

Proyecto 1

Proyecto de motores e hilos de poliéster



Mi nombre es Abel Alejandro Pacheco Quispe soy estudiante de ingeniería en Sistemas con experiencia en desarrollo de software, especializado en Java y Python. He participado en proyectos como una aplicación de inventarios, una red social de fotos y un software de rutas óptimas, demostrando habilidades en control de existencias, manejo de rutas de viaje y registro avanzado de usuarios. Con una formación sólida y conocimientos en redes, bases de datos SQL y NoSQL, y herramientas como GitHub, me destaco por mi capacidad de adaptación, resolución de problemas y compromiso con la calidad. Busco un entorno dinámico que me permita seguir desarrollando mis competencias y aportar con soluciones innovadoras.

Nombre: Abel Alejandro Pacheco Quispe

Materia: Taller de Simulación de Sistemas

Docente: Ayoroa Cardozo Jose Richard

Contenido

Definición del sistema:	3
Colección de datos:	3
Formulación del modelo:	8
• LEAP	8
• Diagrama de simulación:	9
• Diagrama de simulación ProModel:	9
Implementación del modelo en computadora:.....	10
ProModel:	10
Java:	10
Validación:.....	11
Forma Manual:.....	11
Pro Model:	11
Java:	12
Tabla de comparación:	12
Experimentación:	13
Pro Model:	13
Java:	14
Interpretación:	16
Pro Model:	16
Java:	17
Documentación:.....	19
Archivo de texto Pro Model:	19
Código de java:.....	21
Resultados:.....	28
Pro model:.....	28
Java:	29
Tabla de comparación:	31
Conclusión:.....	31
Videos del proyecto:	31

Definición del sistema:

Ejercicio: Una empresa fabricante de hilo poliéster tiene 50 máquinas estiro torcedoras en su departamento de estiraje. El índice de producción anual es de 90,000 toneladas, el cual no se está logrando debido a fallas en los motores de las estiro torcedoras.

Cada máquina tiene un motor que está sujeto a descomposturas al azar, cuando un motor se descompone se reemplaza por otro de repuesto inmediatamente o cuando haya uno disponible. El motor descompuesto se envía al taller de mantenimiento y una vez reparado se convierte en un motor de repuesto.

En mantenimiento hay dos áreas, una de ellas es de capacidad para 100 motores y permite que los motores que no puedan ser reparados de inmediato esperen en ese lugar hasta que se desocupe alguno de los mecánicos y la segunda es donde los mecánicos hacen la reparación. La distancia que hay entre estiraje y el área de espera de motores antes de ser reparados es de 55 metros, entre el área de espera y el área de reparación hay 5 metros, entre reparación y el área de reserva hay 40 metros y la distancia entre el área de reserva y estiraje es 20 metros.

Los datos de la tabla 1 son una muestra histórica de 50 reparaciones.

En la tabla 2 se tienen los datos históricos del tiempo que transcurre desde que un motor es instalado en la estiro torcedora hasta que sufre una descompostura.

Debido al volumen y peso de los motores, cualquier movimiento de un motor debe hacerse con un montacargas. La velocidad del montacargas es de 150 metros/min si va vacío y 100 metros/min si va cargado. El tiempo para subir y o bajar un motor del m montacargas es de 45 segundos. Actualmente la empresa tiene 3 motores de repuesto, 5 mecánicos y 2 montacargas.

Se estima que cada tonelada se vende a \$2,000. El costo anual de cada mecánico es de \$150,000, el de los montacargas es de \$450,000, y cada motor ocasiona gastos anuales por \$750,000.

El ejercicio en este caso tiene como objetivo realizar el calculo de del beneficio anual real de la empresa de hilos de poliéster teniendo en cuenta el problema que consiste en que los motores fallan cada cierto tiempo, donde los motores fallidos se llevan a una fila de espera para entrar a ser reparados y una vez reparados los motores se dirigen a un almacén donde esperan a ser utilizado en caso de que falle un motor.

El cálculo que realizará la simulación de considerará un lapso de tiempo de 6 meses con días de trabajo de lunes a viernes, con 8 horas de trabajo por día para evitar simulaciones largas y que sean más flexibles.

Colección de datos:

Los datos proporcionados por el problema son los siguientes:

Área de estiro cortadoras: tienen un total de 50 maquinas cada una con un motor y tienen una producción anual de 90000 toneladas de hilo de poliéster.

Convertimos 90000 toneladas/año a minutos/toneladas con:

$$\frac{525600 \text{ min}}{90000 \text{ toneladas}} = 5.84$$

Es decir que aproximadamente llegan 5.84 minutos/toneladas.

Área de espera: Tiene una capacidad para 100 motores

Los montacargas: Nos da los datos de velocidad que son 150 metros/minuto con carga y sin carga es de 100 metros/minuto también nos da el dato de tiempo de descarga y carga que es de 45 segundos.

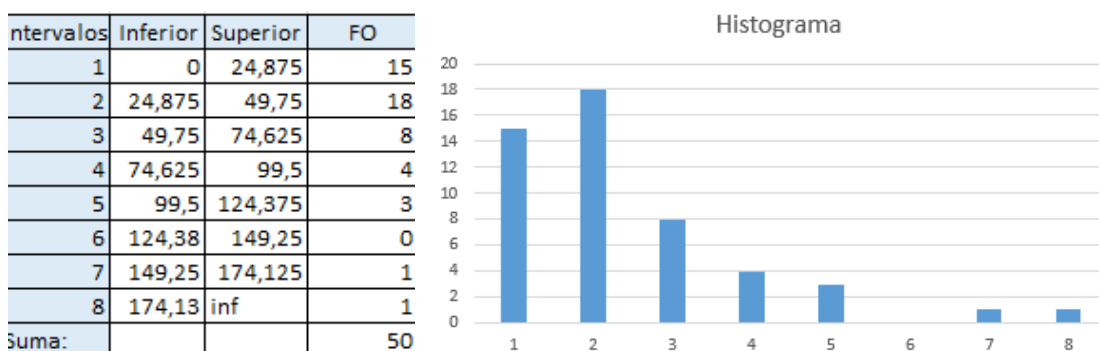
Rutas: Para las rutas tenemos que del área de estiro cortadoras al área de espera es de 55 metros, del área de espera al área de reparación es de 5 metros y del área de reparación al área de almacenamiento es de 60 metros y del área de almacenamiento al área de estiro cortadoras es de 20 metros respectivamente.

Datos de reparación y operación:

También nos da los tiempos de reparación en una tabla:

36.80	48.97	51.15	1.96	49.01
2.86	36.02	58.82	11.70	40.82
26.46	50.62	4.32	78.01	7.37
3.07	5.67	14.70	60.58	51.81
38.70	167.87	26.66	25.96	29.21
118.13	7.89	82.53	29.19	37.03
22.80	111.58	112.10	4.75	9.92
82.83	32.07	85.53	69.27	18.73
198.88	25.53	27.46	16.73	39.86
27.28	41.26	65.79	10.33	59.38

Para hallar la distribución que siguen los valores de la tabla primero ordenamos los números de menor a mayor y calculamos su media y el número de intervalos en el que se dividen con la fórmula de Raíz (número de datos) luego separamos los datos de la tabla en los intervalos y creamos el histograma para ver a que distribución se parece.



Viendo el histograma podemos decir que la distribución tiene forma de la distribución exponencial por tanto procedemos a realizar los cálculos de la frecuencia esperada para realizar la prueba de ajuste de bondad chi cuadrado.

FO	P(X)	P(X)% FE	Error
15	0,4224	21,1202	1,77353
18	0,24398	12,1989	2,75861
8	0,14092	7,04605	0,12915
4	0,0814	4,06977	0,0012
3	0,04701	2,35068	0,17936
0	0,02715	1,35774	1,35774
1	0,01568	0,78422	0,05937
1	0,02145	1,07235	0,00488
50	1	50	6,26384

Calculamos la probabilidad con la formula de la distribución exponencial $1 - e^{-\gamma x}$ donde lambda será igual a la media y x será igual al limite superior de cada intervalo, luego una vez halladas las probabilidades de cada intervalo calculamos la frecuencia esperada multiplicando el número de datos con la probabilidad del intervalo.

Finalmente utilizamos la formula de chi-cuadrado para la prueba de ajuste de bondad que es:

$$\chi^2 = \sum_{i=1}^{\infty} \frac{(FE - FO)^2}{FE}$$

Utilizamos la fórmula para hallar el estadístico de prueba que es el valor que compararemos en este caso es 6.26 entonces procedemos a calcular el valor crítico y plantear la hipótesis nula y alternativa.

Media:	45,3194
Intervalos:	7,07107
Tamaño de Intervalo:	24,875
Ho: Es Exponencial	
Ha: Es otra distribución	
Lambda=1/mu	0,02207
X^2(0):	6,26384
Alfa:	0,05
M:	8
K:	1
Grado de libertad:	6
X^2(0.05,6):	12,592

HO: Los datos siguen una distribución exponencial.

HA: Los datos siguen otra distribución.

En este caso utilizando los datos recolectados a lo largo del calculo hallamos en la tabla de distribución de chi cuadrado el valor χ^2 (α , m-k-1) donde Alpha es el valor de 0.05 y m es el numero de intervalos en este caso 8, k el número de parámetros utilizados en este caso solo 1 la media tomando en cuenta eso tenemos que χ^2 (0.05, 8-1-1) entonces χ^2 (0.05, 6) procedemos a buscar en la tabla y nos da el valor 12.592.

Ahora compramos el estadístico con el valor critico que seria $6,26 < 12.592$ si se cumple entonces podemos aceptar la hipótesis nula que dice que los datos siguen una distribución exponencial de media de 45.32 que nos daría un dato de E(45.32) horas.

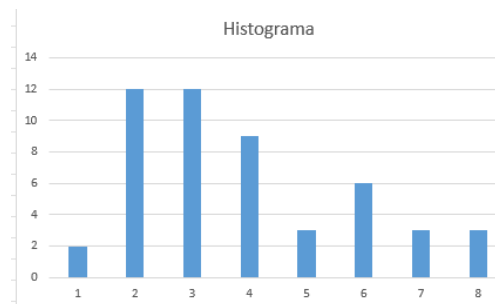
Se puede decir que el tiempo de reparación de los motores sigue una distribución exponencial con media de 45.32.

También nos da los tiempos de operación de los motores en una tabla donde seguimos el mismo procedimiento.

437.65	80.59	355.57	141.20	182.59
98.34	263.45	294.31	97.82	200.28
378.73	130.09	311.03	87.93	195.97
264.56	78.37	80.63	48.07	318.78
143.89	324.54	343.22	118.70	191.50
65.06	134.71	183.02	208.03	119.89
330.00	140.07	111.64	147.79	157.34
203.86	121.53	70.51	109.04	112.96
61.89	102.03	439.97	183.66	290.41
187.15	416.68	256.00	38.32	79.60

Para hallar la distribución que siguen los valores de la tabla primero ordenamos los números de menor a mayor y calculamos su media y el número de intervalos en el que se dividen con la fórmula de Raíz (número de datos) luego separamos los datos de la tabla en los intervalos y creamos el histograma para ver a que distribución se parece.

Intervalos	Inferior	Superior
1	0	55
2	55	110
3	110	165
4	165	220
5	220	275
6	275	330
7	330	385
8	385	INF



Viendo el histograma podemos decir que la distribución tiene forma de la distribución exponencial por tanto procedemos a realizar los cálculos de la frecuencia esperada para realizar la prueba de ajuste de bondad chi cuadrado.

FO	P(X)	P(X)% FE	Error
2	0,2527424	12,637122	8,9536493
12	0,1888637	9,4431847	0,6922775
12	0,1411298	7,0564913	3,4632337
9	0,1054603	5,2730165	2,6342428
3	0,078806	3,9403015	0,2243907
6	0,0588884	2,9444201	3,1709362
3	0,0440048	2,2002402	0,2907027
3	0,1301045	6,5052241	1,8887276
50	1	50	11,318161

Calculamos la probabilidad con la formulad de la distribución exponencial $1 - e^{-\gamma x}$ donde lambda será igual a la media y x será igual al límite superior de cada intervalo, luego una vez halladas las probabilidades de cada intervalo calculamos la frecuencia esperada multiplicando el número de datos con la probabilidad del intervalo.

Finalmente utilizamos la fórmula de chi-cuadrado para la prueba de ajuste de bondad que es:

$$\chi^2 = \sum_{i=1}^{\infty} \frac{(FE - FO)^2}{FE}$$

Utilizamos la fórmula para hallar el estadístico de prueba que es el valor que compararemos en este caso es 11.31 entonces procedemos a calcular el valor crítico y plantear la hipótesis nula y alternativa.

Media:	188,7794
Intervalos:	7,0710678
Tamaño de Intervalo:	55
Varianza	12052,113
Desviación	109,78212
Ho: Es Exponencial	
Ha: Es otra distribución	
Lambda=1/mu	0,0052972
X^2(0):	11,318161
Alfa:	0,05
M:	8
K:	1
Grado de libertad:	6
X^2(0.05,6):	12,592

HO: Los datos siguen una distribución exponencial.

HA: Los datos siguen otra distribución.

Ahora compramos el estadístico con el valor crítico que sería $11.31 < 12.592$ si se cumple entonces podemos aceptar la hipótesis nula que dice que los datos siguen una distribución exponencial de media de 188.78 que nos daría un dato de $E(188.78)$ horas. También nos podemos apoyar de la herramienta de Pro Model llamada Stat Fit para validar o reforzar los resultados.

Project Views	Intervals: 5 P	Project Views	autofit of distributions		
Input		Input	distribution	rank	acceptance
Data	1 437.65	Data	Exponential(38, 151)	71.5	do not reject
Notes	2 80.59	Notes	Lognormal(38, 4.63, 1.16)	70.8	reject
Statistics	3 355.57	Statistics	Uniform(38, 440)	0.000749	reject
Autofit of Distri	4 141.2	Autofit of Distri			
Graphics	5 182.59				
	6 98.34				
	7 263.45				
	8 294.31				
	9 97.82				
	10 200.28				
	11 378.73				
	12 130.09				

Donde lo único que hacemos es poner los datos que nos dan en la tabla y la herramienta detecta a que distribución pertenece, como se puede ver también indica que es una exponencial.

Se puede decir que el tiempo de operación de los motores sigue una distribución exponencial con media de 188.78.

Recursos de la empresa: La empresa cuenta con 3 motores de repuesto, 5 personas en el área de reparación y 2 montacargas.

Ganancias: La empresa gana 2000 dólares por tonelada vendida.

Costos: El costo anual de cada mecánico es de 150,000 dólares, el de los montacargas es de 450,000 dólares y cada motor ocasiona gastos anuales por 750,000 dólares.

Adaptando los costos para 1040 horas de simulación se realiza regla de 3 simple para calcular los nuevos costos.

Mecánicos: si en 8760 el costo es de 150000\$ entonces en 1040 hora el costo será aproximadamente de 17808.22\$.

Montacargas: si en 8760 el costo es de 450000\$ entonces en 1040 hora el costo será aproximadamente de 53424.66\$.

Motores: si en 8760 el costo es de 750000\$ entonces en 1040 hora el costo será aproximadamente de 89041.09\$.

Formulación del modelo:

- **LEAP**

Locaciones:

- Área de estiro cortadoras: capacidad = 50; unidades = 1.
- Área de espera: capacidad = 100; unidades = 1.
- Área de reparación: capacidad = 5; unidades = 1.
- Área de reserva = capacidad = 50; unidades = 1.
- Producción: capacidad = 1; unidades = 1.

Entidades:

- Motor descompuesto
- Motor reparado
- Motor de reserva
- Hilo

Arribos/Llegadas:

Arribos en área de estiro cortadoras:

- Entidad = motor reserva
- Locación = Área de estiro torcedoras
- Capacidad Arribo = 50
- Primera vez = 0
- Ocurrencia = 1
- Frecuencia = 0

Arribos en área de reserva:

- Entidad = motor reparado
- Locación = Área de reserva
- Capacidad Arribo = 1
- Primera vez = 0
- Ocurrencia = 3
- Frecuencia = 0

Arribos en área de producción:

- Entidad = hilo
- Locación = Área de estiro torcedoras
- Capacidad Arribo = 1
- Primera vez = 0
- Ocurrencia = Infinito
- Frecuencia = 5.84 min

Procesos:

1. Motor de reserva está en área de estiro cortadoras un tiempo de E(189) horas y pasa a área de espera.
2. Motor descompuesto está en área de espera y pasa a área de reparación.
3. Motor descompuesto está en área de reparación un tiempo de E(45) horas y pasa a área de reserva.
4. Motor reparado está en área de reserva y pasa a área de estiro cortadoras.
5. Hilo está en área de producción y pasa un tiempo de 5.83 minutos y luego sale.

Diagrama de simulación:

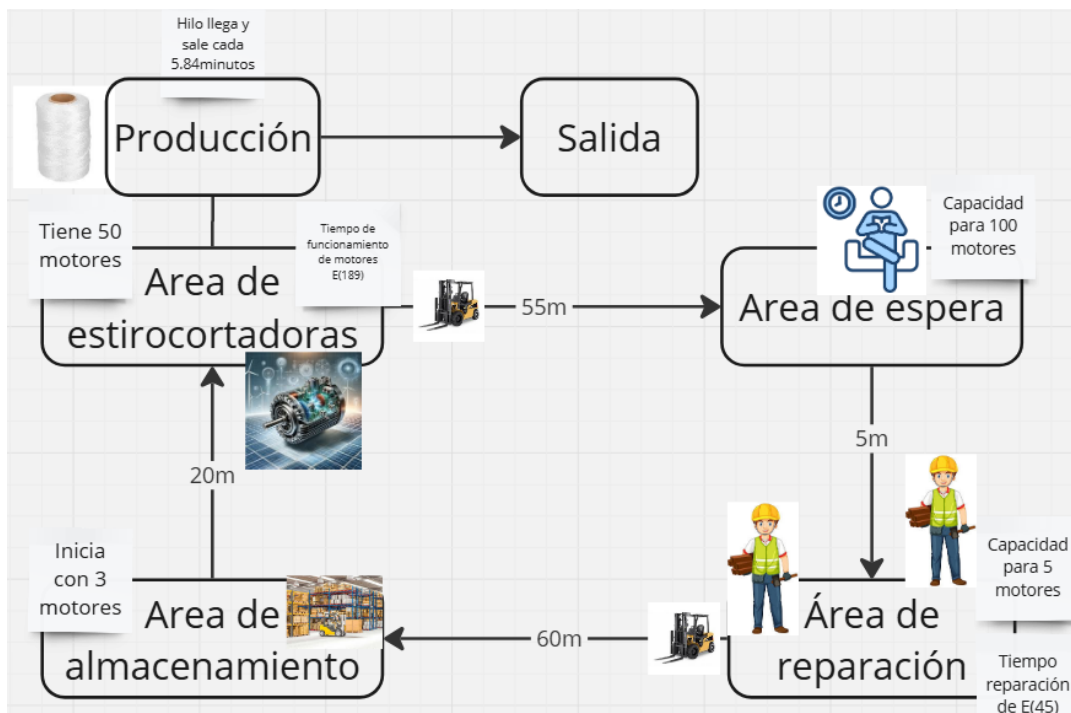
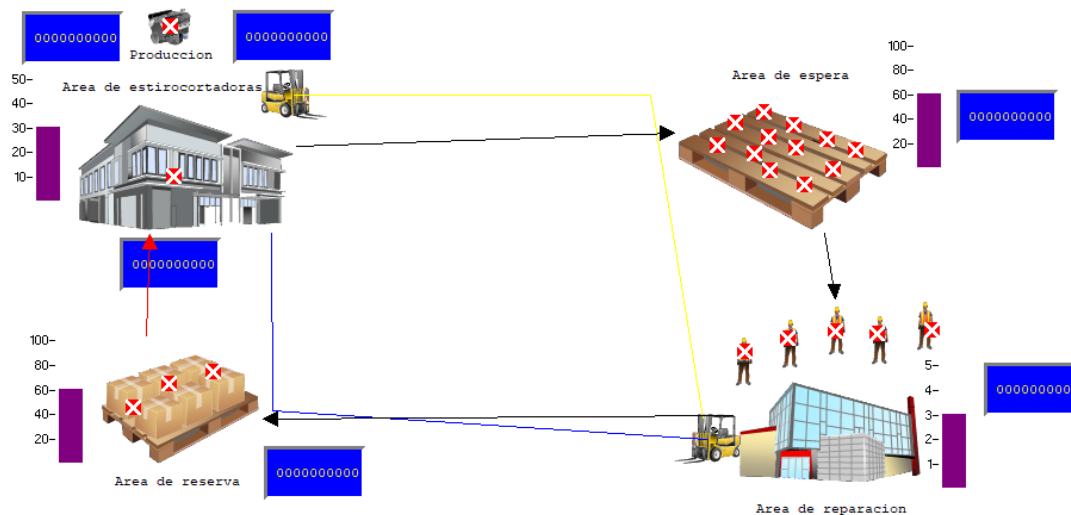


Diagrama de simulación ProModel:



Implementación del modelo en computadora:

ProModel y Java son herramientas útiles en el desarrollo de simulaciones y la creación de interfaces gráficas para visualizar resultados, cada una con características que se adaptan a diferentes necesidades en proyectos de simulación y modelado.

ProModel:

ProModel es un software especializado en la simulación de procesos y sistemas. Se utiliza principalmente para modelar, analizar y optimizar procesos complejos de flujo, como los que se encuentran en la manufactura, logística, salud y servicios. Su diseño visual permite modelar de manera rápida y precisa escenarios de procesos en los que el flujo de recursos y la toma de decisiones son clave para identificar cuellos de botella y mejorar la eficiencia.

Características Principales:

- **Modelado visual:** Permite crear modelos visuales e interactivos que representan el flujo de trabajo.
- **Optimización:** Cuenta con herramientas para evaluar diferentes escenarios y encontrar la configuración óptima para el proceso.
- **Análisis de resultados:** Genera reportes detallados y gráficos que ayudan en la interpretación de los resultados de la simulación.
- **Interfaz amigable:** Ofrece un entorno intuitivo, especialmente para usuarios con poca experiencia en programación.

Java:

Java es un lenguaje de programación versátil que permite desarrollar aplicaciones completas con interfaces gráficas personalizadas y lógica de negocio robusta. En el contexto de simulaciones, Java es útil para crear aplicaciones que requieran una interfaz de usuario y operaciones matemáticas complejas, así como para personalizar funcionalidades de la simulación que no son posibles o fáciles de implementar en herramientas especializadas como ProModel.

Características Principales:

- **Interfaz gráfica (GUI):** Java, junto con librerías como Swing y JavaFX, permite desarrollar aplicaciones con interfaces gráficas que facilitan la interacción con los datos de simulación.
- **Portabilidad:** Al ser un lenguaje multiplataforma, las aplicaciones desarrolladas en Java pueden ejecutarse en diferentes sistemas operativos sin modificaciones.
- **Integración de cálculos personalizados:** Java permite implementar cálculos específicos y algoritmos personalizados que no siempre se pueden realizar en software de simulación estándar.
- **Extensibilidad:** Ofrece la flexibilidad de integrar módulos adicionales y mejorar la funcionalidad según las necesidades del proyecto.

Validación:

Forma Manual:

Para la simulación manual tomaremos en cuenta el tiempo de simulación y la cantidad de toneladas de hilo de poliéster que supuestamente se deberían producir que en este caso es de 90000 toneladas por año o 8760 horas que es lo mismo.

Un mes tiene aproximadamente 4.33 semanas.

Entonces 6 meses serían aproximadamente $6 \times 4.33 = 25.98$ semanas.

Dado que cada semana laboral tiene 5 días $25.98 \text{ semanas} \times 5 \text{ días/semana} = 129.9$ días de trabajo.

Ahora considerando 8 horas de trabajo por día tenemos que $129.9 \text{ días} \times 8 \text{ horas/día} = 1039.02$ horas.

Redondeando el valor tenemos que el tiempo a simular es de **1040 horas** aproximadamente.

Realizando una regla de 3 simple decimo que si en 8760 horas se producen 90000 toneladas entonces en 1040 horas cuantas toneladas se producen.

$$\frac{8760 \text{ horas}}{1040 \text{ horas}} \quad \frac{90000 \text{ toneladas}}{x} \cong 10684.93 \text{ toneladas}$$

Pro Model:

Primeramente, para validar el modelo tomamos en cuenta que las toneladas de hilo de poliéster que se deberían producir de forma anual son de 90000 toneladas si convertimos el tiempo a horas tenemos que en un año hay 8760 horas por lo que si los motores funcionan según lo esperado en el tiempo de un año se debería producir la cantidad de 90000 toneladas de hilo de poliéster.

Nombre	Baseline
Prom. Duración de Calentamiento (Hr)	0
Prom. Duración de Simulación (Hr)	8760

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual
Cantidad real hilos	90.000,00	5,84	0,00	40.437.040,00	40.437.040,00
Motores funcionando	2.036,00	258,08	10,00	50,00	30,00
Cantidad teorica	90.000,00	5,84	0,00	90.000.000,00	90.000.000,00

Como se puede ver en la imagen para una simulación de un año si se cumple la condición de simulación de que en teoría debería producirse 90000 pero en realidad solo se produce 40437.04 toneladas en ese tiempo ahora si calculamos el tiempo pedido de simulación del medio año de trabajo decimos que:

Realizando la simulación con 1040 horas vemos que:

Nombre	Baseline
Prom. Duración de Calentamiento (Hr)	0
Prom. Duración de Simulación (Hr)	1040

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	10.684,00	5,84	0,00	5.505.100,00	5.505.100,00	3.133.879,92
Motores funcionando	267,00	233,26	12,00	50,00	19,00	25,76
Cantidad teorica	10.684,00	5,84	0,00	10.684.000,00	10.684.000,00	5.341.972,61

Como se puede ver efectivamente al simular 1040 horas la cantidad teórica de hilos es de 10684 toneladas y también vemos que dicha cantidad no se llega a cumplir debido a que la cantidad real de hilo producida es de **5505.1 toneladas** lo que concuerda con el contexto del ejercicio.

Java:

De igual forma en el programa de java debería de dar valores similares para el tiempo de simulación en 8760 horas y 1040 horas por lo que para realizar los cálculos simplemente ejecutamos la simulación de java con esas 2 hora para ver la cantidad de hilos producidos.

Para la simulación de java simulamos todo el año con 8760 horas esperando las 90000 toneladas esperadas.

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	90000,24
Toneladas reales	38152,91

Como se puede ver para la simulación de un año vemos que efectivamente se producen 90000 toneladas con un error de 0.24 lo cual no es mucho y también observar que las toneladas reales producidas son menores a las esperadas por lo que podemos decir que la simulación cumple con los requisitos del problema.

Realizando la simulación con 1040 horas:

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	10684,96
Toneladas reales	5497,82

Vemos que par las 1040 horas si se cumple la condición de la cantidad esperada que es de 10684 toneladas fallando por 0.96 decimales y también viendo que las toneladas reales son menores a lo previsto decimos que si cumple con los requisitos del problema.

Tabla de comparación:

Comparando los resultados obtenidos en las 3 pruebas realizados utilizando el tiempo de simulación y las toneladas esperadas podemos ver que:

	Manual	Pro Model	Java
8760 horas	90000 toneladas	90000 toneladas	90000.24 toneladas
1040 horas	10684.93 toneladas	10684 toneladas	10684.82 toneladas
Cumple producción	no	no	no

Como se puede ver la simulación de las tres formas nos da valores similares en la producción de toneladas de hilo de poliéster por lo que podemos decir que los modelos cumplen con los requisitos del problema.

Experimentación:

Pro Model:

- Prueba 1:**

Probamos con los siguientes datos para 26 mecánicos, 4 montacargas y 10 motores de reserva dando como resultado.

Nombre	Tiempo Programado (Hr)	Capacidad	Total Entradas	Tiempo Por entrada Promedio (Min)	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Area de estirocortadoras	8.760,00	50,00	2.261,00	11.189,62	48,13	50,00	50,00	96,27
Area de espera	8.760,00	100,00	2.211,00	0,78	0,00	2,00	0,00	0,00
Area de reparacion	8.760,00	26,00	2.211,00	2.622,50	11,03	22,00	9,00	42,43

Dando un porcentaje de utilización de 96.27% del área de estiro torcedoras y un 42.43% de utilización del área de reparación y una cantidad producida de toneladas total de 86652.24 toneladas con una cantidad de 50 moteres funcionando al momento.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	90.000,00	5,84	0,00	86.652.240,00	86.652.240,00	43.318.337,71
Motores funcionando	4.422,00	118,85	37,00	50,00	50,00	48,14
Cantidad teorica	90.000,00	5,84	0,00	90.000.000,00	90.000.000,00	44.999.506,85

- Prueba 2:**

Probamos con los siguientes datos para 26 mecánicos, 2 montacargas y 1 motores de reserva dando como resultado.

Nombre	Tiempo Programado (Hr)	Capacidad	Total Entradas	Tiempo Por entrada Promedio (Min)	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Area de estirocortadoras	8.760,00	50,00	1.940,00	11.241,18	41,49	50,00	40,00	82,98
Area de espera	8.760,00	100,00	1.900,00	0,78	0,00	2,00	0,00	0,00
Area de reparacion	8.760,00	26,00	1.900,00	2.624,54	9,49	23,00	11,00	36,49

Dando un porcentaje de utilización de 92.98% del área de estiro torcedoras y un 36.49% de utilización del área de reparación y una cantidad producida de toneladas total de 74692.58 toneladas con una cantidad de 40 moteres funcionando al momento.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	90.000,00	5,84	0,00	74.692.580,00	74.692.580,00	37.359.203,67
Motores funcionando	3.790,00	138,59	28,00	50,00	40,00	41,50
Cantidad teorica	90.000,00	5,84	0,00	90.000.000,00	90.000.000,00	44.999.506,85

- Prueba 3:**

Probamos con los siguientes datos para 13 mecánicos, 2 montacargas y 6 motores de reserva dando como resultado.

Nombre	Tiempo Programado (Hr)	Capacidad	Total Entradas	Tiempo Por entrada Promedio (Min)	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Area de estirocortadoras	8.760,00	50,00	2.106,00	11.298,56	45,27	50,00	42,00	90,54
Area de espera	8.760,00	100,00	2.064,00	132,50	0,52	9,00	1,00	0,52
Area de reparacion	8.760,00	13,00	2.063,00	2.570,28	10,09	13,00	13,00	77,60
Area de reserva	8.760,00	50,00	2.056,00	26,37	0,10	6,00	0,00	0,21
Produccion	8.760,00	1,00	90.001,00	5,80	0,99	1,00	1,00	99,32

Dando un porcentaje de utilización de 90.54% del área de estiro torcedoras y un 77.60% de utilización del área de reparación y una cantidad producida de toneladas total de 81498.3 toneladas con una cantidad de 42 moteres funcionando al momento.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	90.000,00	5,84	0,00	81.498.300,00	81.498.300,00	41.017.909,08
Motores funcionando	4.120,00	127,56	34,00	50,00	42,00	45,28
Cantidad teorica	90.000,00	5,84	0,00	90.000.000,00	90.000.000,00	44.999.506,85

- **Tabla de comparación:**

Las pruebas se realizaron con el periodo de tiempo de 1 año.

(Mecánicos, montacargas, motores)	Utilización de Área estiraje	Utilización de Área de reparación	Motores funcionando	Toneladas producidas
Prueba 1 (26, 4, 10)	96.27%	42.43%	50	86652.24 T
Prueba 2 (26, 2, 1)	92.98%	36.49%	40	74692.58 T
Prueba 3 (13, 2, 6)	90.54%	70.60%	42	81498.3 T

Viendo la tabla podemos decir que entre más mecánicos y más motores de reserva se aumenten la producción llega cada vez más a lo previsto que son las 90000 toneladas anuales, mientras que si solo aumentamos mecánicos estos no logran reparar a tiempo los motores por lo que el rendimiento también disminuye así que es necesario tener cierto equilibrio entre los 2 recursos.

Java:

- **Prueba 1:**

Probamos con los siguientes datos para 5 mecánicos, 2 montacargas y 3 motores de reserva y un tiempo de simulación de 8760 horas dando como resultado.

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	90000.24
Toneladas reales	38152.91
Motores en estirocortadoras	24
Motores en espera	24
Motores en reparacion	5
Motores en almacen	0
Min. motores estirocortadoras	8
Max. motores en espera	40
Utilización estirocortadoras	42.39%
Utilización espera	26.81%
Utilización reparación	99.72%

Que las toneladas teóricas producidas son de 90000.24 como pide originalmente el ejercicio y las toneladas reales son del 38152.91 dando como resultado que si efectivamente no se cumple la demanda y teniendo un porcentaje de utilización de 42.39% del área de estiro cortadoras y un 99.72% del área de reparación, también tenemos que la cantidad de motores es de 24 en estiro cortadoras, 24 en el área de espera y 5 en el área de reparación dando como resultado los 53 motores.

- **Prueba 2:**

Probamos con los siguientes datos para 13 mecánicos, 2 montacargas y 6 motores de reserva que son los datos que nos arrojó el Sin Runner de pro model dando como resultado.

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	10684.96
Toneladas reales	10130.57
Motores en estirocortadoras	46
Motores en espera	0
Motores en reparacion	10
Motores en almacen	0
Min. motores estirocortadoras	41
Max. motores en espera	2
Utilización estirocortadoras	94.81%
Utilización espera	0.04%
Utilización reparación	64.06%

Que las toneladas teóricas producidas son de 10684.96 como pide originalmente el ejercicio y las toneladas reales son del 10130.57 dando como resultado que si efectivamente no se cumple la demanda y teniendo un porcentaje de utilización de 94.81% del área de estiro cortadoras y un 64.06% del área de reparación, también tenemos que la cantidad de motores es de 46 en estiro cortadoras, 0 en el área de espera y 10 en el área de reparación dando como resultado los 56 motores.

- **Prueba 3:**

Probamos con los siguientes datos para 10 mecánicos, 3 montacargas y 6 motores de reserva dando como resultado.

Descripción	Valor
Montacargas utilizados	3
Toneladas teoricas	10684,96
Toneladas reales	9257,7
Motores en estirocortadoras	36
Motores en espera	10
Motores en reparacion	10
Motores en almacen	0
Min. motores estirocortadoras	31
Max. motores en espera	15
Utilización estirocortadoras	86.64%
Utilización espera	3.28%
Utilización reparación	93.06%

Que las toneladas teóricas producidas son de 10684.96 como pide originalmente el ejercicio y las toneladas reales son del 9257.7 dando como resultado que si efectivamente no se cumple la demanda y teniendo un porcentaje de utilización de 86.64% del área de estiro cortadoras y un 93.06% del área de reparación, también tenemos que la cantidad de motores es de 36 en estiro cortadoras, 10 en el área de espera y 10 en el área de reparación dando como resultado los 56 motores.

- **Tabla de comparación:**

(Mecánicos, montacargas, motores)	Utilización de Área estiraje	Utilización de Área de reparación	Motores funcionando	Toneladas producidas
Prueba 1 (5, 2, 3) 1año	42.39%	99.72%	24	38152.91 T
Prueba 2 (13, 2, 6)	94.81%	64.06%	46	10130.57 T
Prueba 3 (10, 3, 6)	86.64%	93.06%	36	9257.7 T

Como se puede observar en la tabla se puede ver vemos que el comportamiento del área de estiro cortadoras no cambia mucho de acuerdo al tiempo por que sigue sin cumplir la demanda y también vemos que para el tiempo establecido entre menos mecánicos hay el porcentaje de utilización del área de estiro cortadora baja sin importar los motores de repuesto ya que para reponer los faltantes se tiene que esperar a que se reparen los motores arruinados reflejando el resultado en las toneladas reales producidas.

Interpretación:

Pro Model:

Primeramente, nos encargamos de revisar las variables creadas que son 6 en este caso son la cantidad teórica de hilos, cantidad real de hilos, motores funcionando, motores en espera, motores en reparación y motores en reserva.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	10.684,00	5,84	0,00	5.505.100,00	5.505.100,00	3.133.879,92
Motores funcionando	267,00	233,26	12,00	50,00	19,00	25,76
Cantidad teorica	10.684,00	5,84	0,00	10.684.000,00	10.684.000,00	5.341.972,61
M espera	269,00	231,53	0,00	36,00	29,00	22,29
M reparacion	235,00	262,85	0,00	5,00	5,00	4,91
M reserva	233,00	265,11	0,00	3,00	0,00	0,04

Los datos recolectados nos dicen que para el periodo de simulación de 1040 horas la cantidad de hilos producida realmente es de 5505.1 toneladas, la cantidad de motores funcionando en ese momento solo es de 19 y la cantidad que en teoría efectivamente nos dice que si se producen es de 10684 toneladas, también tenemos los motores en espera que son 29, los motores en reparación que son 5 y los motores en reserva que son 0, si sumamos los totales nos da que los motores en sistema son 53 lo que coincide con el enunciado de 50 motores trabajando y 3 de reserva.

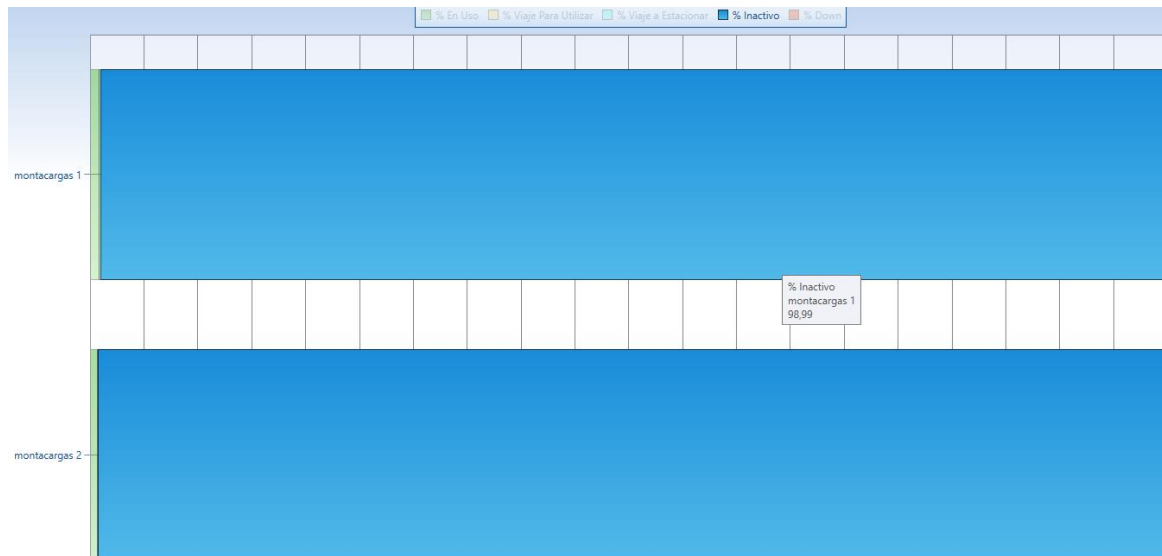
También cabe notar que el mínimo de motores que hubo funcionado es de 12 lo cual es muy poco y contribuye a la suposición de que por fallas no se está produciendo la cantidad estimada de toneladas como ya se vio en los datos.

Nombre	Tiempo Programado (Hr)	Capacidad	Total Entradas	Tiempo Por entrada Promedio (Min)	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Area de estirocortadoras	1.040,00	50,00	168,00	9.568,46	25,76	50,00	19,00	51,52
Area de espera	1.040,00	100,00	149,00	9.332,54	22,28	36,00	29,00	22,28
Area de reparacion	1.040,00	5,00	120,00	2.550,16	4,90	5,00	5,00	98,08
Area de reserva	1.040,00	50,00	118,00	22,01	0,04	3,00	0,00	0,08
Produccion	1.040,00	1,00	10.685,00	5,80	0,99	1,00	1,00	99,32

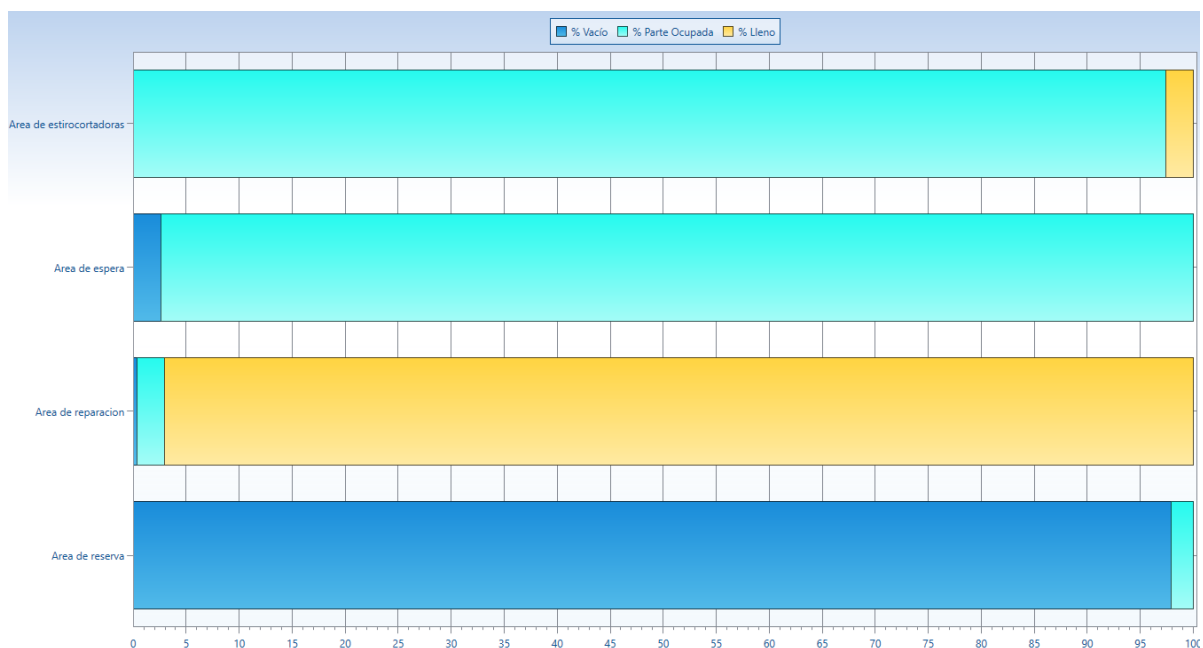
También podemos ver en las locaciones que el área de estiro cortadoras solo tiene un 51.52% de utilización mientras que el área de reparación esta siendo utilizada el 98.08% del tiempo es decir que esta área trabaja más que la principal, también tenemos el área de espera que trabaja el 22.28% del tiempo y el área de reserva que casi no se utiliza con un 0.08% de utilización.

Nombre	Total Salidas	Cantidad actual En Sistema	Tiempo En Sistema Promedio (Min)
motor descompuesto	0,00	34,00	0,00
motor reparado	0,00	0,00	0,00
motor reserva	0,00	19,00	0,00
hilo	10.684,00	1,00	5,80

En las entidades vemos que efectivamente el total de hilos que se produjeron en un caso de que no hay fallas es de 10684 toneladas lo que apoya la suposición y también vemos que los motores aparecen en cero los datos por que los motores nunca salen del sistema y simplemente se quedan en un ciclo de trabajo que es descomposición, reparación y de nuevo a trabajar.



Para las gráficas de los montacargas vemos que el montacargas 1 apenas se utilizaron el 0.79% del tiempo de la simulación y el montacargas 2 apenas se utilizó el 0.67% del tiempo de la simulación, ya que en la mayor parte de su trabajo de transportar los moteros no les toma mas de 1 min realizar el viaje completo.



Para las locaciones podemos ver que el área de estiro cortadoras esta ocupada el 97.42% del tiempo, el área de espera esta ocupada el 97.4% del tiempo ya que siempre pasan los motores por los lugares, en cambio el área de reparación anda lleno el 97.05% del tiempo es decir que no puede recibir mas motores y el área de reserva anda vacía el 97.88% del tiempo debido a que el área de estiramiento siempre necesita motores.

Java:

Primero nos encargamos de revisar los resultados arrojados de la simulación principal que es de 1040 horas con 3 motores de repuesto, con 5 mecánicos en el área de reparación y 2 montacargas.

Procedemos a revisar los resultados arrojados en la siguiente tabla que son los datos más importantes para responder las preguntas planteadas por el ejercicio.

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	10684,96
Toneladas reales	4635,22
Motores en estirocortadoras	14
Motores en espera	34
Motores en reparacion	5
Motores en almacen	0
Min. motores estirocortadoras	11
Max. motores en espera	37

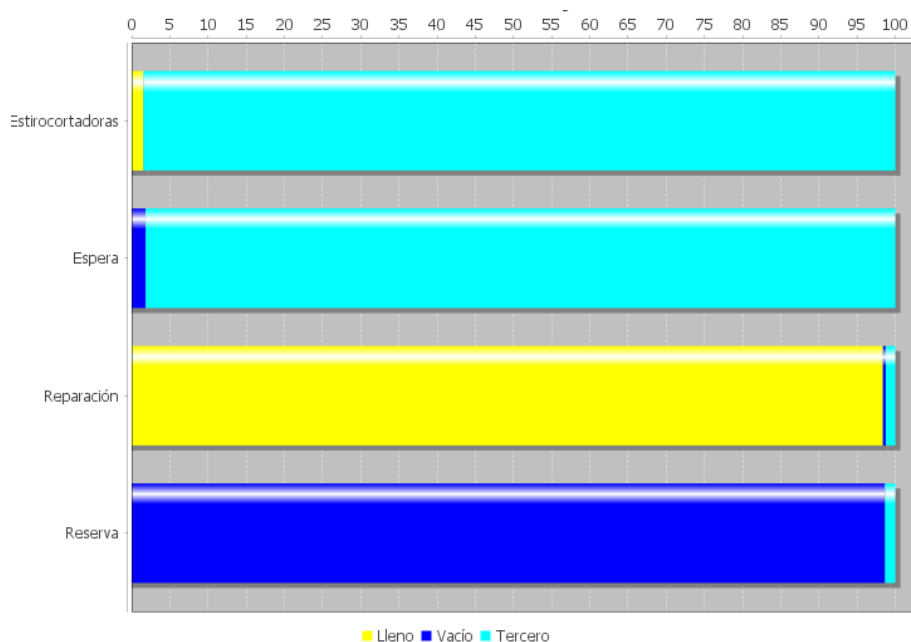
Primero vemos que la cantidad teórica de toneladas producidas es de 10684.96 que está muy cerca de las toneladas arrojadas por el pro model también tenemos que la cantidad real de toneladas producidas es de 4635.22 que también cumple con el criterio de que en realidad debido a la falla de los motores no se llega a cumplir la cantidad mencionada.

También tenemos la cantidad de motores en ese momento que es de 14 en el área de estiro cortadoras, 34 en el área de espera, 5 en el área de reparación y 0 en el área de almacenamiento y haciendo los cálculos tenemos que efectivamente se encuentran los 53 motores introducidos para el ejercicio, luego tenemos datos extras como el número mínimo de motores en el área de estiro cortadoras que es de 11 y en el área de espera una cantidad máxima que es de 37 motores en espera.

Utilización estirocortadoras	43.38%
Utilización espera	26.34%
Utilización reparación	98.81%
Utilización reserva	0.06%

Para los porcentajes de utilización tenemos que el área de estiro cortadoras se utiliza el 43.38% del tiempo, el área de espera se utiliza el 26.34% del tiempo y el área de reparación se utiliza el 98.81% del tiempo y el área de reserva el 0.06% del tiempo.

También tenemos las graficas de las diferentes áreas que representan su porcentaje de vacío llenos y ocupado.



Estirocortadoras lleno	1.54%
Estirocortadoras vacío	0.0%
Espera lleno	0.0%
Espera vacío	1.83%
Reparación lleno	98.37%
Reparación vacío	0.38%
Almacén lleno	0.0%
Almacén vacío	98.65%

En la grafica de barras se representa los porcentajes mostrados de forma grafica con lo que podemos ver es que el área de estiro cortadoras esta ocupada un 98.46% del tiempo, el área de espera esta ocupada un 98.17% de tiempo, el área de reparación esta llena un 98.37% del tiempo y vacía el 0.38% del tiempo, también tenemos que el área de almacenes para vacía el 98.65% del tiempo.

Documentación:

Archivo de texto Pro Model:

El archivo de texto generado por el modelo de simulación de Pro Model es el siguiente.

```

Listado del modelo formateado:
d:\umss\semestre 2-2024\tss\proyecto1\motores2sinrumpr.mod
*****

Unidades de Tiempo:      Minutos
Unidades de Distancia:   Pies

*****
Locaciones
*****

Nombre      Cap      Unidades Estadist      Reglas      Costos
-----
Area_de_estirocortadoras 50      1      Series de tiempo Más Tiempo, ,
Area_de_espera      100      1      Series de tiempo Más Tiempo, ,
Area_de_reparacion      Macro_mecanicos 1      Series de tiempo Más Tiempo, ,
Area_de_reserva      50      1      Series de tiempo Más Tiempo, ,
Produccion      1      1      Series de tiempo Más Tiempo, ,

*****
Entidades
*****

Nombre      Velocidad (Ppm) Estadist      Costos
-----
motor_descompuesto 150      Series de tiempo
motor_reparado      150      Series de tiempo
motor_reserva      150      Series de tiempo
hilo      150      Series de tiempo

```

Redes de Ruta

*

Nombre	Tipo	T/V	Desde	Hasta	BI	Distancia/Tiempo	Factor de Velocidad
ruta_1	Sobrepasar	Velocidad & Distancia	N1	N2	Bi	55	1
			N2	N3	Bi	5	1
ruta_2	Sobrepasar	Velocidad & Distancia	N1	N2	Bi	40	1
			N2	N3	Bi	20	1

Interfaces

*

Red	Nodo	Locación
ruta_1	N1	Area_de_estirocortadoras
	N2	Area_de_espera
	N3	Area_de_reparacion
ruta_2	N1	Area_de_reparacion
	N2	Area_de_reserva
	N3	Area_de_estirocortadoras

* Recursos *

Nombre	Unidades Estadíst	Rec Buscar	Ent Buscar	Ruta	Movimiento	Costos
montacargas_1	1	Por Unidad Más Cercano	Más Tiempo	ruta_1 Home: N1 (Regresar)	Vacío: 150 Ppm Lleno: 100 Ppm Tomar: 45 Segundos Dejar: 45 Segundos	
montacargas_2	1	Por Unidad Más Cercano	Más Tiempo	ruta_2 Home: N1 (Regresar)	Vacío: 150 Ppm Lleno: 100 Ppm Tomar: 45 Segundos Dejar: 45 Segundos	

* Procesamiento *

Proceso				Enrutamiento			
Entidad	Locación	Operación	Blk	Salida	Destino	Regla	Lógica de Movimiento
motor_reserva	Area_de_estirocortadoras	Wait E(189) hr	1	motor_descompuesto	Area_de_espera	FIRST 1	Move With montacargas_1 Then Free Dec Motores_funcionando ,1 Inc M_espera ,1
motor_descompuesto	Area_de_espera		1	motor_descompuesto	Area_de_reparacion	FIRST 1	Move With montacargas_1 Then Free Dec M_espera ,1 Inc M_reparacion ,1
motor_descompuesto	Area_de_reparacion	Wait E(45) hr	1	motor_reparado	Area_de_reserva	FIRST 1	Move With montacargas_2 Then Free Dec M_reparacion ,1 Inc M_reserva ,1
motor_reparado	Area_de_reserva		1	motor_reserva	Area_de_estirocortadoras	FIRST 1	Move With montacargas_2 Then Free Inc M_reserva ,1
motor_reparado	Area_de_reserva		1	motor_reserva	Area_de_estirocortadoras	FIRST 1	Move With montacargas_2 Then Free Inc Motores_funcionando , 1 Dec M_reserva ,1
hilo	Produccion	Wait 5.8 min	1	hilo	EXIT	FIRST 1	Inc Cantidad_teorica , 1000 If Motores_funcionando = 50 Then Inc Cantidad_real_hilos , 1000 If Motores_funcionando = 49 Then Inc Cantidad_real_hilos , 980 E If Motores_funcionando = 48 Then Inc Cantidad_real_hilos , 960 E If Motores_funcionando = 47 Then Inc Cantidad_real_hilos , 940 E If Motores_funcionando = 46 Then Inc Cantidad_real_hilos , 920 E If Motores_funcionando = 45 Then Inc Cantidad_real_hilos , 900 E If Motores_funcionando = 44 Then Inc Cantidad_real_hilos , 880 E If Motores_funcionando = 43 Then Inc Cantidad_real_hilos , 860 E If Motores_funcionando = 42 Then Inc Cantidad_real_hilos , 840 E If Motores_funcionando = 41 Then Inc Cantidad_real_hilos , 820 E If Motores_funcionando = 40

```

*****
                        Arribos
*****

Entidad      Locación      Cant. por Arribo  Primera Vez  Ocurrencias  Frecuencia Lógica
-----
motor_reserva Area_de_estirocortadoras 50           0           1
motor_reparado Area_de_reserva          1           0           Macro_motores
hilo          Produccion              1           0           INF           5.84 min

*****
                        Variables (global)
*****

ID            Tipo      Valor Inicial Estadist
-----
Cantidad_real_hilos Integer    0           Series de tiempo
Motores_funcionando Integer    50          Series de tiempo
Cantidad_teorica Integer    0           Series de tiempo
M_espera      Integer    0           Series de tiempo
M_reparacion  Integer    0           Series de tiempo
M_reserva     Integer    Macro_motores Series de tiempo

*****
                        Macros
*****

ID            Texto
-----
Macro_motores 3
Macro_montacargas 2
Macro_mecanicos 5

```

Código de java:

Interfaz de datos:

Primero para la interfaz de datos tenemos el siguiente formato que pide los datos de tiempo de simulación en horas, el delay que tendrá la simulación el numero de reparadores que tendremos, el numero de montacargas y el numero de motore de repuesto y también la media de las distribuciones para los tiempos de reparación y los tiempos de operación.

En la parte inferior tendrá un panel que muestre que es lo que pasa en tiempo real mediante mensajes sobre los motores y que es lo que están haciendo.

```

public class Ejer1UI extends javax.swing.JFrame {

    // Áreas de texto para mostrar resultados
    private JTextArea textAreaA;
    private JTextArea textAreaB;
    private JTextArea textAreaC;

    public Ejer1UI() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() { ...130 lines }

    private void simularEjercicioA(java.awt.event.ActionEvent evt, JTextField simulacionHoras, JTextField delay, JTextField numeroReparadores, JTextField numeroMotoresRepuesto, JTextField numeroMontacargas, JTextField mediaOperacion, JTextField mediaReparacion) {
        try {
            int duracionHoras = Integer.parseInt(simulacionHoras.getText());
            int delaySegundos = Integer.parseInt(delay.getText());
            int reparadores = Integer.parseInt(numeroReparadores.getText());
            int motoresReserva = Integer.parseInt(numeroMotoresRepuesto.getText());
            int montacargas = Integer.parseInt(numeroMontacargas.getText());
            int mediaOp = Integer.parseInt(mediaOperacion.getText());
            int mediaRep = Integer.parseInt(mediaReparacion.getText());

            Ejecucion1 hilo = new Ejecucion1(textAreaA, textAreaB, duracionHoras, delaySegundos, reparadores, motoresReserva, montacargas, mediaOp, mediaRep);
            hilo.start(); // iniciamos el programa

        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Por favor, introduce valores numéricos válidos.", "Error de entrada", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

En esta parte no encargamos de recuperar los datos de la interfaz y mandarlos como parámetros a nuestro simulador también es donde iniciamos el hilo que es el que realizara la simulación en tiempo real.

Entidad Motor:

```

public class Entidad {
    private String nombre;
    private int id;
    private double tiempo1;
    private double tiempo2;
    private double tiempo3;
    private boolean inAreal;
    private boolean inArea2;

    // Constructor
    public Entidad(String nombre, int id, boolean inAreal) {
        this.nombre = nombre;
        this.id = id;
        this.inAreal = inAreal;
    }

    // Retornos
    public String getNombre() {return nombre;}
    public int getId() {return id;}
    public double getTiempo1() {return tiempo1;}
    public double getTiempo2() {return tiempo2;}
    public double getTiempo3() {return tiempo3;}
    public boolean getInAreal() {return inAreal;}
    public boolean getInArea2() {return inArea2;}

    // Acciones
    public void cambiarNombre(String nombre) {this.nombre = nombre;}
    public void setTiempo1(double tiempo1) {this.tiempo1 = tiempo1;}
    public void setTiempo2(double tiempo2) {this.tiempo2 = tiempo2;}
    public void setTiempo3(double tiempo2) {this.tiempo2 = tiempo2;}
    public void setInAreal(boolean inAreal) {this.inAreal=inAreal;}
    public void setInArea2(boolean inArea2) {this.inArea2=inArea2;}
}

```

Luego creamos un objeto llamado entidad que se encargara de guardar unos datos importantes para la simulación como tiempos y marcadores de área que nos indicaran si la entidad está en el área respectiva junto con sus sets para cambiar y modificar los valores y sus gets para poder ver los valores en cualquier momento.

Simulación de Motores:

```
public class Ejecucion1 extends Thread {
    private JTextArea textAreaA;
    private JTextArea textAreaB;
    private int duracion;
    private int delay;
    //Variables a recibir
    private int Nreparadores;
    private int Nreservas;
    private int Nmontacargas;
    private int mediaOperacion;
    private int mediaReparacion;
    private SimulacionPanel simPanel;

    DecimalFormat df = new DecimalFormat("#.##"); //variable para 2 decimales

    public Ejecucion1(JTextArea textAreaA, JTextArea textAreaB, int duracion, int delay, int Nreparadores, int Nreservas, int Nmontacargas, int mediaOperacion, int mediaReparacion) {
        this.textAreaA = textAreaA;
        this.textAreaB = textAreaB;
        this.duracion = duracion;
        this.delay = delay;
        this.Nreparadores = Nreparadores;
        this.Nreservas = Nreservas;
        this.Nmontacargas = Nmontacargas;
        this.mediaOperacion = mediaOperacion;
        this.mediaReparacion = mediaReparacion;
        initGrafico(); //iniciar grafica
    }
}
```

Iniciamos con la inicialización del modelo que es donde reciben los datos enviados por la interfaz y los guarda en sus propias variables para ser utilizados en la simulación también al último manda a llamar a la gráfica para que también inicie.

```
private void initGrafico() {
    JFrame frame = new JFrame("Grafica de motores");
    simPanel = new SimulacionPanel();
    frame.setSize(900, 500);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.add(simPanel);
    frame.setVisible(true);
}

@Override
public void run() {
    int tiempoSimulacion = duracion; //convertir llamamos el tiempo de duración
    //Locaciones
    List<Entidad> estirocortadoras = new ArrayList<>(); //Locacion estirocortadoras
    List<Entidad> espera = new ArrayList<>(); //Locacion area de espera
    List<Entidad> reparacion = new ArrayList<>(); //Locacion area de reparacion
    List<Entidad> almacen = new ArrayList<>(); //Locacion area de almacen
    List<Entidad> montacarga1 = new ArrayList<>(); //Recurso montacarga 1
    List<Entidad> montacarga2 = new ArrayList<>(); //Recurso montacarga 2
    //Variables para calcular los datos
    double toneladasEsp = 0; //toneladas teoricas
    double toneladasRea = 0; //toneladas reales
    int estirocortadoraMin = 0; //Para guardar el minimo de motores
    int esperaMax = 0; //Para maximo de motores en area espera
    List<Double> porcentajeUti = new ArrayList<>(); //guardar porcentajes
    double porUti = 0; //mostrar porcentaje al final
    List<Double> porcentajeUti2 = new ArrayList<>(); //guardar porcentajes
    double porUti2 = 0; //mostrar porcentaje al final
    List<Double> porcentajeUti3 = new ArrayList<>(); //guardar porcentajes
    double porUti3 = 0; //mostrar porcentaje al final
    List<Double> porcentajeUti4 = new ArrayList<>(); //guardar porcentajes
    double porUti4 = 0; //mostrar porcentaje al final
    int contadorArealL = 0;
    int contadorArealV = 0;
}
```

En el método `initGrafico()` lo único que se ase es hacer visible el grafico y dale su tamaño y titulo respectivo, luego en el método `run()` es donde ya inicia la simulación primero creamos las locaciones como listas que recibirán entidades, luego inicializamos variable que nos serán utiliza para realizar el calculo de los diferentes datos que pide el ejercicio como la cantidad teórica de toneladas, la cantidad real los porcentajes de utilización y otros.

```

for(int i = 1; i <= tiempoSimulacion; i++){//simular las horas
//Arribos
if(i == 1){//Es inicio de simulacion
    arriboEstirocortadoras(50, estirocortadoras); //Arribo de 50 motores de recerva a estirocortadoras
    arriboAlmacen(Nreservas, almacen, 50); //Arribo de 3 motores reparados a almacen
    textAreaA.append("Motores en estirocortadora: "+estirocortadoras.size()+"\n"); //muestra motores al inicio
    textAreaA.append("Motores en Almacen: "+almacen.size()+"\n"); //muestra motores en almacen al inicio
    estirocortadoraMin = estirocortadoras.size();
}
//Calculo de toneladas esperadas y reales
toneladasEsp = dosDecimales(toneladasEsp + 10274.0);//Aumentamos las toneladas teoricas cada hora
toneladasRea = dosDecimales(toneladasRea + calcularCantidadReal(estirocortadoras.size()));//aumentamos las toneladas r
//Calcular minimos y maximos
if(estirocortadoras.size() < estirocortadoraMin){//si el numero de motores es menor al anterior lo guardamos
    estirocortadoraMin = estirocortadoras.size();
}
if(espera.size() > esperaMax){//si el numero de motores es mayor al anterior lo guardamos
    esperaMax = espera.size();
}
//Calcular porcentajes de utilización
porcentajeUti.add(dosDecimales((((double)estirocortadoras.size())/50)*100));
porcentajeUti2.add(dosDecimales((((double)espera.size())/100)*100));
porcentajeUti3.add(dosDecimales((((double)reparacion.size())/Nreparadores)*100));
porcentajeUti4.add(dosDecimales((((double)almacen.size())/50)*100));
//calcular si esta lleno o vacio
if(estirocortadoras.size() == 50){contadorArea1L++;}
if(espera.size() == 100){contadorArea2L++;}
if(reparacion.size() == Nreparadores){contadorArea3L++;}
if(almacen.size() == 50){contadorArea4L++;}
if(estirocortadoras.isEmpty()){contadorArea1V++;}
if(espera.isEmpty()){contadorArea2V++;}
if(reparacion.isEmpty()){contadorArea3V++;}
if(almacen.isEmpty()){contadorArea4V++;}
simPanel.actualizarToneladas(toneladasEsp,toneladasRea);
}
}

```

Iniciamos la simulación con un bucle for para simular las horas y como primer paso procedemos a dar los arribos de datos respectivos a las áreas de estiro cortadoras y a al área de reserva, luego procedemos a calcular las cantidades de toneladas de los hilos producidos tanto los teóricos y los reales después calculamos los calores mínimos y máximos de las áreas de espera y de estiro cortadoras y también procedemos a guardar los datos actuales en listas de los porcentajes para sacar su media al final.

```

//Simular entre hora
if (i != tiempoSimulacion) { // Si no es la última hora
    BigDecimal j = new BigDecimal("0.00"); //valor preciso
    BigDecimal one = new BigDecimal("1.0"); //valor preciso

    while (j.compareTo(one) < 0) { //j es menor a one
        BigDecimal horaExacta = new BigDecimal(i).add(j).setScale(2, RoundingMode.HALF_UP); //i+j para hora
        double tiempoActual = horaExacta.doubleValue(); //convierte a doble y guardar el valor de hora actual
        //textAreaA.append("Hora " + tiempoActual+"\n"); //imprime hora

        //Simulacion en almacen
        if(!almacen.isEmpty()){ //esta no vacia
            for(int a = 1; a <= almacen.size(); a++){ //recorremos cada motor
                Entidad motor = almacen.get(a - 1); //llamamos al motor que toca
                if(estirocortadoras.size() < 50){ //hay espacio en estiraje
                    if(montacarga2.isEmpty()){ //montacargas 2 esta libre
                        almacen.remove(motor); //eliminamos de almacen
                        montacarga2.add(motor); //esta en montacargas 2
                        simPanel.llegadaMotor("montacarga2");
                        simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacarga2", 1);
                        try {
                            Thread.sleep(100); // Agregar retraso de 1 segundo
                        } catch (InterruptedException v) {
                            Thread.currentThread().interrupt();
                            return;
                        }
                        montacarga2.remove(motor); //eliminamos de montacargas 2
                        motor.setInAreal(true); //motor esta en estiraje verdadero
                        estirocortadoras.add(motor); //agregamos motor a estiraje.
                        simPanel.llegadaMotor("estirocortadora");
                        simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacarga2", 1);
                        try {
                            Thread.sleep(100); // Agregar retraso de 1 segundo
                        } catch (InterruptedException v) {
                            Thread.currentThread().interrupt();
                            return;
                        }
                    }
                }
            }
        }
    }
}

```


En pesamos la simulación entre hora con un while que avanzara de 0.01 desde 0.00 hasta ser menor a 1.00, luego procedemos a simular todo lo que pasa en almacén primero preguntando si el almacén está o no vacío si no está entonces preguntamos si hay espacio en el área estiro cortadora para moverlos con el montacargas 2 a ese lugar si no seguir esperando y se procede a actualizar la grafica cada vez que un motor llega a una locación.

```
//Simulacion en reparacion
if(!reparacion.isEmpty()){//reparacion esta no vacia
    for(int r = 1; r<=reparacion.size();r++){//recorremos cada motor
        Entidad motor = reparacion.get(r - 1);//llamamos al motor que toca
        if(motor.getTiempo2() == tiempoActual){//si el tiempo de salida es el actual
            textAreaA.append("Motor " + motor.getId() + " reparado " + motor.getTiempo2()+"\n"); //mensaje reparacion
            motor.setTiempo2(0);//recetamos el tiempo de reparacion
            reparacion.remove(motor);//eliminamos de reparacion
            if(almacen.size() < 50){ //hay espacio en almacen
                montacarga2.add(motor);//va a montacargas 2
                simPanel.llegadaMotor("montacarga2");
                simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacarga2", 460, 370);
                try {
                    Thread.sleep(100); // Agregar retraso de 1 segundo
                } catch (InterruptedException v) {
                    Thread.currentThread().interrupt();
                    return;
                }
                montacarga2.remove(motor); //sale de montacargas2
                almacen.add(motor);//llega al almacen
                simPanel.llegadaMotor("almacen");
                simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacarga2", 160, 370);
                try {
                    Thread.sleep(100); // Agregar retraso de 1 segundo
                } catch (InterruptedException v) {
                    Thread.currentThread().interrupt();
                    return;
                }
            }
        }
    }
}
```

De igual forma en el área de reparación preguntamos si esta vacía si no preguntamos a cada motor que existe en el área si ya es su hora de salir de reparación si es su hora preguntamos si hay espacio en el almacén y si hay lo movemos de igual forma tenemos que utilizar el montacargas 2 y mandamos a actualizar el grafico cada vez que el motor llega a un área.

```
//Simulacion en espera
if(!espera.isEmpty()){//espera esta no vacio
    for(int e = 1; e<=espera.size();e++){//recorremos cada motor
        Entidad motor = espera.get(e - 1);//llamamos al motor que toca
        //reparacion en area de espera
        if(motor.getInArea2() == true){//si motor esta esperando
            if(reparacion.size() < Nreparadores){//hay espacio en reparacion
                textAreaA.append("Motor " + motor.getId() + " sale de del area de espera\n");
                motor.setInArea2(false);//motor ya no esta esperando
                if(montacargal.isEmpty()){ //montacarga esta vacio
                    espera.remove(motor); //motor sale de espera
                    montacargal.add(motor); //motor entra a montacargal
                    simPanel.llegadaMotor("montacargal");
                    simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacargal", 460, 120);
                    try {
                        Thread.sleep(100); // Agregar retraso de 1 segundo
                    } catch (InterruptedException v) {
                        Thread.currentThread().interrupt();
                        return;
                    }
                }
            }
            montacargal.remove(motor); //motor sale de montacarga
            reparacion.add(motor); //motor entra a reparacion
            simPanel.llegadaMotor("reparacion");
            simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacargal", 460, 300);
            try {
                Thread.sleep(100); // Agregar retraso de 1 segundo
            } catch (InterruptedException v) {
                Thread.currentThread().interrupt();
                return;
            }
        }
        repararMotor(motor, tiempoActual); //motor se repara
    }
}
```

Igualmente, para el área de espera preguntamos si esta vacía si no lo mandamos a reparación si hay espacio utilizando el montacargas 1 una vez llegando al área de reparación le damos un nuevo tiempo de reparación para ver ahora a qué hora saldrá el motor de dicha área.

```
//Simular fallos
if(motor.getTiempo1() == tiempoActual){ //llega el tiempo del motor de fallar

    textAreaA.append("motor " + motor.getId() + " se arruino\n");
    motor.setTiempo1(0); //motor deja de funcionar
    if(montacargal.isEmpty()){ //montacarga esta desocupado
        estirocortadoras.remove(motor); //Se elimina de area estiro
        montacargal.add(motor); //se da al montacargal
        simPanel.llegadaMotor("montacargal");
        simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacargal", 160, 120);
        try {
            Thread.sleep(100); // Agregar retraso de 1 segundo
        } catch (InterruptedException v) {
            Thread.currentThread().interrupt();
            return;
        }
        motor.setInAreal(false); //motor ya no esta en area estiro
        motor.cambiarNombre("motor arruinado"); //se cambia el nombre a motor arruinado
        if(espera.size() < 100){ //hay espacio en espera
            montacargal.remove(motor); //motor sale de montacargal
            espera.add(motor); //motor entra a espera
            simPanel.llegadaMotor("espera");
            simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacargal", 460, 120);
            try {
                Thread.sleep(100); // Agregar retraso de 1 segundo
            } catch (InterruptedException v) {
                Thread.currentThread().interrupt();
                return;
            }
        }
        if(reparacion.size() < Nreparadores){ //Hay espacio en reparacion
            if(montacargal.isEmpty()){ //montacarga esta vacio
                espera.remove(motor); //motor sale de espera
                montacargal.add(motor); //motor entra a montacargal
                simPanel.llegadaMotor("montacargal");
            }
            if(reparacion.size() < Nreparadores){ //Hay espacio en reparacion
                if(montacargal.isEmpty()){ //montacarga esta vacio
                    espera.remove(motor); //motor sale de espera
                    montacargal.add(motor); //motor entra a montacargal
                    simPanel.llegadaMotor("montacargal");
                }
                montacargal.remove(motor); //motor sale de montacarga
                reparacion.add(motor); //motor entra a reparacion
                simPanel.llegadaMotor("reparacion");
                simPanel.actualizarEstadoAreas(estirocortadoras, espera, reparacion, almacen, "montacargal", 460, 300);
                try {
                    Thread.sleep(100); // Agregar retraso de 1 segundo
                } catch (InterruptedException v) {
                    Thread.currentThread().interrupt();
                    return;
                }
                repararMotor(motor, tiempoActual); //reapara el motor
            } else {
                motor.setInArea2(true); //motor esta esperando
                textAreaA.append("Motor " + motor.getId() + " esta en area de espera\n");
            }
        }
    }
}

j = j.add(new BigDecimal("0.01")).setScale(2, RoundingMode.HALF_UP); //j++
try {
    Thread.sleep(delay); // Agregar retraso de 1 segundo
} catch (InterruptedException e) {
    Thread.currentThread().interrupt();
    return;
}
```

Igualmente, para el área de estiro cortadora preguntamos y su tiempo de operación ya se cumplió para marcar el motor como arruinado y si es así preguntamos si hay espacio en espera si hay lo mandamos con el montacarga 1 y si también hay espacio en reparación lo mandamos directamente a reparar, pero si no hay espacio le ponemos su estado en espera y lo añadimos al área de espera.

```

textAreaB.append("Área de almacen esta lleno el: " + dosDecimales(((double)contadorArea4L/tiempoSimulacion)*100) + "%\n");
textAreaB.append("Área de almacen esta vacío el: " + dosDecimales(((double)contadorArea4V/tiempoSimulacion)*100) + "%\n");
mostrarResultadosEnTabla(Nmontacargas, dosDecimales(toneladasEsp/1000), dosDecimales(toneladasRea/1000), estirocortadoras.
    reparacion.size(), almacen.size(), estirocortadoraMin, esperaMax, porUti, porUti2, porUti3, porUti4,
    area1L, area1V, area2L, area2V, area3L, area3V, area4L, area4V, tiempoSimulacion);
BarChartExample example = new BarChartExample(area1L, area1V, area2L, area2V, area3L, area3V, area4L, area4V);
example.setVisible(true);
}

```

Y finalmente enviamos los datos para que se muestren en tabla y en las graficas respectivas con los métodos de mostrar resultados en tabla y los de bar char.

```

public void arriboEstirocortadoras (int capacidad, List<Entidad> locacion){
    for(int i = 1; i <= capacidad; i++){//creamos la cantidad de motores requerida
        Entidad entidad = new Entidad("motor reserva", i, true);//con sus datos
        locacion.add(entidad);//y los ponemos en estiraje
    }
}

public void arriboAlmacen (int capacidad, List<Entidad> locacion, int idAnterior){
    for(int i = 1; i <= capacidad; i++){//creamos la cantidad de motores requerida
        Entidad entidad = new Entidad("motor reparado", idAnterior+i, false);//con sus datos
        locacion.add(entidad);//y los ponemos en la almacen
    }
}

public void operacionMotor(Entidad motor, double tiempoActual){
    ExponentialDistribution distribution = new ExponentialDistribution(mediaOperacion); //iniciamos la distribucion exponencial
    double expoPrueba = distribution.sample()+1; //guardamos el valor de la distribución
    BigDecimal roundedExpoPrueba = new BigDecimal(expoPrueba + tiempoActual).setScale(2, RoundingMode.HALF_UP);//lo convertimos a
    double tiempoOperacion = roundedExpoPrueba.doubleValue();//lo convertimos en double para calculos
    motor.setTiempo1(tiempoOperacion); // le damos el valor al motor
    textAreaA.append("Motor " + motor.getId() + " funciona " + motor.getTiempo1()+"\n");
}

public void repararMotor (Entidad motor, double valorActual){
    ExponentialDistribution distribution = new ExponentialDistribution(mediaReparacion); //iniciamos la distribucion exponencial
    double reparacionP = distribution.sample()+1; //guardamos el valor de la distribución
    BigDecimal roundedExpoPrueba = new BigDecimal(reparacionP).setScale(2, RoundingMode.HALF_UP);//lo convertimos a 2 decimales
    double tiempoReparacion = roundedExpoPrueba.doubleValue();//lo convertimos en double para calculos
    motor.setTiempo2(tiempoReparacion);//guardamos el tiempo de reparacion
    textAreaA.append("Motor " + motor.getId() + " tiempo de reparacion " + motor.getTiempo2()+"\n");//mostramos el tiempo reparac
    BigDecimal horasalidaP = new BigDecimal(tiempoReparacion+valorActual).setScale(2, RoundingMode.HALF_UP);//calculamos tiempo s
    double tiempoSalida = horasalidaP.doubleValue();//guardamos tiempo salida
    motor.setTiempo2(tiempoSalida);//lo damos el tiempo salida al motor
    textAreaA.append("Motor " + motor.getId() + " saldra en la hora " + motor.getTiempo2()+"\n"); //mostramos a que hora saldra
}

```

Luego tenemos los métodos mas importantes que son el arribo de estiro cortadoras que se llama al inicio de la simulación para simular la llegada de los 50 motores, también el método arribo en almacén para simular la llegada de los motores de repuesto.

Y los métodos de operación de motor que es el encargado de utilizar una distribución exponencial para simular el tiempo que tardaran los motores en fallar y el método reparar motor que simulara el tiempo en el que se repararan los motores y el tiempo en el que saldrán del área de reparación.

Resultados:

Pro model:

A) El índice real de producción actual:

Para saber el índice de producción actual para una simulación de 1040 horas solo debemos ver la variable toneladas reales que es la que representa el índice de producción real o actual.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	10.684,00	5,84	0,00	5.505.100,00	5.505.100,00	3.133.879,92
Motores funcionando	267,00	233,26	12,00	50,00	19,00	25,76
Cantidad teorica	10.684,00	5,84	0,00	10.684.000,00	10.684.000,00	5.341.972,61

Vemos que el índice real de producción es de **5505.1 toneladas** en un periodo de 1040 horas.

B) La utilidad anual actual.

Para calcular la utilidad actual primero necesitamos el índice de producción real que ya lo tenemos que es de 5505.1 toneladas y los costos de para una simulación de 1040 horas que son de 17808.22\$ para los mecánicos, 53424.66\$ para los montacargas y 89041.09\$ para cada motor.

Ahora calculando tenemos que la ganancia es de.

$$5505.1T \times 2000\$ = 11010200$$

Y para los costos tenemos que, para 3 motores de repuesto, 2 montacargas y 5 mecánicos es de.

$$17808.22\$ \times 5 + 53424.66\$ \times 2 + 89041.09\$ \times 53 = 4915068.19\$$$

Restando los costos de la ganancia tenemos que.

$$11010200\$ - 4915068.19\$ = 6095131.81\$$$

Tenemos que la utilidad actual es de 6095131.81\$

C) El mínimo número de mecánicos a contratar para trabajar en el taller, el número mínimo de montacargas a comprar para el traslado de los motores y el número mínimo de motores de repuesto, para que el departamento de estiraje esté trabajando al 90% de utilización y que le permita obtener un índice de producción anual de 81,000 toneladas. La producción fabricada por arriba de esta cantidad no se puede vender por falta de demanda.

En el Sin Runner queremos tener un porcentaje de utilización de 90% del área de estiro cortadoras entonces lo seleccionamos.

Seleccionamos los macros que son los datos que se modificaran para ir probando que son el numero de motores de reserva, el número de montacargas y el número de reparadores.

Macro_motores : Default = 3.00, Lower = 1, Upper = 10
 Macro_montacargas : Default = 2.00, Lower = 1, Upper = 6
 Macro_mecanicos : Default = 5.00, Lower = 1, Upper = 50

Luego procedemos a simular los posibles escenarios y escogemos el que mejor se adapte a las características solicitadas.

Experiment	Objective Function	Area_de_estirocortadoras: % Utilization	Macro_motores	Macro_montacargas	Macro_mecanicos
16	90.831	90.831	6.000	4.000	26.000
7	90.831	90.831	6.000	2.000	26.000
10	90.831	90.831	6.000	2.000	38.000
1	90.298	90.298	6.000	4.000	13.000

Vemos que hay cuatro pruebas que se adaptan al 90% de utilización que buscamos entonces probamos que más cerca de 90% este es decir el experimento 1.

Probamos el porcentaje con los datos y vemos que efectivamente el porcentaje es de 90.3% de utilización.

Nombre	Tiempo Programado (Hr)	Capacidad	Total Entradas	Tiempo Por entrada Promedio (Min)	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Area de estirocortadoras	1.040,00	50,00	307,00	9.176,91	45,15	50,00	46,00	90,30

Entonces podemos decir que el número **mínimo que necesitamos es de 6 motores de repuesto, 4 montacargas y 13 mecánicos o reparadores.**

D) La utilidad anual con las mejoras realizadas.

Calcular la utilidad con la mejora tenemos que realizadas tenemos que.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Cantidad real hilos	10.684,00	5,84	0,00	9.648.780,00	9.648.780,00	4.810.053,48
Motores funcionando	518,00	119,82	36,00	50,00	46,00	45,15
Cantidad teorica	10.684,00	5,84	0,00	10.684.000,00	10.684.000,00	5.341.972,61

La cantidad real de hilos producida es de 9648.78 toneladas multiplicando con 2000\$ de ganancia por tonelada tenemos que la ganancia es de.

$$9648.78T \times 2000\$ = 19297560\$$$

Y para calcular los costos tenemos que para 13 mecánicos, 4 montacargas y 6 motores de reserva es de.

$$17808.22\$ \times 13 + 53424.66\$ \times 4 + 89041.09\$ \times 6 = 5433186.54\$$$

Restando la ganancia de los costos tenemos que la utilidad es de.

$$19297560\$ - 5433186.54\$ = 13864373.46\$$$

Tenemos que la utilidad es de 13864373.46\$

Java:

A) El índice real de producción actual.

Para la simulación en java de 1040 horas con 5 mecánicos, 2 montacargas y 3 motores de reserva tenemos que.

Descripción	Valor
Montacargas utilizados	2
Toneladas teoricas	10684,96
Toneladas reales	5273,23

Vemos que la cantidad de hilo real producido es de 5273.23 toneladas.

B) La utilidad anual actual.

Para calcular la utilidad actual primero necesitamos la cantidad real de hilos producida que es de 5273.23 tonelada multiplicando con 2000\$ tenemos.

$$5273.23 T \times 2000\$ = 10546460\$$$

Y los costos que se calcularon anteriormente para 5 mecánicos, 2 montacargas y 3 motores de reserva es de 4915068.19\$

Realizamos la resta y tenemos que la utilidad actual real es de.

$$10546460\$ - 4915068.19\$ = 5631391.81\$$$

Que también coincide con la cantidad calculada por el programa.

Ganancia real	10546460\$
costos	4915068,19\$
Utilidas	5631391,81\$

Por tanto, la utilidad es de 5631391.81\$

- C) El mínimo número de mecánicos a contratar para trabajar en el taller, el número mínimo de montacargas a comprar para el traslado de los motores y el número mínimo de motores de repuesto, para que el departamento de estiraje esté trabajando al 90% de utilización y que le permita obtener un índice de producción anual de 81,000 toneladas. La producción fabricada por arriba de esta cantidad no se puede vender por falta de demanda.

Como java no tiene el Sin Runner para hacer experimentos probamos directamente los datos arrojados por pro model que son número mínimo que necesitamos es de 6 motores de repuesto, 4 montacargas y 13 mecánicos o reparadores.

Utilización estirocortadoras	90.59%
Utilización espera	0.83%

Tenemos que efectivamente el porcentaje de utilización es del 90.59% por lo que podemos decir que también aplican los calores mínimos.

Entonces podemos decir que el número **mínimo que necesitamos es de 6 motores de repuesto, 4 montacargas y 13 mecánicos o reparadores.**

- D) La utilidad anual con las mejoras realizadas.

Descripción	Valor
Montacargas utilizados	4
Toneladas teoricas	10684,96
Toneladas reales	9679,96

De igual forma para calcular la utilidad necesitamos el numero de toneladas producidas actual 9679.96 y lo multiplicamos por 2000\$.

$$9679.96T \times 2000\$ = 19359920\$$$

Luego calculamos los costos que ya los tenemos del Pro Model para esas cantidades.

$$17808.22\$ \times 13 + 53424.66\$ \times 4 + 89041.09\$ \times 56 = 5433186.54\$$$

Luego restamos el costo de la ganancia para hallar la utilidad que es de.

$$19359920\$ - 5433186.54\$ = 13926733.46\$$$

Vemos que la utilidad con las mejoras realizadas es de 13926733.46\$

Tabla de comparación:

En la siguiente tabla se realiza una comparación de los resultados de los incisos en los diferentes modelos Java y Pro Model.

	Pro Model	Java
Inciso A) Producción real	5505.1 Toneladas	5273.23 Toneladas
Inciso B) Utilidad real	6,095,131.81\$	5,631,391.81\$
Inciso C) 90% de utilización	6 motores de repuesto 4 montacargas 13 mecánicos	6 motores de repuesto 4 montacargas 13 mecánicos
Inciso D) Utilidad con mejoras	13,864,373.46\$	13,926,733.46\$

Como se puede ver en la tabla ambos modelos arrojan resultados similares para responder los incisos eso quiere decir que los modelos si simulan el escenario planteado por lo que se puede utilizar para realizar más pruebas.

Conclusión:

En conclusión, viendo los resultados de los incisos y los demás resultados arrojados por los modelos podemos decir que si bien tener mas mecánicos y motores de repuesto a primera vista suena a que solo nos genera costos, como se vio en las simulaciones aumentar su número en especial en número de mecánicos ayuda mucho a mejorar la producción del área de estiro cortadoras lo que genera una ganancia mayor a solo mantener un número mínimo de mecánicos.

Videos del proyecto:**Pro Model:**

<https://youtu.be/8tc5azGBbqY>

Java:

<https://youtu.be/1jYOGZZxTQM>