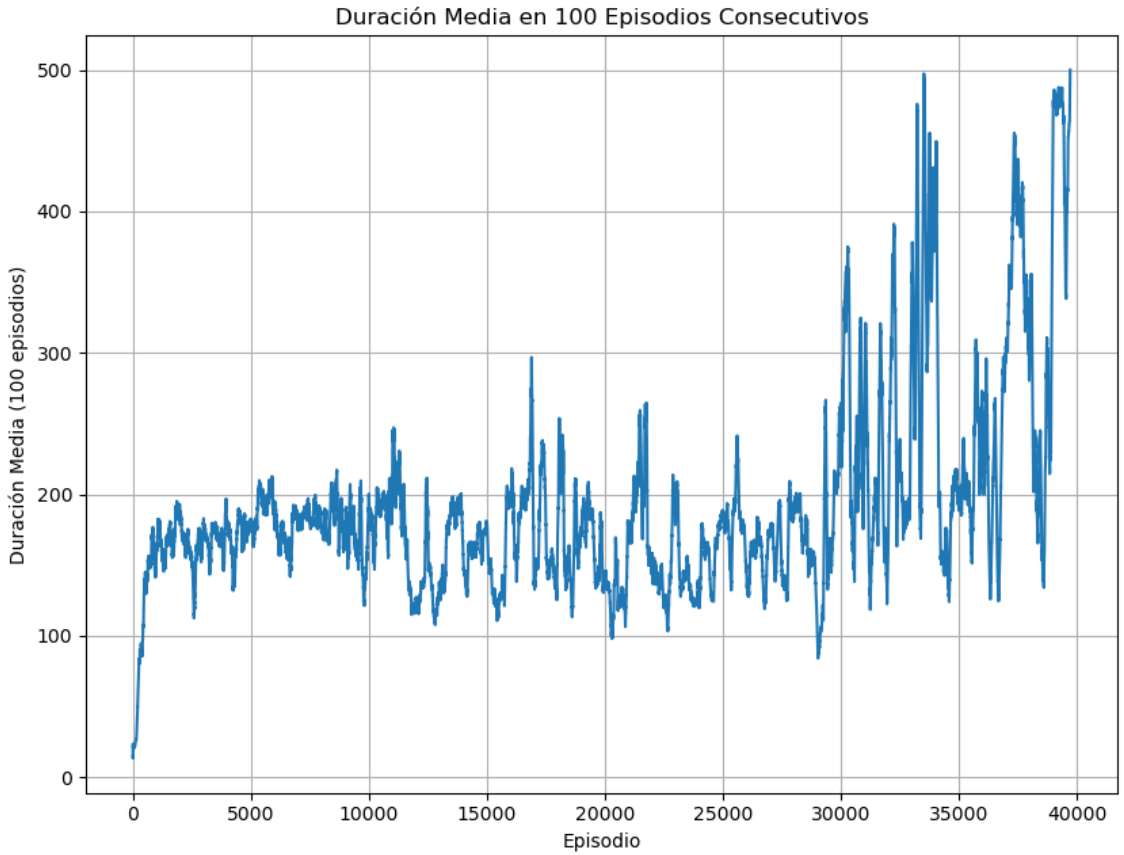
```
Episodio,RecompensaxTotalxEpisodio,RecompensaPromedio100Epidosios
1,15.0,0.00
2,12.0,0.00
3,30.0,0.00
4,19.0,0.00
5,30.0,0.00
6,25.0,0.00
7,35.0,0.00
8,17.0,0.00
9,22.0,0.00
10,14.0,0.00
...
39685,500.0,463.96
39686,500.0,464.82
39687,500.0,466.31
39688,500.0,468.49
39689,500.0,471.24
39690,500.0,473.91
39691,500.0,475.06
39692,500.0,477.48
39693,500.0,477.48
39694,500.0,479.65
39695,500.0,482.76
39696,500.0,484.47
39697,500.0,487.14
39698,500.0,489.44
39699,500.0,491.28
39700,500.0,492.83
39701,500.0,495.14
39702,500.0,495.14
39703,500.0,495.14
39704,500.0,495.14
39705,500.0,497.03
39706,500.0,500.00
```

Experimentación 13

2. Algoritmo SARSA

Gráfico





Problema resuelto en 39706 episodios con una duración media de 500.00

Resultados

Tabla resumen de las experimentaciones con Q-Learning y SARSA con diferentes parámetros

Experimento	Algoritmo	Alpha	Gamma	Epsilon	Epsilon_Decay	Episodios	Pasos x episodio	Resuelto	Nro Pasos	Tiempo (segundos)
1	Q-Learning	0.1	0.99	1	0.995	10000	5000	SI	8491	
2	Q-Learning	0.1	0.99	1	0.995	10000	5000	NO	10000	90.44
3	Q-Learning	0.1	0.99	1	0.995	20000	5000	SI	11455	97.73
4	Q-Learning	0.2	0.99	1	0.995	20000	5000	SI	8427	83.84
5	Q-Learning	0.3	0.99	1	0.995	20000	5000	NO	20000	201.12
6	Q-Learning	0.4	0.99	1	0.995	20000	5000	NO	20000	186.48
7	Q-Learning	0.5	0.99	1	0.995	20000	5000	NO	20000	157.67
8	Q-Learning	0.5	0.99	1	0.995	20000	5000	NO	20000	157.63
9	Q-Learning	0.5	0.99	1	0.995	30000	5000	NO	30000	218.35
10	Q-Learning	0.5	0.99	1	0.995	30000	5000	NO	30000	281.16
1	SARSA	0.1	0.99	1	0.995	10000	5000	NO	10000	83.4
2	SARSA	0.1	0.99	1	0.995	10000	5000	NO	10000	50.07
3	SARSA	0.1	0.99	1	0.995	20000	5000	NO	20000	113.97
5	SARSA	0.1	0.99	1	0.995	30000	5000	NO	30000	270.81
7	SARSA	0.1	0.99	1	0.995	30000	10000	NO	30000	290.23
9	SARSA	0.2	0.99	1	0.995	30000	10000	NO	30000	345.31
11	SARSA	0.5	0.99	1	0.995	30000	10000	NO	30000	216.55
12	SARSA	0.7	0.99	1	0.995	30000	10000	NO	30000	143.38
13	SARSA	0.1	0.99	1	0.995	100000	10000	SI	39706	307.33

Las modificaciones de parámetros se encuentran en rojo

Resultados

- Las experimentaciones con Q-Learning lograron en menos de 20000 episodios, con Alpha = 0.1 en menos de 100 segundos.
- En Q-Learning cuando se incrementaron los valores de Alpha no se pudo lograr la solución dentro de los 30000 episodios.
- Con SARSA con Alha = 0.1, se encontró la solución en casi 40000 episodios, con un tiempo de 307 segundos.
- Con SARSA se realizaron aumentos de Alpha desde 0.1 hasta 0.7, también se aumentaron los episodios desde 10000 hasta 30000 y los números de pasos por episodio se subió a 10000, pero no se logró llegar a la solución.

7. Conclusiones

Conclusiones

Habiéndose usado la configuración de Cartpole-v1 para Q-Learning y SARSA de los resultados se puede concluir:

- Q-Learning usa el valor máximo de la siguiente acción, no depende de la política, lo que le lleva a una convergencia más rápida en entornos donde los estados iniciales estén cerca de las correctas. En la tabla anterior se observa que Q-Learning encuentra la solución usando menos episodios.
- SARSA toma la acción dependiendo de la política actual, lo que le lleva a realizar una exploración más detallada y para al final hallar la solución, esto se observa en la tabla donde la cantidad de episodios pueden ser varias veces más que de Q-Learning para encontrar la solución.
- SARSA es mejor para entornos que requieran una minuciosa o detallada exploración.
- Para realizar experimentaciones con Cartpole-v1 es recomendable usar Q-Learning.

8. Bibliografía

Bibliografía

- **Gymnasium Cart-Pole Environment**:

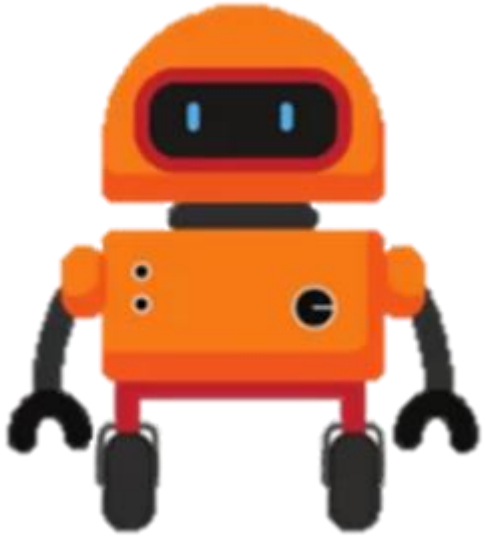
  Cart-Pole Environment in Gymnasium

  https://gymnasium.farama.org/environments/classic_control/cart_pole/

  Gymnasium Documentation

  https://gymnasium.farama.org/
- **Aprendizaje por Refuerzo:**

  Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto.



GRACIAS . . .