

Proyecto 2

Fábrica de piezas de auto



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:	4
• LEAP	4
• Diagrama de simulación:	6
• Diagrama de simulación ProModel:	6
Implementación del modelo en computadora:	7
ProModel:	7
Java:	7
Validación:	8
Forma manual:	8
Pro model:	8
Java:	10
Tabla de comparación:	10
Experimentación:	11
Pro Model:	11
Java:	13
Interpretación:	15
Pro Model:	15
Graficas Pro Model:	16
Java:	17
Graficas java:	18
Documentación:	20
Archivo de texto Pro Model:	20
Código de java:	22
Resultados:	30
Pro Model:	30
Java:	31
Tabla de comparación:	32
Conclusión:	32
Videos del proyecto:	32

Definición del sistema:

La empresa Engranes de Acero SA fabrica piezas automotrices en una celda flexible de manufactura. El diagrama muestra la distribución de los equipos en la celda. Las piezas entran del exterior a un almacén de materia prima a una tasa de 12 piezas/h con distribución Poisson y posteriormente son procesados en el horno, la capacidad del horno es de 10 piezas e inicia su operación cuando está lleno.

El flujo de producción es el siguiente:

1. Horno
2. Carga
3. Torneado
4. Fresado
5. Taladrado
6. Rectificado
7. Descarga
8. Inspección

Para cada movimiento entre las operaciones se requiere de las bandas transportadoras, el robot o la grúa viajera de acuerdo con el diagrama cada locación y recurso tiene sus datos de procesamiento y velocidad de movimiento respectivamente.

Con esta información desarrolle un modelo de simulación. Realice 3 réplicas de 1000 horas cada uno y determine:

- a) La utilización de los equipos.
- b) La producción de la celda.
- c) Identifique la estación cuello de botella.

Determine el número mínimo de máquinas que deberían ser instaladas en cada estación para maximizar la producción y que la estación considerada como cuello de botella trabaje a una utilización lo más cercana al 95%.

Como se puede ver el problema trata de saber la utilización total de los equipos en una celda y la cantidad de piezas que esta puede producir, también nos pide experimentar con las cantidades de máquinas en las áreas de trabajo para poder maximizar la producción y el porcentaje de utilización de una locación que se considera cuello de botella.

Colección de datos:**Llegada de piezas:**

La llegada de las piezas sigue una distribución de Poisson de 12 piezas por hora y la capacidad de procesamiento del horno es de 10 piezas.

Tiempos de procesamiento:

Para los tiempos de procesamiento y las cantidades de máquinas de las diferentes locaciones tenemos la siguiente tabla:

Operación	Tiempo (min/pieza)	Número de maquinas
Carga y descarga	0.5	Infinito
Torneado	5.2	1
Fresado	9.17	1
Taladrado	1.6	1
Rectificado	2.85	1
Inspección	Exponencial (3)	1
Horno	100	1

Velocidad de los equipos de transporte:

Para la velocidad de los equipos de transporte que se utilizan en el ejercicio tenemos la siguiente tabla.

Equipo	Velocidad (pies/min)	Cantidad de equipos
Robot	45	1
Bandas (longitud 30ft)	30	2
Grúa	25	1

Distancia entre los procesos:

Para la distancia entre los procesos de la celda tenemos la siguiente tabla.

Estación	Estación	Distancia (pies)
Almacén inicial	Horno	10
Horno	Banda 1	15
Torneado y rectificado	Fresado y taladrado	15
Carga y descarga	Torneado y rectificado	20

Tiempo de simulación:

Para el tiempo de simulación el problema pide realizar una simulación de 1000 horas y de esa simulación realizar 3 réplicas.

Formulación del modelo:

- **LEAP**

Locaciones:

- **Almacén:** capacidad = Infinito; unidades = 1.
- **Horno:** capacidad = 10; unidades = 1.
- **Banda 1:** capacidad = Infinito; unidades = 1.
- **Área de carga:** capacidad = Infinito; unidades = 1.

- **Torneado:** capacidad = 1; unidades = 1.
- **Fresado:** capacidad = 1; unidades = 1.
- **Taladrado:** capacidad = 1; unidades = 1.
- **Rectificado:** capacidad = 1; unidades = 1.
- **Área de descarga:** capacidad = Infinito; unidades = 1.
- **Banda 2:** capacidad = Infinito; unidades = 1.
- **Inspección:** capacidad = 1; utilidades = 1.

Entidades:

- Pieza

Arribos/Llegadas:

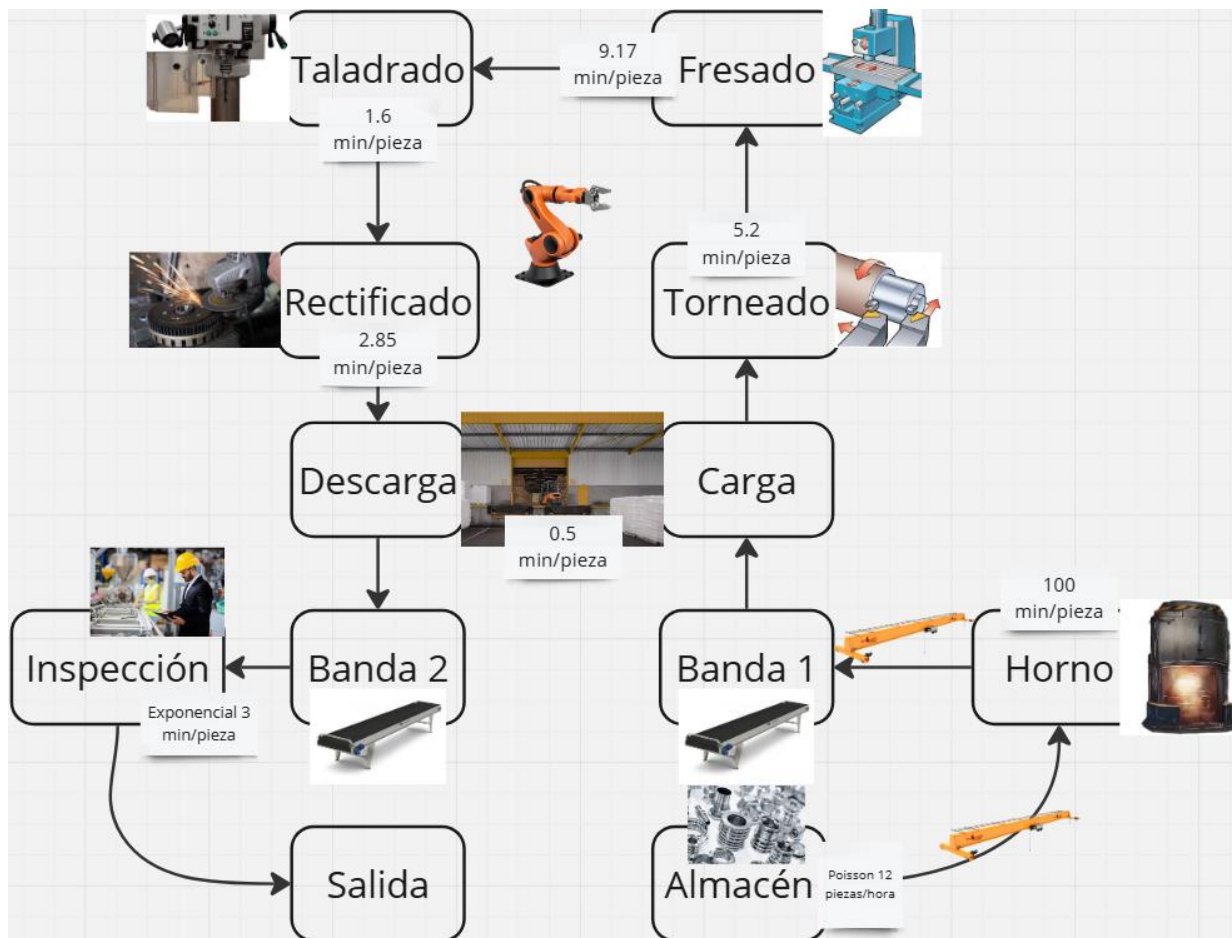
Arribos a almacén:

- Entidad = Pieza
- Locación = Almacén
- Capacidad Arribo = Poisson (12)
- Primera vez = 0
- Ocurrencia = Infinito
- Frecuencia = 1 hora

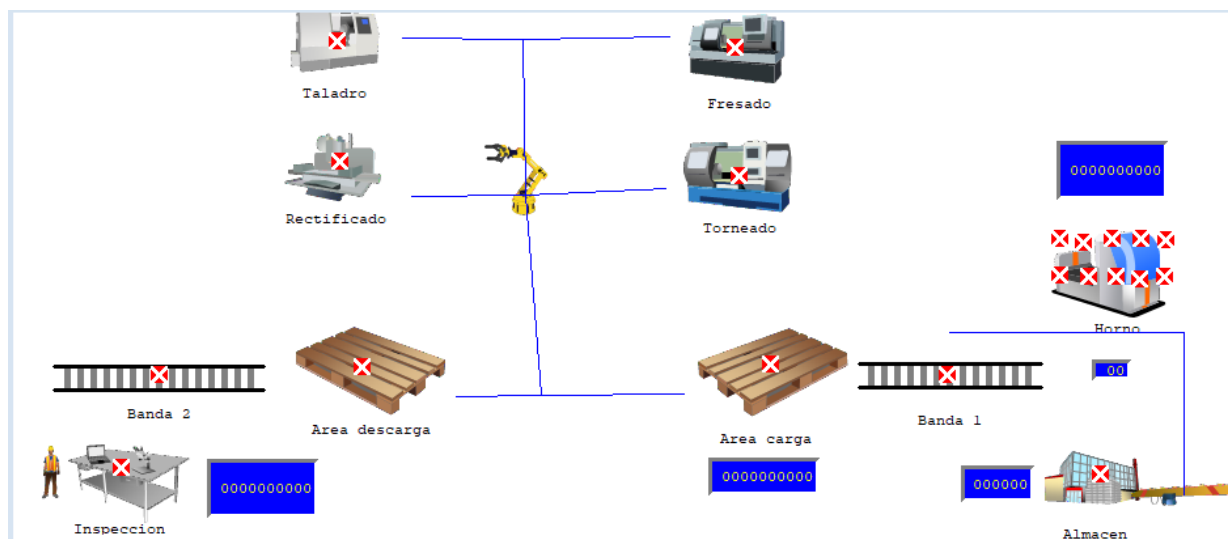
Procesos:

1. Pieza llega a el almacén y se dirige al horno.
2. Pieza esta en el horno, el horno espera a tener 10 piezas en él, pieza espera un tiempo de 100 minutos para pasar a banda 1.
3. Pieza esta en banda 1 y pasa al área de carga.
4. Pieza está en el área de carga y espera un tiempo de 0.5 minutos antes de pasar a torneado.
5. Pieza esta en torneado y espera un tiempo de 5.2 minutos antes de pasar a Fresado.
6. Pieza esta en fresado y espera un tiempo de 9.17 minutos antes de pasar a taladrado.
7. Pieza está en taladrado y espera un tiempo de 1.6 minutos antes de pasar a rectificado.
8. Pieza esta en rectificado y espera un tiempo de 2.85 minutos antes de pasar a área de descarga.
9. Pieza está en área de descarga y espera un tiempo de 0.5 minutos antes de pasar a banda 2.
10. Pieza esta en banda 2 y pasa al área de inspección.
11. Pieza está en el área de inspección un tiempo exponencial de 3 minutos y sale del sistema.

- 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:**

Primeramente, para la forma manual realizamos el calculo de piezas que deberían pasar al sistema en un tiempo de 1000 horas y viendo que el horno recibe 10 piezas cada 100 min realizando una regla de 3 podemos decir que:

$$1000 \text{ horas} \times 60 \text{ minutos} = 60000 \text{ minutos}$$

Ahora realizamos una regla de 3 simple para poder saber el numero aproximado de motores en un tiempo de 60000 minutos.

$$\frac{100 \text{ minutos}}{60000 \text{ minutos}} \quad \frac{10 \text{ piezas}}{X} \cong 6000 \text{ piezas}$$

Como se puede ver en los cálculos aproximadamente deberían de salir 6000 piezas en un tiempo de 1000 horas.

Pero también debemos tomar en cuenta el tiempo que tarda la grúa en llenar y descargar las 10 piezas del horno por lo que al hacer cálculos podemos decir que la gura tarda un promedio de 0.4 minutos en ir al horno y a la banda tarda un promedio de 0.6 minutos y tomando en cuenta que hay que ir y venir por pieza multiplicamos por 20 ese tiempo y tenemos que.

$$\frac{25 \text{ pies}}{10 \text{ pies}} \quad \frac{1 \text{ minuto}}{X} \cong 0.4 \text{ minutos} \times 20 = 8 \text{ minutos}$$

$$\frac{25 \text{ pies}}{15 \text{ pies}} \quad \frac{1 \text{ minuto}}{X} \cong 0.6 \text{ minutos} \times 20 = 12 \text{ minutos}$$

Entonces el tiempo total que tarda el horno en sacar piezas es de 120min. Y si hacemos cálculos tenemos que aproximada mente el horno produce 5000 piezas en un tiempo de 1000 horas.

$$\frac{120 \text{ minutos}}{60000 \text{ minutos}} \quad \frac{10 \text{ piezas}}{X} \cong 5000 \text{ piezas}$$

Ahora también pondremos un ejemplo de un tiempo de simulación de 10 horas que son 600 minutos para poder analizar la cantidad de piezas que en teoría se deberían de producir.

$$\frac{120 \text{ minutos}}{600 \text{ minutos}} \quad \frac{10 \text{ piezas}}{X} \cong 50 \text{ piezas}$$

Pro model:

Para una simulación de 1000 hora vemos las siguientes estadísticas.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.975,00	12,06	0,00	4.975,00	4.975,00	2.482,46
Piezas Carga	9.957,00	6,03	0,00	8,00	3,00	2,87
horno vacio	997,00	60,14	0,00	1,00	0,00	0,07

Podemos ver que el total de piezas que se procesan es de 4975 piezas y la cantidad de piezas todavía en sistema es de:

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	12.070,00	3.570,88	7.080,00	7.080,00	0,36
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	2,86	8,00	3,00	0,00
Torneado	1,00	4.977,00	0,73	1,00	1,00	72,51
Fresado	1,00	4.976,00	0,78	1,00	1,00	77,76
Taladro	1,00	4.975,00	0,14	1,00	0,00	13,93
Rectificado	1,00	4.975,00	0,24	1,00	0,00	23,63
Area descarga	999.999,00	4.975,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.975,00	0,09	2,00	0,00	0,39
Inspeccion	1,00	4.975,00	0,25	1,00	0,00	25,03

La cantidad de piezas en el área de carga, el área de torneado y fresado es de 3, 1 y 1 respectivamente por lo que la cantidad de piezas procesadas es de 4980 piezas aproximadamente lo que coincide con los cálculos manuales.

Para una simulación de 10 horas vemos que las estadísticas son las siguientes.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	99,00	6,06	0,00	10,00	1,00	9,15
Piezas totales	40,00	14,55	0,00	40,00	40,00	16,18
Piezas Carga	90,00	6,66	0,00	8,00	6,00	2,35
horno vacio	9,00	54,35	0,00	1,00	0,00	0,07

Como se puede ver el numero de piezas que salieron es de las 40 piezas y contando las piezas que se encuentran en sistema tenemos que:

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	134,00	36,36	84,00	84,00	0,00
Horno	10,00	50,00	9,10	10,00	1,00	90,99
Banda 1	999.999,00	49,00	0,08	1,00	1,00	0,33
Area carga	999.999,00	48,00	2,35	8,00	6,00	0,00
Torneado	1,00	42,00	0,60	1,00	1,00	59,65
Fresado	1,00	41,00	0,63	1,00	1,00	63,07
Taladro	1,00	40,00	0,11	1,00	0,00	11,20
Rectificado	1,00	40,00	0,19	1,00	0,00	19,00
Area descarga	999.999,00	40,00	0,03	1,00	0,00	0,00
Banda 2	999.999,00	40,00	0,07	1,00	0,00	0,29
Inspeccion	1,00	40,00	0,19	1,00	0,00	19,20

En el horno hay una pieza en la banda hay 1 pieza en el área de carga hay 6 y en el ara de torneado y fresado hay 1 y 1 respectivamente por lo que podemos decir que la cantidad procesado por el horno es de 50 piezas lo que coincide con la prueba manual.

Java:

Para una simulación de 1000 hora vemos las siguientes estadísticas.

Descripción	Valor
Piezas totales producidas	4985
Piezas en almacén	7063
Piezas en horno	10
Piezas en carga	3
Piezas en torneado	1
Piezas en fresado	1
Piezas en taladro	0
Piezas en rectificado	0
Piezas en descarga	0
Piezas en inspección	0

Vemos que las piezas totales producidas son de 4985 piezas, en el horno actualmente hay 10 y en las otras áreas tenemos que hay 3 en el área de carga, 1 en torneado y 1 en el área de fresado por lo que podemos decir que el total de piezas procesadas es de 4990 piezas.

Para una simulación de 10 hora vemos las siguientes estadísticas.

Descripción	Valor
Piezas totales producidas	35
Piezas en almacén	70
Piezas en horno	10
Piezas en carga	3
Piezas en torneado	1
Piezas en fresado	1
Piezas en taladro	0
Piezas en rectificado	0
Piezas en descarga	0
Piezas en inspección	0

Vemos que las piezas totales producidas son de 35 piezas, en el horno actualmente hay 10 y en las otras áreas tenemos que hay 3 en el área de carga, 1 en torneado y 1 en el área de fresado por lo que podemos decir que el total de piezas procesadas es de 40 piezas.

Tabla de comparación:

	Manual	Pro Model	Java
1000 horas	5000 piezas	4975 piezas	4985 piezas
10 horas	50 piezas	40 piezas	35 piezas

Como se puede ver en la tabla podemos decir que los modelos si siguen lo planteado por el ejercicio con respecto a la producción de piezas debido a que los cálculos de la simulación en las diferentes modalidades dan resultados similares hallando en un rango de 10 piezas, también se puede ver que la diferencia en la producción de piezas es debido al tiempo que el horno espera para llenar las 10 piezas dentro de el donde entra mas grande es el tiempo de simulación ese tiempo de espera aumenta considerablemente reduciendo el procesamiento de piezas respectivamente.

Experimentación:

Pro Model:

Para la experimentación probaremos los resultados arrojados por Sin Runner para la consigna de maximización de producción y un porcentaje de utilización lo mas cercano al 95% para el área considerada cuello de botella que en este caso es el área de fresado.

Experiment	Objective Function	Fresado: % Utilization	Pieza: Total Exits	Mtorneado	Mfresado	Mtaladro	Mrectificado
4	5052.758	77.758	4975.000	4.000	1.000	3.000	5.000
6	5052.758	77.758	4975.000	3.000	1.000	4.000	5.000
11	5052.758	77.758	4975.000	3.000	1.000	5.000	2.000
15	5052.758	77.758	4975.000	3.000	1.000	4.000	3.000
22	5052.758	77.758	4975.000	4.000	1.000	5.000	5.000
23	5052.758	77.758	4975.000	1.000	1.000	1.000	1.000
13	5019.046	39.046	4980.000	3.000	2.000	1.000	2.000
5	5017.585	38.585	4979.000	2.000	2.000	3.000	2.000
12	5015.908	38.908	4977.000	1.000	2.000	1.000	3.000
25	5006.347	26.347	4980.000	2.000	3.000	1.000	1.000
1	5005.822	25.822	4980.000	3.000	3.000	2.000	1.000
3	5000.230	20.230	4980.000	2.000	4.000	1.000	2.000
9	5000.197	20.197	4980.000	4.000	4.000	1.000	2.000
2	5000.197	20.197	4980.000	5.000	4.000	1.000	3.000
19	4999.823	19.823	4980.000	3.000	4.000	1.000	2.000
8	4999.412	19.412	4980.000	2.000	4.000	5.000	5.000
14	4999.348	19.348	4980.000	3.000	4.000	3.000	3.000

- **Prueba 1:**

Probando el modelo con 4 torneado, 4 fresado, 1 taladrado y 2 rectificado tenemos los siguientes resultados:

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.980,00	12,05	0,00	4.980,00	4.980,00	2.484,82

Vemos que el número de piezas producidas para esa cantidad de máquinas en cada área es de 4980 piezas en todo el proceso.

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	12.069,00	3.566,17	7.079,00	7.079,00	0,36
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	0,16	2,00	0,00	0,00
Torneado	4,00	4.980,00	0,66	4,00	0,00	16,41
Fresado	4,00	4.980,00	0,81	4,00	0,00	20,20
Taladro	1,00	4.980,00	0,15	1,00	0,00	14,54
Rectificado	2,00	4.980,00	0,24	2,00	0,00	12,13
Area descarga	999.999,00	4.980,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.980,00	0,45	8,00	0,00	1,95
Inspeccion	1,00	4.980,00	0,25	1,00	0,00	24,77

Como se puede ver las principales áreas de producción tienen un porcentaje de utilización de un 16.41% para el área de torneado, 20.20% para el área de fresado, 14.54% para el área de taladrado y un 12.13% para el área de rectificado y un 24.77% para el área de inspección.

- **Prueba 2:**

Probando el modelo con 3 torneado, 2 fresado, 1 taladrado y 2 rectificado tenemos los siguientes resultados:

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.980,00	12,05	0,00	4.980,00	4.980,00	2.484,26

Vemos que el número de piezas producidas para esa cantidad de máquinas en cada área es de 4980 piezas en todo el proceso.

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	11.962,00	3.500,47	6.972,00	6.972,00	0,35
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	0,69	5,00	0,00	0,00
Torneado	3,00	4.980,00	0,97	3,00	0,00	32,49
Fresado	2,00	4.980,00	0,78	2,00	0,00	39,05
Taladro	1,00	4.980,00	0,14	1,00	0,00	13,72
Rectificado	2,00	4.980,00	0,24	2,00	0,00	12,10
Area descarga	999.999,00	4.980,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.980,00	0,21	5,00	0,00	0,88
Inspeccion	1,00	4.980,00	0,24	1,00	0,00	24,45

Como se puede ver las principales áreas de producción tienen un porcentaje de utilización de un 32.49% para el área de torneado, 39.05% para el área de fresado, 13.12% para el área de taladrado y un 12.10% para el área de rectificado y un 24.45% para el área de inspección.

- **Prueba 3:**

Probando el modelo con 3 torneado, 1 fresado, 4 taladrado y 5 rectificado tenemos los siguientes resultados:

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.975,00	12,06	0,00	4.975,00	4.975,00	2.482,46

Vemos que el número de piezas producidas para esa cantidad de máquinas en cada área es de 4975 piezas en todo el proceso.

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	12.070,00	3.570,88	7.080,00	7.080,00	0,36
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	1,61	6,00	1,00	0,00
Torneado	3,00	4.979,00	1,98	3,00	3,00	65,94
Fresado	1,00	4.976,00	0,78	1,00	1,00	77,76
Taladro	4,00	4.975,00	0,14	1,00	0,00	3,45
Rectificado	5,00	4.975,00	0,24	1,00	0,00	4,73
Area descarga	999.999,00	4.975,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.975,00	0,09	2,00	0,00	0,39
Inspeccion	1,00	4.975,00	0,25	1,00	0,00	25,03

Como se puede ver las principales áreas de producción tienen un porcentaje de utilización de un 65.94% para el área de torneado, 77.76% para el área de fresado, 3.45% para el área de taladrado y un 4.73% para el área de rectificado.

- **Tabla de comparación:**

Para la tabla de comparación se realizarán las comparaciones de los resultados para las 3 pruebas.

(To, Fre, Ta, Rec)	Producción	Torneado %	Fresado %	Taladrado %	Rectificado %
Prueba 1	4980	16.41%	20.20%	14.54%	12.13%
Prueba 2	4980	32.49%	39.05%	13.12%	12.10%
Prueba 3	4975	65.94%	77.76%	3.45%	4.73%

Como se puede ver en las pruebas realizadas al momento de aumentar el numero de maquinas de fresado podemos ver que el porcentaje de utilización reduce cada vez mas y se aleja del 95% de utilización que se busca.

También notar que la cantidad de piezas producidas no cambia mucho debido a que como la cantidad de piezas entran al proceso de 10 en 10 no importa cuantas maquinas tengamos la producción no aumentara.

Java:

Para java igual probaremos los datos arrojados por la herramienta de experimentos Sin Runner de Pro Model

- **Prueba 1:**

Probando el modelo con 4 torneado, 4 fresado, 1 taladrado y 2 rectificado tenemos los siguientes resultados:

Descripción	Valor
Piezas totales producidas	4982
Piezas en almacén	7007
Piezas en horno	10
Piezas en carga	0
Piezas en torneado	3
Piezas en fresado	2
Piezas en taladro	1
Piezas en rectificado	0
Piezas en descarga	1
Piezas en inspección	1
Porcentajes de utilización de las locaciones	
% Utilización horno	93.51%
% Utilización área de carga	2.07%
% Utilización de torneado	86.08%
% Utilización de fresado	43.66%
% Utilización de taladro	14.97%
% Utilización de rectificado	12.47%
% Utilización de área de descarga	0.15%
% Utilización de inspección	30.38%

Tenemos que la cantidad de piezas producidas es de 4982 y los porcentajes de utilización de las máquinas de las áreas son de un 93.51% para el horno, 2.07% para el área de carga, para el torneado un 86.08%, para el fresado un 43.66%, para el taladro un 14.97%, para el rectificado un 12.47% del tiempo, para el área de descarga un 0.15% y un 30% para el área de inspección.

- **Prueba 2:**

Probando el modelo con 3 torneado, 2 fresado, 1 taladrado y 2 rectificado tenemos los siguientes resultados:

Descripción	Valor
Piezas totales producidas	4982
Piezas en almacén	6922
Piezas en horno	10
Piezas en carga	3
Piezas en torneado	3
Piezas en fresado	2
Piezas en taladro	0
Piezas en rectificado	0
Piezas en descarga	0
Piezas en inspección	0
Porcentajes de utilización de las locaciones	
% Utilización horno	93.51%
% Utilización área de carga	3.15%
% Utilización de torneado	90.48%
% Utilización de fresado	87.3%
% Utilización de taladro	14.96%
% Utilización de rectificado	12.47%
% Utilización de área de descarga	0.06%
% Utilización de inspeccion	29.6%

Tenemos que la cantidad de piezas producidas es de 4982 y los porcentajes de utilización de las máquinas de las áreas son de un 93.51% para el horno, 3.15% para el área de carga, para el torneado un 90.48%, para el fresado un 87.3%, para el taladro un 14.96%, para el rectificado un 12.47% del tiempo, para el área de descarga un 0.06% y un 29.6% para el área de inspección.

- **Prueba 3:**

Probando el modelo con 3 torneado, 1 fresado, 4 taladrado y 5 rectificado tenemos los siguientes resultados:

Descripción	Valor
Piezas totales producidas	4978
Piezas en almacén	7241
Piezas en horno	10
Piezas en carga	0
Piezas en torneado	2
Piezas en fresado	0
Piezas en taladro	0
Piezas en rectificado	0
Piezas en descarga	0
Piezas en inspección	0
Porcentajes de utilización de las locaciones	
% Utilización horno	93.51%
% Utilización área de carga	3.14%
% Utilización de torneado	90.38%
% Utilización de fresado	74.74%
% Utilización de taladro	3.74%
% Utilización de rectificado	4.98%
% Utilización de área de descarga	0.07%
% Utilización de inspeccion	29.9%

Tenemos que la cantidad de piezas producidas es de 4978 y los porcentajes de utilización de las máquinas de las áreas son de un 93.51% para el horno, 3.14% para el área de carga, para el torneado un 90.38%, para el fresado un 74.74%, para el taladro un 3.74%, para el rectificado un 4.98% del tiempo, para el área de descarga un 0.07% y un 29.9% para el área de inspección.

- **Tabla de comparación:**

Para la tabla de comparación se realizarán las comparaciones de los resultados para las 3 pruebas.

(To, Fre, Ta, Rec)	Producción	Torneado %	Fresado %	Taladrado %	Rectificado %
Prueba 1	4982	86.08%	43.66%	14.97%	12.47%
Prueba 2	4982	90.48%	87.3%	14.96%	12.47%
Prueba 3	4978	90.38%	74.74%	3.74%	4.98%

Como se puede observar en la tabla que compara los experimentos podemos ver que entre mas maquinas se aumentan es menor el tiempo de utilización también resaltar que de igual forma las piezas producidas son similares debido a que no importa la maquinas que haya si el horno solo produce 10 la producción está limitada.

Interpretación:

Pro Model:

Primeramente, nos encargamos de mostrar el número de piezas producidas por la celda de producción para poder saber la cantidad de piezas.

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.975,00	12,06	0,00	4.975,00	4.975,00	2.482,46
Piezas Carga	9.957,00	6,03	0,00	8,00	3,00	2,87
horno vacio	997,00	60,14	0,00	1,00	0,00	0,07

Como se puede ver la cantidad de piezas totales producidas por la celda es de 4975 piezas y también en el horno se están procesando 10 piezas actualmente y en el almacén hay 6130 piezas actualmente.

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	12.070,00	3.570,88	7.080,00	7.080,00	0,36
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	2,86	8,00	3,00	0,00
Torneado	1,00	4.977,00	0,73	1,00	1,00	72,51
Fresado	1,00	4.976,00	0,78	1,00	1,00	77,76
Taladro	1,00	4.975,00	0,14	1,00	0,00	13,93
Rectificado	1,00	4.975,00	0,24	1,00	0,00	23,63
Area descarga	999.999,00	4.975,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.975,00	0,09	2,00	0,00	0,39
Inspeccion	1,00	4.975,00	0,25	1,00	0,00	25,03

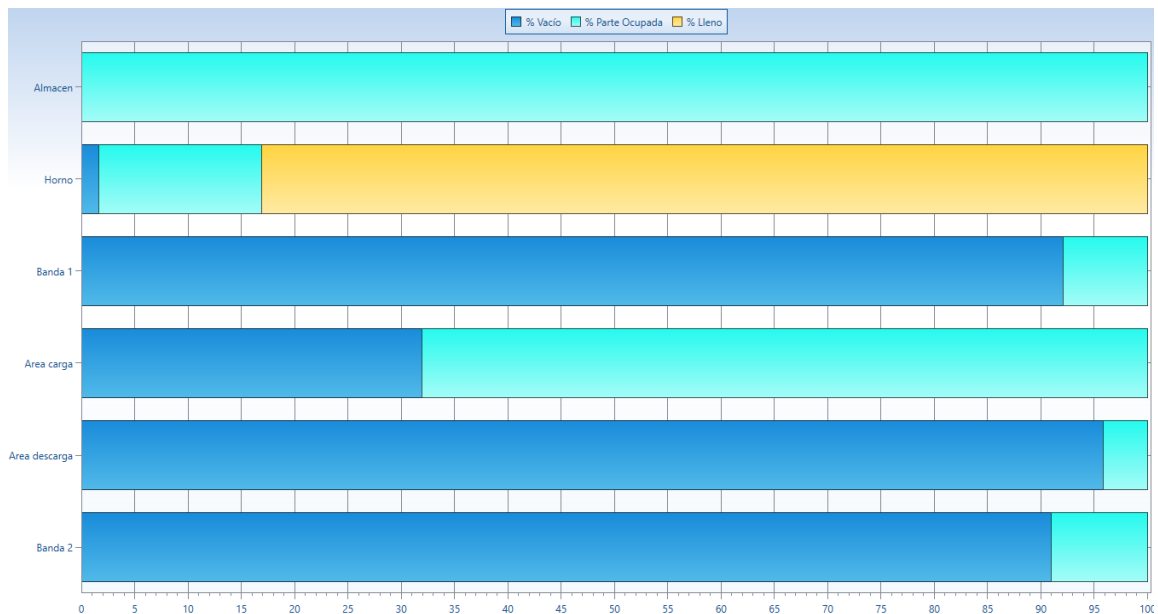
También como se puede ver los porcentajes de utilización de las máquinas de cada área de procesamiento es de 72.51% para el área de torneado, 77.76% para el área de fresado, 13.93%

para el área de taladrado y 23.63% para el área de rectificado y un 25.03% para el área de inspección.

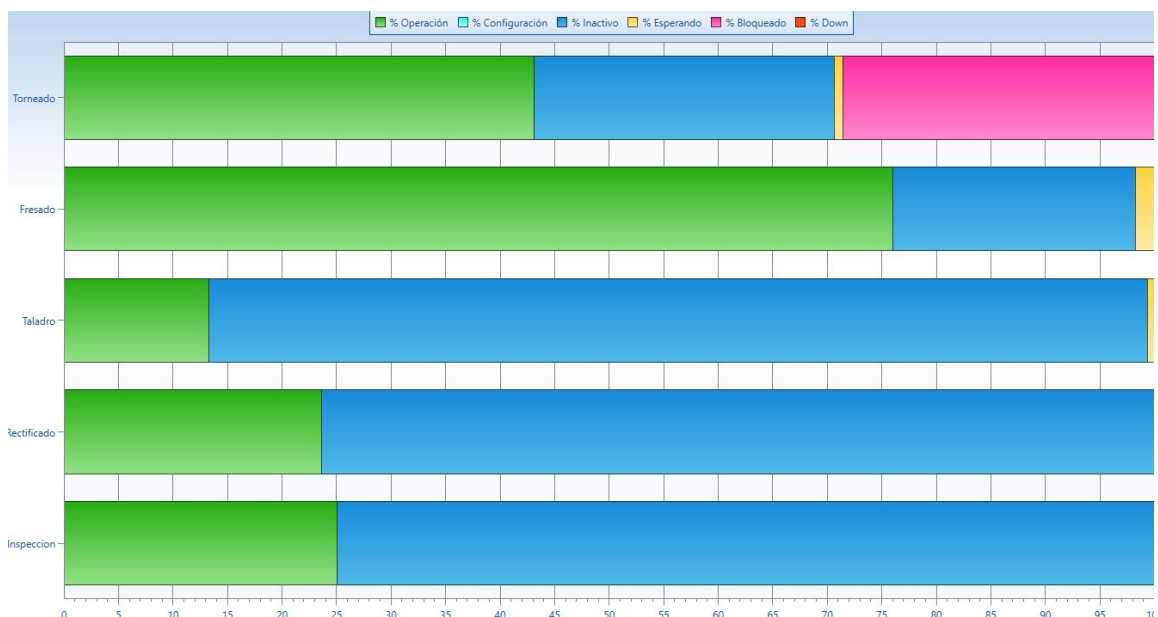
Notar que en el área de carga existen actualmente 3 piezas y en las áreas de torneado y fresado 1 y 1 respectivamente y viendo la producción de 4975 piezas podemos decir que si se cumple lo que las piezas entran de 10 en 10 a la celda.

Graficas Pro Model:

En las graficas de las locaciones podemos observar que:

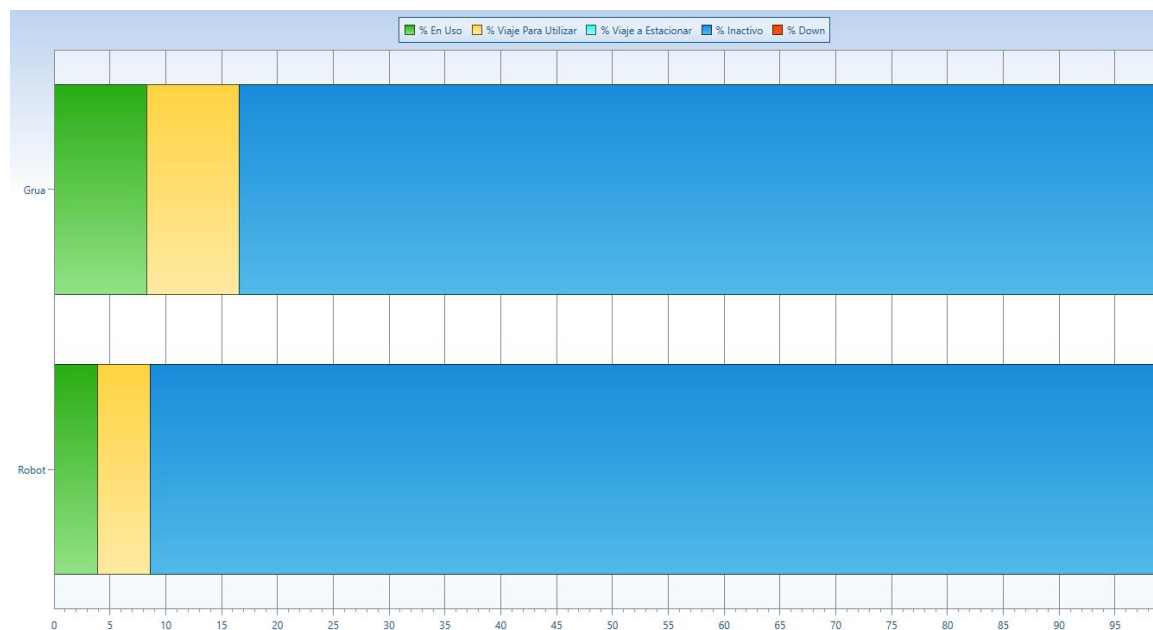


Como se puede ver el área de almacén anda ocupada el 100% del tiempo el horno esta ocupado del 15.27% y también siempre anda lleno el 83.07% del tiempo y en especial viendo que las áreas después de el horno como la banda 1 están vacías el 92.06% del tiempo, el área de carga el 31.91%, el área de descarga el 95.85% y la banda 2 es de 90.98% debido a que esperan a que las piezas salgan del horno.



También viendo la grafica de barras de las áreas de procesamiento de piezas podemos ver que en el área de torneado tiene un bloqueo del 28.55% del tiempo lo que quiere decir es que en el área de procesamiento siguiente que es el área de fresado tarda en procesar su pieza por lo que tiene que esperar a que se termine de procesar esa pieza para poder entrar al fresado.

También vemos que tienen un porcentaje de inactividad debido a que esperan a las piezas del área de fresado y también a que las piezas entran de 10 en 10 por lo que tienen que esperar al horno para poder procesarlos.



Vemos que los porcentajes de utilización de la grúa son del 8.31% del tiempo y del robot es de 3.91% del tiempo, también notar que solo se utilizan al mover las piezas por lo que andan inactivos la mayor parte del tiempo.

Java:

Primeramente, nos encargamos de mostrar el número de piezas producidas por la celda de producción para poder saber la cantidad de piezas.

Descripción	Valor
Piezas totales producidas	4980
Piezas en almacén	7344
Piezas en horno	10
Piezas en carga	0
Piezas en torneado	0
Piezas en fresado	0
Piezas en taladro	0
Piezas en rectificado	0
Piezas en descarga	0
Piezas en inspección	0

Vemos que la cantidad producida de piezas es de 4980 la cantidad actual de piezas en el almacén es de 7344 piezas y vemos que actualmente en el horno se encuentra procesando 10 piezas y las demás maquinas están vacías.

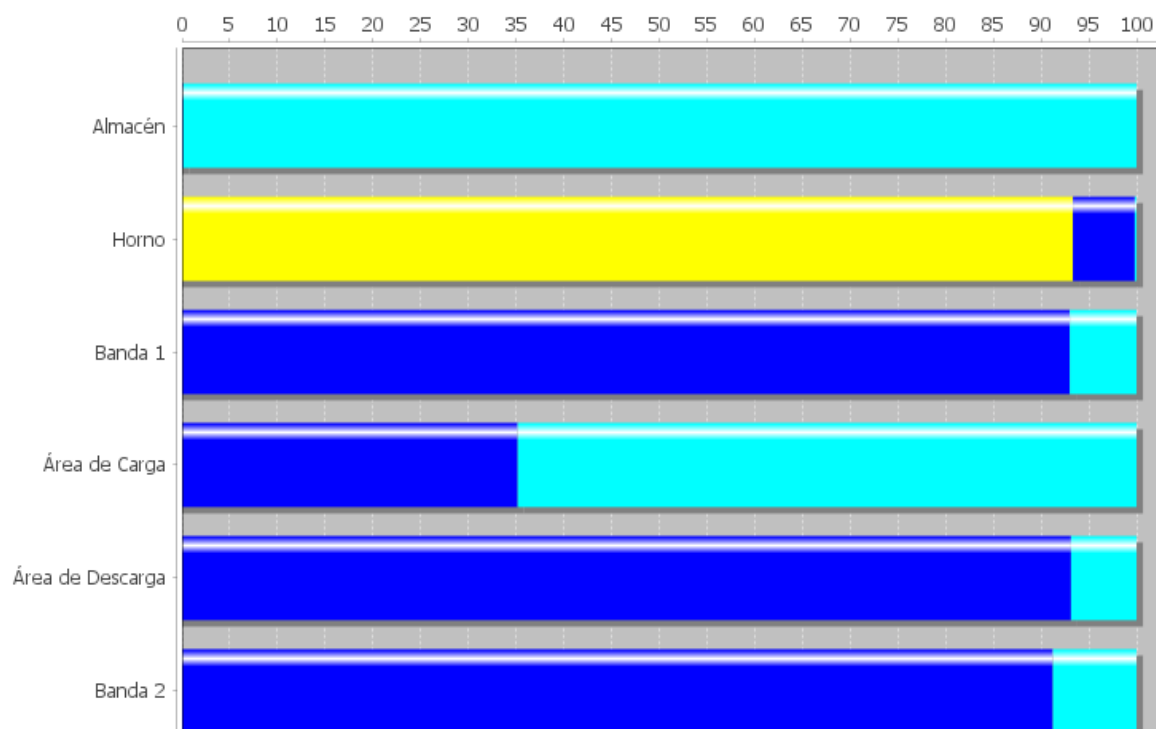
Porcentajes de utilizacion de las locaciones	
% Utilización horno	93.5%
% Utilización área de carga	3.15%
% Utilización de torneado	71.78%
% Utilización de fresado	74.77%
% Utilización de taladro	14.95%
% Utilización de rectificado	24.92%
% Utilización de área de descarga	0.07%
% Utilización de inspeccion	30.11%

Vemos los porcentajes de utilización de las locaciones como del horno que es de un 93.5% del tiempo, la del área de carga que es de 3.15% del tiempo, la de torneado que es del 71.78% del tiempo, la de fresado que es el 74.77% del tiempo, la del taladro que es del 14.95% del tiempo, la del rectificado que es del 24.92%, la del área de carga que es del 0.07% y la del área de inspección que es del 30.11% del tiempo.

Graficas java:

Porcentajes de áreas de gran capacidad:

Estas áreas de gran capacidad son las que no se pueden llenar en ningún momento como el almacén, la banda 1, el área de carga, el área de descarga y la banda 2.

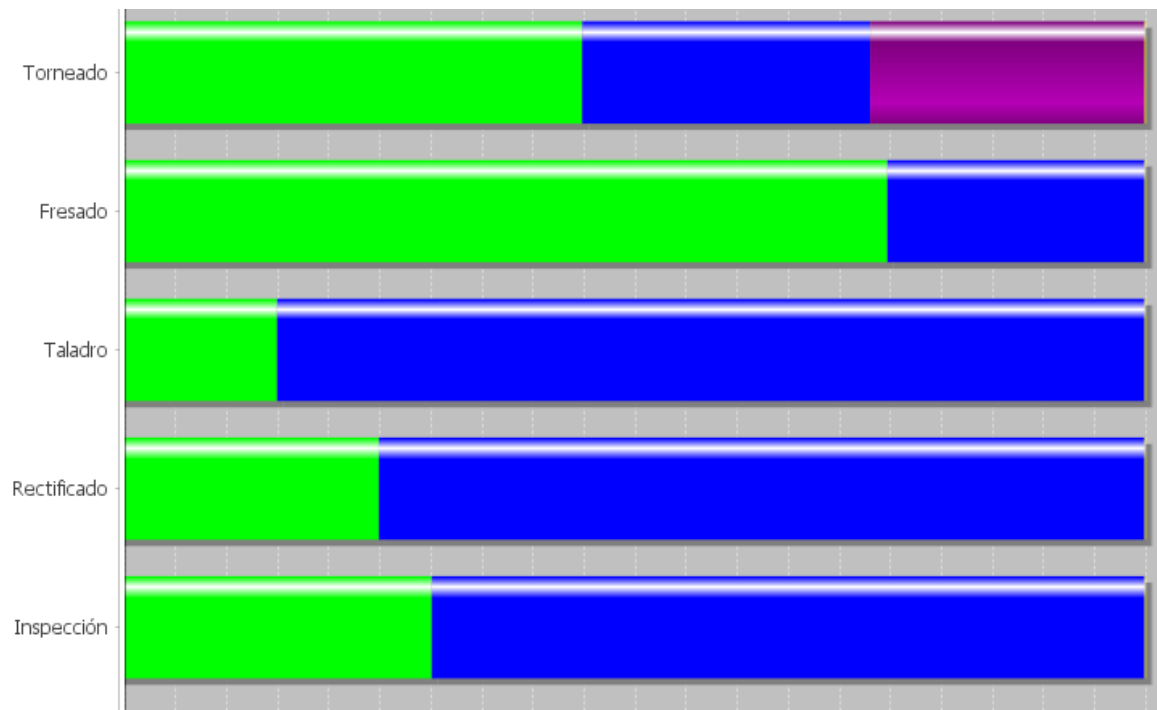


Porcentajes de areas de gran capacidad	
% Almacen lleno	0.0%
% Almacen vacio	0.1%
% Almacen ocupado	99.9%
% Horno lleno	93.33%
% Horno vacio	6.48%
% Horno ocupado	0.19%
% Banda 1 lleno	0.0%
% Banda 1 vacio	93.0%
% Banda 1 ocupado	7.0%
% Area de carga lleno	0.0%
% Area de carga vacio	35.16%
% Area de carga ocupado	64.84%
% Area de descarga lleno	0.0%
% Area de descarga vacio	93.13%
% Area de descarga ocupado	6.87%
% Banda 2 lleno	0.0%
% Banda 2 vacio	91.23%
% Banda 2 ocupado	8.77%

Vemos que el almacén anda ocupado el 99.9% el tiempo, el horno como siempre se debe llenar anda lleno el 93.33% del tiempo, de igual forma la banda 1 anda bacia el 93% del tiempo, el área de carga anda ocupada el 64.84% del tiempo y vacía el 35.16% del tiempo, de igual forma el área de descarga anda vacía el 93.13% del tiempo y ocupada el 6.87% del tiempo y la banda 2 anda vacía el 91.23% del tiempo.

Porcentajes de áreas de capacidad limitada:

Estas áreas de capacidad limitada son las que se pueden llenar en cualquier momento como el torneado, el fresado, el taladro, el rectificado y el área de inspección.



Porcentajes de areas de capacidad limitada	
% Operacion de torneado	44.82%
% Inactividad de torneado	28.19%
% Bloqueo de torneado	26.89%
% espera de torneado	0.1%
% Operacion de fresado	74.7%
% Inactividad de fresado	25.2%
% Bloqueo de fresado	0.0%
% espera de fresado	0.1%
% Operacion de taladro	14.94%
% Inactividad de taladro	84.96%
% Bloqueo de taladro	0.0%
% espera de taladro	0.1%
% Operacion de rectificado	24.9%
% Inactividad de rectificado	75.0%
% Bloqueo de rectificado	0.0%
% espera de rectificado	0.1%
% Operacion de inspeccion	30.08%
% Inactividad de inspeccion	69.82%
% Bloqueo de inspeccion	0.0%
% espera de inspeccion	0.1%

Como se puede ver tenemos que el área de torneado se bloquea el 26.98% del tiempo, y opera el 44.82% del tiempo, el área de fresado no se bloquea, pero está operando el 74.7% del tiempo, el área de talador opera el 14.94 % del tiempo y el área de rectificado el 24.9% del tiempo y por ultimo tenemos que el área de inspección está en operación el 30% del tiempo y esta inactivo el 69.82% 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\proyecto2\fabricasimrun.mod          *
*                                                                                             *
*****

Unidades de Tiempo:              Minutos
Unidades de Distancia:           Pies

*****

*                               Locaciones                                                    *
*****

Nombre      Cap      Unidades Estadist      Reglas      Costos
-----
Almacen      INF      1      Series de tiempo Más Tiempo, ,
Horno        10      1      Series de tiempo Más Tiempo, ,
Banda_1      INFINITE  1      Series de tiempo Más Tiempo, FIFO,
Area_carga   INF      1      Series de tiempo Más Tiempo, ,
Torneado     Mtorneado  1      Series de tiempo Más Tiempo, , Primera
Fresado      Mfresado  1      Series de tiempo Más Tiempo, ,
Taladro      Mtaladro  1      Series de tiempo Más Tiempo, ,
Rectificado  Mrectificado 1      Series de tiempo Más Tiempo, ,
Area_descarga INF      1      Series de tiempo Más Tiempo, ,
Banda_2      INFINITE  1      Series de tiempo Más Tiempo, FIFO,
Inspeccion   1      1      Series de tiempo Más Tiempo, ,

*                               Entidades                                                       *
*****

Nombre      Velocidad (Ppm) Estadist      Costos
-----
Pieza        150      Series de tiempo

*****

*                               Redes de Ruta                                                  *
*****

Nombre      Tipo      T/V      Desde      Hasta      BI      Distancia/Tiempo Factor de Velocidad
-----
Ruta_grua   Sobrepasar   Velocidad & Distancia N1      N2      Bi      10      1
              N2      N3      Bi      15      1
Ruta_robot  Sobrepasar   Velocidad & Distancia N1      N2      Bi      18      1
              N1      N3      Bi      13      1
              N3      N4      Bi      1      1
              N3      N5      Bi      1      1
              N1      N6      Bi      1      1
              N1      N7      Bi      1      1
              N2      N8      Bi      1      1
              N2      N9      Bi      1      1

```


Pieza	Taladro	Wait 1.6 min	1	Pieza	Rectificado	FIRST 1	Move With Robot Then Free
Pieza	Rectificado	Wait 2.85 min	1	Pieza	Area_descarga	FIRST 1	Move With Robot Then Free
Pieza	Area_descarga	Wait 0.5 min	1	Pieza	Banda_2	FIRST 1	
Pieza	Banda_2		1	Pieza	Inspeccion	FIRST 1	
Pieza	Inspeccion	Wait E(3) min	1	Pieza	EXIT	FIRST 1	Inc Piezas_totales ,1

```
*****
*                               Arribos                               *
*****
```

Entidad	Locación	Cant. por Arribo	Primera Vez	Ocurrencias	Frecuencia	Lógica
Pieza	Almacen	P(12)	0	INF	1hr	

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

ID	Tipo	Valor Inicial	Estadist
Piezas_en_horno	Integer	0	Series de tiempo
Piezas_totales	Integer	0	Series de tiempo
Piezas_Carga	Integer	0	Series de tiempo
horno_vacio	Integer	1	Series de tiempo

```
*****
*                               Macros                                   *
*****
```

ID	Texto
Mtorneado	1
Mfresado	1
Mtaladro	1
Mrectificado	1

Código de java:

Interfaz para recibir los datos del problema:

```
public class Ejer2UI extends javax.swing.JFrame {

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

    public Ejer2UI() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() { [...146 lines] }

    private void simularEjercicioA(java.awt.event.ActionEvent evt, JTextField simulacionHoras, JTextField replicas, JTextField delay, JTextField i
        try {
            int duracionHoras = Integer.parseInt(simulacionHoras.getText());
            int numReplicas = Integer.parseInt(replicas.getText());
            int delaySegundos = Integer.parseInt(delay.getText());
            int delayAnimacion = Integer.parseInt(animationDelay.getText()); // Obtención de delay de animación
            int torneado = Integer.parseInt(maquinasTorneo.getText());
            int fresado = Integer.parseInt(maquinasFresado.getText());
            int taladro = Integer.parseInt(maquinasTaladro.getText());
            int rectificado = Integer.parseInt(maquinasRectificado.getText());
            int mediaLleg = Integer.parseInt(mediaLlegada.getText());
            int tiempoInsp = Integer.parseInt(tiempoInspeccion.getText());

            // Ejecución del hilo o lógica de simulación con los datos
            Ejecucion2 hilo = new Ejecucion2(textAreaA, textAreaB, duracionHoras, numReplicas, delaySegundos, delayAnimacion, torneado, fresado,
            hilo.start();
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Por favor, introduce valores numéricos válidos.", "Error de entrada", JOptionPane.ERROR_MESSAGE).
        }
    }
}
```

En esta parte solo se reciben los datos del ejercicio, así como los tiempos de simulación y los tiempos de delay para reproducir la animación del ejercicio y también las distribuciones que utilizan para ser mas específico la cantidad y con esos datos manda a llamar al proceso de simulación que se inicializa como un hilo para poder hacerlo.

Entidad pieza:

```

...
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 double getTiempo4() {return tiempo4;}
public double getTiempo5() {return tiempo5;}
public double getTiempo6() {return tiempo6;}
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 tiempo3) {this.tiempo3 = tiempo3;}
public void setTiempo4(double tiempo4) {this.tiempo4 = tiempo4;}
public void setTiempo5(double tiempo5) {this.tiempo5 = tiempo5;}
public void setTiempo6(double tiempo6) {this.tiempo6 = tiempo6;}
public void setInAreal(boolean inAreal) {this.inAreal=inAreal;}
public void setInArea2(boolean inArea2) {this.inArea2=inArea2;}

```

La entidad es simple tiene unos métodos para poder darle valores de tiempo y que los guarde, así como también poder acceder a ellos y también tienen unos verificadores booleanos para partes específicas del problema.

Proceso de simulación:

```

public class Ejecucion2 extends Thread {
    private JTextArea textAreaA;
    private JTextArea textAreaB;
    private int duracion;
    private int replicas;
    private int delay;
    private int delayAnimacion;
    //Variables a recibir
    private int Mtorneado;
    private int Mfresado;
    private int Mtaladro;
    private int Mrectificado;
    private int mediaLlegadas;
    private int mediaInspect;
    private SimulacionPanel simPanel;

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

    public Ejecucion2(JTextArea textAreaA, JTextArea textAreaB, int duracion, int replicas, int delay, int delayAnimacion,
        int Mtorneado, int Mfresado, int Mtaladro, int Mrectificado, int llegadas, int inspect){ //constructor
        this.textAreaA = textAreaA;
        this.textAreaB = textAreaB;
        this.duracion = duracion;
        this.replicas = replicas;
        this.delay = delay;
        this.delayAnimacion = delayAnimacion;
        //Variables a recibir
        this.Mtorneado = Mtorneado;
        this.Mfresado = Mfresado;
        this.Mtaladro = Mtaladro;
        this.Mrectificado = Mrectificado;
        this.mediaLlegadas = llegadas;
        this.mediaInspect = inspect;
        initGrafico(); //iniciar grafica
    }
}

```

En esta parte del código es donde empieza todo porque es el constructor del proceso es se te encarga de recibir todos los datos enviados por la interfaz de usuario y guardarlos en sus variables respectivas y también manda a iniciar la simulación grafica.

```
private void initGrafico() {
    JFrame frame = new JFrame("Grafica de calda");
    simPanel = new SimulacionPanel();
    frame.setSize(900, 600);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.add(simPanel);
    frame.setVisible(true);
}

@Override
public void run() {
    //Locaciones
    List<Entidad> almacen = new ArrayList<>();
    List<Entidad> grua = new ArrayList<>();
    List<Entidad> horno = new ArrayList<>();
    List<Entidad> bandal = new ArrayList<>();
    List<Entidad> areaCarga = new ArrayList<>();
    List<Entidad> torneado = new ArrayList<>();
    List<Entidad> robot = new ArrayList<>();
    List<Entidad> fresado = new ArrayList<>();
    List<Entidad> taladro = new ArrayList<>();
    List<Entidad> rectificado = new ArrayList<>();
    List<Entidad> descarga = new ArrayList<>();
    List<Entidad> banda2 = new ArrayList<>();
    List<Entidad> inspeccion = new ArrayList<>();
    //Variables auxiliares
    double tiempoSalida = 0;
    double tiempoSalidaH = 0;
    double tiempoEntradaH = 0;
    double tiempoCarga = 0;
    boolean bloqueado = false;
    boolean bloqueado2 = false;
    boolean bloqueado3 = false;
}
```

En esta parte del código construimos la ventana que tendrá la simulación grafica con la función `initGrafico()`, y seguidamente procedemos a inicializar las variables que nos servirán como locaciones en este caso serán listas de entidades y también se inicializaran unas variables auxiliares para realizar cálculos y guardar resultados.

```
//Variables para datos
int piezasTotales = 0; //guarda las piezas totales
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
List<Double> porcentajeUti5 = new ArrayList<>(); //guardar porcentajes
double porUti5 = 0; //mostrar porcentaje al final
List<Double> porcentajeUti6 = new ArrayList<>(); //guardar porcentajes
double porUti6 = 0; //mostrar porcentaje al final
List<Double> porcentajeUti7 = new ArrayList<>(); //guardar porcentajes
double porUti7 = 0; //mostrar porcentaje al final
List<Double> porcentajeUti8 = new ArrayList<>(); //guardar porcentajes
double porUti8 = 0; //mostrar porcentaje al final
int contArea0V = 0;
int contArea1L = 0, contArea1V = 0;
int contArea2L = 0, contArea2V = 0;
int contArea3L = 0, contArea3V = 0;
//dispositivos con capacidad limitada
int contArea4L = 0, contArea4V = 0, contArea4B = 0;
int contArea5L = 0, contArea5V = 0, contArea5B = 0;
int contArea6L = 0, contArea6V = 0, contArea6B = 0;
int contArea7L = 0, contArea7V = 0, contArea7B = 0;
int contArea8L = 0, contArea8V = 0, contArea8B = 0;
```

Esas son todas las variables para guardar los resultados respectivos de la simulación, así como también variables auxiliares para realizar cálculos en la simulación.


```

int contAreaaL = 0, contAreaaV = 0, contAreaaB = 0;
for(int k = 1; k<= replicas; k++){
    for(int i = 1; i <= duracion; i++){//simular las horas
        arriboAlmacen(almacen);// Arribos por hora a almacen
        simPanel.llegadaPieza("almacen");
        //Simular entre hora
        if (i != duracion) { // 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
                //Guardar porcentajes
                porcentajeUti.add(dosDecimales((((double)horno.size())/10)*100));
                porcentajeUti2.add(dosDecimales((((double)areaCarga.size())/100)*100));
                porcentajeUti3.add(dosDecimales((((double)torneado.size())/Mtorneado)*100));
                porcentajeUti4.add(dosDecimales((((double)fresado.size())/Mfresado)*100));
                porcentajeUti5.add(dosDecimales((((double)taladro.size())/Mtaladro)*100));
                porcentajeUti6.add(dosDecimales((((double)rectificado.size())/Mrectificado)*100));
                porcentajeUti7.add(dosDecimales((((double)descarga.size())/100)*100));
                porcentajeUti8.add(dosDecimales((((double)inspeccion.size())/1)*100));
                //maquinas con gran capacidad
                if(horno.size() == 10){contAreaaL++;}
                if(areaCarga.size() == 100){contAreaa2L++;}
                if(descarga.size() == 100){contAreaa3L++;}
                if(almacen.isEmpty()){contAreaa0V++;}
                if(horno.isEmpty()){contAreaa1V++;}
                if(areaCarga.isEmpty()){contAreaa2V++;}
                if(descarga.isEmpty()){contAreaa3V++;}
                //Maquinas con capacidad limitada
                if(!torneado.isEmpty()){contAreaa4L++;}
                if(!fresado.isEmpty()){contAreaa5L++;}
                if(!taladro.isEmpty()){contAreaa6L++;}
            }
        }
    }
}

```

Seguidamente tenemos el inicio de la simulación con un bucle for para el numero de replicas y otro para simular las horas en este caso 100 horas, luego seguidamente como primer acto tenemos que realizar los arribos al almacén, eso es por hora luego simulamos el tiempo entre hora donde haremos los cálculos correspondientes a cada área, primero siempre actualizamos los datos de las variables de los resultados con los del tiempo actual.

```

//Area Inspeccion
if(!inspeccion.isEmpty()){ //inspeccion no esta vacio
    Entidad pieza = inspeccion.get(0); //guardamos la pieza
    if(pieza.getTiempo6() == tiempoActual){ //si es tiempo de salida
        pieza.setTiempo6(0); //reseteamos tiempo
        inspeccion.remove(pieza); //eliminamos de inspeccion
        simPanel.llegadaPieza("inspeccionVacio");
        textAreaA.append("Pieza salio de inspeccion\n");
        piezasTotales++; //aumentamos el contador de piezas totales
    }
}

//Area descarga
if(!descarga.isEmpty() && inspeccion.isEmpty()){//descarga no esta vacia y hay espacio en inspeccion
    for(int c = 0; c < descarga.size(); c++){ //Recorremos cada pieza
        Entidad pieza = descarga.get(c); //guardamos la pieza
        descarga.remove(pieza); //eliminamos de descarga
        banda2.add(pieza); //añadimos a banda 2
        simPanel.llegadaPieza("banda2");
        banda2.remove(pieza); //eliminamos de banda 2
        inspeccion.add(pieza); //añadimos a inspeccion
        if(inspeccion.size()==1){ //si inspeccion esta lleno
            simPanel.llegadaPieza("inspeccionLleno");
            procesarAreaIns(pieza, tiempoActual);//procesamos en inspeccion
            break; //detenemos el ciclo
        }
    }
}

if(descarga.isEmpty()){simPanel.llegadaPieza("descargaVacio");}
}

```

Tenemos primero el área de inspección donde si no esta vacía comparamos si es tiempo de que la pieza salga de inspección y si es así removemos de inspección y aumentamos el contador de piezas totales, luego tenemos el área de descarga que primero pregunta si hay piezas y luego si hay espacio en inspección para mandar las piezas una vez mandadas las piezas es cuando se le da el tiempo de operación en la siguiente área.

```
//Area rectificado
if(!rectificado.isEmpty()){ // taladro no esta vacio
    for(int r = 0; r < rectificado.size(); r++){ //recorremos cada pieza
        Entidad pieza = rectificado.get(r); //guardamos la pieza actual
        if(pieza.getTiempo4() == tiempoActual){ //es tiempo de salida de taladro
            pieza.setTiempo4(0); //recetamos tiempo salida
            textAreaA.append("Pieza salio de rectificado\n");
            rectificado.remove(pieza); //eliminamos de rectificado
            robot.add(pieza); //añadimos a robot
            simPanel.llegadaPieza("rectificadoVacio");
            simPanel.moverMaquina("robot", 310, 190);
            simPanel.llegadaPieza("robotEnd");
            retrasoGrafica(delayAnimacion);
            robot.remove(pieza); //eliminamos de robot
            descarga.add(pieza); //añadimos a descarga
            simPanel.moverMaquina("robot", 310, 330);
            simPanel.llegadaPieza("robotSal");
            simPanel.llegadaPieza("descargaLleno");
            procesarAreaD(pieza, tiempoActual); //procesamos en descarga
        }
    }
}
```

En el área de rectificado de igual forma preguntamos si esta no vacío luego recorremos cada pieza en caso de haber mas de una luego si su tiempo coincide sale del área y se dirige al área de descarga las partes sin comentarios están relacionadas a la animación donde pasamos los tiempos de delay y ubicaciones de movimiento.

Las demás áreas funcionan de forma similar por lo que las pondremos de forma seguida:

```
//Area taladro
if(!taladro.isEmpty()){ // taladro no esta vacio
    for(int ta = 0; ta < taladro.size(); ta++){ //recorremos cada pieza
        Entidad pieza = taladro.get(ta); //guardamos la pieza actual
        if(pieza.getTiempo3() == tiempoActual){ //es tiempo de salida de fresado
            if(rectificado.size() < Mrectificado){ //hay espacio en rectificado
                bloqueado3 = false; //desbloqueamos
                pieza.setTiempo3(0); //recetamos tiempo salida
                textAreaA.append("Pieza salio de taladro\n");
                taladro.remove(pieza); //eliminamos de taladro
                robot.add(pieza); //añadimos a robot
                simPanel.llegadaPieza("taladroVacio");
                simPanel.moverMaquina("robot", 310, 50);
                simPanel.llegadaPieza("robotEnd");
                retrasoGrafica(delayAnimacion);
                robot.remove(pieza); //eliminamos de robot
                rectificado.add(pieza); //añadimos a rectificado
                simPanel.moverMaquina("robot", 310, 190);
                simPanel.llegadaPieza("robotSal");
                simPanel.llegadaPieza("rectificadoLleno");
                procesarAreaR(pieza, tiempoActual); //procesamos en rectificado
            } else {
                bloqueado3 = true; //bloqueamos taladro
                textAreaA.append("Pieza bloqueada en taladrado\n");
                pieza.setTiempo3(dosDecimales(tiempoActual+0.01)); //actualizamos el tiempo
            }
        }
    }
}
```

```

//Area fresado
if(!fresado.isEmpty()){ // fresado no esta vacio
    for(int f = 0; f < fresado.size(); f++){ //recorremos cada pieza
        Entidad pieza = fresado.get(f); //guardamos la pieza actual
        if(pieza.getTiempo2() == tiempoActual){ //es tiempo de salida de torneado
            if(taladro.size() < Mtaladro){ //hay espacio en taladro
                bloqueado2 = false; //desbloqueamos
                pieza.setTiempo2(0); //receteamos tiempo salida
                textAreaA.append("Pieza salio de Fresado\n");
                fresado.remove(pieza); //eliminamos de fresado
                robot.add(pieza); //añadimos a robot
                simPanel.llegadaPieza("fresadoVacio");
                simPanel.moverMaquina("robot", 410, 50);
                simPanel.llegadaPieza("robotEnd");
                retrasoGrafica(delayAnimacion);
                robot.remove(pieza); //eliminamos de robot
                taladro.add(pieza); //añadimos a taladro
                simPanel.moverMaquina("robot", 310, 50);
                simPanel.llegadaPieza("robotSal");
                simPanel.llegadaPieza("taladroLleno");
                procesarAreaTa(pieza, tiempoActual); //procesamos en taladro
            }else{
                bloqueado2 = true; //bloqueamos fresado
                textAreaA.append("Pieza bloqueada en fresado\n");
                pieza.setTiempo2(dosDecimales(tiempoActual+0.01)); //actualizamos el tiempo
            }
        }
    }
}

```

Cabe resaltar que en estas áreas si no hay espacio en el área siguiente la pieza se considera bloqueada esos datos van para las variables de resultados.

```

//Area torneado
if(!torneado.isEmpty()){ // torneado no esta vacio
    for(int t = 0; t < torneado.size(); t++){ //recorremos cada pieza
        Entidad pieza = torneado.get(t); //guardamos la pieza actual
        if(pieza.getTiempo1() == tiempoActual){ //es tiempo de salida de carga
            if(fresado.size() < Mfresado){ //hay espacio en fresado
                bloqueado = false; //desbloqueamos
                pieza.setTiempo1(0); //receteamos tiempo salida
                textAreaA.append("Pieza salio de torneado\n");
                torneado.remove(pieza); //eliminamos de torneado
                robot.add(pieza); //añadimos a robot
                simPanel.llegadaPieza("torneadoVacio");
                simPanel.llegadaPieza("robotEnd");
                simPanel.moverMaquina("robot", 410, 190);
                retrasoGrafica(delayAnimacion);
                robot.remove(pieza); //eliminamos de robot
                fresado.add(pieza); //añadimos a fresado
                simPanel.moverMaquina("robot", 410, 50);
                simPanel.llegadaPieza("robotSal");
                simPanel.llegadaPieza("fresadoLleno");
                procesarAreaF(pieza, tiempoActual); //procesamos en fresado
            }else{
                bloqueado = true; //bloqueamos torneado
                textAreaA.append("Pieza bloqueada en torneado\n");
                pieza.setTiempo1(dosDecimales(tiempoActual+0.01)); //actualizamos el tiempo
            }
        }
    }
}

```

```

//Area carga
if(!areaCarga.isEmpty()){ // carga no esta vacio
    for(int c = 0; c < areaCarga.size(); c++){ //recorremos cada pieza
        Entidad pieza = areaCarga.get(c); //guardamos la pieza actual
        if(torneado.size() < Mtorneado){ //si torneado tiene espacio
            areaCarga.remove(pieza); //eliminamos de carga
            robot.add(pieza); //añadimos a robot
            simPanel.llegadaPieza("robotEnd");
            simPanel.moverMaquina("robot", 410, 330);
            retrasoGrafica(delayAnimacion);
            robot.remove(pieza); //removemos de robot
            torneado.add(pieza); //añadimos a torneado
            simPanel.moverMaquina("robot", 410, 190);
            simPanel.llegadaPieza("robotSal");
            simPanel.llegadaPieza("torneadoLleno");
            procesarAreaT(pieza, tiempoActual); //procesamos en torneado
        }else{
            break; //si no hay espacio paramos la revision de piezas
        }
    }
}
}
}

//Horno
if(horno.size() == 10 && tiempoActual == tiempoEntradaH){ //tiempo es el tiempo de entrada y horno lleno
    textAreaA.append("El horno se llenó \n");
    tiempoSalida = procesar(tiempoActual); //tiempo de proceso del horno
    tiempoSalidaH = dosDecimales(tiempoSalida + 0.2); //tiempo donde el horno estara vacio
    textAreaA.append("El horno termina en la hora: " + tiempoSalida + "\n");
    textAreaA.append("El horno esta vacio en la hora: " + tiempoSalidaH + "\n");
}

//Primer llenado
if(horno.size() < 10 && tiempoEntradaH == 0){ // Horno tiene espacio?
    for (int a = 0; a < almacen.size(); a++) { // Recorre desde 0 hasta almacen.size() - 1
        retrasoGrafica(delayAnimacion);
        Entidad pieza = almacen.get(a); // Obtén el elemento actual
        almacen.remove(pieza); // Elimina del almacen
        grua.add(pieza); // Añade a la grua
        simPanel.llegadaPieza("gruaEnd");
        simPanel.moverMaquina("grua", 720, 350);
        retrasoGrafica(delayAnimacion);
        grua.remove(pieza); // Elimina de la grua
        horno.add(pieza); // Añade al horno
        simPanel.llegadaPieza("gruaSal");
        simPanel.moverMaquina("grua", 720, 480);
        if (horno.size() == 10) { // Verifica si el horno se llenó
            simPanel.llegadaPieza("hornoLleno");
            tiempoEntradaH = dosDecimales(tiempoActual + 0.13); //actualizamos tiempo entrada
            break; // Detén el bucle si el horno está lleno
        }
        a--; // Retrocede el índice para compensar la eliminación del elemento
    }
}
}

```

Aquí es donde tenemos el funcionamiento del horno donde si hay espacio en el horno entonces se mandan las 10 piezas si no esperan de igual forma se utiliza la grúa para dicho propósito y si el horno se llena se detiene el proceso del almacén para evitar fallos al ejecutar.

```

        j = j.add(new BigDecimal("0.01")).setScale(2, RoundingMode.HALF_UP); //j++
        retrasoGrafica(delay);
    }
}
if(i==duracion){simPanel.actualizarEstado(almacen, horno, areaCarga, torneado, fresado, taladro, rectificado, piezasTotales);}
}
}

//resultados
porUti = dosDecimales(calcularPorUti(porcentajeUti));
porUti2 = dosDecimales(calcularPorUti(porcentajeUti2));
porUti3 = dosDecimales(calcularPorUti(porcentajeUti3));
porUti4 = dosDecimales(calcularPorUti(porcentajeUti4));
porUti5 = dosDecimales(calcularPorUti(porcentajeUti5));
porUti6 = dosDecimales(calcularPorUti(porcentajeUti6));
porUti7 = dosDecimales(calcularPorUti(porcentajeUti7));
porUti8 = dosDecimales(calcularPorUti(porcentajeUti8));

double area0V = dosDecimales(((double)contArea0V/(duracion*100))*100);
double area1L = dosDecimales(((double)contArea1L/(duracion*100))*100);
double area1V = dosDecimales(((double)contArea1V/(duracion*100))*100);
double area2L = dosDecimales(((double)contArea2L/(duracion*100))*100);
double area2V = dosDecimales(((double)contArea2V/(duracion*100))*100);
double area3L = dosDecimales(((double)contArea3L/(duracion*100))*100);
double area3V = dosDecimales(((double)contArea3V/(duracion*100))*100);

BarChartExample example = new BarChartExample(
    0.0, area0V, // Almacén
    area1L, area1V, // Horno
    0.0, 93.0, // Banda 1
    area2L, area2V, // Área de Carga
    area3L, area3V, // Área de Descarga
    0.0, 91.23 // Banda 2
);

```

Al terminar la simulación mandamos los datos a imprimir en las tablas y en las graficas para su posterior análisis.

```

public void arriboAlmacen (List<Entidad> locacion){
    PoissonDistribution llegada = new PoissonDistribution(mediallegadas);
    double llegadas = llegada.sample();
    textAreaA.append("Llegaron " + llegadas + " piezas\n"); //imprime hora
    for(int i = 1; i <= llegadas; i++){//creamos la cantidad de motores requerida
        Entidad entidad = new Entidad("Pieza", i, true);//con sus datos
        locacion.add(entidad);//y los ponemos en estiraje
    }
}

public double procesar(Double tiempoActual){...3 lines }

public void procesarAreaC(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaT(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaF(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaTa(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaR(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaD(Entidad pieza, double tiempoActual){...6 lines }

public void procesarAreaIns(Entidad pieza, double tiempoActual){
    textAreaA.append("La pieza tarda un tiempo exponencial \n");
    ExponentialDistribution inspec = new ExponentialDistribution(0.05);//Llamamos a la distribucion exponencial
    double tiempoSalida = dosDecimales(inspec.sample()+0.01);// procesamos en tiempo de horas
    tiempoSalida = dosDecimales(tiempoSalida + tiempoActual);// leguardamos el valor de salida
    pieza.setTiempo6(tiempoSalida); // le damos el tiempo a la pieza
    textAreaA.append("La pieza sale de inspeccion en la hora: "+tiempoSalida+"\n");
}

```

Luego tenemos los métodos auxiliares donde resaltamos a `arriboAlmacen()` y a `procesarAreaIns()` donde el arribo a almacén utiliza la distribución poisson para decidir el numero de piezas que llegaran al almacén por hora y el proceso de inspección que utiliza una

distribución exponencial para determinar el tiempo de inspección y asignarlo los demás métodos funcionan de forma similar solo que con valores fijos y nos con distribuciones.

Resultados:

Pro Model:

a) La utilización de los equipos

Para el porcentaje de utilización de los equipos tenemos que los equipo tienen los siguientes porcentajes de utilización.

Nombre	Capacidad	Total Entradas	Contenido Promedio	Contenido Máximo	Contenido Actual	% Utilización
Almacen	999.999,00	12.070,00	3.570,88	7.080,00	7.080,00	0,36
Horno	10,00	4.990,00	9,07	10,00	10,00	90,71
Banda 1	999.999,00	4.980,00	0,08	1,00	0,00	0,34
Area carga	999.999,00	4.980,00	2,86	8,00	3,00	0,00
Torneado	1,00	4.977,00	0,73	1,00	1,00	72,51
Fresado	1,00	4.976,00	0,78	1,00	1,00	77,76
Taladro	1,00	4.975,00	0,14	1,00	0,00	13,93
Rectificado	1,00	4.975,00	0,24	1,00	0,00	23,63
Area descarga	999.999,00	4.975,00	0,04	1,00	0,00	0,00
Banda 2	999.999,00	4.975,00	0,09	2,00	0,00	0,39
Inspeccion	1,00	4.975,00	0,25	1,00	0,00	25,03

El almacén se utiliza un 0.36% del tiempo, el horno se utiliza el 90.71% del tiempo, la banda 1 se utiliza el 34% del tiempo, el área de carga no se utiliza, **el área de torneado se utiliza el 72.51%, el área de fresado se utiliza el 77.76%, el área de taladrado se utiliza el 13.93%, el área de Rectificado se utiliza el 23.63%**, el área de descarga no se utiliza del tiempo, la banda 2 se utiliza el 0.39% y el área de inspección se utiliza el 25.03%.

b) La producción de la celda

Para el total de piezas producidas por la celda de la fábrica tenemos que:

Nombre	Total Cambios	Tiempo Por cambio Promedio (Min)	Valor Mínimo	Valor Máximo	Valor Actual	Valor Promedio
Piezas en horno	9.970,00	6,01	0,00	10,00	10,00	9,12
Piezas totales	4.975,00	12,06	0,00	4.975,00	4.975,00	2.482,46

El total de piezas producidas es de **4975 piezas** para es momento en la simulación es decir en la hora 1000.

c) Identifique la estación cuello de botella

Analizando las gráficas podemos ver que:



En el área de torneado las piezas se bloquean que se representa con el color morado eso quiere decir que el área siguiente que es el área de fresado es el cuello de botella por que las piezas tienen que esperar a que esta se libere para poder seguir y eso es un cuello de botella.

- d) **Determine el número mínimo de máquinas que deberían ser instaladas en cada estación para maximizar la producción y que la estación considerada como cuello de botella trabaje a una utilización lo más cercana al 95%.**

Para responder esta pregunta tenemos que utilizar el Sin Runner para poder hacer los experimentos y los resultados que nos arroja son.

Experiment	Objective Function	Fresado: % Utilization	Pieza: Total Exits	Mtorneado	Mfresado	Mtaladro	Mrectificado
4	5052.758	77.758	4975.000	4.000	1.000	3.000	5.000
6	5052.758	77.758	4975.000	3.000	1.000	4.000	5.000
11	5052.758	77.758	4975.000	3.000	1.000	5.000	2.000
15	5052.758	77.758	4975.000	3.000	1.000	4.000	3.000
22	5052.758	77.758	4975.000	4.000	1.000	5.000	5.000
23	5052.758	77.758	4975.000	1.000	1.000	1.000	1.000
13	5019.046	39.046	4980.000	3.000	2.000	1.000	2.000

Como se puede ver en la parte marcada con azul podemos decir que el número mínimo de máquinas en cada área necesarias para maximizar la producción de la celda y hacer que el área considerada cuello de botella este en una utilización lo mas cercana al 95% es de 1 en el área de torneado, 1 en el área de fresado, 1 en el área de taladrado y 1 en el área de rectificado esto debido a que si aumentamos en 1 el área de fresado este reduce su porcentaje de utilización significativamente por lo que no sirve para el ejercicio.

Java:

- a) **La utilización de los equipos**

Para el porcentaje de utilización de los equipos tenemos que los equipo tienen los siguientes porcentajes de utilización.

Porcentajes de utilizacion de las locaciones	
% Utilización horno	93.5%
% Utilización área de carga	3.15%
% Utilización de torneado	71.78%
% Utilización de fresado	74.77%
% Utilización de taladro	14.95%
% Utilización de rectificado	24.92%
% Utilización de área de descarga	0.07%
% Utilización de inspeccion	30.11%

Vemos que el porcentaje de utilización de los equipos es de 93.5% para el horno, 3.15% para el área de carga, **71.78% para el área de torneado, 74.77% para el área de fresado, 14.95% para el área de taladro, 24.92% para el área de rectificado**, para el área de descarga el 0.07% y para el área de inspección 30.11%.

- b) **La producción de la celda**

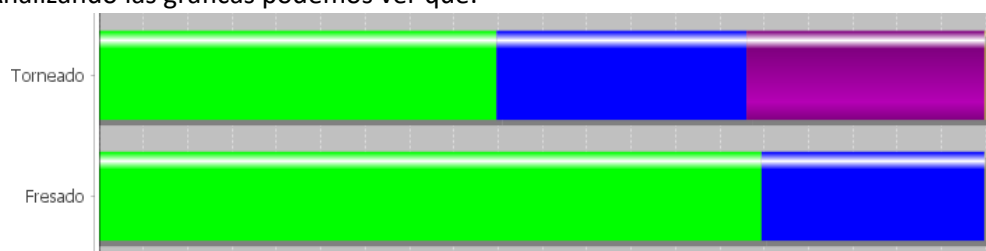
Para el total de piezas producidas por la celda de la fábrica tenemos que:

Piezas totales producidas	4980
Piezas en almacén	7344

Vemos que la celda produjo la cantidad de 49080 piezas.

- c) **Identifique la estación cuello de botella**

Analizando las gráficas podemos ver que:



Vemos que el área del torneado tiene color morado lo que significa que esta bloqueado lo que quiere decir que espero a que el área siguiente se desocupe para poder seguir su trabajo por tanto el área siguiente es el cuello de botella en este caso el **área de fresado**.

- d) **Determine el número mínimo de máquinas que deberían ser instaladas en cada estación para maximizar la producción y que la estación considerada como cuello de botella trabaje a una utilización lo más cercana al 95%.**

Como no se implemento el experimentador en el programa en java tomaremos los mismos datos arrojados por la herramienta de experimentos de Pro Model que son 1 en el área de torneado, 1 en el área de fresado, 1 en el área de taladro y 1 en el área de rectificado.

Tabla de comparación:

	Pro Model	Java
A) Utilización de equipos	Torneado = 72.51% fresado = 77.76% taladrado = 13.93% Rectificado = 23.63%	Torneado = 71.78% fresado = 74.77% taladrado = 14.95% Rectificado = 24.92%
B) Producción de celda	4975 piezas	4980 piezas
C) Cuello de botella	Área de Fresado	Área de Fresado
D) Mínimo de maquinas	Torneado = 1 fresado = 1 taladrado = 1 Rectificado = 1	Torneado = 1 fresado = 1 taladrado = 1 Rectificado = 1

Conclusión:

En conclusión, podemos decir que la celda esta bien como está actualmente ya que si aumentamos el nmero de máquinas la producción no varia casi nada y solo seria un aumento de costos innecesario para la fabrica también notar que el tiempo que espera el horno hasta llenarse cuando es en un tiempo prolongado si puede hacer variar la producción de piezas de forma significativa y tam bien que si quiere aumentar la producción seria aumentar la capacidad de piezas del horno.

Videos del proyecto:

Pro Model:

<https://youtu.be/ExluR3bjqSY>

Java:

<https://youtu.be/hpQmTX1b0pk>