

# **Análisis de mejora del sistema de una lavandería**

*Mario E. Ferreyra*

**17 de Mayo del 2016**

# Índice

1. Introducción .....	1
2. Algoritmo y descripción de las variables .....	2
2.1. Sistema actual .....	2
2.1.1 Pseudocódigo .....	3
2.1.2. Código Python .....	4
2.2. Sistema con Opción 1: Incrementar los técnicos .....	5
2.2.1 Pseudocódigo .....	5
2.2.2. Código Python .....	7
2.3. Sistema con Opción 2: Incrementar las lavadoras de repuesto .....	8
3. Resultados .....	9
3.1. Resultado de las simulaciones .....	9
3.1.1. Como se obtuvieron los de resultado de simulaciones .....	10
3.2. Histogramas .....	11
3.2.1 Sistema actual .....	11
3.2.2 Sistema con Opción 1: Incrementar los técnicos .....	12
3.2.3 Sistema con Opción 2: Incrementar las lavadoras de repuesto ....	13
3.2.4 Comparación del Sistema con la Opción 1 y la Opción 2 .....	14
4. Conclusiones .....	15

# 1. Introducción

El problema que se plantea, es la mejora de un sistema perteneciente a una lavandería. En dicha lavandería, hay 5 lavadoras en servicio y 2 lavadoras de repuesto para el caso en el que alguna de las 5 lavadoras funcionando se descomponga, además la lavandería cuenta con un técnico el cual se encarga de reparar las lavadoras descompuestas, dicho técnico solamente es capaz de reparar una maquina a la vez.

El sistema funciona de la siguiente manera, las 5 lavadoras en servicio, comienzan a funcionar, si alguna de estas lavadoras deja de funcionar, se manda a una cola de reparación para que el técnico se encargue de volver a hacerla funcionar, una vez que la lavadora entra en la cola de reparación, algunas de las lavadoras de repuesto, entra en funcionamiento, obviamente al entrar en servicio dicha lavadora, la cantidad de repuestos disminuye. Una vez que el técnico termina de reparar una lavadora esta pasa a ser una lavadora de repuesto.

El sistema de la lavandería deja de ser operativo, es decir, que falla cuando hay menos de 5 lavadoras en servicio, o lo que es lo mismo, hay mas de 2 lavadoras para ser reparadas por el técnico.

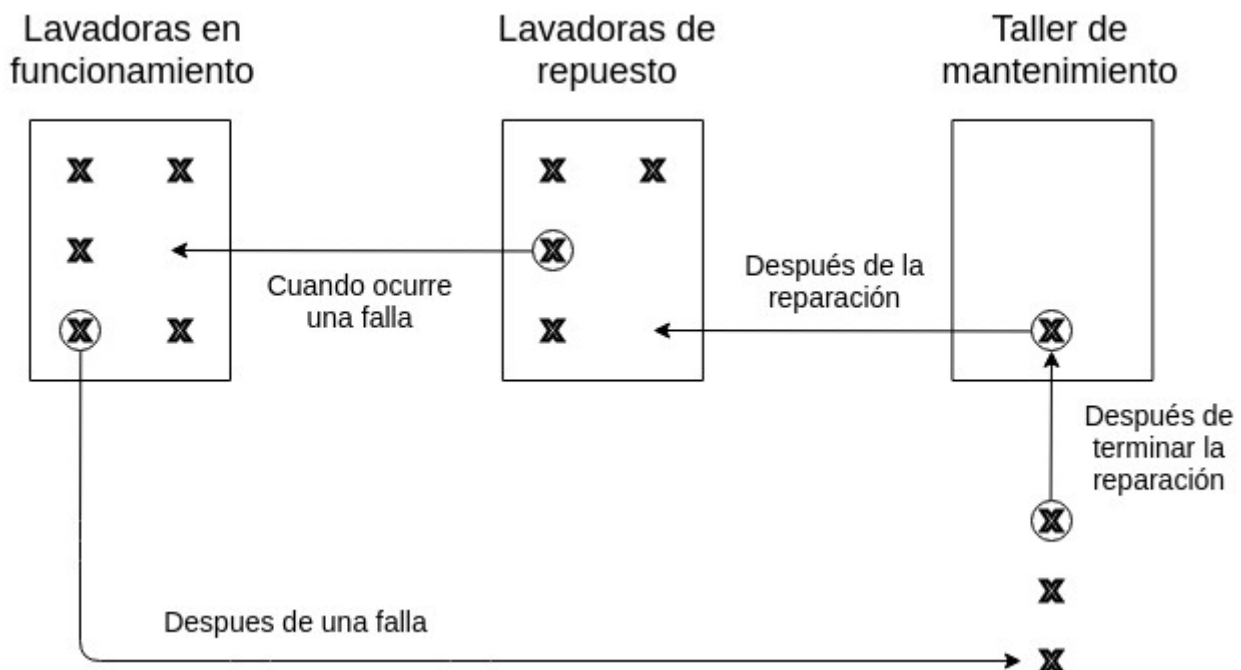
El dueño de la lavandería necesita saber tiempo en el cual el sistema de su lavandería dejara de ser operativo y también opciones para mejorar su sistema. Se exploraron opciones de mejora para el sistema, entre las cuales se destacan las siguientes que mencionaremos:

## Opción 1: Incrementar los técnicos

Se evalúa la posibilidad de contratar otro técnico, es decir, se tendrán 2 técnicos para reparar las lavadoras, estos trabajarían de forma paralela.

## Opción 2: Incrementar las lavadoras de repuesto

Se evalúa la posibilidad de compra de otra lavadora, es decir, se tendrán 3 lavadoras de repuesto, para cuando falle alguna de las que están funcionando.



## 2. Algoritmo y descripción de las variables

El algoritmo que planteamos usa como datos la cantidad de lavadoras en servicio, la cantidad de lavadoras de repuesto, el tiempo falla ( $T_f$ ) y el tiempo reparación ( $T_r$ ) de una lavadora.

Se tiene, como dato, que los tiempos de falla y reparación son variables aleatorias exponenciales, con tiempo medio de falla de 1 mes y tiempo medio de reparación de 1/8 mes. Vamos a expresar todos los tiempos de falla y reparación usando como unidad el mes.

Dicho de otra forma:

$$E[T_f] = 1$$

$$E[T_r] = 1/8$$

Por lo tanto, el tiempo de falla y reparación son variables aleatorias exponenciales de parámetro 1 y 8 respectivamente.

$$T_f \sim \text{Exponencial}(1)$$

$$T_r \sim \text{Exponencial}(8)$$

Primero presentaremos un pseudocódigo del algoritmo, y luego una implementación hecha en lenguaje de programación Python.

Pasaremos a describir las variables que serán usadas en el algoritmo:

- $N$  : Lavadoras en servicio.
- $S$  : Lavadoras de repuesto.
- $T$  : Tiempo en que falla el sistema.
- $t$  : Variable de tiempo.
- $r$  : Lavadoras descompuestas en el instante  $t$ .
- $t^*$  : Tiempo en que la lavadora en reparación vuelve a funcionar.
- $\text{lavadoras}[]$  : Lista ordenada de forma decreciente que contiene los de tiempos de falla de las lavadoras.

Vale aclarar que la variable “ $\text{lavadoras}[]$ ” es usada en el algoritmo implementado Python.

Como la cantidad de lavadoras descompuestas cambiará a medida que una lavadora en servicio falle o que una lavadora que este en reparación sea reparada, tendremos en cuenta estos dos casos hasta que veamos que el sistema a fallado.

Como sabemos la lavandería tiene en principio 5 lavadoras en servicio y 2 lavadoras de repuesto. Tendremos en cuenta esto para inicialización de nuestro algoritmo.

### 2.1. Sistema actual

Mostraremos el pseudocódigo y el algoritmo en Python del sistema actual que se esta usando en la lavandería.

## 2.1.1. Pseudocódigo

# Inicialización

$N \leftarrow 5$

$S \leftarrow 2$

$T \leftarrow 0$

$t \leftarrow 0$

$r \leftarrow 0$

$t^* \leftarrow \text{infinito}$  // Tiempo en el que la Lavadora en reparación vuelve a funcionar

Generar  $Tf_1, Tf_2, Tf_3, Tf_4, Tf_5$  v.a. i.i.d. tal que  $Tf_i \sim \text{Exponencial}(1)$  con  $i = 1, 2, 3, 4, 5$

Ordenar de forma decreciente los  $Tf_i$  # Notar que  $Tf_i$  es el tiempo de falla de las lavadora  $i$

**while** True **do**

    # Caso 1: Lavadora falla antes de que se repare alguna

**if**  $Tf_1 < t^*$  **then**

$t \leftarrow Tf_1$

$r \leftarrow r + 1$  # Fallo una lavadora

        # No hay lavadoras de repuesto

**if**  $r = S + 1$  **then**

$T \leftarrow t$

            STOP WHILE

**end**

        # Se agrega la lavadora de repuesto, ya que fallo alguna

**if**  $r < S + 1$  **then**

            Generar  $Tf \sim \text{Exponencial}(1)$  # Tiempo hasta fallar de la lavadora de repuesto

            Ordenar de forma decreciente  $Tf_2, Tf_3, Tf_4, Tf_5, Tf$

**end**

        // Se comienza a reparar lavadora rota.

**if**  $r = 1$  **then**

            Generar  $Tr \sim \text{Exponencial}(8)$  # Tiempo de reparación de la lavadora rota

$t^* \leftarrow t + Tr$  # Tiempo en que concluirá la reparación de la lavadora rota

**end**

**end**

    # Caso 2: Lavadora que estaba en reparación, esta disponible

**if**  $Tf_1 \geq t^*$  **then**

$t \leftarrow t^*$

$r \leftarrow r - 1$

        # Hay una o mas lavadoras para Reparar

**if**  $r > 0$  **then**

            Generar  $Tr \sim \text{Exponencial}(8)$  # Tiempo de reparación de la lavadora rota

$t^* \leftarrow t + Tr$  # Tiempo en que concluirá la reparación de la Lavadora rota

**end**

        # No hay lavadoras que Reparar

**if**  $r = 0$  **then**

$t^* \leftarrow \text{infinito}$  # No hay lavadoras que reparar

**end**

**end**

**end**

**return**  $T$

## 2.1.2. Código Python

```
# -*- coding: utf-8 -*-
```

```
import random
import math
```

```
INFINITO = float("inf") # Constante Infinito
```

```
def exponencial(lamda):
    """
```

```
    Genera una v.a. X con distribución Exponencial de parámetro lamda.
     $X \sim \text{Exp}(\text{lamda})$ .
    """
```

```
    u = random.random()
    x = -(1/float(lamda))*math.log(u)
```

```
    return x
```

```
def lavadero(N, S):
    """
```

```
    N = Lavadoras en servicio
    S = Lavadoras de repuesto
    """
```

```
    T = 0 # Tiempo en que falla el sistema
    t = 0 # Variable de tiempo
    r = 0 # Numero de Lavadoras rotas en el instante t
```

```
    t_estrella = INFINITO # Tiempo en el que la Lavadora en reparación vuelve a funcionar
```

```
    lavadoras = [] # Lista de tiempos de falla de las Lavadoras
```

```
    # Generamos N tiempos de falla (uno para cada maquina)
```

```
    for _ in xrange(N):
        F = exponencial(1) # Tiempo hasta Fallar
        lavadoras.append(F)
```

```
    lavadoras.sort() # Ordenamos los tiempos de falla de las Lavadoras
```

```
    while True:
```

```
        # Lavadora falla antes de que se repare alguna
```

```
        if lavadoras[0] < t_estrella:
```

```
            t = lavadoras[0]
```

```
            r += 1 # Se rompió una Lavadora
```

```
            # Si hay mas de S Lavadoras descompuestas (no hay repuestos)
```

```
            if r == S+1:
```

```
                T = t
```

```
                break
```

```
            # Hay Lavadoras de repuesto para reponer. Se agrega la Lavadora de repuesto, ya que fallo alguna
```

```
            if r < S+1:
```

```
                X = exponencial(1) # Tiempo hasta fallar de la lavadora de repuesto
```

```
                lavadoras.pop(0) # Quitamos la Lavadora que fallo
```

```
                lavadoras.append(t+X) # Agregamos la nueva Lavadora al sistema (tiempo actual + tiempo de falla)
```

```
                lavadoras.sort() # Ordenamos los tiempos en que fallan las Lavadoras
```

```
            # La Lavadora rota es la única descompuesta, entonces se comienza a reparar
```

```
            if r == 1:
```

```
                Y = exponencial(8) # Tiempo de reparación de la Lavadora rota
```

```
                t_estrella = t + Y # Tiempo en que concluirá la reparación de la Lavadora rota
```

```

# Lavadora que estaba en reparación, esta disponible
elif lavadoras[0] >= t_estrella:
    t = t_estrella
    r -= 1 # Se reparo la Lavadora que estaba rota
    # Hay una o mas Lavadoras para Reparar
    if r > 0:
        Y = exponencial(8) # Tiempo de reparación de la Lavadora rota
        t_estrella = t + Y # Tiempo en que concluirá la reparación de la Lavadora rota
    # No hay Lavadoras que Reparar
    if r == 0:
        t_estrella = INFINITO # Técnico no tiene nada que reparar

return T

```

```

print lavadero(5, 2) # N = 5 y S = 2

```

## 2.2. Sistema con Opción 1: Incrementar los técnicos

Para esta opción de mejora, el algoritmo cambiará, ahora la variable “t\*” pasara a ser un lista de dos valores, con lo que el algoritmo quedará de la siguiente forma:

### 2.2.1 Pseudocódigo

```

# Inicialización
N ← 5
S ← 2
T ← 0
t ← 0
r ← 0
t* ← [infinito, infinito] # Tiempo en el que la Lavadora en reparación vuelve a funcionar
Generar Tf1, Tf2, Tf3, Tf4, Tf5 v.a. i.i.d. tal que Tfi ~ Exponencial(1) con i = 1, 2, 3, 4, 5
Ordenar de forma decreciente los Tfi # Notar que Tfi es el tiempo de falla de las lavadora i

while True do
    # Caso 1: Lavadora falla antes de que se repare alguna
    if Tf1 < t*[0] then
        t ← Tf1
        r ← r + 1 # Fallo una lavadora

        # No hay lavadoras de repuesto
        if r = S + 1 then
            T ← t
            STOP WHILE
        end

        # Se agrega la lavadora de repuesto, ya que fallo alguna
        if r < S + 1 then
            Generar Tf ~ Exponencial(1) # Tiempo hasta fallar de la lavadora de repuesto
            Ordenar de forma decreciente Tf2, Tf3, Tf4, Tf5, Tf
        end
    end
end

```

```

# Se comienza a reparar la primera lavadora rota.
if r = 1 then
    Generar Tr ~ Exponencial(8) # Tiempo de reparación de la lavadora rota
    t*[0] ← t + Tr # Tiempo en que concluirá la reparación de la lavadora rota
end

# Se comienza a reparar la segunda lavadora rota.
if r = 2 then
    Generar Tr ~ Exponencial(8) # Tiempo de reparación de la lavadora rota
    t*[1] ← t + Tr # Tiempo en que concluirá la reparación de la lavadora rota
end

Ordenar de forma decreciente la lista t*
end

# Caso 2: Lavadora que estaba en reparación, esta disponible
if Tf1 ≥ t*[0] then
    t ← t*[0]
    r ← r - 1

    # Hay 2 o más lavadoras para Reparar
    if r > 1 then
        Generar Tr ~ Exponencial(8) # Tiempo de reparación de la lavadora rota
        t*[0] ← t + Tr # Tiempo en que concluirá la reparación de la Lavadora rota
    end

    # Hay 1 o 0 lavadoras para reparar
    if r ≤ 1 then
        t*[0] ← infinito # No hay lavadoras que reparar
    end

    Ordenar de forma decreciente la lista t*
end
end

return T

```



## 2.2.2. Código Python

```
# -*- coding: utf-8 -*-
```

```
import random
import math
```

```
INFINITO = float("inf") # Constante Infinito
```

```
def exponencial(lamda):
```

```
    """
```

```
    Genera una v.a. X con distribución Exponencial de parámetro lamda.
```

```
     $X \sim \text{Exp}(\text{lamda})$ .
```

```
    """
```

```
    u = random.random()
```

```
    x = -(1/float(lamda))*math.log(u)
```

```
    return x
```

```
def lavadero(N, S):
```

```
    """
```

```
    N = Lavadoras en servicio
```

```
    S = Lavadoras de repuesto
```

```
    """
```

```
    T = 0 # Tiempo en que falla el sistema
```

```
    t = 0 # Variable de tiempo
```

```
    r = 0 # Numero de Lavadoras rotas en el instante t
```

```
    t_estrella = [INFINITO, INFINITO] # Tiempo en el que las Lavadoras en reparación vuelve a funcionar
```

```
    lavadoras = [] # Lista de tiempos de falla de las Lavadoras
```

```
    # Generamos N tiempos de falla (uno para cada maquina)
```

```
    for _ in xrange(N):
```

```
        F = exponencial(1) # Tiempo hasta Fallar
```

```
        lavadoras.append(F)
```

```
    lavadoras.sort() # Ordenamos los tiempos
```

```
    while True:
```

```
        # Lavadora falla antes de que se repare alguna
```

```
        if lavadoras[0] < t_estrella[0]:
```

```
            t = lavadoras[0]
```

```
            r += 1 # Se rompio una Lavadora
```

```
            # Si hay mas de S Lavadoras descompuestas (no hay repuestos)
```

```
            if r == S+1:
```

```
                T = t
```

```
                break
```

```
            # Hay Lavadoras de repuesto para reponer. Se agrega la Lavadora de repuesto, ya que fallo alguna
```

```
            if r < S+1:
```

```
                X = exponencial(1) # Tiempo hasta fallar de la lavadora de repuesto
```

```
                lavadoras.pop(0) # Quitamos la Lavadora que fallo
```

```
                lavadoras.append(t+X) # Agregamos la nueva Lavadora al sistema (tiempo actual + tiempo de falla)
```

```
                lavadoras.sort() # Ordenamos los tiempos en que fallan las Lavadoras
```

```
            # Primera Lavadora rota, entonces se comienza a reparar
```

```
            if r == 1:
```

```
                Y = exponencial(8) # Tiempo de reparación de la Primera Lavadora rota
```

```
                t_estrella[0] = t + Y # Tiempo en que concluirá la reparación de la Primera Lavadora rota
```

```

# Segunda Lavadora rota, entonces se comienza a reparar
if r == 2:
    Y = exponencial(8) # Tiempo de reparación de la Segunda Lavadora rota
    t_estrella[1] = t + Y # Tiempo en que concluirá la reparación de la Segunda Lavadora rota

t_estrella.sort() # Ordenamos los tiempos de reparación en orden decreciente (para ver quien tarda menos)

# Lavadora que estaba en reparación, esta disponible
elif lavadoras[0] >= t_estrella[0]:
    t = t_estrella[0]
    r -= 1 # Se reparo una maquina
    # Hay 2 o + Lavadoras para Reparar
    if r > 1:
        Y = exponencial(8) # Tiempo de reparación de la Lavadora para reparar
        t_estrella[0] = t + Y # Tiempo en que concluirá la reparación de la Lavadora rota
        # Se la damos al Técnico que termino primero de reparar
    # Hay 1 o 0 Lavadoras para reparar
    elif r <= 1:
        t_estrella[0] = INFINITO # Técnico queda no tiene nada que reparar
        # El otro Técnico esta reparando la lavadora

t_estrella.sort() # Ordenamos los tiempos de reparación en orden decreciente (para ver quien tarda menos)

return T

print lavadero(5, 2) # N = 5 y S = 2

```

## **2.3. Sistema con Opción 2: Incrementar las lavadoras de repuesto**

Para esta opción de mejora, el algoritmo que sera usado es el del sistema actual con solo una única modificación, que es la de cambiar la inicialización de las variables N y S de la siguiente forma:

$$\begin{aligned}
 N &\leftarrow 5 \\
 S &\leftarrow 2
 \end{aligned}$$

# 3. Resultados

En esta sección se mostrar los resultados obtenidos al ejecutar los algoritmos antes descriptos un numero “n” de simulaciones, para poder estimar el tiempo medio de fallo del sistema ( $\mu$ ), varianza ( $\sigma^2$ ) y desviación estándar ( $\sigma$ ).

Usaremos los siguientes nombres para los algoritmos antes descriptos:

- Algoritmo 1 : Algoritmo con N = 5, S = 2 y 1 Técnico.
- Algoritmo 2 : Algoritmo con N = 5, S = 2 y 2 Técnicos (Opción 1).
- Algoritmo 3 : Algoritmo con N = 5, S = 3 y 1 Técnico (Opción 2).

## 3.1. Resultado de las simulaciones

Algoritmo 1	E[T] ( $\mu$ )	V[T] ( $\sigma^2$ )	$\sqrt{V[T]}$ ( $\sigma$ )
n = 100	1.70724125088	2.19020901067	1.47993547517
n = 1000	1.7404625149	2.48584430421	1.57665605134
n = 10000	1.75134160726	2.5779437092	1.60559761746
n = 100000	1.75451847743	2.56715632376	1.60223479046

Algoritmo 2	E[T] ( $\mu$ )	V[T] ( $\sigma^2$ )	$\sqrt{V[T]}$ ( $\sigma$ )
n = 100	2.08305787499	4.112280164	2.02787577627
n = 1000	2.57737918259	6.32604110246	2.51516224178
n = 10000	2.59454498851	6.15868402747	2.48166960482
n = 100000	2.58903738453	6.07975267469	2.46571544885

Algoritmo 3	E[T] ( $\mu$ )	V[T] ( $\sigma^2$ )	$\sqrt{V[T]}$ ( $\sigma$ )
n = 100	4.36843121127	16.3450689355	4.04290352785
n = 1000	3.57682252785	10.1370199349	3.18386870566
n = 10000	3.56025357143	10.6680875167	3.26620383882
n = 100000	3.60410154483	11.0641450939	3.32628097037

### **3.1.1. Como se obtuvieron los de resultado de simulaciones**

Los resultados  $E[T]$  se obtuvieron mediante la Ley de los Grandes Números, con la cual aproximamos  $E[T]$  con el promedio, es decir:

$$\text{promedio de } T \approx E[T]$$

Por lo cual se hacen “n” simulaciones de T y luego se saca el promedio, vale aclarar que mientras más grande “n” más aproxima el promedio de T a  $E[T]$ .

A continuación mostraremos un pseudocódigo de como obtener  $E[T]$ ,  $V[T]$ ,  $\sqrt{V[T]}$ :

```
input n
suma1 = 0
suma2 = 0

for 1 to n:
  Generar T
  suma1 += T
  suma2 += T^2

esperanza = suma1/n
varianza = suma2/n - esperanza^2 #  $V(T) = E(T^2) - E(T)^2$ 

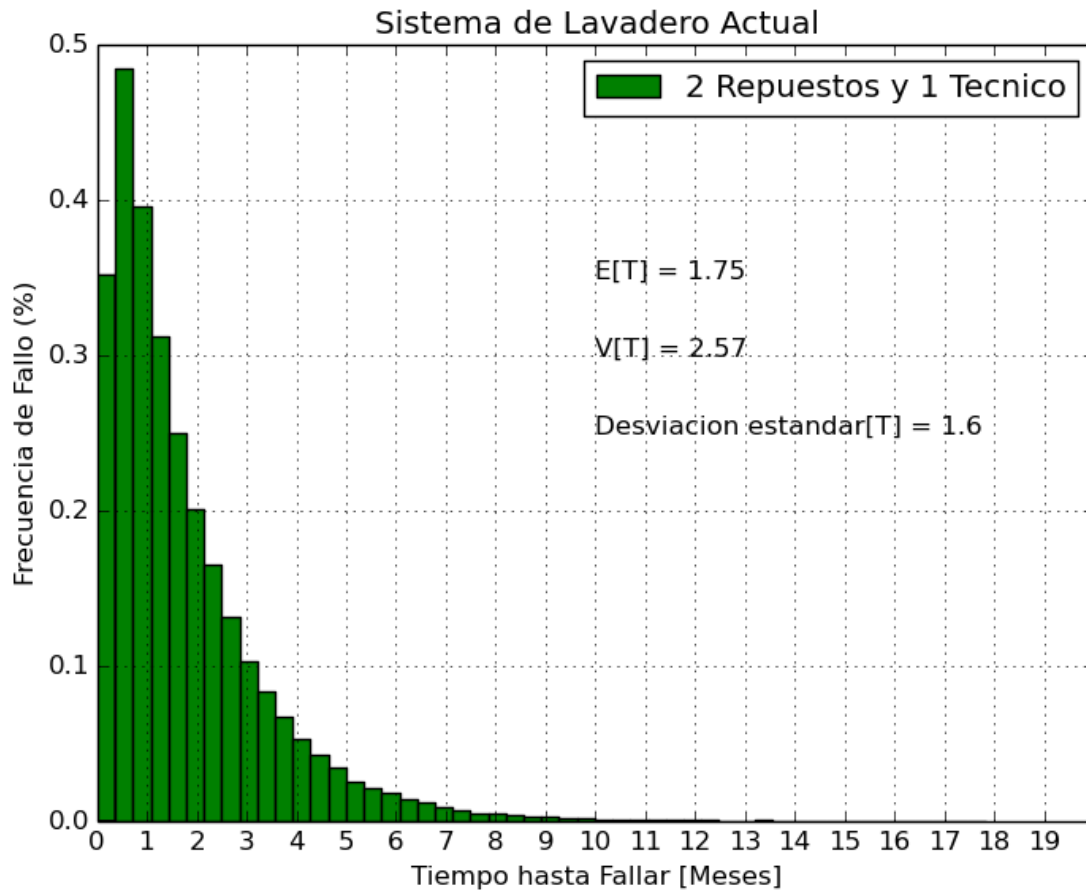
des_est = varianza^(1/2) # Desviación estándar =  $V(T)^{(1/2)}$ 

return esperanza, varianza, des_est
```

## 3.2. Histogramas

### 3.2.1 Sistema Actual

Histograma del sistema actual de la lavandería.

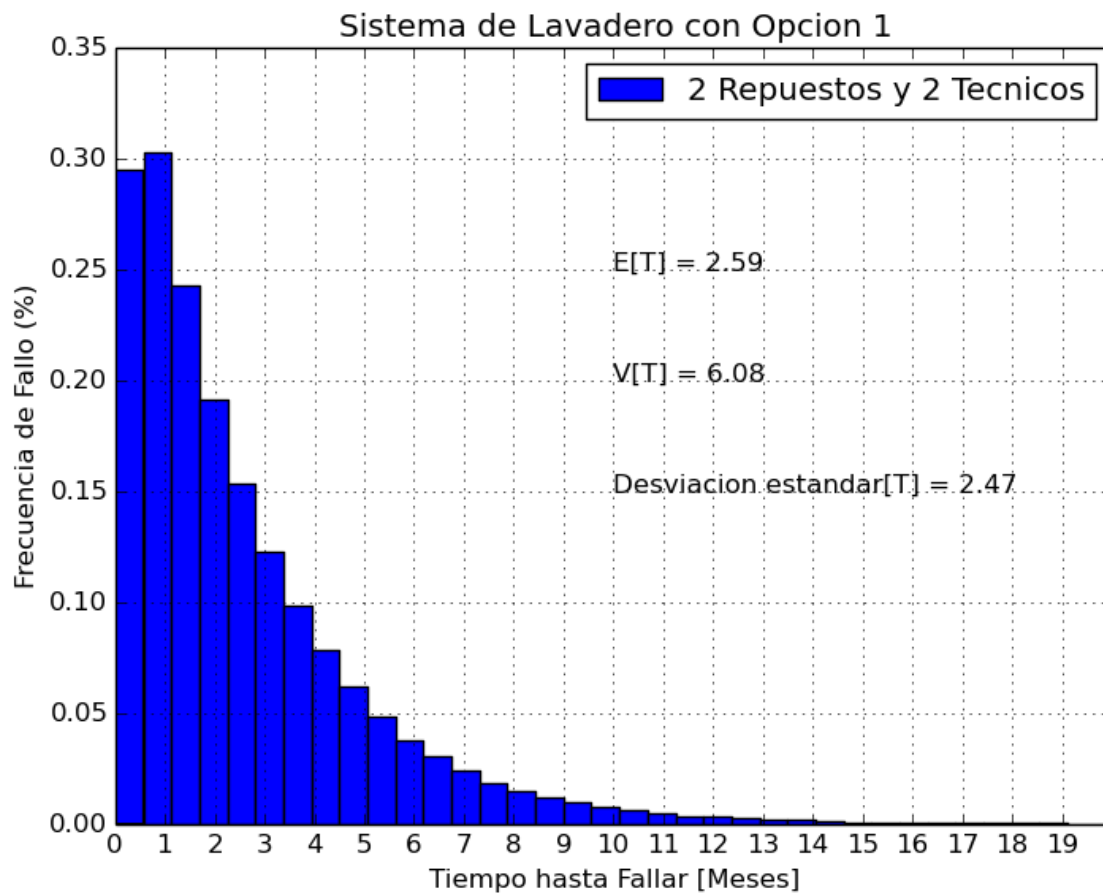


$E[T] \approx 1 \text{ mes}, 22 \text{ días}, 12 \text{ horas}$

$\sigma[T] \approx 1 \text{ mes}, 18 \text{ días}$

### 3.2.2 Sistema con Opción 1: Incrementar los técnicos

Histograma del sistema de la lavandería con la Opción 1.

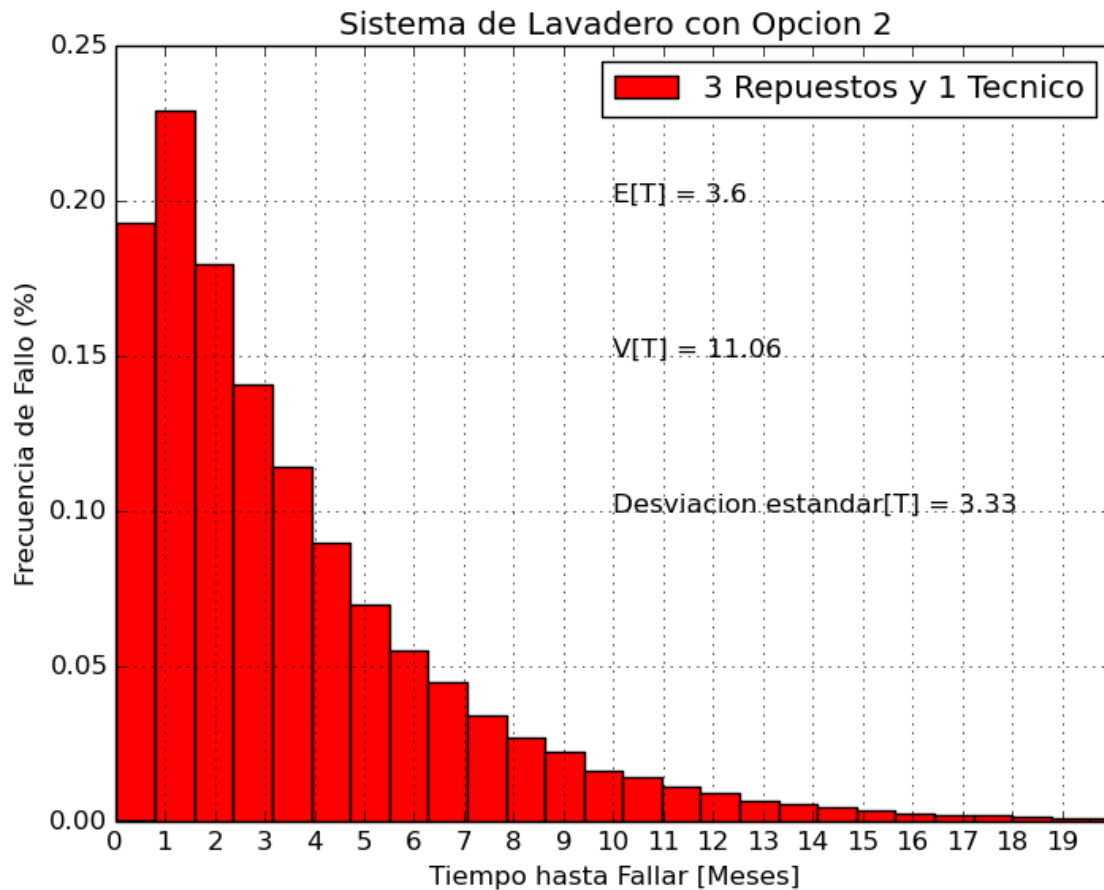


$E[T] \approx 2$  meses, 17 días, 17 horas

$\sigma[T] \approx 2$  meses, 14 días, 2 horas

### 3.2.3 Sistema con Opción 2: Incrementar las lavadoras de repuesto

Histograma del sistema de la lavandería con la Opción 2.

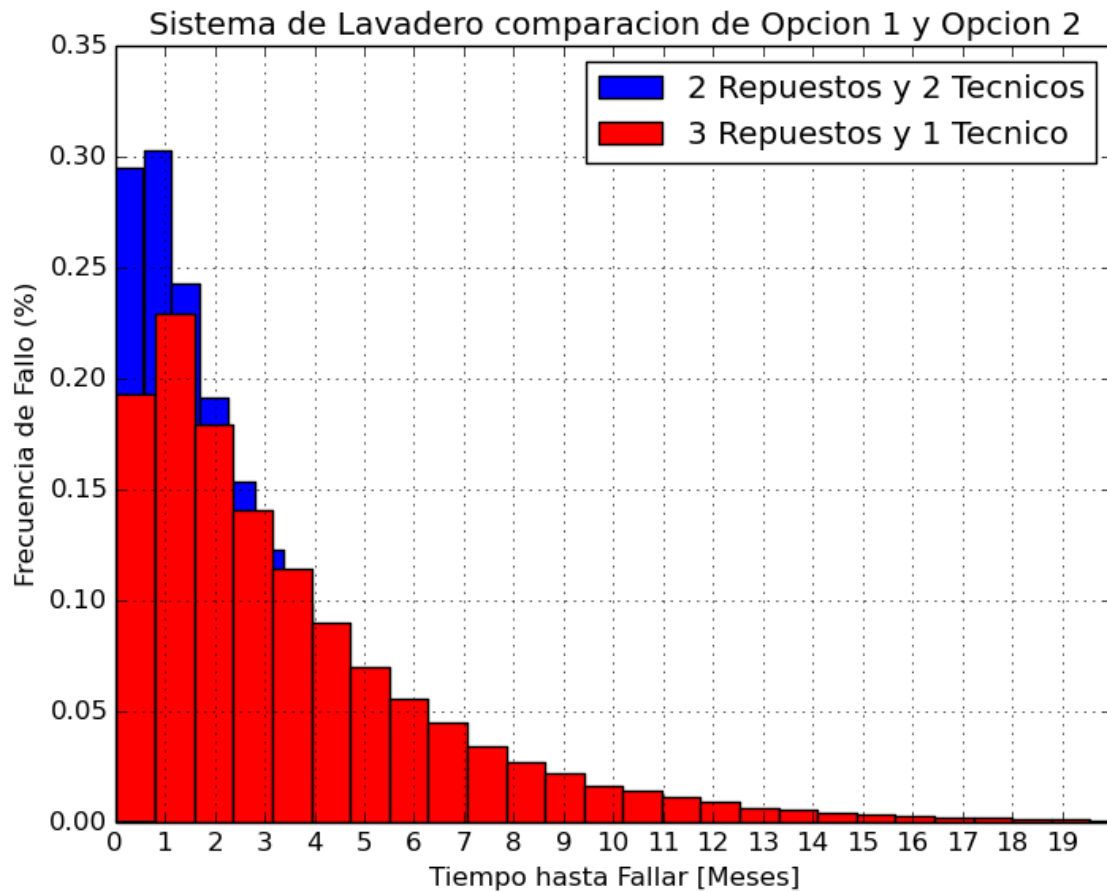


$E[T] \approx 3$  meses, 18 días

$\sigma[T] \approx 9$  meses, 9 días, 22 horas

### 3.2.4 Comparación del Sistema con la Opción 1 y la Opción 2

Comparación de los Histogramas del sistema de la lavandería con la Opción 1 y la Opción 2. Para poder obtener un mejor análisis, de cual opción es mejor aplicar.





## **4. Conclusiones**

Como podemos apreciar que con cualquiera de las dos opciones mejoramos el sistema actual de la lavandería.

Pero analizando el ultimo histograma mostrado, donde comparamos el sistema de la lavandería aplicando la Opción 1 y la Opción 2, vemos claramente que con la Opción 2 se reduce más la frecuencia en que ocurren los fallos y por lo tanto aumenta más el tiempo medio de fallo del sistema que con la Opción 1.

Esto ultimo lo podemos ver claramente analizando los tiempos de medios de falla con cada opción:

Opción 1: El tiempo medio de falla del sistema es de aproximadamente 2 meses, 17 días, 17 horas

Opción 2: El tiempo medio de falla del sistema es de aproximadamente 3 meses, 18 días.

Por lo tanto concluimos que al dueño de la lavandería le conviene aplicar la Opción 2.