

EXEMPLO: AMBIENTE 7 x 1

Ambiente

```
1. class Ambiente:
2.     init(posicao_inicial, r_max=1):
3.         ambiente = ['- ', '.', '.', '.', '.', '.', '+']
4.         posicao_agente = posicao_inicial
5.         this.posicao_inicial = posicao_inicial
6.         this.r_max = r_max

7.     property dim_amb:
8.         return len(ambiente)

9.     reiniciar():
10.         mover_agente(posicao_inicial)
11.         return posicao_agente
```


EXEMPLO: AMBIENTE 7 x 1

Ambiente (cont.)

```
1.    actuar(accao):
2.        posicao = posicao_agente
3.        if accao == Accao.ESQ:
4.            posicao -= 1
5.        elif accao == Accao.DIR:
6.            posicao += 1
7.        if posicao >= 0 and posicao < dim_amb:
8.            mover_agente(posicao)

9.    observar():
10.        return posicao_agente, reforco(posicao_agente)
11.
12.    mover_agente(posicao):
13.        posicao_agente = posicao
```


EXEMPLO: AMBIENTE 7 x 1

Ambiente (cont.)

```
1.  reforco(posicao):
2.      elemento = ambiente[posicao]
3.      if elemento == '+':
4.          r = r_max
5.      elif elemento == '-':
6.          r = -r_max
7.      else:
8.          r = 0
9.      return r

10. mostrar():
11.     for posicao in range(dim_amb):
12.         if posicao == posicao_agente:
13.             print('A')
14.         else:
15.             print(ambiente[posicao])
16.     print_line()
```


EXEMPLO: AMBIENTE 7 x 1

Ambiente (cont.)

```
1.  mostrar_politica(politica):
2.      print_line("\nPolítica:")
3.      for s in range(dim_amb):
4.          accao = politica.get(s)
5.          print(accao)
6.      print_line()

7.  mostrar_valor(valor):
8.      print_line("\nValor:")
9.      for s in range(dim_amb):
10.         vs = valor.get(s)
11.         print(vs)
12.     print_line()
```

Acção

```
1. enum Accao:
2.     ESQ
3.     DIR
```


EXEMPLO: AMBIENTE 7 x 1

Agente com aprendizagem por reforço do tipo *Q-Learning*

```
1. class AgenteAprendRef:
2.     def __init__(ambiente, mec_aprend):
3.         self.ambiente = ambiente
4.         self.mec_aprend = mec_aprend
5.         s = None
6.         a = None
7.
8.     def executar(num_episodios):
9.         num_passos_episodio = []
10.        for num_episodio in range(num_episodios):
11.            print(f"\nEpisódio: {num_episodio + 1}")
12.            s = ambiente.reiniciar()
13.            num_passos = 0
14.            while not fim_episodio():
15.                passo_episodio()
16.                num_passos += 1
17.            num_passos_episodio.append(num_passos)
18.        return num_passos_episodio
```


EXEMPLO: AMBIENTE 7 x 1

Agente com aprendizagem por reforço do tipo *Q-Learning* (cont.)

```
1.     fim_episodio():
2.         _, r = ambiente.observear()
3.         return r > 0
4.
5.     passo_episodio():
6.         a = mec_aprend.seleccionar_accao(s)
7.         ambiente.actuar(a)
8.         sn, r = ambiente.observear()
9.         mec_aprend.aprender(s, a, r, sn)
10.        s = sn
11.        ambiente.mostrar()
```


EXEMPLO: AMBIENTE 7 x 1

Teste de aprendizagem por reforço

```
1.  # Parâmetros de teste
2.  posicao_inicial = 3
3.  num_episod = 10
4.  epsilon = 0.1
5.  # Iniciar ambiente e agente
6.  ambiente = Ambiente(posicao_inicial)
7.  accoes = list(Accao)
8.  mec_aprend_ref = MecAprendRef(accoes, epsilon)
9.  agente = AgenteAprendRef(ambiente, mec_aprend_ref)
10. # Executar agente
11. num_passos_episodio = agente.executar(num_episod)
12. # Mostrar política aprendida
13. ambiente.mostrar_politica(mec_aprend_ref.obter_politica())
14. ambiente.mostrar_valor(mec_aprend_ref.obter_valor())
15. # Mostrar número de passos por episódio
16. print_line(f"\nPassos por episódio: {num_passos_episodio}")
```