DOCUMENTACIÓN

Alumno: Miguel Rodríguez Gallego

Inteligencia artificial

GDDV 4.3

Aprendizaje Reforzado de acciones
Convocatoria ordinaria - 2023/24.

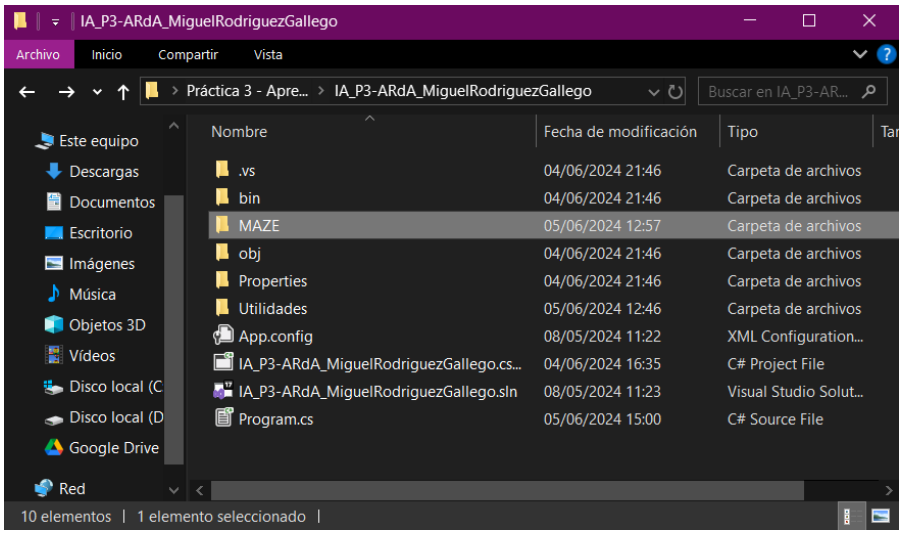
Entrega: 14/04/2024

Contenido

Introducción

En esta práctica desarrollé un sistema de inteligencia artificial por Sarsa y QLearning para escapar de un laberinto.

En la práctica entonces lo que desarrollé fue:

- Gestión del input
- Sarsa
- QLearning
- Sistema de selección de otro escenario personalizado mediante un .txt que el usuario pueda poner una carpeta interna del proyecto llamada MAZE
    - Junto a esto permito la selección de la casilla de inicio y salida, la cual deberá el jugador escoger correctamente o no funcionará.
    - 

Resultados

Estas gráficas son sacadas del proyecto de Visual Studio, en donde creé un Excel para sus respectivas comparaciones.

Gráficas

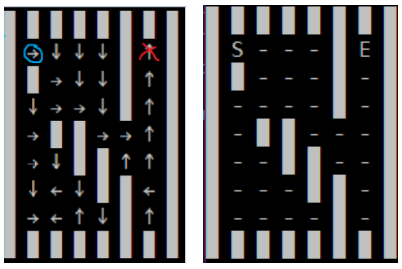
Realicé varias pruebas con estos diferentes mapas como resultados finales de direcciones según la base que se nos propuso la cual trasladé a código:

Utilicé la configuración que recomiendo al ejecutar para cada algoritmo:

- **Tasa de aprendizaje**: 0.1
- **Tasa de descuento**: 0.9
- **Tasa de exploración**: 0.1
- **Recompensa por llegar a la meta**: 100
- **Recompensa por movimiento**: -1
- **Número de entrenamientos**: 100

```
55
56          // Configurar parámetros de aprendizaje
57          double learningRate = 0.1;
58          double discountRate = 0.9;
59          double epsilon = 0.1;
60          int numberOfTrainings = 100; // 100
61          int rewardGoal = 100;
62          int rewardMove = -1;
63
64          // Escoger parámetros interactivos
65          learningRate = GetParameter("Tasa de aprendizaje", 0.1);
66          discountRate = GetParameter("Tasa de descuento", 0.9);
67          epsilon = GetParameter("Tasa de exploración", 0.1);
68          rewardGoal = (int)GetParameter("Recompensa por llegar a la meta", 100);
69          rewardMove = (int)GetParameter("Recompensa por movimiento", -1);
70          numberOfTrainings = (int)GetParameter("Número de entrenamientos", 100);
71
```

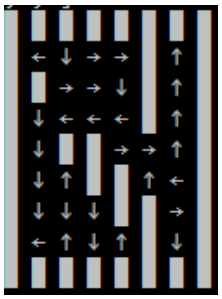
El algoritmo siempre empieza en el punto azul y debe llegar al rojo en los ejemplos que se encuentran en estas pruebas.



Sarsa

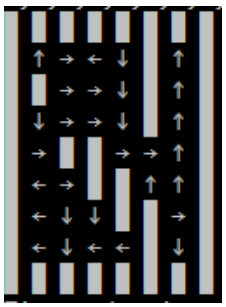
Evolución general

Episodio nº 10



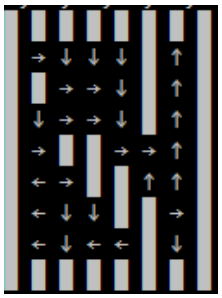
```
Tiempo de ejecución: 4,4715 ms
[95;525;209;338;206;218;61;201;33;157;
```

Episodio nº 50



```
Tiempo de ejecución: 4,4715 ms
[95;525;209;338;206;218;61;201;33;157;85;81;107;36;26;26;130;30;108;17;91;25;17;25;40;113;17;17;13;16;77;88;15;11;27;14;19;17;20;13;16;13;11;13;16;15;13;11;13;13;
```

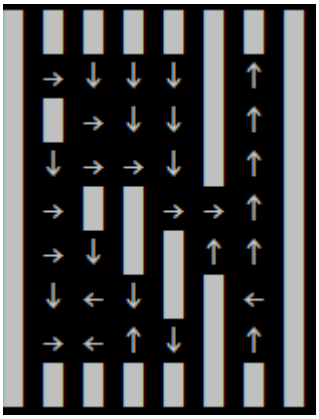
Episodio nº 100

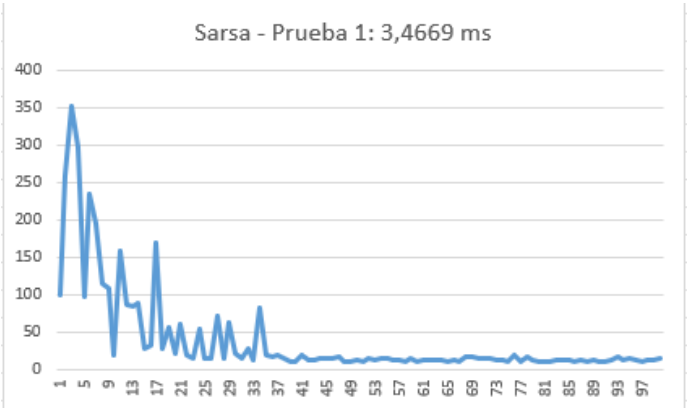


```
Tiempo de ejecución: 4,4715 ms
[95;525;209;338;206;218;61;201;33;157;85;81;107;36;26;26;130;30;108;17;91;25;17;25;40;113;17;17;13;16;77;88;15;11;27;14;19;17;20;13;16;13;11;13;16;15;13;11;13;13;16;11;15;11;14;11;15;14;14;13;11;11;13;11;12;11;1
1;11;13;11;12;11;12;12;12;13;11;13;15;11;14;11;14;15;11;11;12;11;14;16;11;11;11;12;13;11;11;12;11;11]
```

Diferentes ejecuciones completas de 100 episodios

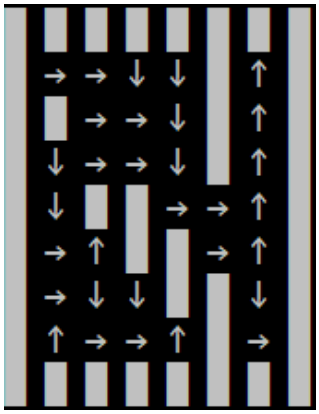
Sarsa 1

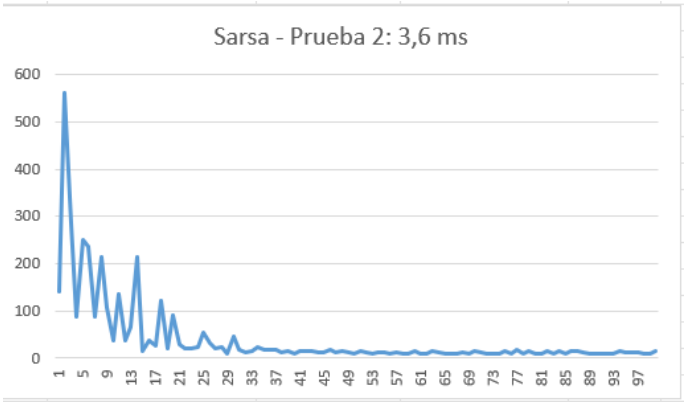




[99;258;352;297;96;234;194;115;108;19;158;86;85;89;28;31;169;27;56;21;59;18;15;54;14;14;71;14;63;21;14;28;12;82;18;16;19;14;11;11;18;13;13;15;15;15;16;11;11;13;11;15;12;15;15;12;12;11;15;11;13;13;12;13;11;13;11;17;16;14;14;14;13;13;11;18;11;16;13;11;11;11;12;13;12;11;13;11;13;11;11;12;16;12;15;13;11;12;12;14]

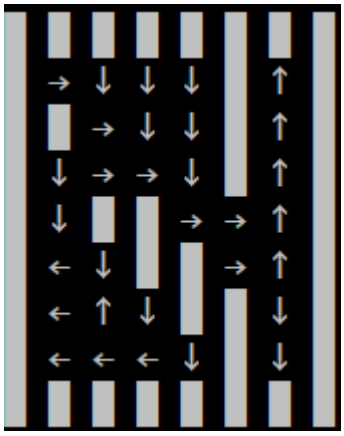
Sarsa 2

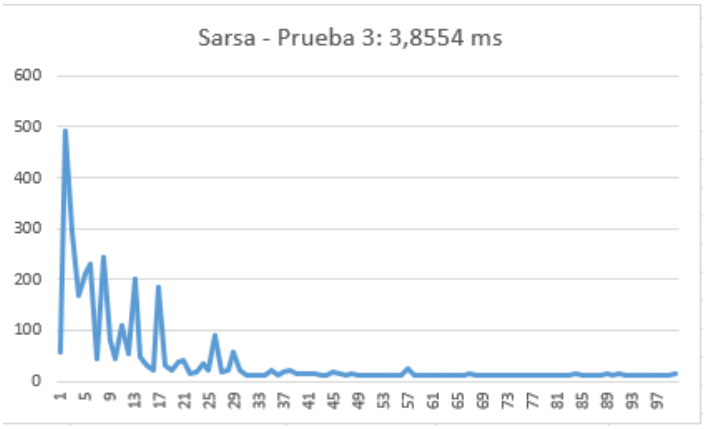




[140;562;288;88;251;236;87;213;109;39;135;37;67;213;14;39;27;122;21;90;29;21;20;23;55;31;21;24;11;45;19;12;14;23;19;17;17;13;14;11;16;15;15;12;13;18;13;14;12;11;14;12;11;13;13;11;12;11;11;15;11;11;16;13;11;11;11;13;11;15;13;11;11;11;14;11;17;11;15;11;11;14;11;16;11;14;15;13;11;11;11;11;11;15;12;13;13;11;11;15]

Sarsa 3





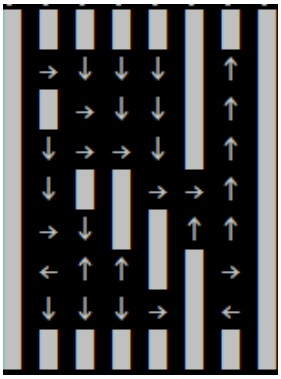
[59;490;297;167;210;230;43;242;81;45;108;55;200;46;31;23;184;32;21;37;42;14;17;35;20;91;18;20;58;21;13;13;13;13;22;13;19;20;16;15;14;15;11;11;18;14;12;14;11;11;12;11;13;13;13;12;24;11;13;12;11;13;13;11;12;11;14;11;11;11;11;11;11;11;13;11;11;11;12;11;11;11;11;15;12;12;13;13;14;13;15;12;11;12;13;13;11;12;11;15]

Qlearning

Evolución general

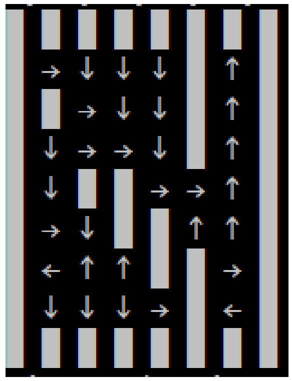
Episodio nº 10

Episodio nº 50



```
Tiempo de ejecución: 4,8933 ms
[219;319;398;125;189;194;224;30;73;91;51;46;174;39;76;66;39;14;68;29;55;29;30;15;17;17;59;18;16;19;14;84;13;18;14;12;16;15;13;15;12;15;16;15;14;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0]
```

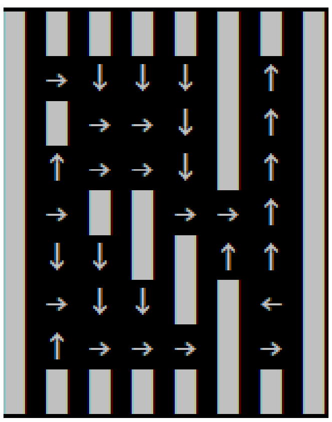
Episodio nº 100

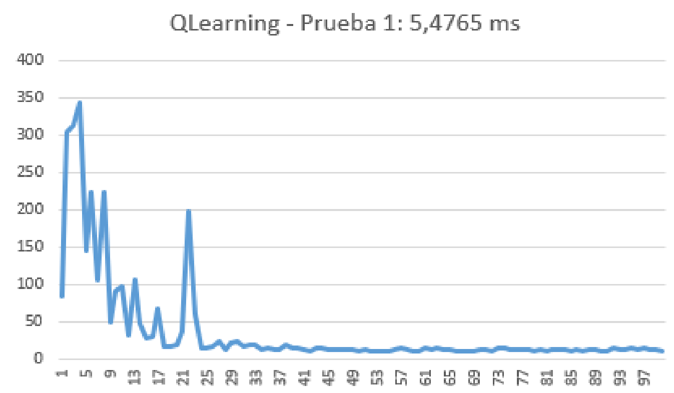


```
Tiempo de ejecución: 4,8933 ms
[219;319;398;125;189;194;224;30;73;91;51;46;174;39;76;66;39;14;68;29;55;29;30;15;17;17;59;18;16;19;14;84;13;18;14;12;16;15;13;15;12;15;16;15;14;11;13;12;15;13;13;14;14;11;11;13;16;11;14;11;11;11;11;11;11;11;13;14;12;11;12;14;13;12;15;14;11;11;13;17;11;12;11;12;11;15;11;15;11;12;13;12;11;11;17;11;11;11;11;12]
```

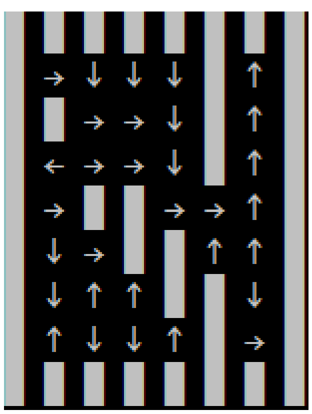
Diferentes ejecuciones completas de 100 episodios

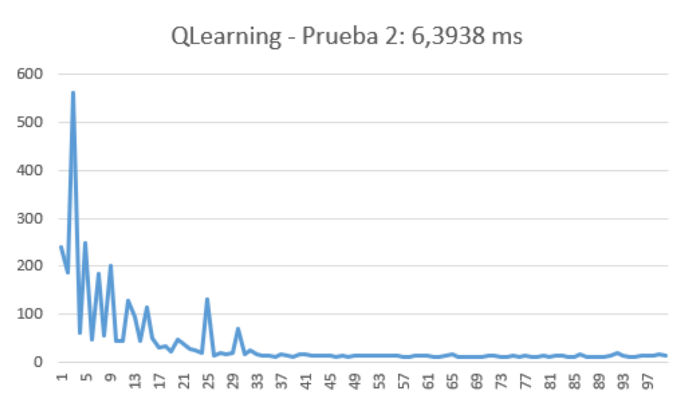
QLearning 1





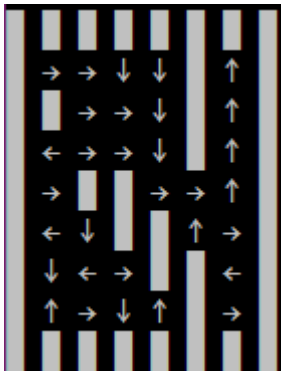
QLearning 2

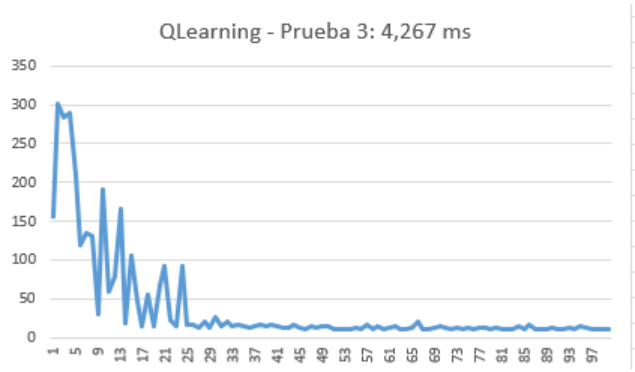




[241;186;563;61;247;46;183;56;200;45;43;127;93;45;115;50;29;34;21;46;38;27;23;20;130;14;20;17;19;68;15;23;15;14;12;11;16;13;11;16;16;12;13;14;13;11;12;11;14;12;12;13;13;14;12;13;11;11;12;12;13;11;11;12;16;11;11;11;11;11;14;13;11;11;12;11;12;11;11;12;11;12;13;11;11;17;11;11;11;11;12;18;13;11;11;13;13;13;16;12]

QLearning 3





[155;302;283;290;211;119;135;131;31;190;60;78;165;18;106;48;14;55;14;67;92;23;15;92;16;16;12;21;13;27;15;20;15;17;14;12;14;16;14;16;14;13;13;17;13;11;14;12;14;14;11;11;11;11;13;11;16;11;14;11;12;14;11;11;12;20;11;11;12;15;13;11;13;11;12;11;12;12;11;12;11;11;11;14;11;17;11;11;11;13;11;11;12;11;15;13;11;11;11;11]

Conclusiones de las comparativas

En las prácticas de evolución general de ambos algoritmos, se aprecia una mejora del episodio 10, al 50 y luego al 100 en ambos rápida visualmente en muchas casillas del mapa viendo las direcciones que aplican como más óptimas.

	Sarsa						
Time:	3,4669						
Action Number:	99	258	352	297	96	234	194
Total episodios:	100						
Action values Media:	38,72						
Action/Time:	28,84421241						
Final Media = ValuesMedia/(Action/Time)	1,34238368						
Time:	3,6						
Action Number:	140	562	288	88	251	236	87
Total episodios:	100						
Action values Media:	39,77						
Action/Time:	27,77777778						
Final Media = ValuesMedia/(Action/Time)	1,43172						
Time:	3,8554						
Action Number:	59	490	297	167	210	230	43
Total episodios:	100						
Action values Media:	38,4						
Action/Time:	25,9376459						
Final Media = ValuesMedia/(Action/Time)	1,4804736						

	Qlearning				
Time:	5,4765				
Action Number:	85	304	312	342	145
Total episodios:	100				
Action values Media:	36,44				
Action/Time:	18,25983749				
Final Media = ValuesMedia/(Action/Time)	1,9956366				
Time:	6,3938				
Action Number:	241	186	563	61	247
Total episodios:	100				
Action values Media:	36,91				
Action/Time:	15,6401514				
Final Media = ValuesMedia/(Action/Time)	2,35995158				
Time:	4,267				
Action Number:	155	302	283	290	211
Total episodios:	100				
Action values Media:	36,92				
Action/Time:	23,43566909				
Final Media = ValuesMedia/(Action/Time)	1,5753764				

Además, realicé tablas con los datos recogidos de los ejemplos dispersos de 3 ejecuciones de ambos algoritmos.

Hice una media final en base a los valores con los que trabajé y los resultados finales del ejercicio.

En cómputo general es más estable y eficiente es el método Sarsa según la tabla que hice, pero si es cierto que su uso puede diferir según lo que queramos hacer, ya que me paré a investigar más allá de los datos y medias que recogí aquí.

Retomando las bases del funcionamiento de cada algoritmo podemos ver esto mismo.

SARSA es un algoritmo “on-policy”, lo que significa que actualiza su política mientras explora el entorno.

Actualiza los valores Q utilizando la acción real tomada en el siguiente estado y es más conservador, tiende a evitar acciones riesgosas durante la exploración. Por lo tanto, es apropiado para problemas donde la exploración es costosa o peligrosa y maneja mejores espacios de acción continuos.

Sin embargo, requiere ajustar los parámetros de exploración para lograr una política óptima.

Q-Learning por el otro lado es un algoritmo “off-policy”, lo que significa que aprende directamente la política óptima sin considerar la acción tomada.

En Q-Learning, se actualizan los valores Q utilizando el máximo valor Q posible para el siguiente estado. Además, tiene mayor varianza por muestra y puede tener problemas de convergencia debido a esto.

Además, aprende directamente la política óptima y es menos dependiente de la exploración, pero puede ignorar penalizaciones por movimientos exploratorios.

En conclusión y según los datos recogidos, si se busca una política cercana a la óptima mientras se explora, es mejor SARSA, pero si se desea aprender directamente la política óptima, Q-Learning es más apropiado.