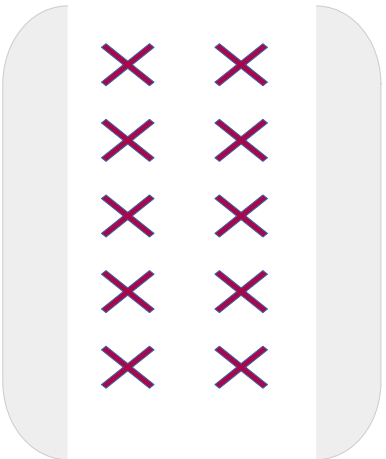
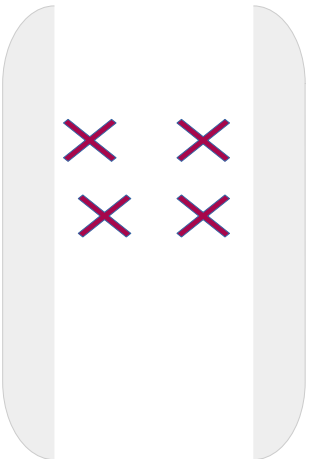
PROBLEMA DE REPARACIÓN

MAQUINAS EN FUNCIONAMIENTO



n

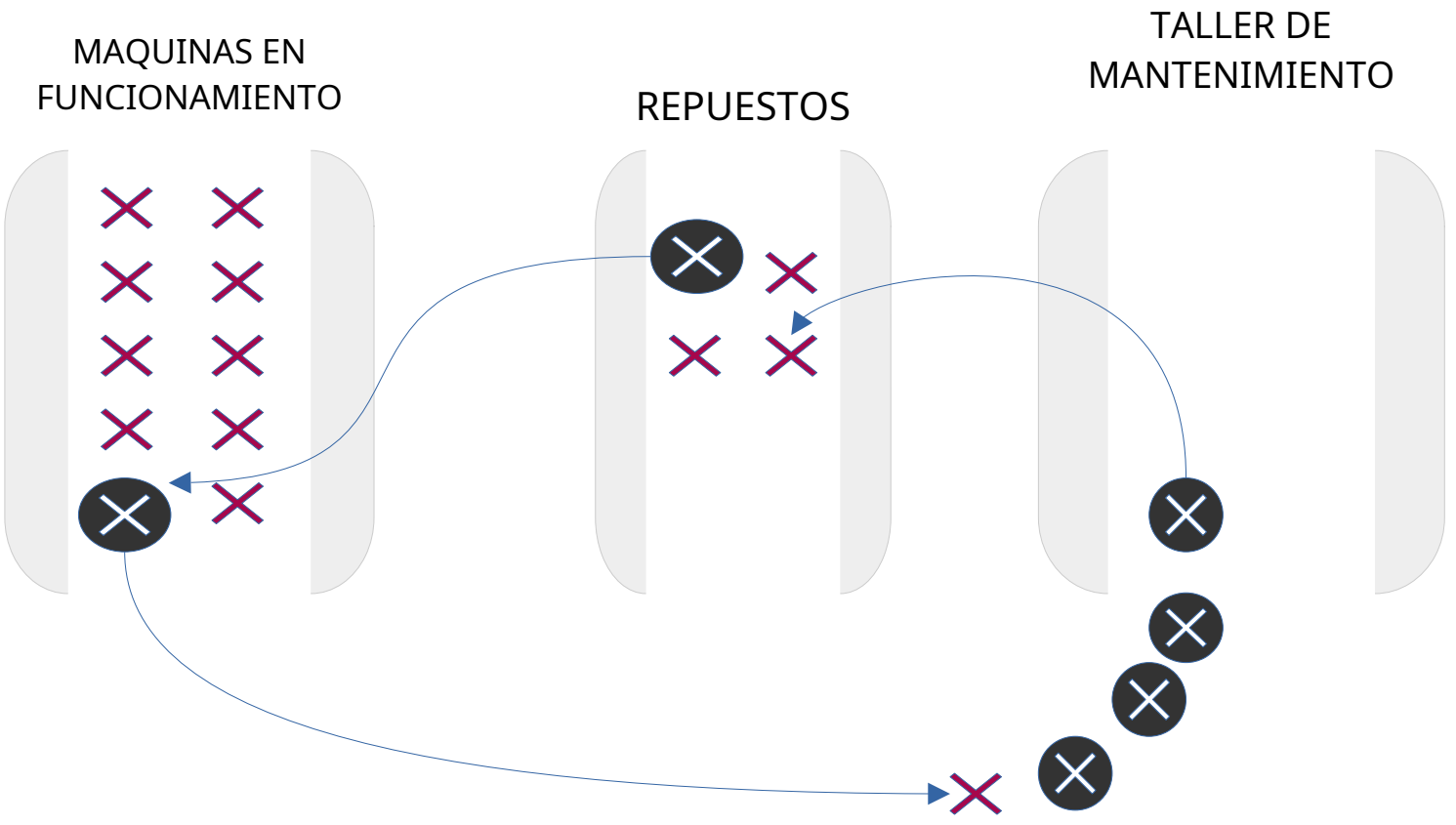
REPUESTOS



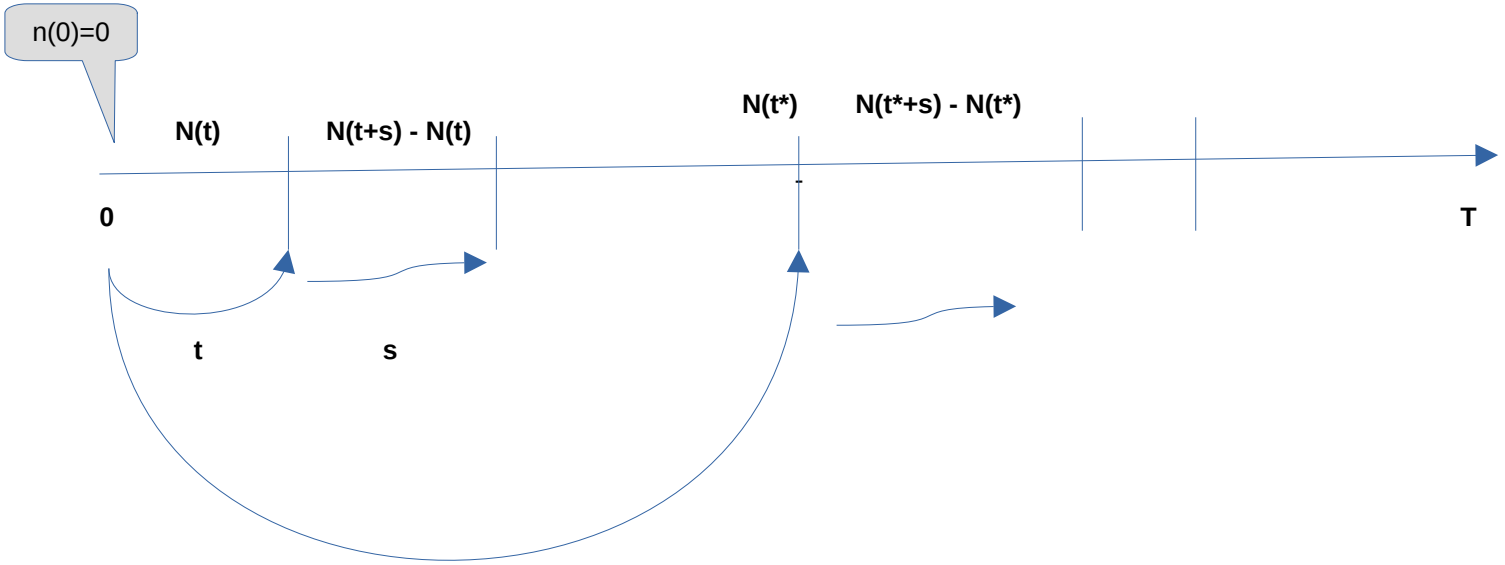
re

TALLER DE MANTENIMIENTO

des


MAQUINAS EN FUNCIONAMIENTO
REPUESTOS
TALLER DE MANTENIMIENTO

Proceso de poisson homogéneo



1,si ocurre un evento en el intervalo i

Lim h → 0 P(N(h) = 1) = λ h

Lim h → 0 P(N(h) >= 2) = 0

Xi =

0 si no ocurre

h = T/n

P(Xi = 1) = λT/n

N(t) = X1 +...Xn
Es una suma de bernoulli → Binomial(n,p) → Poisson(np = n λt/n = λt)

P(X1>t) = P(N(t) =0) = e^(-λt)

Inicialización

Sean $t = r = 0$, $t^* = \infty$.
Generar $X_1, \ldots, X_n$, variables aleatorias independientes, cada una con distribución F. Ordenar estos valores, de modo que t_i sea el i-ésimo menor, $i = 1, \ldots, n$.
Sea la lista de eventos: $t_1, \ldots, t_n, t^*$.

La actualización del sistema se realiza de acuerdo con los dos casos siguientes.

Caso 1 $t_1 < t^*$

Restablecer: $t = t_1$.

Restablecer: $r = r + 1$ (pues ha fallado otra máquina).

Si $r = s + 1$, detener esta ejecución y reunir los datos $T = t$ (pues como hay $s + 1$ máquinas descompuestas, no hay repuestos).

Si $r < s + 1$, generar una variable aleatoria X con distribución F. Esta variable aleatoria representará el tiempo de trabajo del repuesto que entrará en funciones.

Ahora reordenamos los valores $t_2, t_3, \ldots, t_n, t + X$ y sea t_i el i-ésimo menor de ellos, $i = 1, \ldots, n$.
Si $r = 1$, generar una variable aleatoria Y con función de distribución G y restablecer $t^* = t + Y$. (Esto es necesario debido a que en este caso, la máquina que acaba de fallar es la única descompuesta y por lo tanto de inmediato se comienza a reparar; Y será su tiempo de reparación y por lo tanto su reparación se concluirá en el instante $t + Y$.)

Caso 2 $t^* \leq t_1$

Restablecer: $t = t^*$.
Restablecer: $r = r - 1$.
Si $r > 0$, generar una variable aleatoria Y con función de distribución G, la cual representa el tiempo de reparación de la máquina que acaba de ingresar a servicio, y restablecer $t^* = t + Y$.
Si $r = 0$, hacer $t^* = \infty$.

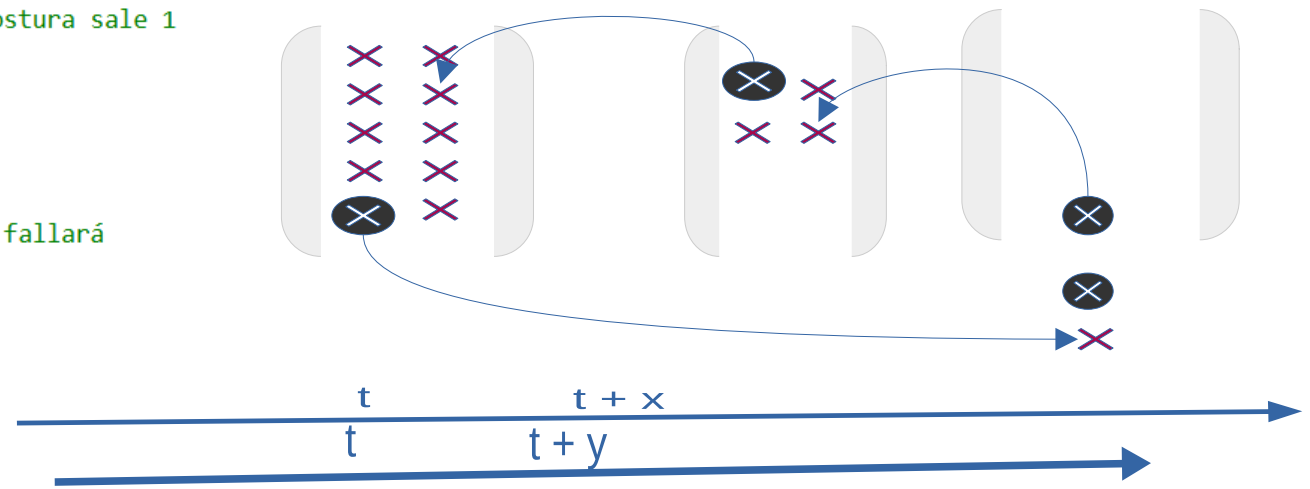
Inicialización

```
t = 0 ; des = 0; t_re =  float('inf'); re = 2
n = 10
X = []
for i in range(n):
    U = generar_u_aleatorio()
    x = -float(np.log(U))/lam
    X.append(x)
X = np.array(X)
t_ord = np.sort(X)
X = X.tolist()
t_ord = t_ord.tolist()
print(f"t_fallos{t_ord}")
```

0.07,	0.19	0.39	0.6,	0.62	0.64	0.71	1.29	1.47	2.25

Caso 1 t_falla < t_reparación

```
while True:
    t1 = t_ord[0]
    if t1 < t_re : # caso 1
        t = t1 ; des = des + 1
        t_ord.pop(0)
        if des == re + 1: # porque cuando hay descompostura sale 1
            T.append(t)
            break
        if des < re + 1:
            U = generar_u_aleatorio()
            x = -float(np.log(U))/lam
            t_ord.append(t + x) # porque en el futuro fallará
            t_ord = np.array(t_ord)
            t_ord = np.sort(t_ord)
            t_ord = t_ord.tolist()
        if des == 1 :
            U = generar_u_aleatorio()
            Y = -float(np.log(U))/lam
            t_re = t + Y
```



Caso 2 : t_reparación <= t_falla

```
elif t_re <= t1: #caso 2
    t = t_re ; des = des - 1
    if des > 0 :
        U = generar_u_aleatorio()
        Y = -float(np.log(U))/lam
        t_re = t + Y
    if des == 0:
        t_re = float('inf')
```

calculamos la media para todos los “sub-intervalos” desde 0→j con j:0→i para cada i-esima simulacion , donde i:0→N-1 simulaciones

```
falla = []
for i in range(len(T)):
    suma = 0
    for j in range(i+1):
        suma = suma + T[j]
    falla.append(suma/(i+1))
Tsimulaciones.append(falla[-1])
I = np.arange(len(T))
I = np.array(I)
I = I + 1
```

El siguiente bloque de codigo permite visualizar la cantidad de interés mientras el modelo crece en el tiempo

```
        plt.figure()
        plt.bar(I,falla,width= 0.5,color='green')
        plt.xlabel("n-ésima simulacion")
        plt.ylabel("media T-falla")
        plt.title(f"{i+1} simulaciones")
        plt.pause(1.5)
        plt.close()
print(f"i-esimo tiempo de falla promedio del sistema hasta la i-esima simulacion:")
print(",".join(f"{e:.4f}" for e in Tsimulaciones))
plt.bar(np.arange(len(Tsimulaciones)),Tsimulaciones,width=0.5,color="green")
plt.xlabel("n-ésima simulacion")
plt.ylabel("media T-falla")
plt.title(f"{i+1} simulaciones")
plt.show()
```

Las ordenadas son los tiempos promedio de falla hasta la n-esima simulacion