



# Aprendizaje

## TPN°8: Ejercicios 1, 2, 3, 4

Barraquero Ignacio, Campo Camila, Villarreal Francisco, Marzari Agustina

Facultad de Ingeniería



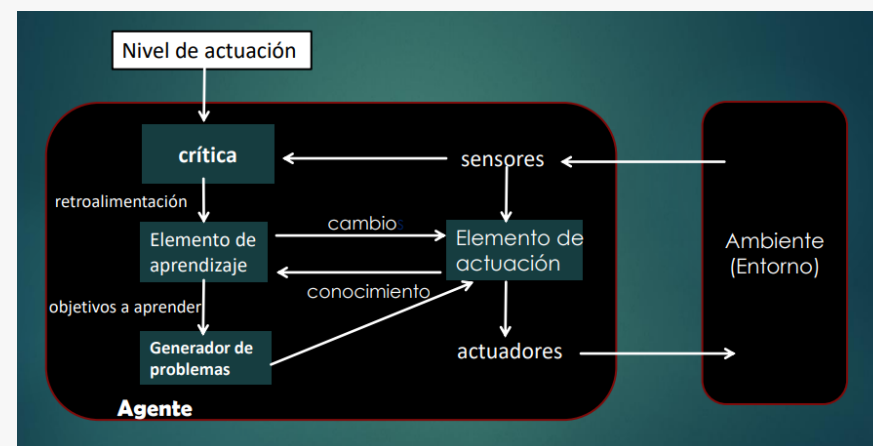
### Introducción al aprendizaje

El **aprendizaje** en Inteligencia Artificial se refiere a la capacidad de los agentes para:

- Mejorar su comportamiento a partir de experiencias pasadas.
- Formular predicciones sobre el futuro.
- Descubrir patrones en grandes volúmenes de datos mediante algoritmos matemáticos y estadísticos.

**Componentes de los Agentes que Aprenden**

1. **Elemento de Actuación:** Selección de acciones externas.
2. **Elemento de Aprendizaje:** Mejora continua según críticas.
3. **Crítica:** Evalúa el desempeño respecto a un estándar.
4. **Generador de Problemas:** Sugiere acciones nuevas para explorar y aprender.



El aprendizaje se clasifica en:

- **No Supervisado**
- **Supervisado**
- **Por Refuerzo**



**Aprendizaje No Supervisado Características:**

- No existen etiquetas para los ejemplos.
- Objetivo: encontrar estructuras o patrones ocultos.

**Algoritmo Clave: K-means**

- Representa cada clúster por su centroide.
- Itera asignación y recalclo hasta converger.
- Aplicaciones: segmentación de clientes, compresión de imágenes, IoT.

**Aprendizaje Supervisado Características:**

- Existen ejemplos etiquetados.
- Objetivo: aprender una función que relacione entradas con salidas.

**Algoritmos Clave:**

- **Aprendizaje Bayesiano:** actualiza hipótesis con Teorema de Bayes.
- **KNN:** clasificación por los vecinos más cercanos.

**Aprendizaje por Refuerzo Características:**

- No hay etiquetas fijas.
- El agente aprende interactuando con el entorno.
- Recibe recompensas o castigos según las acciones.

**Algoritmo Clave: Q-Learning**

- Aprende una política óptima maximizando recompensas acumuladas.
- Utiliza una tabla  $Q(s, a)$  para almacenar valores de acción.

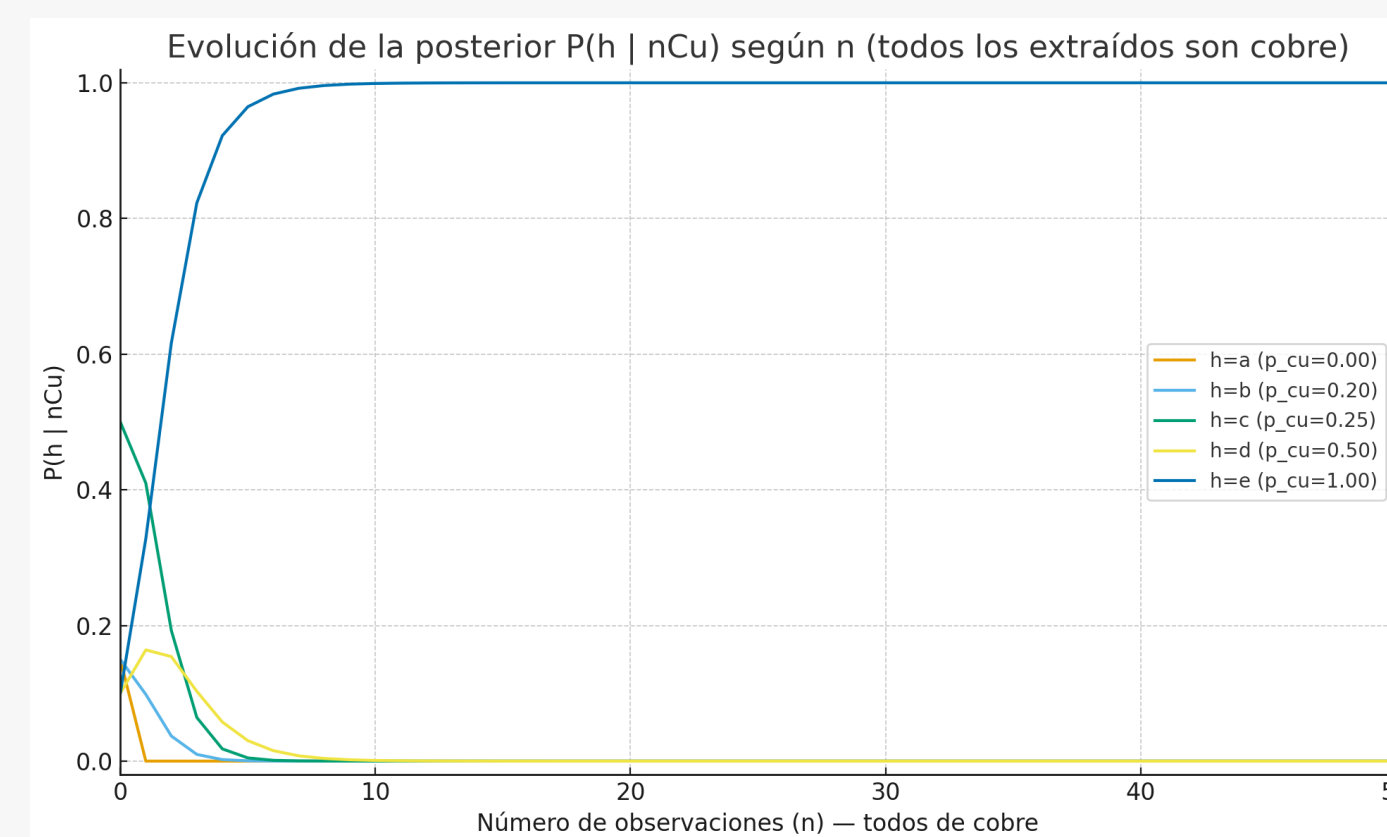
### Ejercicio 1: Aprendizaje Bayesiano

- Dados los datos, planteamos el estado hipotesis:  $P(h)=0.15; 0.15; 0.50; 0.10; 0.10$
- $h = a, b, c, d, e$

Luego nos piden el calculo de la probabilidad de las hipotesis  $h$  si extraemos 10 tornillos con reemplazo. Esto se puede generalizar para  $n$  tornillos graficando la verosimilitud Para graficar la evolución de la verosimilitud, vamos a suponer:

- Extracciones con reposición
- extraemos  $n$  tornillos recubiertos de cobre

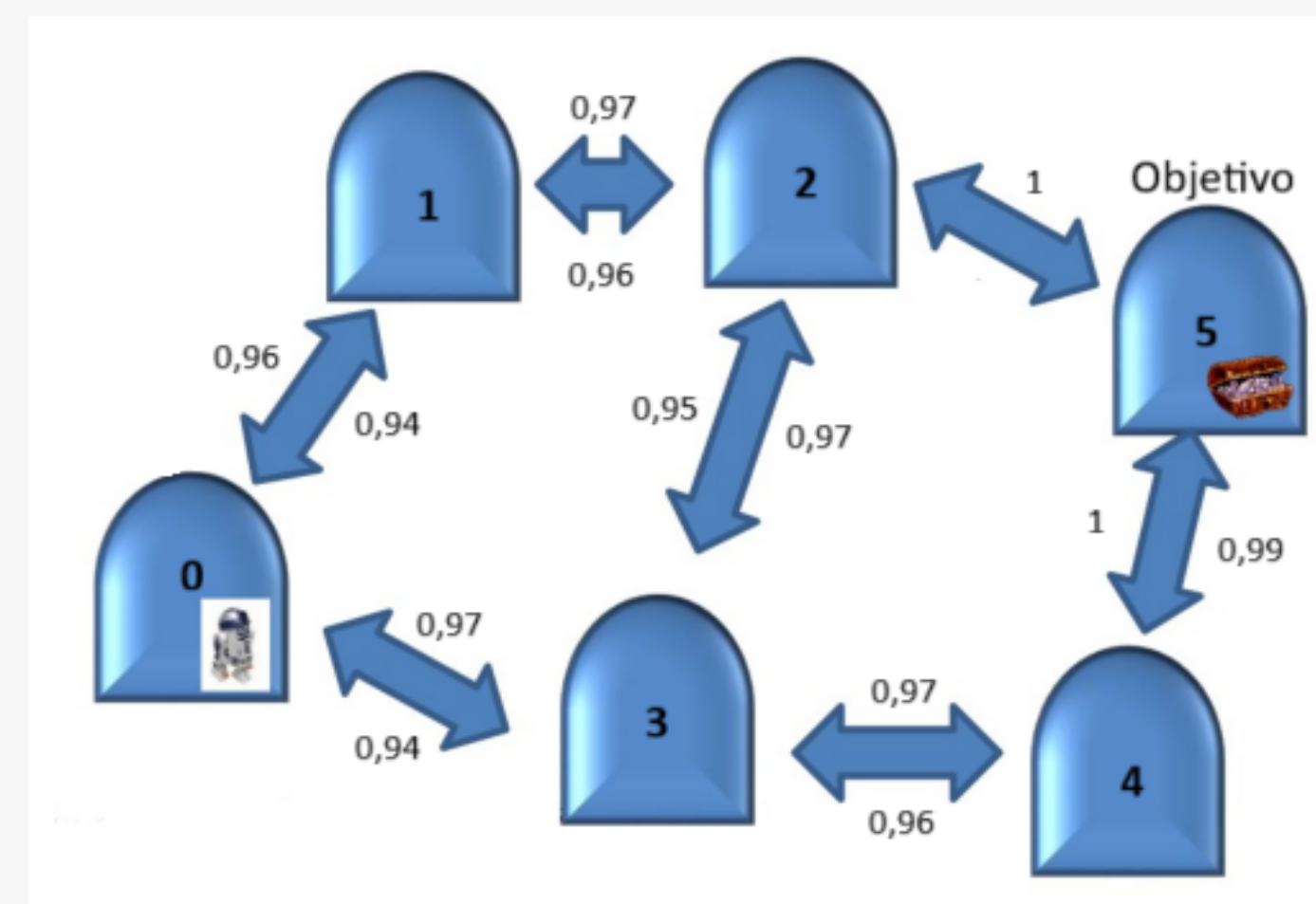
$$P(h_i | nCu) = \frac{P(h_i) * P(Cu|h_i)^n}{\sum_j P(h_j) * P(Cu|h_j)^n}$$



Concluimos que si de una misma caja extraemos más de 10 tornillos y todos son de cobre, tendremos una probabilidad de 99 entre 100 de que sea la caja de todos los tornillos recubiertos de cobre.

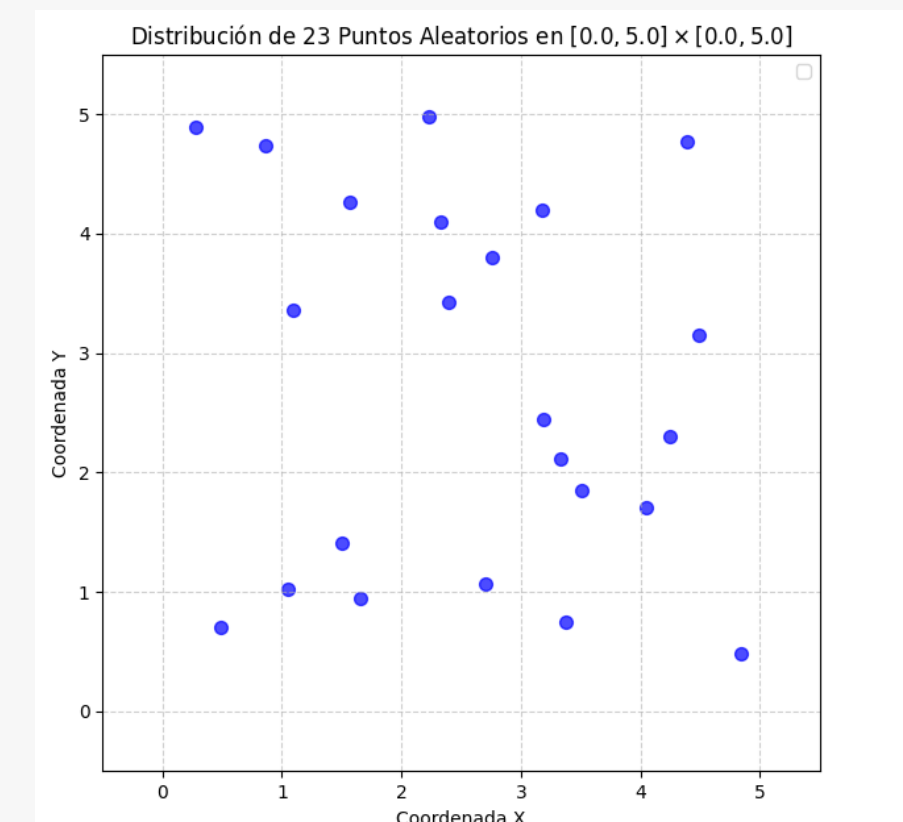
### Ejercicio 4: Pablo.png, Aprendizaje Q-Learning

Se implementó un algoritmo Q-learning para definir los caminos más óptimos posibles para que el **arturito** llegue a la caja de herramientas



### Ejercicio 3: Aprendizaje K-means y Knn

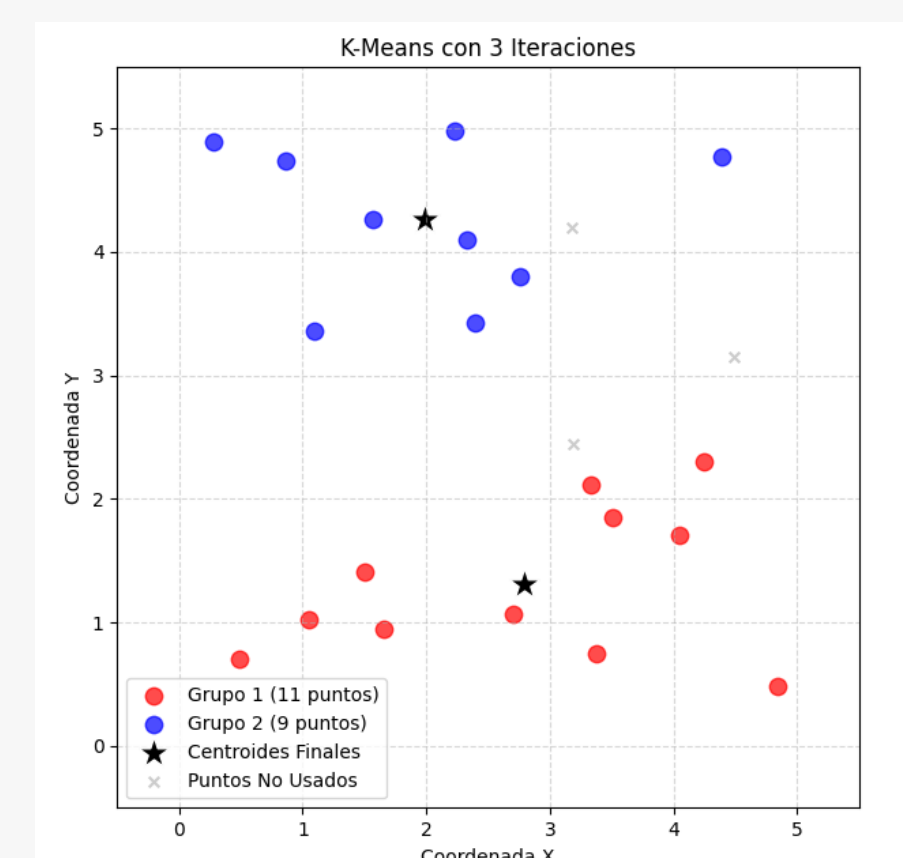
Lo primero que se hizo fue generar 23 puntos aleatorios con coordenadas (x,y) entre [0,5] y graficalas



Luego se implementó el algoritmo K-means; como semilla de los centroides se eligieron los primeros dos puntos generados aleatoriamente.

Se calculó la distancia Euclídeana de 20 puntos a estos centroides para hacer una clasificación entre puntos **rojos** y puntos **azules**. Con esa primera aproximación se recalcularon los centroides por lo que también se redefinió la clasificación de los puntos. Este proceso se repite hasta que los centroides tengan un error de variación pequeño o se quede sin iteraciones el programa.

```
Centroides Iniciales (Semillas):
[[4.24944016 2.38427274]
 [4.38589489 4.77532987]]
Iteración 1: Variación = 8.558584
Iteración 2: Variación = 0.185883
Iteración 3: Variación = 0.000080
Convergencia alcanzada en la iteración 3.
```



A los tres puntos restantes se los clasificó con el algoritmo Knn; el  $k$  se ingresa por consola y se exige que sea un numero impar para evitar conflictos