

FALLAS EN EL SERVIDOR

Nombres: Daniela Moreno, Sofia Torres y Samuel Avellaneda

1. ¿Qué medimos?

Tiempo hasta que el servidor falla

Medimos esto porque nos ayuda a saber cuánto dura funcionando antes de caerse. Es importante para saber si el sistema es estable o si falla muy seguido.

Número de fallas por día

Esto nos permite ver qué tan frecuente se cae el servidor. Si en un día falla muchas veces, claramente hay un problema.

Tiempo de reparación

No solo importa cuándo falla, sino cuánto tarda en volver a funcionar. Si tarda mucho en repararse, afecta a los usuarios y puede generar pérdidas.

2. Tipo de distribución: Poisson

Elegimos **Poisson** porque esta distribución se usa cuando queremos contar cuántas veces ocurre algo en un tiempo determinado. Es decir, estamos contando cuántas fallas ocurren en un día o en una hora.

Tiene sentido porque:

- Las fallas no se pueden predecir exactamente.
- Pueden ocurrir en cualquier momento.
- Lo que nos interesa es cuántas pasan en cierto tiempo.

3. ¿Qué representa?

Este modelo representa:

- Qué tan confiable es el servidor.
Si casi no falla, es confiable. Si falla mucho, no lo es.
- Cada cuánto tiempo tiende a fallar.
Nos da una idea de si el servidor es estable o inestable.
- El riesgo de que falle en cierto intervalo.
Por ejemplo, cuál es la probabilidad de que falle 3 veces en un día.
- Si la tasa λ es alta → el servidor falla con frecuencia.
Si λ es pequeña → el servidor es más estable y confiable.

(λ = promedio de fallas en un intervalo de tiempo.)

4. ¿Qué pasaría si usamos otra distribución?

- Normal:

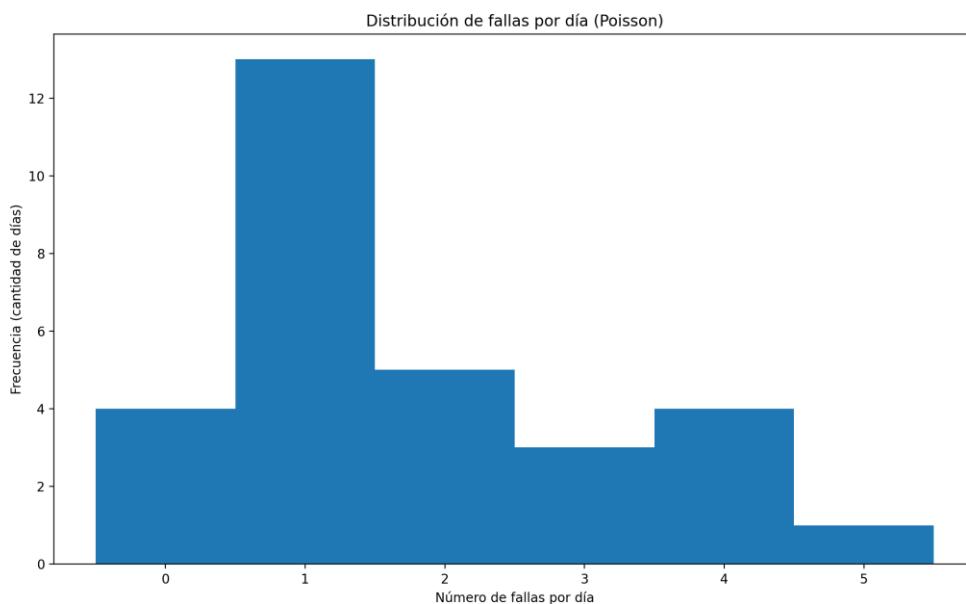
Puede dar valores negativos, y el tiempo no puede ser negativo.

No sirve mucho para modelar ya que se necesita esperar hasta que algo ocurra.

-Uniforme:

Significa que cualquier momento tiene exactamente la misma probabilidad de falla.

Eso no es muy realista, porque normalmente hay horas donde el servidor trabaja más y puede fallar más.



```
fallas.py > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  fallas = 2
5  dias = 30
6
7  fallas_por_dia = np.random.poisson(fallas, dias)      Use a "numpy.ran
8
9  print("Número de fallas por día durante 30 días:", fallas_por_dia)
10 print("Promedio real obtenido en la simulación:")
11 print(np.mean(fallas_por_dia))
12
13 plt.figure()
14 plt.hist(fallas_por_dia,
15           bins=range(0, max(fallas_por_dia)+2),
16           align='left')
17
18 plt.xlabel('Número de fallas por día')
19 plt.ylabel('Frecuencia (cantidad de días)')
20 plt.title('Distribución de fallas por día (Poisson)')
21 plt.show()
```