Introducci?n al Aprendizaje por Refuerzos

June 1, 2017

1 Introducción al Aprendizaje por Refuerzos

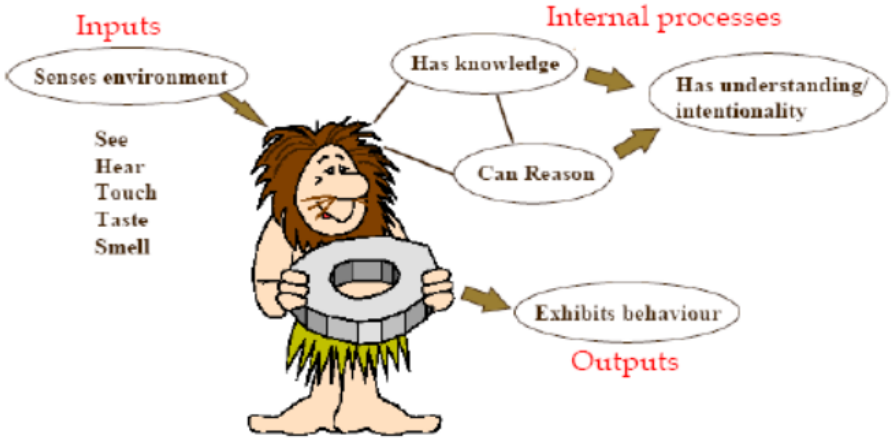
- Introducción. Modelo Agente Entorno. Agente Situado. Arquitectura Actor-Crítico.
- Aprendizaje por Refuerzos. Elementos. Ciclo del Aprendizaje por Refuerzos. Definición Formal.
- Procesos de Decisión de Markov. Función de Valor. Ecuación de Bellman. Optimalidad.
- Aproximaciones al Aprendizaje. Model Free y Model Based.
  - Iteración de Política.
  - Iteración de Valor.
- Ejercicios.

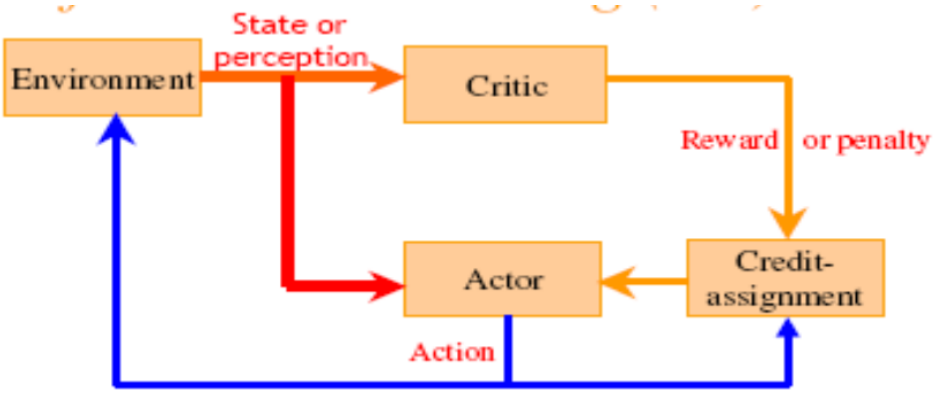
1.1 5to año - Ingeniería en Sistemas de Información

1.1.1 Facultad Regional Villa María

1.2 Introducción: Entidad Inteligente -> Agente Situado

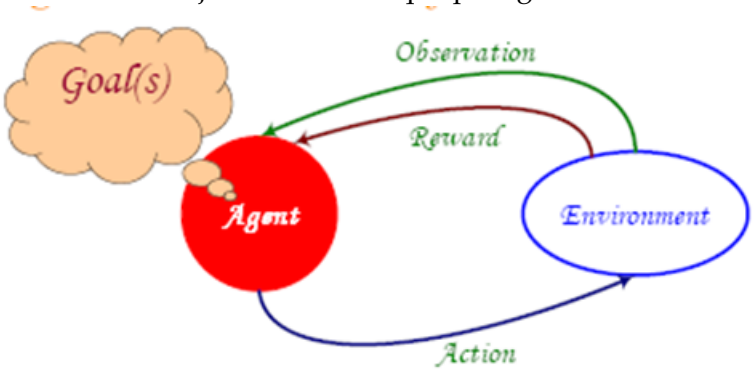
- El desarrollo de la inteligencia requiere que la entidad o el agente esté situada/o en un entorno **(Measuring universal intelligence: Towards an anytime intelligence test, Hernandez-Orallo & Dowe, Artificial Intelligence, 2010).**





1.3 Agent-Environment Framework

- El agente y su entorno interactúan a través de la ejecución de acciones, observación de estados y señales rewards. La inteligencia tendrá efecto sólo si el agente tiene claramente definidos objetivos o metas que persigue activamente mientras ocurre la interacción.



1.4 Arquitectura Actor-Crítico

1.5 Aprendizaje por Refuerzos

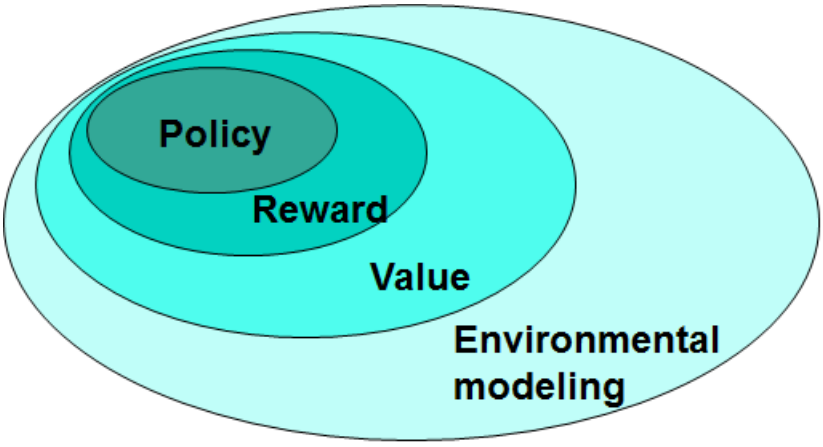
- La toma de decisiones secuencial involucra aprender sobre nuestro entorno y elegir acciones que maximizan el retorno esperado. El RL computacional, inspirado por estas ideas, las formalizo y produjo un impacto importante en robótica, machine learning y neurociencias.

- El Aprendizaje por Refuerzos (RL) consiste en un agente que se encuentra en algún estado $s \in S$ inmerso en un entorno E y toma acciones $a \in A$ en busca de una meta. El agente puede ser modelado formalmente como una función f, que toma un historial de interacción como entrada, y devuelve una acción a tomar. Una manera conveniente para representar el agente es una medida de probabilidad sobre el set A de acciones, en base a un historial de interacción:

$$f(a_n|s_1a_1r_2s_2a_2...r_ns_n)$$

que representa la probabilidad de la acción a en el ciclo n dado un historial de interacción.

- Problema RL: ¿Cómo el agente produce la distribución de probabilidad sobre las acciones?



- Dilema de exploración - explotación: debido a que el Agente no recibe ejemplos de entrenamiento, debe probar alternativas, procesar los resultados de sus acciones y modificar su comportamiento en algún sentido. ¿Cuándo explotar este conocimiento vs. cuándo probar nuevas estrategias?

1.6 Elementos del Aprendizaje por Refuerzos

- **Policy (Política):**

Una política define la manera de comportarse de un agente, en cualquier momento de tiempo dado. Basicamente, es un mapeo de un estado o percepción s a una acción a, pudiendo ser estocásticas.

- **Reward Function (Función de Recompensa)**

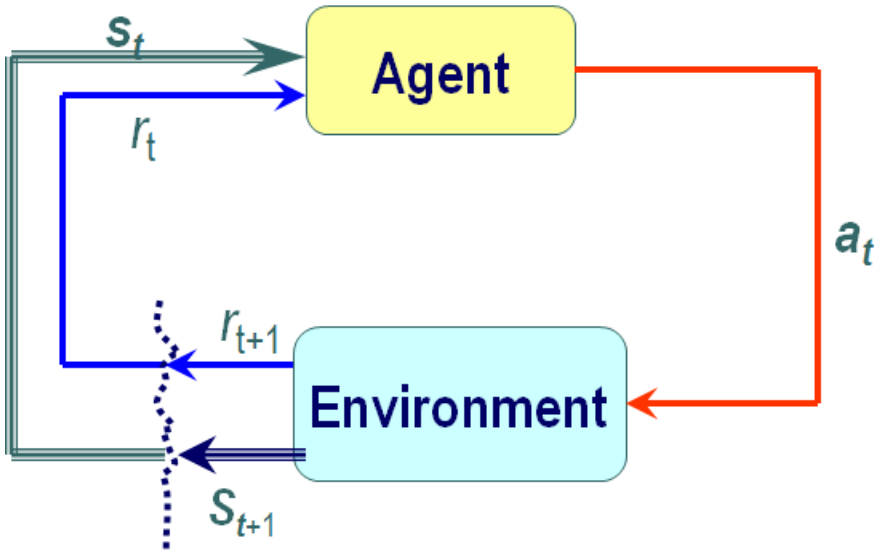
Define cuantitativamente el objetivo del agente. Es un mapeo de un par estado-acción a un número real que indica "cuán deseable" es ejecutar dicha acción en ese estado. Asimismo, el único objetivo del agente es maximizar la recompensa total que recibe a lo large del tiempo. Cabe mencionar que, si bien la función de reward no puede ser alterada por el agente, provee las bases para cambiar la política del mismo.

- **Value function (Función de Valor)**

La función de valor se diferencia de la función de reward en el sentido de que indica "cuán deseable" es, a largo plazo, ejecutar una acción en un determinado estado. Así, el valor de un estado s es la cantidad total de reward que el agente espera obtener a futuro comenzando la interacción en el estado s.

- **Environment (Entorno)**

El entorno se encuentra constitutido por todo aquel elemento (real o simulado) que el agente no puede controlar. Es con quién el agente interactúa a partir de la ejecución de acciones de control.



1.7 Ciclo del Aprendizaje por Refuerzos

1.7.1 Definición formal

- Si el problema de RL dado tiene un conjunto finito de estados y acciones y satisface la propiedad de Markov entonces puede definirse como un Proceso de Decisión de Markov

$$MDPFinito = (S, A, P(.), R(.), \gamma) \tag{1}$$

donde

$$S = s_1, s_2, ..., s_n$$

es un conjunto finito de estados.

$$A = a_1, a_2, ..., a_m$$

es un conjunto finito de acciones.

$$P_a(s, s') = P(s_{t+1} = s | s_t = s, a_t = a)$$

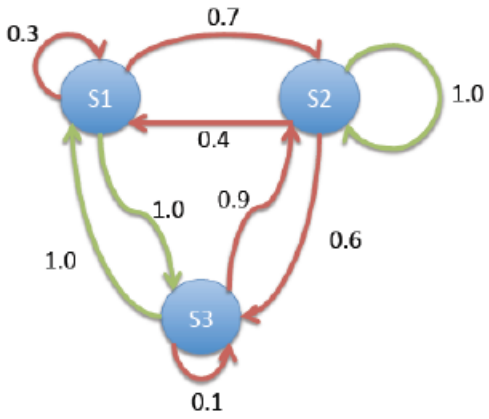
es la probabilidad de que la acción a tomada en tiempo t y en estado s lleve al agente al estado s' en tiempo t+1

$$R_a(s, s')$$

es la recompensa inmediata recibido tras transicionar, luego de tomar la acción a, desde el estado s al estado s'

$$\gamma \in [0, 1]$$

es el factor de descuento, representando la diferencia en la importancia de la recompensa a corto plazo vs la recompensa a largo plazo.



R(s1) = +1
R(s2) = 0
R(s3) = -1

Función de transición T:

S	A	S'	p
S1	R	S1	0.3
	R	S2	0.7
	V	S3	1.0
S2	R	S1	0.4
	R	S2	0.6
	V	S2	1.0
S3	V	S1	1.0
	R	S3	0.1
	R	S2	0.9

- Un episodio (instancia) de este MDP forma una secuencia finita

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, ..., s_{n-1}, a_{n-1}, r_n, s_n$$

donde

$$s_n$$

es un estado final (o n es el tiempo de corte).

- La recompensa total del episodio está dado por

$$R = r_1 + r_2 + ... + r_n$$

- En consecuencia, la recompensa a futuro partiendo del tiempo t está dado por

$$R_t = r_t + r_{t+1} + ...$$

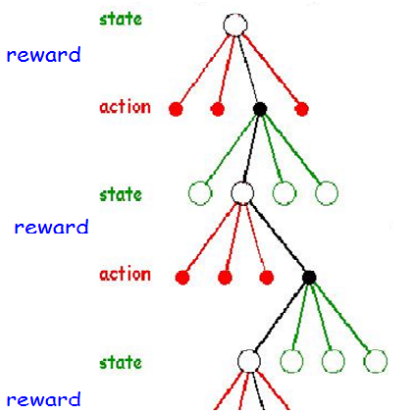
- Hay que considerar que el ambiente es estocástico en la mayor parte de los entornos reales y, por tanto, la recompensa suele diverger mientras más alejado se encuentre el instante de tiempo considerado. Es por esto que se utiliza un parámetro γ llamado *factor de descuento*, para descontar el valor de las recompensas futuras. De esta manera,

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... = r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3}...) = r_t + \gamma R_{t+1} \quad (2)$$

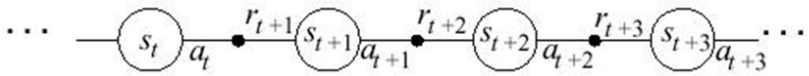
- Si utilizamos $\gamma = 0$, el agente priorizará sólo la recompensa inmediata, mientras que $\gamma = 1$ hará que considere todas los recompensas de la misma manera, independientemente del momento en donde las

Función de Estado - Valor para la Política π :

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\}$$



Predicción y control en secuencias *estado-acción-recompensa* , con un objetivo a largo plazo.



reciba.

Given an MDP $\langle S, A, T, R \rangle$, a policy is a computable function that outputs for each state $s \in S$ an action $a \in A$ (or $a \in A(s)$). Formally, a *deterministic* policy π is a function defined as $\pi : S \to A$. It is also possible to define a *stochastic* policy as $\pi : S \times A \to [0,1]$ such that for each state $s \in S$, it holds that $\pi(s,a) \geq 0$ and $\sum_{a \in A} \pi(s,a) = 1$

1.8 Procesos de Decisión de Markov

1.8.1 Función de Valor

- El valor de un estado es el retorno esperado por el agente, comenzando la interacción en dicho estado, dependiendo de la política ejecutada por el agente.

- El valor de la ejecución de una acción en un estado es el retorno esperado por el agente, comenzando la interacción en dicho estado a partir de la ejecución de dicha acción, dependiendo de la política ejecutada por el agente.

Una propiedad fundamental de las funciones de valor es que satisfacen ciertas propiedades recursivas. Para cualquier política π y cualquier estado s, V(s) y Q(s,a) pueden ser definidas recursivamente en términos de la denominada *Ecuación de Bellman* ** (Bellman, 1957) **

1.8.2 Ecuación de Bellman

- La idea básica es:

Función de Acción - Valor para la Política π :

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\}$$

$$\begin{aligned} R_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \cdots \\ &= r_{t+1} + \gamma\left(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} \cdots\right) \\ &= r_{t+1} + \gamma R_{t+1} \end{aligned}$$

- Entonces,

- O, sin el operador de valor esperado:

La ecuación anterior refleja el hecho de que el valor de un estado se encuentra definido en términos de la recompensa inmediata y los valores de los estados siguientes ponderados en función de las probabilidades de transición, y adicionalmente un factor de descuento.

1.8.3 Ecuación de Optimalidad de Bellman

La Ecuación de Optimalidad de Bellman refleja el hecho de que el Valor de un estado bajo la política óptima debe ser igual al retorno esperado para la mejor acción en dicho estado:

Al mismo tiempo, la acción óptima para un estado s dada la función de valor, puede obtenerse mediante:

La política anterior se denomina **Política Greedy**, dado que selecciona la mejor acción para cada estado, teniendo en cuenta la función de valor V(s). De manera análoga, la función de acción-valor óptima puede expresarse como:

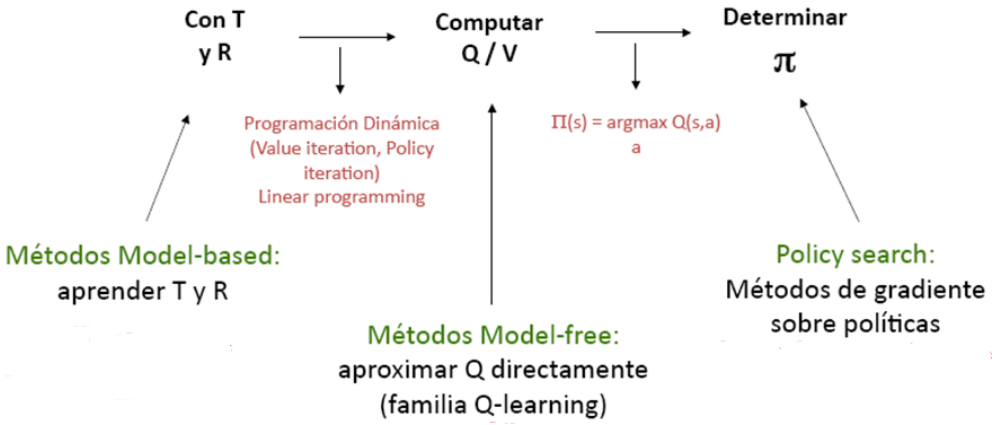
$$\begin{aligned} V^{\pi}(s) &= E_{\pi}\{R_t | s_t = s\} \\ &= E_{\pi}\{r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s\} \end{aligned}$$

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{\pi}(s') \right]$$

$$V^{*}(s) = \max_{a \in A} \sum_{s' \in S} T(s,a,s') \left(R(s,a,s') + \gamma V^{*}(s') \right)$$

$$\pi^{*}(s) = \arg\max_{a} \sum_{s' \in S} T(s,a,s') \left(R(s,a,s') + \gamma V^{*}(s') \right)$$

$$Q^{*}(s,a) = \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma \max_{a'} Q^{*}(s',a') \right)$$



Initialize array v arbitrarily (e.g., $v(s) = 0$ for all $s \in \mathcal{S}^+$)

Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $temp \leftarrow v(s)$
    $v(s) \leftarrow \max_a \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma v(s')]$
    $\Delta \leftarrow \max(\Delta, |temp - v(s)|)$
until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, π, such that
  $\pi(s) = \arg\max_a \sum_{s'} p(s'|s,a)\left[r(s,a,s') + \gamma v(s')\right]$

1.9 Aproximaciones para el aprendizaje de V y Q

1.9.1 Model Based vs. Model Free

- Model-free aprende Q/V directamente y presenta muy baja complejidad computacional.
- Model-based aprende T y R y usa un algoritmo de planning para encontrar la política. Uso eficiente de los datos/experiencia. Alto costo computacional.

1.9.2 Programación Dinámica: Iteración de Valor e Iteración de Política (Model Based)

Iteración de Valor

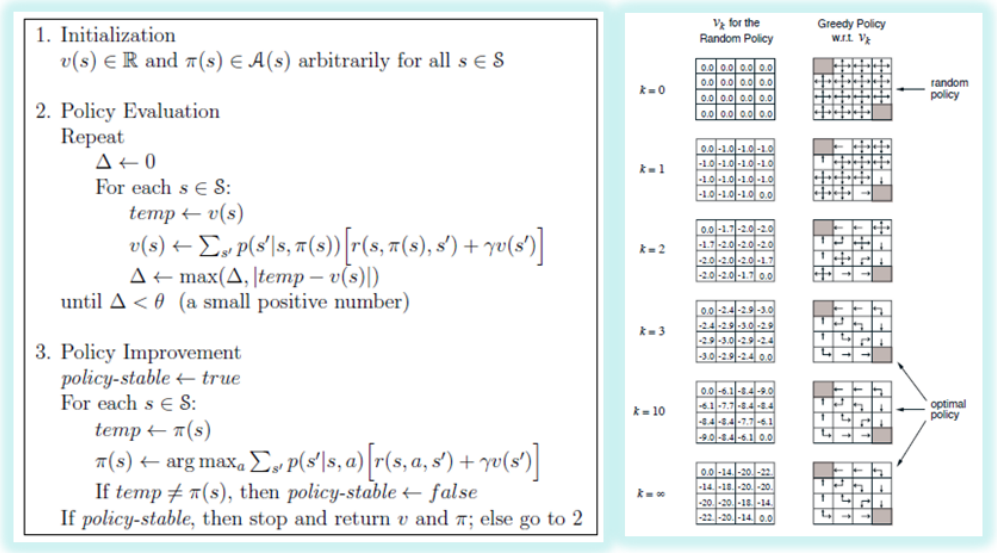
Iteración de Política

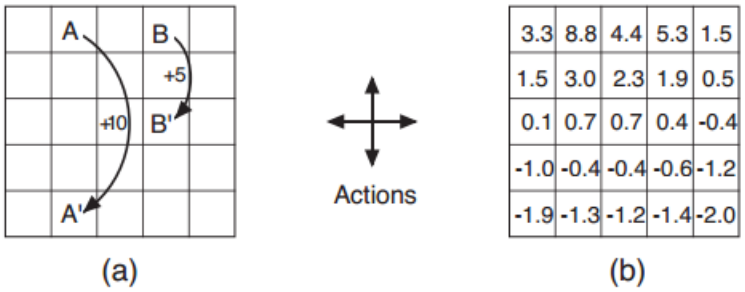
1.10 Ejercicios

Fecha de entrega: **14/06/2017**

Nota: la resolución de los ejercicios es **individual**; en el caso de que dos ejercicios enviados contengan un código igual o muy similar (sin considerar los comentarios), se los considerará a ambos como desaprobados. La reutilización del código de los notebooks está permitida (por ejemplo para confeccionar gráficos).

1. Un entorno denominado **"gridworld"** consiste en un agente que se mueve en una grilla formada por un conjunto de celdas, cada una de las cuales se corresponde con un estado. En cada una de las celdas, el agente puede ejecutar una entre cuatro acciones posibles: norte, sur, este y oeste, las que producen el efecto de mover el agente hacia la celda adyacente de acuerdo a la acción ejecutada (de manera determinística). Aquella acción que lleva al agente





fuera de la grilla, tiene el efecto de mantener al mismo en la misma celda, pero producen una recompensa de -1. Las demás acciones producen una recompensa de 0, excepto aquellas que mueven al agente fuera de los estados especiales denominados A y B. Desde el estado A, las cuatro acciones producen una recompensa de 10, y el efecto es que el esado siguiente siempre es A'. Lo mismo ocurre con el estado B, excepto que la recompensa es 5 y el estado siguiente es B' (Ver figura inferior a)).

La parte b) de la figura, muestra la función de valor calculada para un agente que actúa de manera aleatoria, es decir, aquel que siempre elige las acciones de manera equiprobable, obtenidas a partir de la aplicación de la siguiente implementación de Iteración de Valor:

```
In [1]: from __future__ import print_function
        from python_utils.import_ import import_global
        import numpy as np

        # Límites de la grilla y posiciones especiales
        WORLD_SIZE = 5
        A_POS = [0, 1]
```

```
A_PRIME_POS = [4, 1]
B_POS = [0, 3]
B_PRIME_POS = [2, 3]
discount = 0.9

world = np.zeros((WORLD_SIZE, WORLD_SIZE))

# acciones left, up, right, down
actions = ['L', 'U', 'R', 'D']

# se agrega en actionProb la probabilidad de las acciones para la política
actionProb = []
for i in range(0, WORLD_SIZE):
    actionProb.append([])
    for j in range(0, WORLD_SIZE):
        actionProb[i].append(dict({'L':0.25, 'U':0.25, 'R':0.25, 'D':0.25}))

# se setea la función de transición y la función de reward
nextState = []
actionReward = []
for i in range(0, WORLD_SIZE):
    nextState.append([])
    actionReward.append([])
    for j in range(0, WORLD_SIZE):
        next = dict()
        reward = dict()
        if i == 0:
            next['U'] = [i, j]
            reward['U'] = -1.0
        else:
            next['U'] = [i - 1, j]
            reward['U'] = 0.0

        if i == WORLD_SIZE - 1:
            next['D'] = [i, j]
            reward['D'] = -1.0
        else:
            next['D'] = [i + 1, j]
            reward['D'] = 0.0

        if j == 0:
            next['L'] = [i, j]
            reward['L'] = -1.0
        else:
            next['L'] = [i, j - 1]
            reward['L'] = 0.0

        if j == WORLD_SIZE - 1:
```

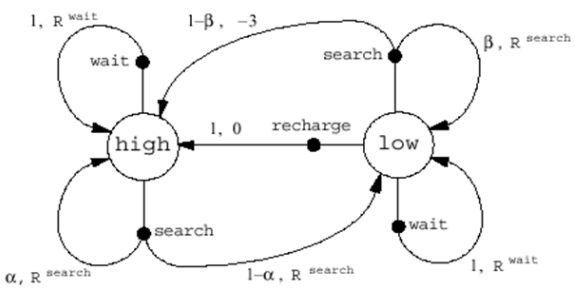
```
                next['R'] = [i, j]
                reward['R'] = -1.0
            else:
                next['R'] = [i, j + 1]
                reward['R'] = 0.0

            if [i, j] == A_POS:
                next['L'] = next['R'] = next['D'] = next['U'] = A_PRIME_POS
                reward['L'] = reward['R'] = reward['D'] = reward['U'] = 10.0

            if [i, j] == B_POS:
                next['L'] = next['R'] = next['D'] = next['U'] = B_PRIME_POS
                reward['L'] = reward['R'] = reward['D'] = reward['U'] = 5.0

            nextState[i].append(next)
            actionReward[i].append(reward)

    #Iteración de Valor
    while True:
        # Se itera hasta lograr la convergencia
        newWorld = np.zeros((WORLD_SIZE, WORLD_SIZE))
        for i in range(0, WORLD_SIZE):
            for j in range(0, WORLD_SIZE):
                for action in actions:
                    newPosition = nextState[i][j][action]
                    # Actualización basada en Bellman
                    newWorld[i, j] += actionProb[i][j][action] * (actionReward[
        if np.sum(np.abs(world - newWorld)) < 1e-4:
            print('Política Aleatoria')
            print(newWorld)
            break
        world = newWorld
```

```
Política Aleatoria
[[ 3.30902999  8.78932551  4.42765281  5.3224012   1.49221235]
 [ 1.52162172  2.9923515   2.25017358  1.90760531  0.5474363 ]
 [ 0.05085614  0.73820423  0.67314689  0.35821982 -0.40310755]
 [-0.97355865 -0.43546179 -0.35484864 -0.58557148 -1.18304148]
 [-1.8576669  -1.34519762 -1.22923364 -1.42288454 -1.97514545]]
```

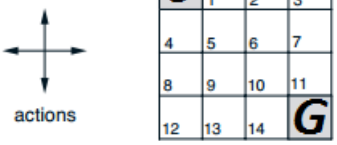
1.1 Genere una gráfica que muestre la evolución del cálculo "np.sum(np.abs(world - newWorld))", para cada paso de actualización realizado hasta lograr la convergencia.

1.2 Modifique el algoritmo anterior para encontrar el valor de la política óptima. Genere una gráfica que muestre la evolución del cálculo "np.sum(np.abs(world - newWorld))", para cada paso de actualización realizado hasta lograr la convergencia.

2. Un robot de reciclaje de residuos debe decidir, en cada instante de tiempo, si busca activamente un contenedor de residuos, si permanece en el lugar en que se encuentra a la espera de



s	s'	a	$p(s'\|s,a)$	$r(s,a,s')$
high	high	search	α	r_{search}
high	low	search	$1-\alpha$	r_{search}
low	high	search	$1-\beta$	-3
low	low	search	β	r_{search}
high	high	wait	1	r_{wait}
high	low	wait	0	r_{wait}
low	high	wait	0	r_{wait}
low	low	wait	1	r_{wait}
low	high	recharge	1	0
low	low	recharge	0	0.



que alguien le traiga un contenedor de residuos, o bien si debe volver a su base para recargar la batería. La mejor forma de encontrar contenedores es buscarlos, pero dicha acción reduce la carga de la batería, mientras que la acción de esperar no. Por otra parte, en cualquier caso en que el robot se encuentre buscando contenedores, existe la posibilidad de que la batería se agote. En este caso, el robot se apaga y necesita ser rescatado (produciendo una recompensa muy baja). Asuma que el problema puede ser modelado de la manera que se muestra en la figura siguiente (Diagrama y función de transición de estados), en donde R_{search} es el número esperado de contenedores que se espera encontrar mientras se ejecuta search, y R_{wait} es el número esperado de contenedores que se espera recibir mientras se ejecuta wait, y $R_{search} > R_{wait}$.

2.1 Implemente un algoritmo de Iteración de Valor para obtener la política óptima del robot de reciclaje.

2.2 Utilice el algoritmo implementado en (2.1) para evaluar cómo cambia el valor de la política óptima a partir de alterar α, β para un valor de $R_{search}=5$ y $R_{wait} = 2$. Para dicha evaluación, emplee una gráfica que permita determinar cuáles son los valores de dichas variables que maximizan el retorno esperado por el agente en cada estado. AYUDA: Varíe algoritmicamente los valores de α y β y calcule la política óptima correspondiente. La gráfica debería presentar tres ejes: α, β, y una variable que totalice los valores de los estados.

2.3 Con los mejores valores de α y β obtenidos en 2.2, realice la misma operación variando R_{search} con un tope de 10, manteniendo R_{wait} = en 4.

3. Un agente debe aprender a llegar en la menor cantidad de pasos desde la posición S a la posición G en una grilla como la que sigue.

Las acciones disponibles en cada estado son las mismas que las descriptas para el agente del Ejercicio 1. El efecto de una acción que llevaría al agente fuera de la grilla, es que el agente vuelve al estado S.

3.1 Plantee una función de recompensa que permita al agente aprender a lograr el objetivo expresado en 3.

3.2 Implemente un algoritmo basado en Iteración de Valor para aprender la política óptima en el entorno especificado.

3.3 Realice una gráfica que permita evaluar como cambia el valor de la política óptima en relación al factor de descuento γ.

4. Implemente la solución de los ejercicios anteriores empleando Iteración de Política. Puede utilizar como base de implementación el siguiente código fuente:

```
In [2]: import numpy as np

        """Este es el pseudocódigo correspondiente al algoritmo implementado más ab

        1: Procedure Policy_Iteration(S,A,P,R)
        2:           Inputs
        3:                     S is the set of all states
        4:                     A is the set of all actions
        5:                     P is state transition function specifying P(s'|s,a)
        6:                     R is a reward function R(s,a,s')
        7:           Output
        8:                     optimal policy π
        9:           Local
        10:                     action array π[S]
        11:                     Boolean variable noChange
        12:                     real array V[S]
        13:           set π arbitrarily
        14:           repeat
        15:                     noChange ←true
        16:                     Solve V[s] = ∑s'∈S P(s'|s,π[s])(R(s,a,s')+γV[s'])
        17:                     for each s∈S do
        18:                               Let QBest=V[s]
        19:                               for each a ∈A do
        20:                                         Let Qsa=∑s'∈S P(s'|s,a)(R(s,a,
        21:                                         if (Qsa > QBest) then
        22:                                                   π[s]←a
        23:                                                   QBest ←Qsa
        24:                                                   noChange ←false
        25:           until noChange
        26:           return π
        """

        # aquí se setea un entorno de ejemplo
        states = [0,1,2,3,4]
        actions = [0,1]
        N_STATES = len(states)
        N_ACTIONS = len(actions)
        P = np.zeros((N_STATES, N_ACTIONS, N_STATES))  # Probabilidades de Transici
        R = np.zeros((N_STATES, N_ACTIONS, N_STATES))  # Rewards

        P[0,0,1] = 1.0
        P[1,1,2] = 1.0
        P[2,0,3] = 1.0
```

```
P[3,1,4] = 1.0
P[4,0,4] = 1.0


R[0,0,1] = 1
R[1,1,2] = 10
R[2,0,3] = 100
R[3,1,4] = 1000
R[4,0,4] = 1.0

# factor de descuento
gamma = 0.75

# inicializa la política y la función de valor
policy = [0 for s in range(N_STATES)]
V = np.zeros(N_STATES)

print("Política Inicial")
print(policy)

is_value_changed = True
iterations = 0
while is_value_changed:
    is_value_changed = False
    iterations += 1
    # corre la iteración de valor para cada estado
    for s in range(N_STATES):
        V[s] = sum([P[s,policy[s],s1] * (R[s,policy[s],s1] + gamma*V[s1]) f

    # realiza la mejora de la política
    for s in range(N_STATES):
        q_best = V[s]
        for a in range(N_ACTIONS):
            q_sa = sum([P[s, a, s1] * (R[s, a, s1] + gamma * V[s1]) for s1
            if q_sa > q_best:
                print("State", s, ": q_sa", q_sa, "q_best", q_best)
                policy[s] = a
                q_best = q_sa
                is_value_changed = True

    print ("Iteracion:", iterations)

print ("Política Final")
print (policy)
print (V)
```

```
Política Inicial
[0, 0, 0, 0, 0]
```

```
State 1 : q_sa 85.0 q_best 0.0
State 3 : q_sa 1000.75 q_best 0.0
State 4 : q_sa 1.75 q_best 1.0
Iteracion: 1
State 0 : q_sa 64.75 q_best 1.0
State 2 : q_sa 850.5625 q_best 100.0
State 3 : q_sa 1001.3125 q_best 1000.75
State 4 : q_sa 2.3125 q_best 1.75
Iteracion: 2
State 1 : q_sa 647.921875 q_best 85.0
State 2 : q_sa 850.984375 q_best 850.5625
State 3 : q_sa 1001.734375 q_best 1001.3125
State 4 : q_sa 2.734375 q_best 2.3125
Iteracion: 3
State 0 : q_sa 486.94140625 q_best 64.75
State 1 : q_sa 648.23828125 q_best 647.921875
State 2 : q_sa 851.30078125 q_best 850.984375
State 3 : q_sa 1002.05078125 q_best 1001.734375
State 4 : q_sa 3.05078125 q_best 2.734375
Iteracion: 4
State 0 : q_sa 487.178710938 q_best 486.94140625
State 1 : q_sa 648.475585938 q_best 648.23828125
State 2 : q_sa 851.538085938 q_best 851.30078125
State 3 : q_sa 1002.28808594 q_best 1002.05078125
State 4 : q_sa 3.2880859375 q_best 3.05078125
Iteracion: 5
State 0 : q_sa 487.356689453 q_best 487.178710938
State 1 : q_sa 648.653564453 q_best 648.475585938
State 2 : q_sa 851.716064453 q_best 851.538085938
State 3 : q_sa 1002.46606445 q_best 1002.28808594
State 4 : q_sa 3.46606445313 q_best 3.2880859375
Iteracion: 6
State 0 : q_sa 487.49017334 q_best 487.356689453
State 1 : q_sa 648.78704834 q_best 648.653564453
State 2 : q_sa 851.84954834 q_best 851.716064453
State 3 : q_sa 1002.59954834 q_best 1002.46606445
State 4 : q_sa 3.59954833984 q_best 3.46606445313
Iteracion: 7
State 0 : q_sa 487.590286255 q_best 487.49017334
State 1 : q_sa 648.887161255 q_best 648.78704834
State 2 : q_sa 851.949661255 q_best 851.84954834
State 3 : q_sa 1002.69966125 q_best 1002.59954834
State 4 : q_sa 3.69966125488 q_best 3.59954833984
Iteracion: 8
State 0 : q_sa 487.665370941 q_best 487.590286255
State 1 : q_sa 648.962245941 q_best 648.887161255
State 2 : q_sa 852.024745941 q_best 851.949661255
State 3 : q_sa 1002.77474594 q_best 1002.69966125
```

```
State 4 : q_sa 3.77474594116 q_best 3.69966125488
Iteracion: 9
State 0 : q_sa 487.721684456 q_best 487.665370941
State 1 : q_sa 649.018559456 q_best 648.962245941
State 2 : q_sa 852.081059456 q_best 852.024745941
State 3 : q_sa 1002.83105946 q_best 1002.77474594
State 4 : q_sa 3.83105945587 q_best 3.77474594116
Iteracion: 10
State 0 : q_sa 487.763919592 q_best 487.721684456
State 1 : q_sa 649.060794592 q_best 649.018559456
State 2 : q_sa 852.123294592 q_best 852.081059456
State 3 : q_sa 1002.87329459 q_best 1002.83105946
State 4 : q_sa 3.8732945919 q_best 3.83105945587
Iteracion: 11
State 0 : q_sa 487.795595944 q_best 487.763919592
State 1 : q_sa 649.092470944 q_best 649.060794592
State 2 : q_sa 852.154970944 q_best 852.123294592
State 3 : q_sa 1002.90497094 q_best 1002.87329459
State 4 : q_sa 3.90497094393 q_best 3.8732945919
Iteracion: 12
State 0 : q_sa 487.819353208 q_best 487.795595944
State 1 : q_sa 649.116228208 q_best 649.092470944
State 2 : q_sa 852.178728208 q_best 852.154970944
State 3 : q_sa 1002.92872821 q_best 1002.90497094
State 4 : q_sa 3.92872820795 q_best 3.90497094393
Iteracion: 13
State 0 : q_sa 487.837171156 q_best 487.819353208
State 1 : q_sa 649.134046156 q_best 649.116228208
State 2 : q_sa 852.196546156 q_best 852.178728208
State 3 : q_sa 1002.94654616 q_best 1002.92872821
State 4 : q_sa 3.94654615596 q_best 3.92872820795
Iteracion: 14
State 0 : q_sa 487.850534617 q_best 487.837171156
State 1 : q_sa 649.147409617 q_best 649.134046156
State 2 : q_sa 852.209909617 q_best 852.196546156
State 3 : q_sa 1002.95990962 q_best 1002.94654616
State 4 : q_sa 3.95990961697 q_best 3.94654615596
Iteracion: 15
State 0 : q_sa 487.860557213 q_best 487.850534617
State 1 : q_sa 649.157432213 q_best 649.147409617
State 2 : q_sa 852.219932213 q_best 852.209909617
State 3 : q_sa 1002.96993221 q_best 1002.95990962
State 4 : q_sa 3.96993221273 q_best 3.95990961697
Iteracion: 16
State 0 : q_sa 487.86807416 q_best 487.860557213
State 1 : q_sa 649.16494916 q_best 649.157432213
State 2 : q_sa 852.22744916 q_best 852.219932213
State 3 : q_sa 1002.97744916 q_best 1002.96993221
```

```
State 4 : q_sa 3.97744915955 q_best 3.96993221273
Iteracion: 17
State 0 : q_sa 487.87371187 q_best 487.86807416
State 1 : q_sa 649.17058687 q_best 649.16494916
State 2 : q_sa 852.23308687 q_best 852.22744916
State 3 : q_sa 1002.98308687 q_best 1002.97744916
State 4 : q_sa 3.98308686966 q_best 3.97744915955
Iteracion: 18
State 0 : q_sa 487.877940152 q_best 487.87371187
State 1 : q_sa 649.174815152 q_best 649.17058687
State 2 : q_sa 852.237315152 q_best 852.23308687
State 3 : q_sa 1002.98731515 q_best 1002.98308687
State 4 : q_sa 3.98731515224 q_best 3.98308686966
Iteracion: 19
State 0 : q_sa 487.881111364 q_best 487.877940152
State 1 : q_sa 649.177986364 q_best 649.174815152
State 2 : q_sa 852.240486364 q_best 852.237315152
State 3 : q_sa 1002.99048636 q_best 1002.98731515
State 4 : q_sa 3.99048636418 q_best 3.98731515224
Iteracion: 20
State 0 : q_sa 487.883489773 q_best 487.881111364
State 1 : q_sa 649.180364773 q_best 649.177986364
State 2 : q_sa 852.242864773 q_best 852.240486364
State 3 : q_sa 1002.99286477 q_best 1002.99048636
State 4 : q_sa 3.99286477314 q_best 3.99048636418
Iteracion: 21
State 0 : q_sa 487.88527358 q_best 487.883489773
State 1 : q_sa 649.18214858 q_best 649.180364773
State 2 : q_sa 852.24464858 q_best 852.242864773
State 3 : q_sa 1002.99464858 q_best 1002.99286477
State 4 : q_sa 3.99464857985 q_best 3.99286477314
Iteracion: 22
State 0 : q_sa 487.886611435 q_best 487.88527358
State 1 : q_sa 649.183486435 q_best 649.18214858
State 2 : q_sa 852.245986435 q_best 852.24464858
State 3 : q_sa 1002.99598643 q_best 1002.99464858
State 4 : q_sa 3.99598643489 q_best 3.99464857985
Iteracion: 23
State 0 : q_sa 487.887614826 q_best 487.886611435
State 1 : q_sa 649.184489826 q_best 649.183486435
State 2 : q_sa 852.246989826 q_best 852.245986435
State 3 : q_sa 1002.99698983 q_best 1002.99598643
State 4 : q_sa 3.99698982617 q_best 3.99598643489
Iteracion: 24
State 0 : q_sa 487.88836737 q_best 487.887614826
State 1 : q_sa 649.18524237 q_best 649.184489826
State 2 : q_sa 852.24774237 q_best 852.246989826
State 3 : q_sa 1002.99774237 q_best 1002.99698983
```

```
State 4 : q_sa 3.99774236963 q_best 3.99698982617
Iteracion: 25
State 0 : q_sa 487.888931777 q_best 487.88836737
State 1 : q_sa 649.185806777 q_best 649.18524237
State 2 : q_sa 852.248306777 q_best 852.24774237
State 3 : q_sa 1002.99830678 q_best 1002.99774237
State 4 : q_sa 3.99830677722 q_best 3.99774236963
Iteracion: 26
State 0 : q_sa 487.889355083 q_best 487.888931777
State 1 : q_sa 649.186230083 q_best 649.185806777
State 2 : q_sa 852.248730083 q_best 852.248306777
State 3 : q_sa 1002.99873008 q_best 1002.99830678
State 4 : q_sa 3.99873008291 q_best 3.99830677722
Iteracion: 27
State 0 : q_sa 487.889672562 q_best 487.889355083
State 1 : q_sa 649.186547562 q_best 649.186230083
State 2 : q_sa 852.249047562 q_best 852.248730083
State 3 : q_sa 1002.99904756 q_best 1002.99873008
State 4 : q_sa 3.99904756219 q_best 3.99873008291
Iteracion: 28
State 0 : q_sa 487.889910672 q_best 487.889672562
State 1 : q_sa 649.186785672 q_best 649.186547562
State 2 : q_sa 852.249285672 q_best 852.249047562
State 3 : q_sa 1002.99928567 q_best 1002.99904756
State 4 : q_sa 3.99928567164 q_best 3.99904756219
Iteracion: 29
State 0 : q_sa 487.890089254 q_best 487.889910672
State 1 : q_sa 649.186964254 q_best 649.186785672
State 2 : q_sa 852.249464254 q_best 852.249285672
State 3 : q_sa 1002.99946425 q_best 1002.99928567
State 4 : q_sa 3.99946425373 q_best 3.99928567164
Iteracion: 30
State 0 : q_sa 487.89022319 q_best 487.890089254
State 1 : q_sa 649.18709819 q_best 649.186964254
State 2 : q_sa 852.24959819 q_best 852.249464254
State 3 : q_sa 1002.99959819 q_best 1002.99946425
State 4 : q_sa 3.9995981903 q_best 3.99946425373
Iteracion: 31
State 0 : q_sa 487.890323643 q_best 487.89022319
State 1 : q_sa 649.187198643 q_best 649.18709819
State 2 : q_sa 852.249698643 q_best 852.24959819
State 3 : q_sa 1002.99969864 q_best 1002.99959819
State 4 : q_sa 3.99969864272 q_best 3.9995981903
Iteracion: 32
State 0 : q_sa 487.890398982 q_best 487.890323643
State 1 : q_sa 649.187273982 q_best 649.187198643
State 2 : q_sa 852.249773982 q_best 852.249698643
State 3 : q_sa 1002.99977398 q_best 1002.99969864
```

```
State 4 : q_sa 3.99977398204 q_best 3.99969864272
Iteracion: 33
State 0 : q_sa 487.890455487 q_best 487.890398982
State 1 : q_sa 649.187330487 q_best 649.187273982
State 2 : q_sa 852.249830487 q_best 852.249773982
State 3 : q_sa 1002.99983049 q_best 1002.99977398
State 4 : q_sa 3.99983048653 q_best 3.99977398204
Iteracion: 34
State 0 : q_sa 487.890497865 q_best 487.890455487
State 1 : q_sa 649.187372865 q_best 649.187330487
State 2 : q_sa 852.249872865 q_best 852.249830487
State 3 : q_sa 1002.99987286 q_best 1002.99983049
State 4 : q_sa 3.9998728649 q_best 3.99983048653
Iteracion: 35
State 0 : q_sa 487.890529649 q_best 487.890497865
State 1 : q_sa 649.187404649 q_best 649.187372865
State 2 : q_sa 852.249904649 q_best 852.249872865
State 3 : q_sa 1002.99990465 q_best 1002.99987286
State 4 : q_sa 3.99990464867 q_best 3.9998728649
Iteracion: 36
State 0 : q_sa 487.890553487 q_best 487.890529649
State 1 : q_sa 649.187428487 q_best 649.187404649
State 2 : q_sa 852.249928487 q_best 852.249904649
State 3 : q_sa 1002.99992849 q_best 1002.99990465
State 4 : q_sa 3.99992848651 q_best 3.99990464867
Iteracion: 37
State 0 : q_sa 487.890571365 q_best 487.890553487
State 1 : q_sa 649.187446365 q_best 649.187428487
State 2 : q_sa 852.249946365 q_best 852.249928487
State 3 : q_sa 1002.99994636 q_best 1002.99992849
State 4 : q_sa 3.99994636488 q_best 3.99992848651
Iteracion: 38
State 0 : q_sa 487.890584774 q_best 487.890571365
State 1 : q_sa 649.187459774 q_best 649.187446365
State 2 : q_sa 852.249959774 q_best 852.249946365
State 3 : q_sa 1002.99995977 q_best 1002.99994636
State 4 : q_sa 3.99995977366 q_best 3.99994636488
Iteracion: 39
State 0 : q_sa 487.89059483 q_best 487.890584774
State 1 : q_sa 649.18746983 q_best 649.187459774
State 2 : q_sa 852.24996983 q_best 852.249959774
State 3 : q_sa 1002.99996983 q_best 1002.99995977
State 4 : q_sa 3.99996983024 q_best 3.99995977366
Iteracion: 40
State 0 : q_sa 487.890602373 q_best 487.89059483
State 1 : q_sa 649.187477373 q_best 649.18746983
State 2 : q_sa 852.249977373 q_best 852.24996983
State 3 : q_sa 1002.99997737 q_best 1002.99996983
```

```
State 4 : q_sa 3.99997737268 q_best 3.99996983024
Iteracion: 41
State 0 : q_sa 487.89060803 q_best 487.890602373
State 1 : q_sa 649.18748303 q_best 649.187477373
State 2 : q_sa 852.24998303 q_best 852.249977373
State 3 : q_sa 1002.99998303 q_best 1002.99997737
State 4 : q_sa 3.99998302951 q_best 3.99997737268
Iteracion: 42
State 0 : q_sa 487.890612272 q_best 487.89060803
State 1 : q_sa 649.187487272 q_best 649.18748303
State 2 : q_sa 852.249987272 q_best 852.24998303
State 3 : q_sa 1002.99998727 q_best 1002.99998303
State 4 : q_sa 3.99998727213 q_best 3.99998302951
Iteracion: 43
State 0 : q_sa 487.890615454 q_best 487.890612272
State 1 : q_sa 649.187490454 q_best 649.187487272
State 2 : q_sa 852.249990454 q_best 852.249987272
State 3 : q_sa 1002.99999045 q_best 1002.99998727
State 4 : q_sa 3.9999904541 q_best 3.99998727213
Iteracion: 44
State 0 : q_sa 487.890617841 q_best 487.890615454
State 1 : q_sa 649.187492841 q_best 649.187490454
State 2 : q_sa 852.249992841 q_best 852.249990454
State 3 : q_sa 1002.99999284 q_best 1002.99999045
State 4 : q_sa 3.99999284058 q_best 3.9999904541
Iteracion: 45
State 0 : q_sa 487.89061963 q_best 487.890617841
State 1 : q_sa 649.18749463 q_best 649.187492841
State 2 : q_sa 852.24999463 q_best 852.249992841
State 3 : q_sa 1002.99999463 q_best 1002.99999284
State 4 : q_sa 3.99999463043 q_best 3.99999284058
Iteracion: 46
State 0 : q_sa 487.890620973 q_best 487.89061963
State 1 : q_sa 649.187495973 q_best 649.18749463
State 2 : q_sa 852.249995973 q_best 852.24999463
State 3 : q_sa 1002.99999597 q_best 1002.99999463
State 4 : q_sa 3.99999597282 q_best 3.99999463043
Iteracion: 47
State 0 : q_sa 487.89062198 q_best 487.890620973
State 1 : q_sa 649.18749698 q_best 649.187495973
State 2 : q_sa 852.24999698 q_best 852.249995973
State 3 : q_sa 1002.99999698 q_best 1002.99999597
State 4 : q_sa 3.99999697962 q_best 3.99999597282
Iteracion: 48
State 0 : q_sa 487.890622735 q_best 487.89062198
State 1 : q_sa 649.187497735 q_best 649.18749698
State 2 : q_sa 852.249997735 q_best 852.24999698
State 3 : q_sa 1002.99999773 q_best 1002.99999698
```

```
State 4 : q_sa 3.99999773471 q_best 3.99999697962
Iteracion: 49
State 0 : q_sa 487.890623301 q_best 487.890622735
State 1 : q_sa 649.187498301 q_best 649.187497735
State 2 : q_sa 852.249998301 q_best 852.249997735
State 3 : q_sa 1002.9999983 q_best 1002.99999773
State 4 : q_sa 3.99999830104 q_best 3.99999773471
Iteracion: 50
State 0 : q_sa 487.890623726 q_best 487.890623301
State 1 : q_sa 649.187498726 q_best 649.187498301
State 2 : q_sa 852.249998726 q_best 852.249998301
State 3 : q_sa 1002.99999873 q_best 1002.9999983
State 4 : q_sa 3.99999872578 q_best 3.99999830104
Iteracion: 51
State 0 : q_sa 487.890624044 q_best 487.890623726
State 1 : q_sa 649.187499044 q_best 649.187498726
State 2 : q_sa 852.249999044 q_best 852.249998726
State 3 : q_sa 1002.99999904 q_best 1002.99999873
State 4 : q_sa 3.99999904433 q_best 3.99999872578
Iteracion: 52
State 0 : q_sa 487.890624283 q_best 487.890624044
State 1 : q_sa 649.187499283 q_best 649.187499044
State 2 : q_sa 852.249999283 q_best 852.249999044
State 3 : q_sa 1002.99999928 q_best 1002.99999904
State 4 : q_sa 3.99999928325 q_best 3.99999904433
Iteracion: 53
State 0 : q_sa 487.890624462 q_best 487.890624283
State 1 : q_sa 649.187499462 q_best 649.187499283
State 2 : q_sa 852.249999462 q_best 852.249999283
State 3 : q_sa 1002.99999946 q_best 1002.99999928
State 4 : q_sa 3.99999946244 q_best 3.99999928325
Iteracion: 54
State 0 : q_sa 487.890624597 q_best 487.890624462
State 1 : q_sa 649.187499597 q_best 649.187499462
State 2 : q_sa 852.249999597 q_best 852.249999462
State 3 : q_sa 1002.9999996 q_best 1002.99999946
State 4 : q_sa 3.99999959683 q_best 3.99999946244
Iteracion: 55
State 0 : q_sa 487.890624698 q_best 487.890624597
State 1 : q_sa 649.187499698 q_best 649.187499597
State 2 : q_sa 852.249999698 q_best 852.249999597
State 3 : q_sa 1002.9999997 q_best 1002.9999996
State 4 : q_sa 3.99999969762 q_best 3.99999959683
Iteracion: 56
State 0 : q_sa 487.890624773 q_best 487.890624698
State 1 : q_sa 649.187499773 q_best 649.187499698
State 2 : q_sa 852.249999773 q_best 852.249999698
State 3 : q_sa 1002.99999977 q_best 1002.9999997
```

```
State 4 : q_sa 3.99999977322 q_best 3.99999969762
Iteracion: 57
State 0 : q_sa 487.89062483 q_best 487.890624773
State 1 : q_sa 649.18749983 q_best 649.187499773
State 2 : q_sa 852.24999983 q_best 852.249999773
State 3 : q_sa 1002.99999983 q_best 1002.99999977
State 4 : q_sa 3.99999982991 q_best 3.99999977322
Iteracion: 58
State 0 : q_sa 487.890624872 q_best 487.89062483
State 1 : q_sa 649.187499872 q_best 649.18749983
State 2 : q_sa 852.249999872 q_best 852.24999983
State 3 : q_sa 1002.99999987 q_best 1002.99999983
State 4 : q_sa 3.99999987243 q_best 3.99999982991
Iteracion: 59
State 0 : q_sa 487.890624904 q_best 487.890624872
State 1 : q_sa 649.187499904 q_best 649.187499872
State 2 : q_sa 852.249999904 q_best 852.249999872
State 3 : q_sa 1002.9999999 q_best 1002.99999987
State 4 : q_sa 3.99999990433 q_best 3.99999987243
Iteracion: 60
State 0 : q_sa 487.890624928 q_best 487.890624904
State 1 : q_sa 649.187499928 q_best 649.187499904
State 2 : q_sa 852.249999928 q_best 852.249999904
State 3 : q_sa 1002.99999993 q_best 1002.9999999
State 4 : q_sa 3.99999992824 q_best 3.99999990433
Iteracion: 61
State 0 : q_sa 487.890624946 q_best 487.890624928
State 1 : q_sa 649.187499946 q_best 649.187499928
State 2 : q_sa 852.249999946 q_best 852.249999928
State 3 : q_sa 1002.99999995 q_best 1002.99999993
State 4 : q_sa 3.99999994618 q_best 3.99999992824
Iteracion: 62
State 0 : q_sa 487.89062496 q_best 487.890624946
State 1 : q_sa 649.18749996 q_best 649.187499946
State 2 : q_sa 852.24999996 q_best 852.249999946
State 3 : q_sa 1002.99999996 q_best 1002.99999995
State 4 : q_sa 3.99999995964 q_best 3.99999994618
Iteracion: 63
State 0 : q_sa 487.89062497 q_best 487.89062496
State 1 : q_sa 649.18749997 q_best 649.18749996
State 2 : q_sa 852.24999997 q_best 852.24999996
State 3 : q_sa 1002.99999997 q_best 1002.99999996
State 4 : q_sa 3.99999996973 q_best 3.99999995964
Iteracion: 64
State 0 : q_sa 487.890624977 q_best 487.89062497
State 1 : q_sa 649.187499977 q_best 649.18749997
State 2 : q_sa 852.249999977 q_best 852.24999997
State 3 : q_sa 1002.99999998 q_best 1002.99999997
```

```
State 4 : q_sa 3.9999999773 q_best 3.99999996973
Iteracion: 65
State 0 : q_sa 487.890624983 q_best 487.890624977
State 1 : q_sa 649.187499983 q_best 649.187499977
State 2 : q_sa 852.249999983 q_best 852.249999977
State 3 : q_sa 1002.99999998 q_best 1002.99999998
State 4 : q_sa 3.99999998297 q_best 3.9999999773
Iteracion: 66
State 0 : q_sa 487.890624987 q_best 487.890624983
State 1 : q_sa 649.187499987 q_best 649.187499983
State 2 : q_sa 852.249999987 q_best 852.249999983
State 3 : q_sa 1002.99999999 q_best 1002.99999998
State 4 : q_sa 3.99999998723 q_best 3.99999998297
Iteracion: 67
State 0 : q_sa 487.89062499 q_best 487.890624987
State 1 : q_sa 649.18749999 q_best 649.187499987
State 2 : q_sa 852.24999999 q_best 852.249999987
State 3 : q_sa 1002.99999999 q_best 1002.99999999
State 4 : q_sa 3.99999999042 q_best 3.99999998723
Iteracion: 68
State 0 : q_sa 487.890624993 q_best 487.89062499
State 1 : q_sa 649.187499993 q_best 649.18749999
State 2 : q_sa 852.249999993 q_best 852.24999999
State 3 : q_sa 1002.99999999 q_best 1002.99999999
State 4 : q_sa 3.99999999282 q_best 3.99999999042
Iteracion: 69
State 0 : q_sa 487.890624995 q_best 487.890624993
State 1 : q_sa 649.187499995 q_best 649.187499993
State 2 : q_sa 852.249999995 q_best 852.249999993
State 3 : q_sa 1002.99999999 q_best 1002.99999999
State 4 : q_sa 3.99999999461 q_best 3.99999999282
Iteracion: 70
State 0 : q_sa 487.890624996 q_best 487.890624995
State 1 : q_sa 649.187499996 q_best 649.187499995
State 2 : q_sa 852.249999996 q_best 852.249999995
State 3 : q_sa 1003.0 q_best 1002.99999999
State 4 : q_sa 3.99999999596 q_best 3.99999999461
Iteracion: 71
State 0 : q_sa 487.890624997 q_best 487.890624996
State 1 : q_sa 649.187499997 q_best 649.187499996
State 2 : q_sa 852.249999997 q_best 852.249999996
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999697 q_best 3.99999999596
Iteracion: 72
State 0 : q_sa 487.890624998 q_best 487.890624997
State 1 : q_sa 649.187499998 q_best 649.187499997
State 2 : q_sa 852.249999998 q_best 852.249999997
State 3 : q_sa 1003.0 q_best 1003.0
```

```
State 4 : q_sa 3.99999999773 q_best 3.99999999697
Iteracion: 73
State 0 : q_sa 487.890624998 q_best 487.890624998
State 1 : q_sa 649.187499998 q_best 649.187499998
State 2 : q_sa 852.249999998 q_best 852.249999998
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.9999999983 q_best 3.99999999773
Iteracion: 74
State 0 : q_sa 487.890624999 q_best 487.890624998
State 1 : q_sa 649.187499999 q_best 649.187499998
State 2 : q_sa 852.249999999 q_best 852.249999998
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999872 q_best 3.9999999983
Iteracion: 75
State 0 : q_sa 487.890624999 q_best 487.890624999
State 1 : q_sa 649.187499999 q_best 649.187499999
State 2 : q_sa 852.249999999 q_best 852.249999999
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999904 q_best 3.99999999872
Iteracion: 76
State 0 : q_sa 487.890624999 q_best 487.890624999
State 1 : q_sa 649.187499999 q_best 649.187499999
State 2 : q_sa 852.249999999 q_best 852.249999999
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999928 q_best 3.99999999904
Iteracion: 77
State 0 : q_sa 487.890624999 q_best 487.890624999
State 1 : q_sa 649.187499999 q_best 649.187499999
State 2 : q_sa 852.249999999 q_best 852.249999999
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999946 q_best 3.99999999928
Iteracion: 78
State 0 : q_sa 487.890625 q_best 487.890624999
State 1 : q_sa 649.1875 q_best 649.187499999
State 2 : q_sa 852.25 q_best 852.249999999
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.9999999996 q_best 3.99999999946
Iteracion: 79
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.9999999997 q_best 3.9999999996
Iteracion: 80
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
```

```
State 4 : q_sa 3.99999999977 q_best 3.9999999997
Iteracion: 81
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999983 q_best 3.99999999977
Iteracion: 82
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999987 q_best 3.99999999983
Iteracion: 83
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.9999999999 q_best 3.99999999987
Iteracion: 84
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999993 q_best 3.9999999999
Iteracion: 85
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999995 q_best 3.99999999993
Iteracion: 86
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999996 q_best 3.99999999995
Iteracion: 87
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999997 q_best 3.99999999996
Iteracion: 88
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
```

```
State 4 : q_sa 3.99999999998 q_best 3.99999999997
Iteracion: 89
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999998 q_best 3.99999999998
Iteracion: 90
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999999 q_best 3.99999999998
Iteracion: 91
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999999 q_best 3.99999999999
Iteracion: 92
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999999 q_best 3.99999999999
Iteracion: 93
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 3.99999999999 q_best 3.99999999999
Iteracion: 94
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 3.99999999999
Iteracion: 95
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 96
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
```

```
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 97
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 98
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 99
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 100
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 101
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 102
State 0 : q_sa 487.890625 q_best 487.890625
State 1 : q_sa 649.1875 q_best 649.1875
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 103
State 0 : q_sa 487.890625 q_best 487.890625
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 104
State 1 : q_sa 649.1875 q_best 649.1875
State 2 : q_sa 852.25 q_best 852.25
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 105
State 0 : q_sa 487.890625 q_best 487.890625
```

```
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 106
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 107
State 2 : q_sa 852.25 q_best 852.25
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 108
State 1 : q_sa 649.1875 q_best 649.1875
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 109
State 0 : q_sa 487.890625 q_best 487.890625
State 3 : q_sa 1003.0 q_best 1003.0
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 110
State 2 : q_sa 852.25 q_best 852.25
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 111
State 1 : q_sa 649.1875 q_best 649.1875
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 112
State 0 : q_sa 487.890625 q_best 487.890625
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 113
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 114
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 115
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 116
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 117
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 118
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 119
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 120
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 121
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 122
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 123
State 4 : q_sa 4.0 q_best 4.0
Iteracion: 124
Iteracion: 125
Política Final
```

```
[0, 1, 0, 1, 0]
[  487.890625   649.1875     852.25      1003.          4.      ]
```