



UNIVERSIDAD NACIONAL DE COLOMBIA

**MEDICIÓN DEL IMPACTO DEL MANTENIMIENTO PREVENTIVO Y
CORRECTIVO EN EL NIVEL DE PRODUCCIÓN MEDIANTE SIMULACIÓN
DISCRETA EN LA EMPRESA COLGAS DE OCCIDENTE**

CÉSAR AUGUSTO VARGAS YUMAYUZA

Universidad Nacional de Colombia
Facultad de ingeniería y arquitectura, Departamento de Ingeniería Industrial
Manizales, Colombia
2016

**MEDICIÓN DEL IMPACTO DEL MANTENIMIENTO PREVENTIVO Y
CORRECTIVO EN EL NIVEL DE PRODUCCIÓN MEDIANTE
SIMULACIÓN DISCRETA EN LA EMPRESA COLGAS DE OCCIDENTE**

CÉSAR AUGUSTO VARGAS YUMAYUZA

Trabajo final presentado como requisito parcial para optar al título de:
Magister en Ingeniería Industrial con énfasis en Dirección de Producción y Operaciones

Director:

Ph.D. Jaime Alberto Giraldo García.

Universidad Nacional de Colombia
Facultad de ingeniería y arquitectura, Departamento de Ingeniería Industrial
Manizales, Colombia

2016

Dedicatoria

A mis padres, esposa e hijo y a todos aquellos que por su apoyo y comprensión, lograron que todo este esfuerzo se hiciera realidad. Gracias a ustedes y al Dios todopoderoso

Resumen

El sistema de configuración productiva del sistema de producción de la empresa Colgas de Occidente S.A. E.S.P planta Manizales, es híbrido entre un sistema de producción con flujo en línea acompañada por el equipo y un flujo en línea acompañado por operarios en el cual se dan volúmenes medios del producto con un número diferente o familias de productos, esto da lugar a que la tasa de producción dependa del producto en particular que se produzca, del número de operarios asignados a la línea y la eficacia de operación en los equipos. Los cilindros son llenados y transportados al área de almacenamiento de producto en proceso, en la que cuatro operarios son responsables de inspeccionar los cilindros para descartar la no conformidad del producto según el programa de calidad con la que cuenta la empresa, después de esta operación en plataforma los cilindros aprobados son sellados en su válvula para garantizar que fueron inspeccionados y cumplen las especificaciones del programa de calidad, posteriormente el proceso de envasado culmina con el almacenamiento del producto en carros aptos para movilizar el producto a sus diferentes centros de distribución o comercialización (puntos de ventas, expendios y estaciones de servicio).

La empresa Colgas de Occidente S.A.E.S.P; ha optado por evaluar el desempeño de su sistema actual teniendo en cuenta los paros programados y no programados en los equipos que intervienen en el llenado de cilindros, ya que se estima que los tiempos entre fallos/mantenimientos y los tiempos de reparación/mantenimiento, reducen el rendimiento productivo del envasado, tal situación refleja, bajos volúmenes del envasado, tiempos variables en el envasado, bajo rendimiento de los equipos (Básculas) dada la disponibilidad del equipo cuando éste se requiere para su funcionalidad y limitaciones en el cumplimiento de sus pedidos. Situación que preocupa a la empresa ya que obstaculiza el cumplimiento de objetivos y metas trazadas en el corto, mediano y largo plazo de la organización.

Frente a esta problemática se pretende estudiar en detalle el sistema real, con el fin de determinar cuánto, cuándo y qué, equipos son los que más afectan el rendimiento de producción del envasado de cilindros en relación a los paros presentes en el sistema productivo por medio del estudio de simulación; la metodología empleada para este proyecto con simulación discreta, fue planteada y orientada siguiendo los principios y prácticas de la administración de proyectos propuestas por Giraldo (2014), este inició con la formulación del problema, el establecimiento de objetivos con su respectivo plan general del estudio, la definición del sistema real a modelar, la conceptualización del modelo, la recolección de la información, la codificación del modelo, verificación del modelo, la validación del modelo, su diseño experimental, el análisis de salida y su implementación.

Finalmente el trabajo concluye con los resultados obtenidos con el estudio de simulación, donde se evidencia como los paros tanto preventivos como correctivos afectan el nivel del envasado. De este modo con los escenarios expuestos se permitió analizar y evaluar la producción total del sistema, teniendo en cuenta el número de operarios que intervienen en el proceso de envasado y los índices de disponibilidad y confiabilidad de cada báscula como se ven afectados frente a los paros programados y no programados.

Palabras clave: Confiabilidad, Mantenibilidad, Disponibilidad, GLP, Simulación, Modelo, validez.

**MEASUREMENT OF THE IMPACT OF PREVENTIVE MAINTENANCE CORRECTIVE
BY DISCRETE SIMULATION AT THE LEVEL OF PRODUCTION OF THE COMPANY
OCCIDENT COLGAS PLANT MANIZALES**

Abstract

The productive system configuration Colgas of Occidente S.A. E.S.P plant Manizales is a hybrid between a production system with rhythmic flow line equipment and a rhythmic flow line in which media workers volumes of the product are given a different number or product families, this leads to the production rate depends on the particular product to occur, the number of operators assigned to the line and the operation efficiency of equipment. The cylinders are filled and transported to the storage area in progress, in which four workers are responsible for inspecting the cylinders to eliminate the non-conformity of the product of the quality program that the company, after this operation the platform approved cylinders are sealed in the valve to ensure that were inspected and meet the specifications of the quality program, then the packaging process culminates with the product storage tanks suitable for mobilizing the product to its various centers of distribution or marketing (points sales outlets and service stations).

The company Colgas Occidente S.A.E.S.P; it was decided to evaluate the performance of your current system taking into account scheduled shutdowns and unscheduled equipment involved in the filling of cylinders, and it is estimated that the time between failures / maintenance and repair times / maintenance, reduce the yield of packaging reflects this situation, low volume packaging, varying times in packaging, poor performance of the equipment (scales) given the availability of the equipment when it is required for its functionality and limitations in fulfilling their orders. Situation that worries the company and hinders the achievement of objectives and goals set in the short, medium and long-term organization.

Faced with this problem it is to study in detail the real system, in order to determine how, when and what, teams are those that most affect the performance of product packaging cylinder in relation to the present stoppages in the production system through the simulation study; the methodology used for this project with discrete simulation, was raised and oriented according to

the principles and practices of project management proposed by Giraldo (2014), this began with the formulation of the problem, setting goals with its own general plan of study the definition of the actual modeling system, the conceptualization of the model, data collection, coding model, model verification, validation of the model, experimental design, analysis and implementation output.

Finally, the paper concludes with the results obtained with the simulation study, which is evident as both preventive and corrective work stoppages affecting the level of packaging. Thus the exposed scenarios possible to analyze and evaluate the total output of the system, taking into account the number of workers involved in the packaging process and rates availability and reliability of each scale as affected against strikes scheduled and unscheduled.

Keywords: Reliability, Maintainability, Availability, LPG, Simulation. Model. Validity

Contenido

Resumen.....	VII
Lista de figuras.....	XIV
Lista de tablas.....	XV
Abreviaturas y acrónimos	XVI
1 Capítulo I: Introducción	17
1.1 Antecedentes	17
1.2 Planteamiento del problema.....	18
1.3 Objetivos.....	19
1.3.1 Objetivo general	19
1.3.2 Objetivos específicos.....	19
1.4 Justificación.....	19
2 Capítulo II: Marco teórico referencial	21
2.1 El mantenimiento en la Dirección de Producción y Operaciones.....	22
2.2 Caracterización del mantenimiento industrial.....	23
2.3 Indicadores de Confiabilidad, Mantenibilidad y Disponibilidad - CMD.....	26
2.3.1 Fiabilidad o confiabilidad – R	26
2.3.2 Mantenibilidad - M.....	30
2.3.3 Disponibilidad – D.....	31
2.4 Distribuciones estadísticas aplicadas en mantenimiento.....	32
2.4.1 Distribución Normal o de Gauss	33
2.4.2 Distribución Exponencial	34
2.4.3 Distribución Gama.....	34
2.4.4 Distribución Weibull	36
2.5 La simulación como herramienta para evaluar alternativas de solución.....	37
2.6 Caracterización del sistema de producción de Colgas de Occidente	38
3 Capítulo III: Metodología	40
3.1 Formulación del Problema	41
3.1.1 Enunciado formal del problema	41
3.1.2 Orientación del analista con el sistema o proceso	41
3.1.3 Herramientas para hacer el proceso de orientación	42
3.2 Establecer objetivos y proyecto general del estudio	43
3.2.1 Establecimiento de objetivos	43
3.2.2 Elaboración del plan general del estudio.....	44
3.3 Definición del Sistema	46
3.3.1 Clasificación del sistema	46
3.3.2 Componentes y eventos del sistema	47
3.3.3 Datos de entrada del sistema	48
3.3.4 Datos de salida del sistema.....	49
3.4 Conceptualización del modelo	49

3.4.1	Conceptos preliminares de modelamiento.....	51
3.4.2	Paradigmas de modelamiento.....	51
3.4.3	Elementos estructurales del modelo.....	52
3.4.4	Elementos operacionales del modelo.....	53
3.5	Recolección y análisis de datos.....	59
3.5.1	Directrices a tener en cuenta para la recolección de datos.....	59
3.5.2	Determinar los requerimientos de datos.....	61
3.5.3	Identificar fuentes de datos.....	61
3.5.4	Recolección de datos.....	62
3.5.5	Establecer suposiciones de trabajo.....	64
3.5.6	Análisis estadístico de los datos de entrada.....	64
3.6	Codificación del modelo.....	65
3.6.1	Comienzo de la codificación.....	65
3.6.2	Versión administrativa de la codificación.....	66
3.6.3	Organización del programa de simulación.....	67
3.7	Verificación del modelo.....	68
3.7.1	Medidas preventivas.....	69
3.7.2	Técnicas de verificación.....	69
3.8	Validación del modelo.....	71
3.8.1	Validación visual.....	71
3.8.2	Validación estadística.....	72
3.9	Diseño experimental.....	73
3.10	Documentación y análisis de salida.....	74
3.11	Implementación.....	74
4	Capítulo IV: simulación y análisis de resultados.....	75
4.1	Formulación del problema.....	75
4.2	Establecimiento de objetivos y plan general del estudio.....	75
4.2.1	Establecimiento de objetivos:.....	75
4.2.2	Elaboración del plan general.....	76
4.3	Definición del Sistema de Colgas de Occidente S.A. E.S.P.....	77
4.3.1	Clasificación.....	77
4.3.2	Componentes y eventos.....	78
4.4	Conceptualización del modelo.....	80
4.4.1	Elementos estructurales y operacionales del modelo.....	82
4.5	Recolección de datos.....	83
4.6	Codificación del modelo.....	85
4.7	Verificación del modelo:.....	86
4.8	Validación del modelo.....	87
4.8.1	Validación Visual.....	87
4.8.2	Validación Prueba estadística: T student.....	90
4.9	Diseño experimental.....	96
4.10	Análisis de salida.....	96
4.11	Plan de implementación.....	104

5	Conclusiones y recomendaciones	105
5.1	Conclusiones	105
5.2	Recomendaciones.....	105
A.	Anexo: Estadística tiempo de llenado cilindro 100 libras	110
B.	Anexo: Estadística tiempo de llenado cilindro 40 libras.....	114
C.	Anexo: Estadística tiempo de llenado cilindro 33 libras.....	118
E.	MTBF y MTTR.....	127
F.	Modelo Documental codificación por Objetos	146

Lista de figuras

Figura 2.1-1: Gestión integral de mantenimiento	23
Figura 2.2-1: Clasificación del mantenimiento Industrial.....	24
Figura 2.3-1: Intervalos de tiempo entre fallos.	27
Figura 2.3-2: Siglas que se usan en la medición CMD.	28
Figura 2.3-3: Curva o bañera de Davies.....	30
Figura 2.4-1: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, Normal	33
Figura 2.4-2: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, exponencial.	34
Figura 2.4-3: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, <i>Gamma</i>	35
Figura 2.5-1: Relación modelo y sistema.....	37
Figura 2.6-1: Ideograma del proceso de envasado de cilindros.....	39
Figura 2.6-1: Etapas para realizar un estudio de simulación.....	40
Figura 3.4-1: Simbología para diagramar objetos de flujo.....	50
Figura 3.4-2: Modelo conceptual empleando simbología objetos de flujo.	50
Figura 4.2-1: Diagrama de Gannt del proyecto.....	76
Figura 4.3-1: Diagrama de flujo de alto nivel.....	79
Figura 4.4-1: Modelo conceptual del proceso de envasado de Colgas de Occidente.	81
Figura 4.5-1: Interfaz gráfica de SGC.	83
Figura 4.7-1: Verificación del modelo con la Animación.....	86
Figura 4.10-1: box joking statistic per scenario	103
Figura 4.10-2: Resultados del escenario_1	97
Figura 4.10-3: Resultados del Escenario_2	99
Figura 4.10-4: Resultados del Escenario_3	101

Lista de tablas

Tabla 2-1: Evolución del mantenimiento	21
Tabla 2.3-1 Indicadores de mantenimiento.....	32
Tabla 4.8-1: Reporte SGC de envasado báscula 1	87
Tabla 4.8-2: Reporte SGC de envasado báscula 8	87
Tabla 4.8-3: Reporte SGC de envasado báscula 16	87
Tabla 4.8-4: Reporte SGC de envasado sin fallos báscula 1.	88
Tabla 4.8-5: Reporte SGC de envasado sin fallos báscula 8.	88
Tabla 4.8-6: Reporte SGC de envasado sin fallos báscula 16.	88
Tabla 4.8-7: Reporte SGC de envasado con fallos báscula 1.	88
Tabla 4.8-8: Reporte SGC de envasado sin fallos báscula 8.	89
Tabla 4.8-9: Reporte SGC de salida de envasado sin fallos báscula 14.....	89
Tabla 4.8-10: Reporte de salida de envasado de las 16 básculas con Flexsim.	89
Tabla 4.8-11: Cilindros envasados del sistema real y simulación.....	90
Tabla 4.8-12: Prueba no paramétrica Kolmogorov – Smirnov.....	91
Tabla 4.8-13: Prueba T student para análisis de varianzas y medias.	94
Tabla 4.10-1: Número de Replicas por escenario.	96

Abreviaturas y acrónimos

MP	Mantenimiento preventivo
MC	Mantenimiento correctivo
MTBF	Tiempo medio entre fallos
MTTR	Tiempo medio para reparar
D	Disponibilidad
C	Confiabilidad
M	Mantenibilidad
TPM	Mantenimiento productivo total
GLP	Gas licuado del petróleo
R	Fiabilidad
WIP	Trabajo en proceso
SGC	Sistema Global de Control.

1 Capítulo I: Introducción

Actualmente las organizaciones tanto de manufacturas como de servicios se ven obligadas a estar en un entorno más competitivo, razón por la cual el mercado les exige ser más eficientes en sus sistemas productivos y ofrecer a sus clientes un producto de calidad respetando los tiempos de entrega pactados (LEAD TIME). Este nuevo entorno hace que el mantenimiento juegue un rol muy importante dentro del sistema de producción, ya que debe garantizar el logro de metas y objetivos trazados por la organización para mantenerse a través del tiempo en el mercado. Esto permite reducir tiempos muertos en los equipos, cuestión que posteriormente impacta en la producción, tiempos de entrega, optimización de la calidad en los productos y el aumento de la eficiencia de los recursos productivos de la compañía Colgas De Occidente S.A. E.S.P.

El presente trabajo está desarrollado en cinco capítulos, el primero expone los antecedentes, planteamiento del problema, objetivos y justificación del proyecto; el segundo muestra el marco teórico del mantenimiento, su génesis, evolución y como éste se relaciona con el subsistema de fabricación en el que algunos actores en la materia analizan el estudio de los sistemas de mantenimiento, indicando la necesidad de utilizar técnicas cuantitativas y modelos matemáticos que permiten medir, evaluar y controlar la eficiencia de dichos sistema; en los capítulos tercero y cuarto se plantea y se aplica la metodología para estudiar el problema y posibles soluciones al mismo. Y el capítulo cinco finaliza con conclusiones y recomendaciones sobre el trabajo final.

1.1 Antecedentes

El término de mantenimiento y su aplicabilidad tuvo su origen en el siglo XIX, época de auge de la revolución industrial donde las tareas se centraban en solucionar las fallas de las maquinas/equipos, éstas eran básicamente correctivas y es lo que se conoce actualmente como mantenimiento correctivo. A partir de la primera y segunda guerra mundial se realizaron estudios en investigación y operaciones, en los que se originó un gran avance en los sistemas de producción y por ende en el estudio de prevenir las fallas, las causas y los efectos presentes en la manufactura, esto buscaba incrementar los índices de producción, evitando el reproceso por

averías presentadas en el aparato productor. De este modo aparece el mantenimiento preventivo, predictivo, proactivo y el TPM (Mantenimiento Productivo Total – Cero Averías) que mediante diferentes técnicas y métodos ayudan a optimizar el sistema de producción. Para el caso se analizará el proceso productivo de envasado de la empresa Colgas de Occidente Planta Manizales.

1.2 Planteamiento del problema

La empresa Colgas de Occidente S.A. E.S.P Planta Manizales, tiene como objetivo buscar mejoras al proceso productivo del envasado de cilindros del Gas Licuado del Petróleo GLP. La compañía ha optado por evaluar el desempeño de su sistema actual teniendo en cuenta los paros programados y no programados en los equipos que intervienen en el llenado de cilindros, ya que se estima que los tiempos entre fallos/mantenimientos y los tiempos de reparación/mantenimiento, reducen el rendimiento productivo del envasado, tal situación refleja las circunstancias siguientes:

- Bajos volúmenes en la producción del envasado.
- Tiempos variables en el envasado.
- Bajo rendimiento de los equipos (Básculas) dada la disponibilidad del equipo cuando éste se requiere para su funcionalidad.
- Limitaciones en el cumplimiento de los pedidos que hacen a la empresa.
- Falta de coordinación e instrucción al personal operativo que está relacionado directamente a la operatividad del equipo y no sabe qué hacer cuando un equipo falla.

Las situaciones anteriores hacen que la compañía reduzca su participación en el mercado y obstaculice en el cumplimiento de objetivos y metas trazadas en el corto, mediano y largo plazo. De continuarse esta problemática la empresa podría perder competitividad y rentabilidad del negocio.

Frente a esta problemática se pretende estudiar en detalle el sistema real, con el fin de determinar cuánto, cuándo y qué, equipos son los que más afectan el rendimiento de producción del envasado de cilindros en relación a los paros presentes en el sistema productivo.

1.3 Objetivos.

1.3.1 Objetivo general

Realizar una medición del impacto del mantenimiento preventivo y correctivo en el nivel de producción mediante un estudio con simulación discreta, con el fin de determinar acciones que beneficien el interés organizacional.

1.3.2 Objetivos específicos

- Caracterizar el mantenimiento correctivo.
- Caracterizar el mantenimiento preventivo.
- Describir el sistema actual del proceso productivo de envasado de GLP.
- Analizar los fundamentos teóricos y conceptuales dentro de los cuales se enmarcan la simulación y el mantenimiento.
- Obtener un modelo comparativo del sistema actual frente a un modelo de mejora propuesto como herramienta para la toma de decisiones.

1.4 Justificación

En un ambiente de alta competencia, las empresas para mantenerse en el mercado requieren implementar estrategias de mantenimiento que garanticen una alta disponibilidad y un bajo índice de paradas no programadas en los equipos que hacen parte principal del sistema productivo de cada organización, pero al mismo tiempo permitan optimizar los costos para asegurar una buena rentabilidad para los accionistas e inversionistas.

El mantenimiento preventivo debe poder lograr la reducción de las averías imprevistas o aquellos eventos indeseados en los tiempos de reparación, es decir, propiciar la prolongación de la vida útil de los componentes y/o equipos obteniendo los efectos del ahorro de recursos y con ellos la reducción de los costos asociados al mantenimiento de las instalaciones, lo que

contribuye a mejorar la calidad de los productos y servicios que la empresa Colgas De Occidente que oferta tanto a sus clientes como a sus usuarios.

En términos generales, las fallas en la medida que ocurran tendrán algún tipo de efecto directo o indirecto sobre la seguridad o el comportamiento funcional de la planta. En la medida que ésta resulte afectada, se adjudicará el hecho al contexto operacional del equipo que ha fallado, su comportamiento funcional y los efectos físicos de la causa o modo de la falla ocurrida. Esta combinación de efectos, sugiere que dentro del contexto del mantenimiento cada falla tendrá consecuencias específicas asociadas a ella y sobre todo percutirá en la productividad y competitividad de la organización. Es por eso que mediante el modelo experimental de simulación se pretenderá jugar con aquellas variables en las cuales se encuentran los síntomas (variables dependientes) y las causas (variables independientes), para tratar de contrarrestar y reducir las causas que afectan el sistema productivo de los equipo en el proceso de envasado del GLP.

2 Capítulo II: Marco teórico referencial

El concepto del mantenimiento surge con el advenimiento de las maquinas, época de la revolución industrial en la que empezó el hombre a producir bienes y servicios en masa para cubrir las necesidades de su entorno. Este tipo de producción acarrió lo que actualmente se conoce como automatización industrial, que permitió la fabricación a gran escala de productos (Bautista Paz, 2007). Cabe resaltar que el concepto de mantenimiento se origina como consecuencia de la alta producción, ya que los recursos productivos comenzaron a fallar y se hizo necesario adoptar un enfoque sistémico, concepto central de la teoría general de sistemas-TGS (Ackoff, 2002; Churchman, 1973). Bajo este enfoque se permitió analizar y diagnosticar las causas y los efectos que impactaban al subsistema de producción. Según Mora (2010), los sistemas de mantenimiento y de producción están relacionados directamente y se ubican en una época contemporánea reciente y resumiéndose cronológicamente en la siguiente tabla 2-1.

Tabla 2-1: Evolución del mantenimiento

Etapa	Sucede aproximadamente	Producción - Manufactura		Mantenimiento e ingeniería de fábricas	
		Orientación hacia...	Necesidad específica	Orientación hacia...	Objetivo que pretende
I	antes de 1950	el producto	generar el producto	hacer acciones correctivas	reparar fallos imprevistos
II	entre 1950 y 1959	la producción	estructurar un sistema productivo	aplicar acciones planeadas	prevenir, predecir y reparar fallos
III	entre 1960 y 1980	la productividad	optimizar la producción	establecer tácticas de mantenimiento	gestar y operar bajo un sistema organizado
IV	entre 1981 y 1995	la competitividad	mejorar índices mundiales (Benchmarking)	implementar una estrategia	medir costos, compararse, predecir índices
V	entre 1996 y 2003	la innovación tecnológica			
VI	desde 2004	Gestión y operación integral de activos en forma coordinada de todo el sistema organizacional de la empresa, anticipándose a las necesidades de los equipos y de los clientes de mantenimiento - Predicciones - Pronósticos - Gestión de activos.			

Fuente: Mora, 2010.

2.1 El mantenimiento en la Dirección de Producción y Operaciones

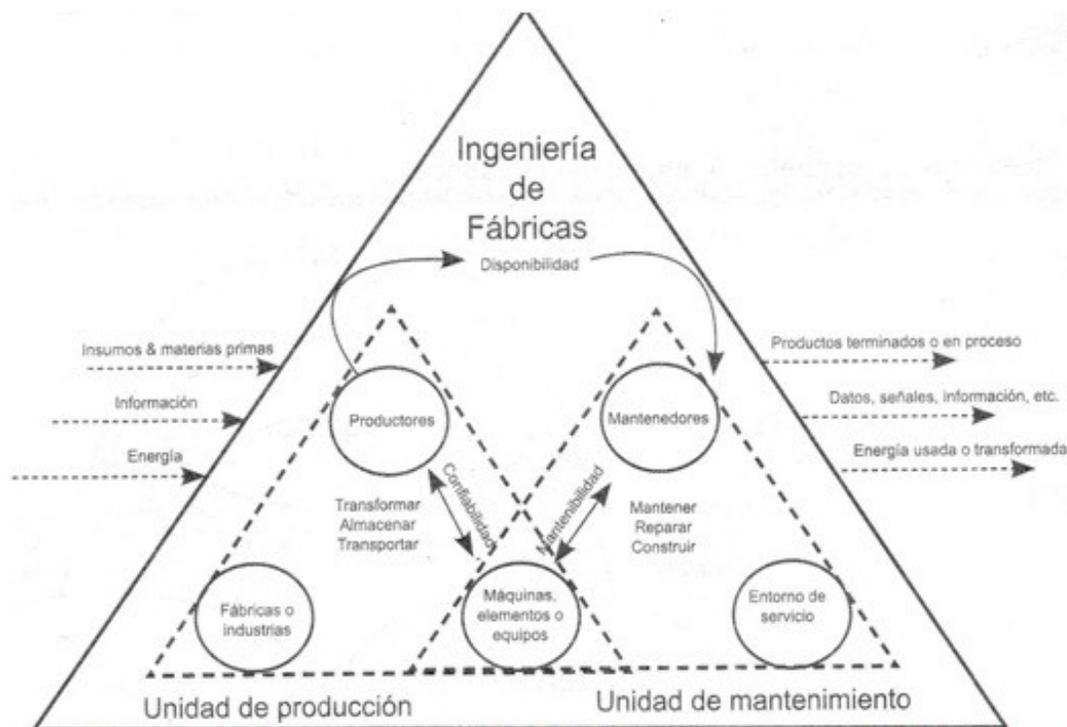
Una buena gestión en el mantenimiento permite una mejor organización en el sistema de producción, ya que permite asegurar la disponibilidad del equipamiento, la calidad del producto, la entrega a tiempo, el precio competitivo y el respeto a las normas de seguridad, reduciendo los riesgos en accidentes de trabajo ocasionados por el mal estado de las máquinas y por ende reflejados en una reducción de costos provocados por paradas no programadas en los procesos productivos (Riis et al. 1997).

El sistema de mantenimiento debe de garantizar la funcionalidad de la infraestructura operativa y productiva del sistema de producción. Ya que ambas deben de ir mejorando la infraestructura productiva de la organización. Este refleja la importancia del mantenimiento en la estructura estratégica de las empresas (Muller et al., 2007; Kardec y Nacif, 2003; Swanson, 2001; Teixeira, 2001).

Es importante señalar que el mantenimiento y la producción deben ser un sistema integral, en el cual se relacionan y engranan todos sus elementos, generando una sinergia entre sus partes y componentes, cuya finalidad es garantizar la confiabilidad y disponibilidad de los recursos productivos de la empresa.

Dentro del sistema integral de mantenimiento y producción existe una serie de inputs (insumos o materias primas, energía, información, mano de obra, etc.) que a través de un proceso de conversión (transformación) den unos outputs con agregación de valor (productos terminados, información, energía y entre otros). De esta manera se puede ver el sistema como un sistema abierto que interactúa constantemente con la empresa y su entorno (Domínguez et ál; 1995).

En la figura 2.1, se puede apreciar como el sistema de producción y el sistema de mantenimiento hay una relación permanente, abierta y dinámica, las cuales se relacionan a través de máquinas y equipos dando como resultado un sistema integrado de ingeniería de fábricas (mantenimiento-máquinas-producción).

Figura 2.1-1: Gestión integral de mantenimiento

Fuente: Mora, 2010.

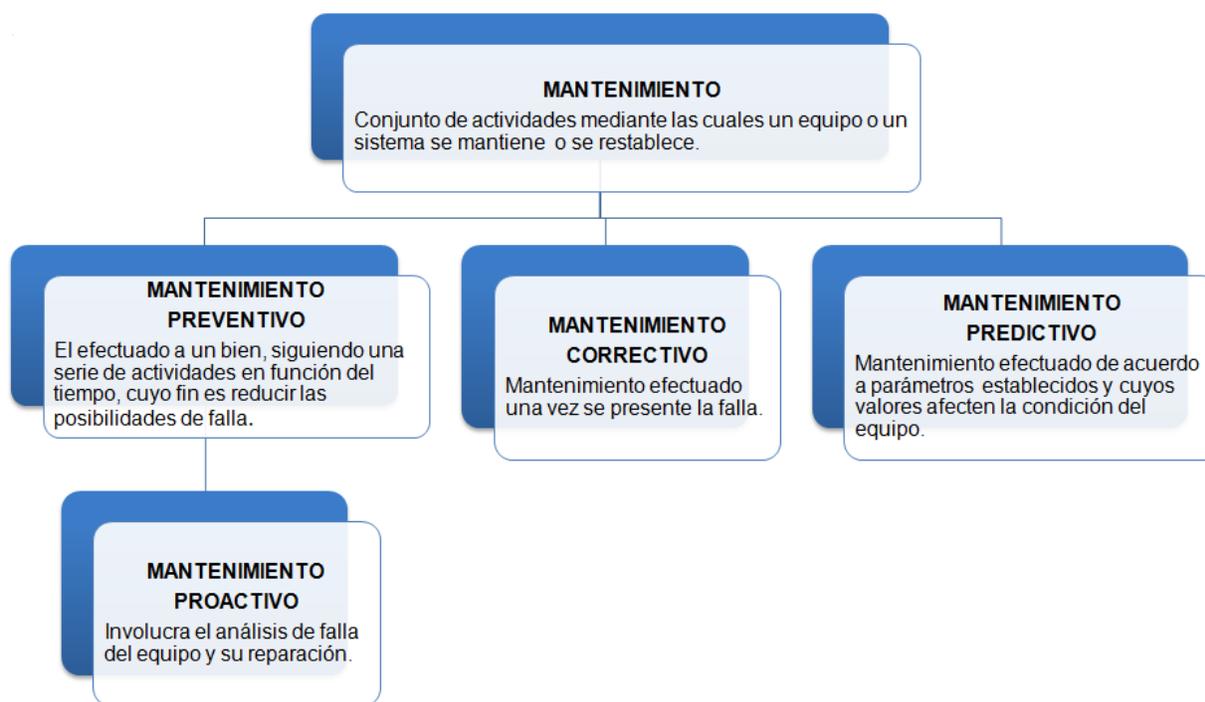
2.2 Caracterización del mantenimiento industrial

El mantenimiento en términos generales se efectúa para incrementar la confiabilidad de los sistemas de producción y de mantener la funcionalidad de los equipos a través del tiempo, su objetivo es conseguir un determinado nivel de disponibilidad de producción en condiciones de calidad exigente, al mínimo costo, con el máximo nivel de seguridad para el personal que lo utiliza y lo mantiene, y con una mínima degradación hacia el medio ambiente (Navarro et al, 1997).

Según (Campbell, Duffuaa, & Raouf, 2010), en su filosofía de mantenimiento, básicamente es la de tener un número necesario de personal capacitado de mantenimiento que sea consistente con la optimización de la producción y la disponibilidad de la planta sin que se comprometa la seguridad.

Para lograr esta filosofía, el autor plantea una serie de estrategias que apuntan a desempeñar un papel eficaz si se aplican en la combinación y forma correcta, algunas de estas estrategias es importante mencionarlas a continuación y se puede ver en la figura 2.2 los diferentes tipos de mantenimientos más usados en la industria, se diferencian el uno del otro por su periodicidad y objetivo de su uso, es por eso que Campell (2010), resalta el rol de los diferentes usos del mantenimiento si cada uno de ellos se combinan, pueden de cierto modo ser una fuerte herramienta de utilidad.

Figura 2.2-1: Clasificación del mantenimiento Industrial.



Fuente: Elaboración propia.

Mantenimiento correctivo o por fallas: el autor plantea que este tipo de mantenimiento se realiza cuando un equipo es incapaz de seguir operando. No hay elemento de planeación para este tipo de mantenimiento. Sin embargo, cuando se presenta se suele aplicar la estrategia de operación-hasta-que-falle, y/o cuando el costo adicional frente a otros tipos de mantenimiento no se puede justificar. Suele aplicarse muy comúnmente en componentes electrónicos.

Mantenimiento preventivo (revisiones y limpiezas periódicas basadas en lista de chequeos): el cual se planifica conforme una planeación en función a las horas de operación o

basado en un calendario establecido, el cual permita minimizar las fallas potenciales dentro un sistema de producción que pueda afectar la funcionalidad de los equipos. Requiere de un alto nivel de planeación y es importante poseer conocimiento acerca de la distribución de las fallas o la confiabilidad del equipo como tal.

La empresa Colgas de Occidente S.A. E.S.P, ha adoptado este tipo de mantenimiento, que mediante una rutina de inspecciones semanales y mensuales se emplean listas de chequeo o *CheckList* (actividades y procedimientos de verificar, ajustar, reemplazar y limpiar), que buscan reducir los mantenimientos correctivos a fin de mantener la disponibilidad y fiabilidad de los equipos.

Mantenimiento predictivo (análisis del estado del equipo mediante el análisis de variables físicas). Se basa en las condiciones, es decir, este se determina vigilando los parámetros establecidos del equipo o cuyos valores se vean afectados por la condición del cual están expuestos y está estrechamente relacionado al mantenimiento preventivo.

Mantenimiento de oportunidad. Como su nombre lo indica, se lleva a cabo cuando surge la oportunidad, es decir, cuando se presenta un periodo de paro considerable originado por un programa de un sistema en particular, y puede utilizarse para efectuar tareas conocidas de mantenimiento dentro de ese lance de tiempo; un caso típico que recomiendan algunos autores consiste en realizar este tipo de mantenimiento cuando el personal operativo esté fuera de servicio, es decir, por variables como suplementos de descansos, capacitaciones del personal o en horario no operativo de los recursos y entre otros relacionados.

Mantenimiento Proactivo. El mantenimiento predictivo tuvo como última innovación el mantenimiento proactivo, que usa gran cantidad de técnicas para alargar la disponibilidad de las máquinas. La idea principal de un mantenimiento proactivo es el análisis de las causas fundamentales de las fallas en las máquinas.

Factores tales como alta confiabilidad, mantenibilidad, disponibilidad y seguridad, además del bajo costo de mantenimiento, alta calidad de los productos elaborados entre otros aspectos relacionados, pueden dar un impacto fundamental a la empresa para ser altamente competitiva en su sistema de producción (Shyjith, 2009).

2.3 Indicadores de Confiabilidad, Mantenibilidad y Disponibilidad - CMD

Los indicadores de mantenimiento y los sistemas de planificación empresarial asociados al área de efectividad permiten evaluar el comportamiento operacional de las instalaciones, sistemas, equipos, dispositivos y componentes que están directamente relacionados a la producción y por lo tanto afectan directamente o indirectamente el sistema productivo, de esta manera será posible implementar un plan de mantenimiento orientado a perfeccionar esta labor.

La Confiabilidad, mantenibilidad y disponibilidad son prácticamente las únicas medidas técnicas y científicas, fundamentadas en los cálculos matemáticos, estadísticos y probabilísticos, que tiene el mantenimiento para soportar su análisis (Mora et al, 2003).

Para una mejor comprensión del CMD, es importante mencionar como eje central del tema de mantenimiento las fallas y entender su compartimiento a través del tiempo, cuya finalidad será introducir mejoras al sistema.

2.3.1 Fiabilidad o confiabilidad – R

Según Ebeling et ál; 2005. La confiabilidad o fiabilidad se define como la probabilidad de un recurso operativo de desempeñarse y funcionar satisfactoriamente para las labores para las cuales fue fabricado en un periodo de tiempo específico, bajo condiciones normales de operación dadas por el fabricante.

La frecuencia con la cual ocurren las fallas en el tiempo se conoce como fiabilidad o confiabilidad. Si no hay fallas, el equipo es ciento por ciento (100%) confiable; si la frecuencia de fallas es muy baja, la confiabilidad del equipo es aún aceptable; pero si la frecuencia de fallas es muy alta, el equipo es poco confiable (Rojas, 1975).

La falla se puede entender como la probabilidad de que un evento degrade un equipo, máquina o elemento de producción, esto incluye:

- Falla completa, al perder disponibilidad y funcionabilidad del equipo.

- Falla parcial, sin generar la pérdida total de disponibilidad, es decir, el equipo puede continuar operando bajo ciertas condiciones.

Dentro de estas dos premisas, las fallas se clasifican como:

Criticas: causando pérdida completa de funcionalidad y por lo tanto el recurso queda inoperativo.

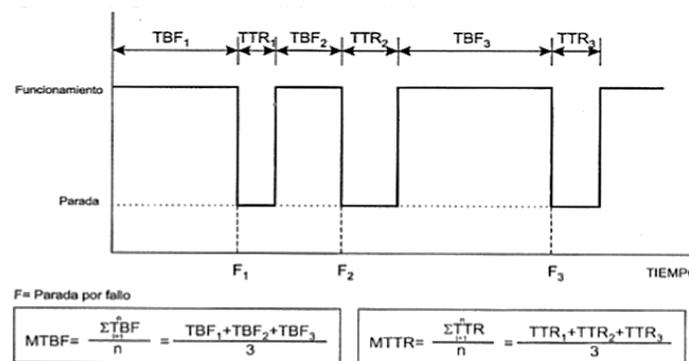
Degradantes: causan pérdida parcial del equipo, afectando las condiciones de la capacidad de producción, sin embargo, el equipo puede seguir funcionando; normalmente este tipo de fallas evolucionan a críticas con el tiempo o uso sino son intervenidas, (Oreda, 1997).

La fiabilidad se suele representar con la letra R (Reliability), y está relacionada con el tiempo medio entre fallos – El MTFB (Mean Time Between Failures) puede definirse en términos del valor esperado de la función de densidad de Fallos $f(t)$, el cual se relaciona en la Ecuación 1 y visto de un intervalo de tiempo expuesto en la figura 2.3.1.

Ecuación 1, Calculo de Tiempo Medio Entre Fallos.

$$MTFB = \int_0^{\infty} R(t)dt = \frac{1}{\lambda}$$

Figura 2.3-1: Intervalos de tiempo entre fallos.



Fuente: Mora, 2010

Un equipo bien diseñado, perfectamente montado, correctamente probado y apropiadamente mantenido no debe fallar nunca en teoría; no obstante, la experiencia ha

Donde:

TTF = Time To Failure = Tiempo hasta Fallar (se usa en equipos que solo fallan una vez, no reparables).

f_i = Falla i -ésima

m = número de fallas ocurridas en el tiempo que se revisa, desde f_1 hasta f_i

TTR = Time To Repair = Tiempo que demora la reparación neta, sin incluir demoras ni tiempos logísticos, ni tiempos invertidos en suministros de repuestos o recursos humanos.

MTTR = Mean Time To Repair = Tiempo Medio para Reparar = $\sum TTR / m$

TBF = Time Between Failures = Tiempo entre Fallas.

MTBF = Mean Time Between Failures = Tiempo Medio entre Fallas = $\sum TBF / m$

UT = Up Time = Tiempo Útil en el que equipo funciona correctamente.

MUT = Mean Up Time = Tiempo Medio de Funcionamiento entre Fallas = $\sum UT / m$

DT = Down Time = Tiempo no operativo.

MDT = Mean Down Time = Tiempo Medio de Indisponibilidad o no funcionamiento entre Fallas = $\sum DT / m$

ADT = Administrative Delay Time = retrasos administrativos exógenos a la actividad propia de reparación, diferentes al tiempo activo neto de la reparación; ejemplos de estos son: suministro de personal especializado, entrenamiento de recursos humanos requeridos para esa reparación, revisión de manuales de mantenimiento u operación, localización de herramientas, cumplimiento de procesos y/o procedimientos internos, etc.

LDT' = Logistics Delay Time = retrasos logísticos la obtención de insumos para la reparación, en los procesos de mantenimiento o de producción, en los tiempos de suministros, etc. como por ejemplo el tiempo requerido para transporte de repuestos, o el tiempo que hay que esperar a que se construya un repuesto especial por parte de los fabricantes, etc.

LDT = ADT + LDT' = Logistic Down Time = Tiempo total logístico que demora la acción propia de reparación o mantenimiento. Son todos los tiempos exógenos al equipo que retrasan el tiempo activo.

MLDT = Mean Logistics Down Time = Tiempo Medio de Tiempos Logísticos

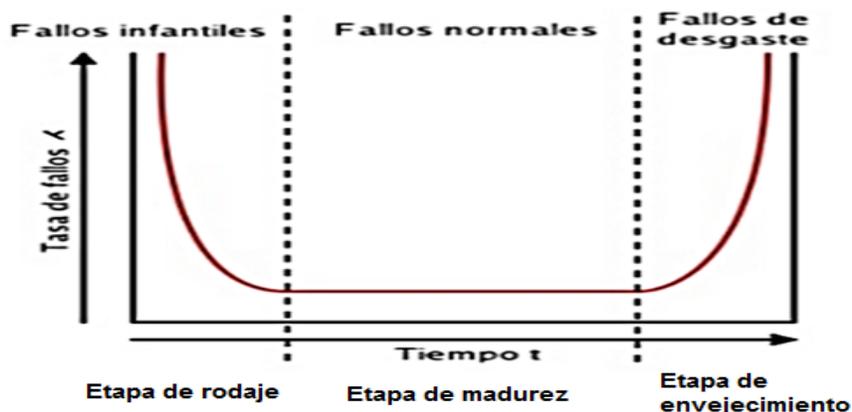
SoFa = State of Failure = Estado de Falla, el equipo no funciona correctamente.

SoFu = State of Functioning = Estado de Funcionamiento correcto.

PM = Planned Maintenances = Mantenimientos Planeados, pueden ser preventivos o predictivos.

Ready Time = Tiempo de Alistamiento = el equipo o sistema está disponible, opera pero no produce, no está en carga operativa.

La confiabilidad distingue tres tipos característicos de fallas excluyendo daños por manejo inadecuado, almacenado u operación inapropiada de los usuarios, que son inherentes al equipo, que ocurren sin ninguna falla por parte del operador y están relacionadas con cada una de las regiones de vida del elemento y el tipo de falla que en ella ocurren; éstas son las tempranas o infantiles, fallos normales y fallos de desgaste. Como se muestra en la curva de la bañera o curva de Davies ver figura 2.3.3, permite analizar los fallos durante el período de vida útil de un equipo o máquina. Tiene forma de bañera debido a que la vida útil de los dispositivos tiene un comportamiento que viene reflejado por tres etapas diferenciadas y que a continuación se pueden apreciar en la siguiente figura.

Figura 2.3-3: Curva o bañera de Davies

Fuente: Mora, 2010.

- **FALLOS INFANTILES:** esta etapa de rodaje se caracteriza por tener una elevada tasa de fallos que descienden rápidamente con el tiempo. Éstos pueden deberse a diferentes razones como equipos defectuosos, instalaciones incorrectas, errores de diseño del equipo, desconocimiento del equipo por parte de los operarios o desconocimiento del procedimiento adecuado a la hora de manipularlos.
- **FALLAS NORMALES:** es una etapa de madurez con una tasa de errores menor y constante. Estos fallos no se producen debido a causas inherentes al equipo sino por causas aleatorias externas, es decir, accidentes fortuitos y a condiciones inadecuadas en la manipulación del equipo.
- **FALLAS POR DESGASTE:** es una etapa de vejez caracterizada por una tasa de errores rápidamente creciente. Los fallos se producen por desgaste natural del equipo debido al ciclo de vida útil del mismo.

2.3.2 Mantenibilidad - M

La rapidez con la cual las fallas o el funcionamiento defectuoso en los equipos son diagnosticados y corregidos, se entiende como mantenibilidad y es definida como la probabilidad para que un elemento, máquina o dispositivo, pueda regresar nuevamente a su estado de funcionamiento normal después de una avería, falla o interrupción productiva, (Mora, 2010). Una medida de la mantenibilidad es el tiempo medio de reparación o sigla conocida en inglés como MTTR (*Mean Time To Repair*).

2.3.3 Disponibilidad – D

La disponibilidad es la probabilidad de que el equipo esté operando satisfactoriamente en el momento en que sea requerido después del comienzo de su operación y se usa bajo condiciones estables; el tiempo total considerado puede incluir el tiempo de operación, tiempo activo de reparación, tiempo inactivo, tiempo en mantenimiento preventivo, tiempo administrativo y tiempo logístico de acuerdo con el tipo de disponibilidad seleccionada según Blanchard et al ,(1995); Vallejo, (2004); Céspedes et al, (2003).

El indicador de disponibilidad es una medida importante y útil en los casos en que el usuario deba tomar decisiones importantes para elegir un equipo entre varias alternativas. Para tomar una decisión objetiva con respecto a la adquisición del nuevo equipo, es necesario utilizar información que abarque todas las características relacionadas, entre ellas la disponibilidad, que es una medida que suministra una imagen más completa sobre el perfil de funcionalidad. Toro et al, (2003). La probabilidad de que el equipo funcione satisfactoriamente en el momento en que sea requerido, se define mediante la siguiente expresión:

Ecuación2. Ecuación general de disponibilidad.

$$D = \frac{UT}{UT + DT} = \frac{MTBF}{MTBF + MTTR}$$

Donde;

UT = Tiempo de funcionamiento (*Up Time*).

DT= Tiempo de parada o Tiempo no operativo. (*Down Time*).

MTBF = Tiempo medio entre fallas (*Mean Time Between Failures*).

MTTR = Tiempo medio para reparar (*Mean Time To Repair*).

De esta manera se obtiene una disponibilidad inherente o intrínseca, considerando que la no funcionalidad de cualquier equipo es inherente no más al tiempo activo de reparación, es decir, que está basada únicamente en la distribución de fallas y en la distribución de tiempos de reparación (Ebeling, 2005). En resumen, la confiabilidad está relacionada directamente a las fallas, la mantenibilidad a reparaciones y la disponibilidad a la posibilidad de generar servicios o productos (Mora, 2010). En la siguiente tabla 2.2 Indicadores de mantenimiento se describen

algunos índices relacionados a mantenimiento en relación a la disponibilidad y confiabilidad en equipos.

Tabla 2 Indicadores de mantenimiento

Indicador	Símbolo	Descripción	Formula
Tasa de Fallo	λ (Lambda)	Es un indicador que se utiliza para determinar el desempeño de un equipo.	$\lambda = \frac{\# \text{ de fallos}}{\text{Horas de operación}}$
Tiempo Medio Entre Fallo	MTBF	Es un indicador de confiabilidad que relaciona la cantidad de fallas frente al tiempo de uso.	$MTBF = \frac{\text{Horas de operación}}{\# \text{ fallos}} = \frac{1}{\lambda}$
Tiempo Medio Para Reparar	MTTR	Este indicador refleja el nivel de gravedad de la avería y la eficacia de las reparaciones.	$MTTR = \frac{\text{Horas de reparación}}{\# \text{ fallos}}$
Disponibilidad	D (t)	Es un indicador que determina el tiempo de actividad del equipo en la operación.	$D = \frac{MTBF}{MTBF + MTTR}$
Confiabilidad o Fiabilidad	R (t)	Este indicador mide la probabilidad que el equipo va a completar una misión en un tiempo sin fallos.	$R(t) = e^{-\lambda t}$

2.4 Distribuciones estadísticas aplicadas en mantenimiento

La presencia de un fallo en cualquier sistema durante un tiempo $t(x)$ es algo que no puede preverse de forma exacta, sino que responde a la acción de múltiples variables de tipo aleatorio, sin embargo, estas se pueden estudiar siguiendo métodos probabilísticos.

La función que da la probabilidad a que se produzca un fallo en un determinado tiempo dado de $t(x)$, se le denomina función de distribución de fallo, dentro de dichas distribuciones se pueden encontrar las siguientes y son muy empleadas en el mantenimiento:

- Distribución Normal.
- Distribución Exponencial.
- Distribución Gamma.

- Distribución *Weibull*.

Estas distribuciones son funciones matemáticas teóricas que se utilizan para realizar previsiones, por lo tanto son útiles en mantenimiento, debido a que ayudan a tomar decisiones en condiciones de incertidumbre y son tenidas en cuenta para la elaboración del presente trabajo y que a continuación se mencionan en detalle.

2.4.1 Distribución Normal o de Gauss

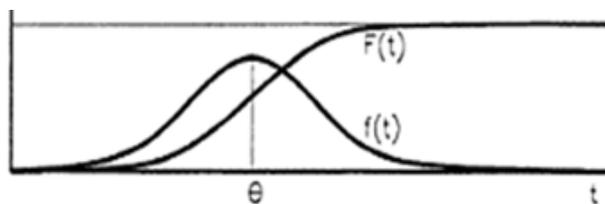
Es una distribución de probabilidad de variable continua y se utiliza como primera aproximación en aquellos sistemas en los que existe un modelo de comportamiento conocido, y en los que, por tanto, existe una estimación de su periodo de vida nominal. Por ejemplo, en elementos mecánicos sujetos a desgaste, tales como rodamientos, cojinetes, engranajes, entre otros relacionados. Se dice que una variable aleatoria continua X sigue una distribución normal de parámetros μ y σ . Su función de densidad de probabilidad está dada por:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Siendo la función de distribución:

$$F(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^x e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Figura 2.4-1: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, Normal



Fuente: Mora, 2010.

Donde los parámetros son: $X \in \mathbb{R}$; μ es la media y σ es la desviación estándar.

2.4.2 Distribución Exponencial

Es una distribución de probabilidad continua con un parámetro estadístico $\lambda > 0$.

Es utilizada en mantenimiento para modelar la fiabilidad en los equipos o sistemas, se considera adecuada para modelar aquellos sistemas cuya probabilidad de fallo es insensible al uso, es decir, no se encuentran afectados por agentes como el desgaste o el envejecimiento. Es muy usada para modelar el tiempo de vida de componentes electrónicos. Genéricamente describe la cantidad de tiempo que transcurre entre eventos, tal como el tiempo entre fallas de equipos electrónicos.

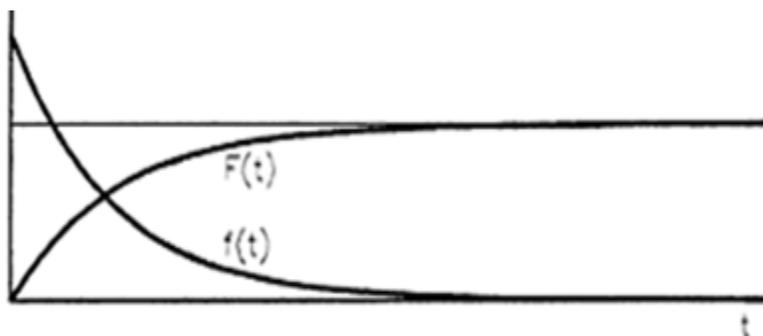
La densidad de probabilidad viene dada por la expresión:

$$f(t) = \lambda e^{-\lambda t}$$

Siendo la función de distribución:

$$F(t) = 1 - e^{-\lambda t}$$

Figura 2.4-2: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, exponencial.



Fuente: Mora, 2010.

Donde el parámetro es: $\theta = \lambda^{-1} = MTBF$

2.4.3 Distribución Gama.

Es una función de distribución continua, sus usos son relevantes en distribución de tiempo de vida entre recalibración, ajustes o mantenimiento de equipos y tiempo de vida de

elementos con sistemas en paralelos o en *Stand-By*, es muy conveniente para caracterizar los tiempos de fallas de equipos durante periodos de rodaje.

La función de densidad de probabilidad es:

$$f(t) = \frac{\lambda}{\Gamma(r)} (\lambda t)^{r-1} e^{-\lambda t}$$

Siendo la función de distribución:

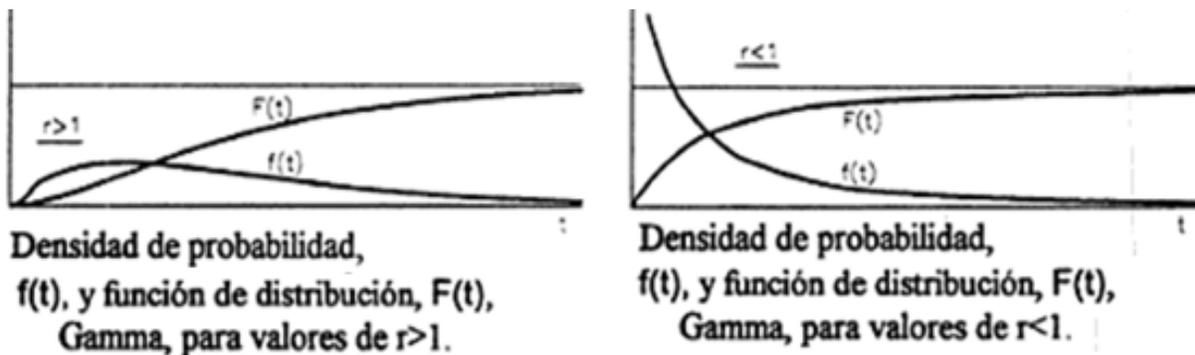
$$F(t) = \frac{\lambda}{\Gamma(r)} \int_0^t (\lambda t)^{r-1} e^{-\lambda t} dt$$

Siendo $\Gamma(r)$ la conocida función integral Gamma de r :

$$\Gamma(r) = \int_0^{\infty} x^{r-1} e^{-x} dx = (r-1)!$$

Esta distribución presenta como caso particular, en el caso que $r = 1$, la distribución *Gamma* se convierte en una exponencial.

Figura 2.4-3: Densidad de probabilidad, $f(t)$, y función de distribución, $F(t)$, *Gamma*



Fuente: Mora, 2010.

2.4.4 Distribución Weibull

Es una distribución de probabilidad continua y triparamétrica, definida por tres parámetros que son:

- Gamma: parámetro de posición (γ). El más difícil de estimar y por este motivo se asume con demasiada frecuencia que vale cero. Indica el lapso en el cual la probabilidad de falla es nula (Forcadas, 1983).
- Eta: Parámetro de escala o características de vida útil (η). Su valor es determinante para fijar la vida útil del producto o del sistema. Cuanto más alto, las maquinas pueden ser más robustas o de trabajos de mayor duración.
- Beta: Parámetro de forma (β): Refleja la dispersión de los datos y determina la forma que toma la distribución. Este parámetro permite a la distribución Weibull tomar diversas formas, es decir, cuando el valor $\beta < 1$ se le denomina que está en fase o etapa de rodaje (Tasa de falla decreciente con el tiempo); cuando $\beta = 1$ se describe en la etapa de madurez (Tasa de falla constante en el tiempo), y si $\beta > 1$ se determina que se encuentra en etapa de envejecimiento o de desgaste (Tasa de falla creciente con el tiempo), como se pudo apreciar en la Figura 5. Curva de Davies. (Díaz, 1992).

Los usos y aplicaciones de la *Weibull* son relevantes en elementos o sistemas con resistencia a la corrosión, distribución de vida de elementos básicos como capacitores, rodamientos, motores entre otros relacionados.

La función de densidad de probabilidad es:

$$f(t) = \frac{\beta}{\eta} \left(\frac{t - \gamma}{\eta} \right)^{\beta-1} e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta}$$

Función de distribución: $F(t) = 1 - e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta}$

La Distribución *Weibull* puede asemejarse a otras distribuciones dados sus parámetros Beta, es decir, cuando:

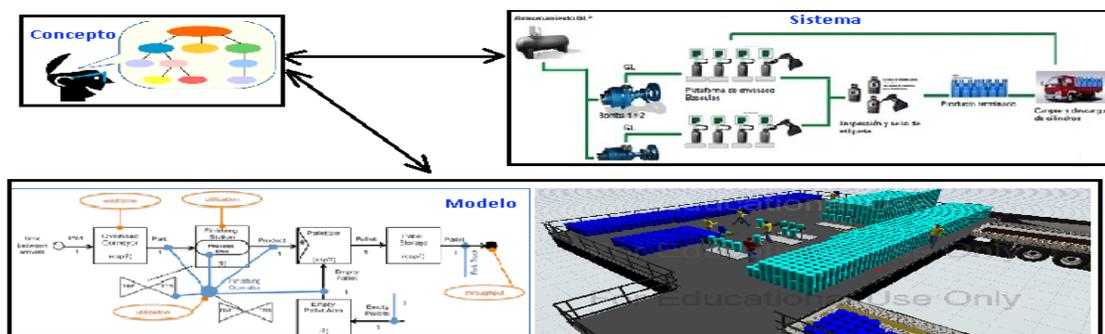
- $\beta=1$; su comportamiento se asemeja a la distribución Exponencial.
- $\beta=2$; Distribución de *Rayleigh*.
- $3 \leq \beta \leq 4$; Distribución Normal.

2.5 La simulación como herramienta para evaluar alternativas de solución

El término simulación lo definen Kelton & Law (2007) como una herramienta que empleando computadoras imitan el sistema real, a fin de describir y entender su comportamiento a través del tiempo. Un modelo de simulación se puede definir como una representación abstracta del sistema real, que mediante el empleo de herramientas matemáticas, estadísticas y el uso de algún *software* de simulación busca imitar el sistema real para comprender las relaciones entre sus componentes y predecir su desempeño.

Como todo sistema debe recibir unas entradas o *Inputs*, los cuales pasan por un proceso de conversión al fin de obtener unas salidas llamadas *Outputs*. Un sistema de producción según Domínguez (1999), “Es un sistema abierto y complejo, en el que los diferentes subsistemas y elementos están convenientemente interrelacionados y organizados, forman un todo unitario y desarrollando una serie de funciones que pretenden la consecución de los objetivos globales de la organización”. La fig. 2.5.1 Ilustra la relación que existe entre un sistema y modelo.

Figura 2.5-1: Relación modelo y sistema



Fuente: Adaptación propia a partir de los aportes de J. Giraldo (2014), Harrell et al. (2004).

Para el desarrollo de este trabajo se hará hincapié en los modelos de simulación discretos, en los cuales las variables de estado sufren cambios instantáneos en puntos separados en el tiempo. Para Banks et al. (2009), este tipo de simulación produce una serie de secuencias instantáneas en las cuales representa la evolución del sistema en el tiempo. Las medidas de desempeño para este tipo de sistema de simulación son:

- Tiempo de espera promedio en cola del *Flow item*.
- Número promedio de *Flow item* en la cola.
- Tasa de utilización de los procesadores.

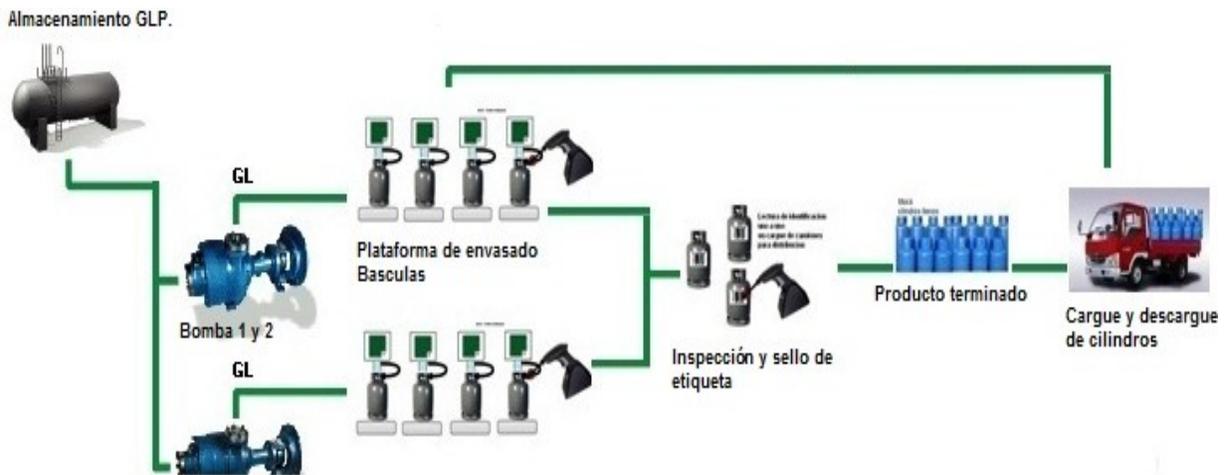
A continuación se caracteriza el sistema de producción de la empresa Colgas de Occidente S.A. E.S.P Planta Manizales.

2.6 Caracterización del sistema de producción de Colgas de Occidente

Colgas de Occidente S.A ESP, en su objeto económico contempla la compra, venta y comercialización al por mayor y al detal del producto Gas Licuado de Petróleo (GLP), que se comercializan en el mercado en cilindros de 20, 33, 40, 100 libras y tanques estacionarios, pero para el caso de estudio se contemplara el análisis como tal del sistema productivo de envasado de cilindros para delimitar su alcance en el modelo de simulación. El sistema de configuración productiva de Colgas de Occidente S.A. E.S.P planta Manizales es híbrido entre un sistema de producción con flujo en línea acompasada por el equipo y un flujo en línea acompasado por operarios en el cual se dan volúmenes medios del producto con un número diferente o familias de productos y su *layout* es lineal, lo cual permite que la tasa de producción dependa del producto en particular que se produzca, del número de operarios asignados a la línea y la eficacia de operación en los equipos. Miltenburg (2004). Para este proceso productivo el envasado de cilindros inicia cuando se tienen existencias del producto en almacenamiento de los tanques estacionarios con capacidad de 30.000 galones cada uno, la empresa actualmente cuenta con seis (6) de ellos, una vez garantizado el suministro y teniendo en stock del producto, se procede a realizar el trasiego con la bomba 1 o bomba 2, para alimentar el sistema de red de los equipos

que hacen parte del envasado en plataforma (16 básculas). En estas dieciséis (16) básculas, dos operarios son responsables de realizar el llenado de los cilindros, de acuerdo a una programación y planeación por parte del área operativa y debidamente aprobado por el administrador de planta. Los cilindros son llenados y transportados al área de almacenamiento de producto en proceso, en la que cuatro operarios son responsables de inspeccionar los cilindros para descartar la no conformidad del producto según el programa de calidad que cuenta la empresa, una vez realizada esta operación en plataforma los cilindros aprobados son sellados en su válvula para garantizar que fueron inspeccionados y cumplen las especificaciones del programa de calidad, posteriormente el proceso de envasado culmina con el almacenamiento del producto en carros aptos para movilizar el producto a sus diferentes centros de distribución o comercialización (puntos de ventas, expendios y estaciones de servicio). La figura 2.6, muestra el ideograma del proceso de producción de envasado de la empresa “Colgas de occidente planta Manizales”.

Figura 2.6-1: Ideograma del proceso de envasado de cilindros.

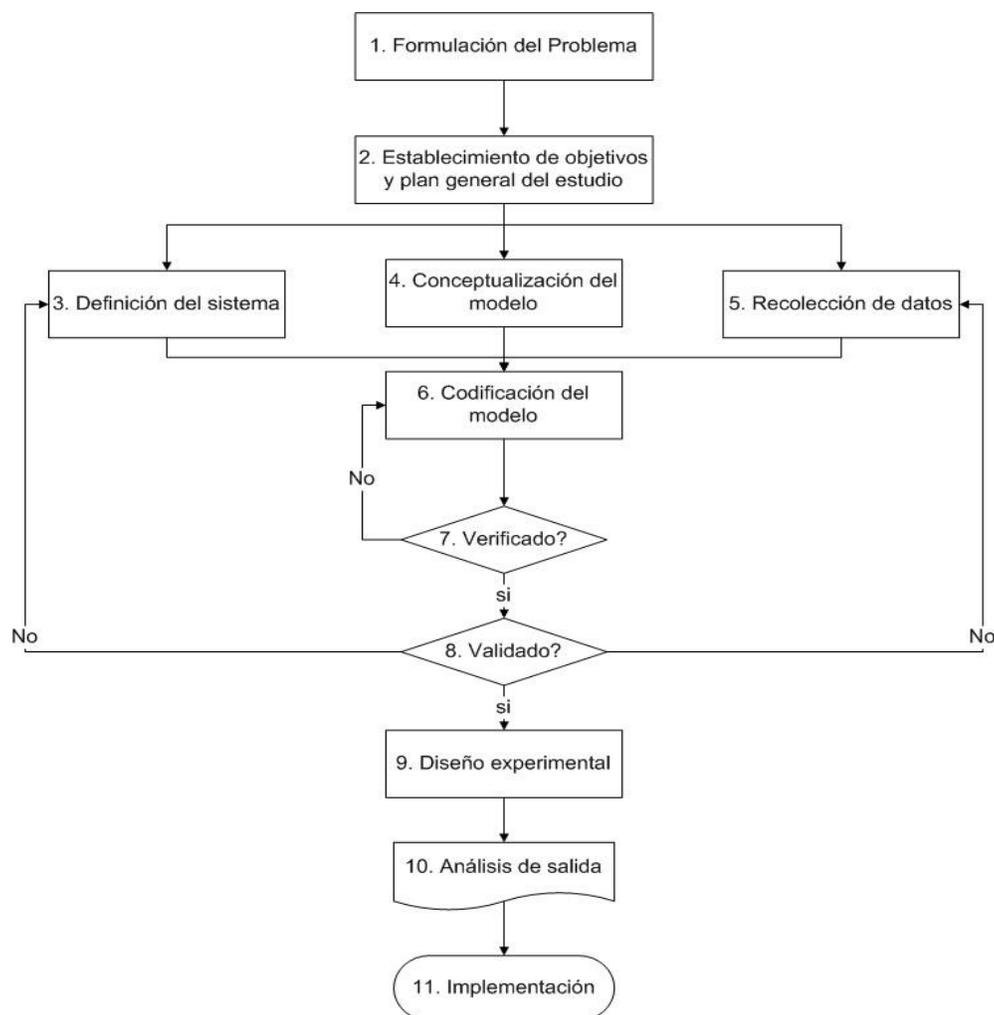


Fuente: Elaboración propia.

3 Capítulo III: Metodología

La decisión de usar herramientas de simulación, se basó considerando la interdependencia y variabilidad que tiene el sistema, adicionalmente de la necesidad de estudiar su comportamiento en el tiempo en el marco de los principios y prácticas de la administración de proyectos, siguiendo de manera general la metodología propuesta por Giraldo (2014) y adaptada por varios autores, según etapas mostradas en la figura 2.6 1. A continuación se describe:

Figura 2.6-1: Etapas para realizar un estudio de simulación



Fuente: Giraldo (2014) a partir de los aportes de Banks et al. (2009), Kelton & Law (2007), Harrell et al. (2006), Chung (2004), Shannon (1988) y Gordon (1978).

3.1 Formulación del Problema

Cada estudio de simulación debe comenzar con el enunciado del problema, este debe ser claramente descrito y entendible. (Chung, 2004), sugiere los siguientes pasos para formular el problema:

3.1.1 Enunciado formal del problema

El enunciado del problema suele incluir textos relacionados con incrementar la satisfacción del cliente, el *throughput*, reducir desperdicios y el trabajo en proceso (WIP – *Work In Process*); estos factores son fundamentales en cualquier sistema que implique operaciones, ya que involucra el *goodwill* y el prestigio de la marca u organización. Están relacionados a la reducción de tiempos en una cola o fila, tiempos de procesamiento, a la reducción de cuellos de botella y la mejora en las operaciones, lo cual se refleja en cumplimientos de entrega pactados y la disminución de costos operacionales. Chung (2004) sugiere dos herramientas disponibles para ayudar a enunciar el problema, conocidas en el entorno industrial como diagrama de causa – efecto y diagrama de Pareto, estas dos herramientas permiten identificar y definir el problema.

3.1.2 Orientación del analista con el sistema o proceso

Esta consiste en familiarizarse con el sistema o proceso, es decir, que información es relevante para la recolección de datos a fin de entender el funcionamiento del sistema para su modelamiento, sus componentes y cómo estos se interrelacionan.

Giraldo (2014), recomienda que durante este proceso de orientación, deba existir un mínimo de tres tipos diferentes de visitas de orientación y sugiere que las visitas se realicen por separado para aumentar la probabilidad de entender el funcionamiento del sistema. Dichos tipos de visitas de orientación son:

- **Visita Inicial:** el propósito de esta visita es obtener una comprensión de alto nivel de las entradas y salidas básicas del sistema. Este tipo de orientación se realiza típicamente como una visita guiada por un miembro de la organización. El analista

debe de ir identificando y documentando los componentes del proceso o sistema relacionado a la problemática a estudiar.

- **Visitas de orientación sobre el flujo detallado:** después que el analista ha obtenido una comprensión básica de alto nivel del sistema, se debe de recoger información detallada sobre el funcionamiento del sistema o proceso mediante el registro de:
 - ✓ Los tipos de entidades que son procesados por el sistema.
 - ✓ El número y tipo de colas.
 - ✓ El número y tipo de recursos del sistema.
 - ✓ La secuencia de los procesos experimentados por las entidades.
 - ✓ Las mediciones de desempeño del sistema.

Durante esta fase es necesario que el analista comience a desarrollar un gráfico de flujo de alto nivel para ir obteniendo una comprensión amplia del sistema.

- **Visita de revisión:** una vez que el analista tiene una comprensión del funcionamiento del sistema, es hora de volver al él para revisión de las visitas de orientación con la información recolectada y su propósito es asegurar que la comprensión del funcionamiento del sistema es consistente con la comprensión que tiene el equipo de trabajo sobre el sistema o diagramas de flujo.

3.1.3 Herramientas para hacer el proceso de orientación

El analista y su equipo de trabajo deben tener un alto grado de acceso al sistema, permitiendo que se empleen dispositivos electrónicos de grabación; el propósito de usar estos dispositivos no es registrar los datos de entrada del sistema, sino permitir que el analista de simulación revise los procesos más complejos cuando este se requiera, con el fin de obtener una comprensión fundamental del sistema. La recolección de datos del sistema se hace en una etapa posterior. El objetivo de la fase de orientación es principalmente entender el sistema a modelar.

3.2 Establecer objetivos y proyecto general del estudio

3.2.1 Establecimiento de objetivos

Para Harrell et al. (2004), el objetivo de una simulación define el propósito o razón para conducir el estudio. Estos deben de ser realistas, claros, completos, alcanzables y precisos. Los objetivos pueden ser agrupados en las siguientes categorías generales:

- Análisis de desempeño: Responde a ¿cuál es el rendimiento de todo el sistema en términos de utilización de los recursos, el tiempo de flujo, tasa de salida, etc.?
- Análisis de restricciones de capacidad: cuando es llevado al máximo, ¿cuál es la capacidad de producción o procesamiento del sistema y dónde se encuentran los cuellos de botella?
- Comparación de configuraciones: ¿qué tan bien una configuración del sistema satisface los objetivos de desempeño comparada con otra?
- Optimización: ¿cuál configuración en particular de las variables de decisión obtienen las metas de desempeño deseadas?
- Análisis de sensibilidad: ¿cuáles variables de decisión son las que más influyen en las medidas de desempeño, y cómo ocurre dicha influencia?
- Comunicación: ¿cómo puede ser efectivamente visualizada la dinámica del sistema?

Chung (2004), también propone las siguientes cuatro categorías de objetivos muy relacionados con los sistemas de producción y de servicios.

- Políticas de operación relacionadas con el desempeño del sistema: ¿cuál configuración de las variables de decisión mejoran el desempeño del sistema?
- Políticas de recursos relacionadas con el desempeño del sistema: ¿cuáles niveles y qué distribución de los recursos mejoran el desempeño del sistema?
- Políticas relacionadas con el costo de los recursos: ¿cuáles niveles y qué distribución de los recursos disminuyen su costo?
- Evaluación de capacidades de equipo: ¿los equipos existentes ofrecen la capacidad suficiente para satisfacer la demanda?

Harrell et al. (2004) señalan que algunas veces los objetivos cambian o se aumentan en el transcurso del proyecto. Esto se debe a que el conocimiento ganado con el proceso de hacer simulación sobre el sistema se convierte en una oportunidad de mejoramiento. La definición de objetivos también debe tener en cuenta el uso último que se le dará al modelo de simulación. Algunos modelos son usados una sola vez, otros son rehusados para análisis “que tal si”, otros simplemente obtienen un estadístico del desempeño, algunos son destinados para que sean usados por gerentes con poco conocimiento de la simulación por lo que requiere una interfaz usuario-sistema amigable, otros son empleados para tomar decisiones de menor consecuencia, etc.

Giraldo (2014), recomienda que los objetivos para esta fase reúnan las siguientes características:

- Tenga un alto impacto (reducir costos, aumentar las unidades producidas, etc).
- Sea loggable (reducir el inventario en un 20%).
- Sea específico y cuantificable, es decir, reducir el tiempo de espera en una particular cola (reducir el tiempo de flujo en un 40%).
- Sea medible (incrementar el rendimiento en un 10%).

3.2.2 Elaboración del plan general del estudio

Los estudios de simulación son proyectos de ingeniería y se desarrollan en un proceso de planificación del proyecto, o sea, con sus respectivas actividades, tiempo de duración, qué recursos son necesarios para el desarrollo del proyecto de simulación, que actividades preceden una de otras y un diagrama de Gantt que ilustra los tiempos de duración de las actividades y la relación entre ellas. Para Chung (2004) el éxito o fracaso en la gestión de un proyecto de simulación depende de los siguientes parámetros:

- **Tiempo:** este parámetro es asociado con la programación del proyecto y frecuentemente implementado como un diagrama de Gantt. Mientras el proyecto se ejecuta, el administrador del proyecto debe regularmente comparar los resultados obtenidos con los resultados programados, esto con el fin de corregir cualquier discrepancia.

- **Costo:** esto significa que hay un presupuesto asociado con el proyecto y estos pueden incluir *hardware*, *software*, recurso humano, entre otros relacionados a fin de completar el proyecto.
- **Desempeño técnico:** el proyecto debe tener objetivos de desempeño técnico, éstos normalmente son especificados en la etapa de establecimiento de los mismos. Pueden requerir que el modelo de simulación sea usado para tomar decisiones operacionales o de inversión de capital. Es por eso que el diseño del modelo de simulación debe representar la realidad lo más fiablemente posible con el objeto de tomar decisiones acertadas.

Según Chung (2004), el proyecto de simulación tiene el siguiente ciclo de vida:

- **Conceptualización:** consiste en la decisión inicial tomada por alguien de la organización, concerniente a considerar la realización del estudio de simulación, por lo que se asignará formalmente a un director del proyecto. El proceso de formulación del problema puede ser completado durante esta fase.
- **Planeación:** durante esta fase el director del proyecto de simulación conformará el equipo de trabajo del proyecto, las actividades y responsabilidades del caso para lograr el objetivo del mismo. Estas pueden ser plasmadas en una estructura de desglose del trabajo (WBS – *Work Breakdown Structure*), un diagrama de responsabilidad lineal (LRC – *Linear Responsibility Chart*) y el diagrama de Gantt que se llevará a cabo en esta fase.
- **Ejecución:** la mayoría de actividades del proyecto de simulación serán completadas durante esta fase. Estas incluyen la definición del sistema, la recolección de datos y su análisis, la translación del modelo, verificación, validación, experimentación y análisis de resultados.
- **Terminación:** esta fase suele realizarse después de que la mayor parte de las actividades del proyecto han sido concluidas, éstas incluyen el informe de resultados del proyecto de simulación y las actividades de presentación.

Para Cleland y de Irlanda (2002), cada proyecto tiene lo que se conoce como partes interesadas o *stakeholder* - grupos de interés. Y como su nombre lo indica son personas

que tienen algún tipo de interés en las operaciones en curso o en la terminación del proyecto. El interés de estas partes podrá dar lugar a alguna acción u omisión que puede afectar el proyecto. Los interesados pueden clasificarse como:

- **Interesados internos:** son los asociados directamente con el equipo del proyecto de simulación, estos incluyen el director del proyecto, analistas, estadísticos, recolectores de datos, gerentes funcionales que supervisan a los miembros del equipo del proyecto, entre otros relacionados.
- **Interesados externos:** son aquellos que no están directamente asociados con el equipo del proyecto de simulación, pero que de cierta forma son importantes para el desarrollo del proyecto, es decir, pueden estar dentro de la organización matriz, pero fuera del equipo del proyecto de simulación o pueden estar completamente ajenos a la organización del mismo.

3.3 Definición del Sistema

Para Chung (2004), esta etapa consiste en determinar: la clasificación del sistema, es decir, si el sistema es discreto, continuo o una combinación de ambos, qué componentes y eventos del sistema se requieren para modelar, qué datos de entrada se deben recolectar y por último qué datos de salida debe generar el modelo.

3.3.1 Clasificación del sistema

Los sistemas pueden ser clasificados con respecto a dos dimensiones. En primer lugar, un sistema puede ser discreto, continuo, o una combinación de ambos. En segundo lugar puede ser con terminación o sin terminación. Estas clasificaciones son importantes, ya que pueden afectar el modo en que el analista lleve a cabo el análisis y modelado del proyecto, (Kelton & Law, 2007).

Los sistemas discretos: son aquellos en el cual las variables de estado cambian sólo en un instante del tiempo, esto significa que el estado de algún componente en el sistema está cambiando en instantes espaciados en el tiempo. Por ejemplo, en las colas de un banco, en las que el analista estará interesado en la estimación de medidas como el tiempo de espera promedio

o la longitud de la línea de espera. Tales medidas sólo cambian cuando un cliente entra o sale del sistema.

Los sistemas continuos: son aquellos en los que las variables de estado cambian en forma continua a través del tiempo. Algunos sistemas de comportamiento continuo usualmente involucran algún tipo de sustancia o líquido como fluido y normalmente pueden ser modelados con ecuaciones diferenciales. Un ejemplo son las plantas de tratamiento de agua e industrias químicas.

Sistemas con terminación: son aquellos en los cuales las entidades son iniciadas, procesadas y terminadas en ese mismo periodo de tiempo, es decir, que comienzan cada periodo sin ningún tipo de influencia del periodo anterior. Un ejemplo son las entidades bancarias, centros comerciales, restaurantes, entidades educativas, entre otros relacionados, lugares en los que no se permiten que los clientes o usuarios se queden en las locaciones durante toda la noche. Los sistemas con terminación deben:

- Tener un evento natural de terminación.
- No mantener entidades en el sistema de un periodo a otro.

Sistemas sin terminación: este tipo de sistemas inicia su operación con las entidades que ya están en el sistema desde periodos anteriores o no existe un periodo de cierre de la operación y operan de forma continua, es decir, 7x24, las 24 horas del día, 7 días a la semana. Por ejemplo: hospitales, estaciones de combustibles, fábricas que operan los 3 turnos del día, entre otros relacionados. De otra parte, en general, los sistemas sin terminación pueden:

- Tener un evento de terminación, pero mantener las entidades del sistema entre los periodos de tiempo.
- No tener un evento de terminación y operar de forma continua.

3.3.2 Componentes y eventos del sistema

Para Chung (2004), una herramienta esencial para representar la definición del sistema es un diagrama de flujo de alto nivel, este diagrama describe gráficamente cómo interactúan los

principales componentes, eventos y como fluyen los procesos. También muestra qué papel desempeñarán los datos de entrada y salida en la conceptualización del modelo.

Algunos componentes comunes en los sistemas de producción y servicios según (Kelton & Law, 2007) son: personal, máquinas, transportadores, bandas transportadoras e instalaciones de fabricación.

El analista de simulación también debe decidir qué eventos y procesos del sistema serán incluidos en el modelo, por lo cual estaría interesado en los siguientes procesos o eventos: tipos de órdenes de trabajo, comportamiento de colas de máquinas, procesamiento con máquinas, movimiento de material, fallos de máquinas, mantenimiento preventivo, inspección de defectos, entre otros relacionados que tengan un impacto significativo en el sistema.

3.3.3 Datos de entrada del sistema

Para Chung (2004), los datos de entrada son los que activan el sistema y los datos de salida son su resultado.

En el proceso de definición del sistema, el analista está interesado en identificar una lista preliminar de los tipos de datos de entrada que afectan las medidas de desempeño. Existen dos categorías generales de datos de entrada y son:

- La primera categoría está relacionada con el concepto de entidad del sistema, que se define como los elementos que son procesado por éste, por ejemplo: clientes, pasajeros, órdenes de trabajo, materia prima, entre otras. Los datos de entrada asociados a las entidades son el tiempo entre arribos al sistema, el número de entidades que arriban a la vez o tamaños de lote y tiempos de procesamiento.
- La segunda categoría son los recursos del sistema. Estos recursos son componentes del sistema que procesan entidades, por ejemplo: operarios, equipos de manufactura y vehículos. Entre los datos de entrada de estos recursos están los tiempos de descanso, tasa de fallos, tiempos de reparación, programación del mantenimiento, capacidades de operación, velocidades y tiempos de movimiento, entre otros relacionados.

3.3.4 Datos de salida del sistema

Para Chung (2004), los datos de salida son usados para medir los niveles de desempeño. Estas medidas de desempeño son usadas para validar el modelo de simulación y realizar experimentación con el mismo. En el proceso de validación del modelo, las medidas de desempeño son recolectadas y comparadas con las medidas de desempeño del sistema real. En el proceso de experimentación, se crean modelos de simulación con diferentes políticas de operación y recursos (llamados escenarios). Entonces se comparan las medidas de salida entre los diferentes modelos con el fin de seleccionar el escenario de mejor desempeño.

- Principales medidas de desempeño: la selección de cuáles medidas emplear en el estudio de simulación depende de los objetivos trazados, por ejemplo: si el objetivo es reducir costos de operación, el analista debe centrarse en la utilización de recursos o si el objetivo es mejorar la satisfacción al cliente, la medida de desempeño apropiada sería el tiempo promedio de permanencia de los clientes en el sistema.

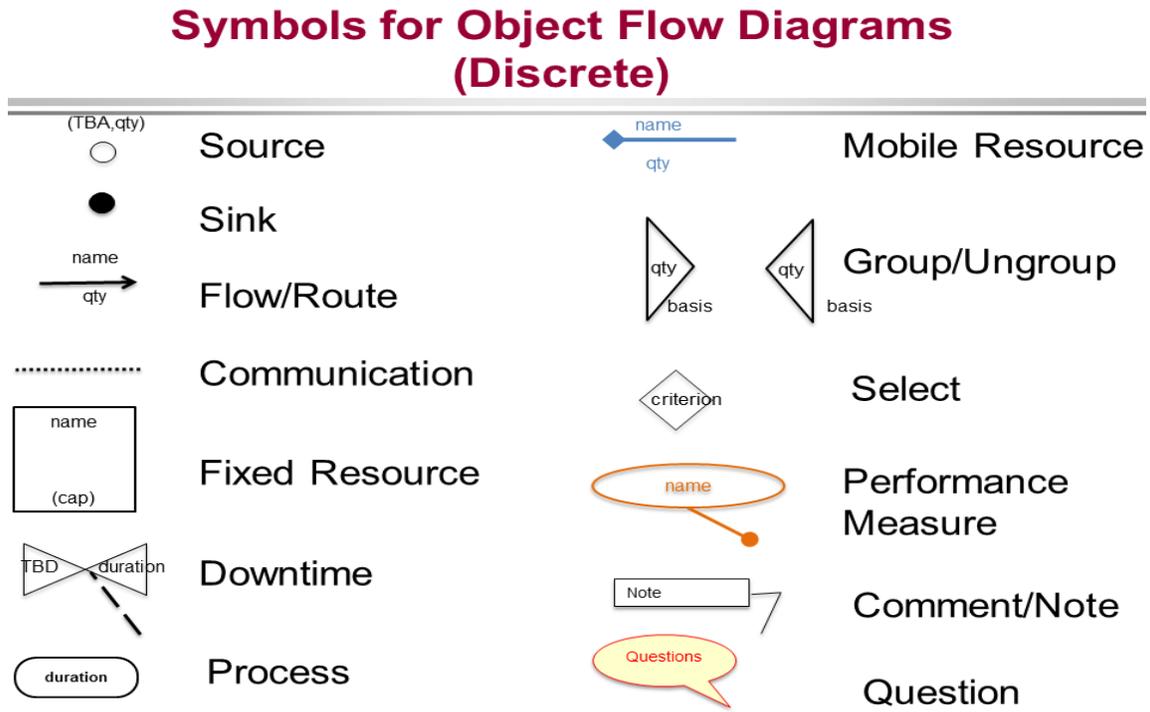
Algunas medidas de desempeño:

- ✓ Tiempo promedio en el sistema.
- ✓ Tiempo promedio en la cola.
- ✓ Número promedio ponderado en el tiempo de entidades en la cola.
- ✓ Tasa promedio de utilización de recursos.
- ✓ Entre otros relacionados.

3.4 Conceptualización del modelo

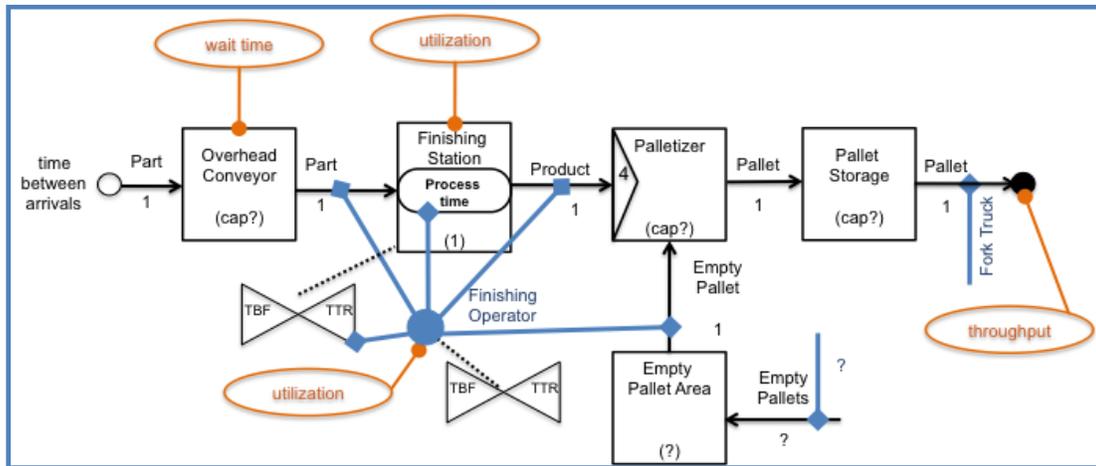
La conceptualización del modelo es tratar de representar visualmente el sistema real en el modelo de simulación, esto comprende la definición de los elementos operacionales y estructurales que caracterizan y componen en esencial al sistema o proceso a ser modelado. En la ilustración figura 3.4.1 y figura 3.4.2., se representa la simbología utilizada para un modelo conceptual empleando diagrama de flujo de objetos usados en el *software* de Flexsim y un ejemplo de su aplicación:

Figura 3.4-1: Simbología para diagramar objetos de flujo.



Fuentes: Beaverstock, Greenwood, Lavery, & Nordgren, 2012

Figura 3.4-2: Modelo conceptual empleando simbología objetos de flujo.



Fuentes: Beaverstock, Greenwood, Lavery, & Nordgren, 2012

3.4.1 Conceptos preliminares de modelamiento

En esta etapa se abstraen las características esenciales del sistema, se seleccionan y modifican las suposiciones que caracterizan al sistema o proceso, se elabora y enriquece el modelo hasta que resulte ser una representación útil. En esta etapa se recomienda iniciar con un modelo simple e ir agregándole mayor complejidad. Para Harrell et al., (2004) a partir de un realista, significativo y bien definido objetivo del estudio de simulación, es posible establecer el alcance y nivel de detalle del modelo a simular.

El alcance se refiere a qué elementos del sistema serán representados en el modelo y la delimitación del sistema a modelar, es decir, hasta dónde está relacionada la amplitud del área problemática.

El nivel de detalle del modelo está relacionado con la profundidad del sistema que representa, este nivel define la granularidad o resolución del modelo, en síntesis, el nivel de detalle se determina en gran medida por el grado de precisión requerido en los resultados.

Harrell et al. (2004), consideran que el modelamiento es más que conocer cómo usar un *software* de simulación. De hecho, el modelamiento es arte y ciencia.

3.4.2 Paradigmas de modelamiento

Para Harrell et al. (2004), un paradigma de modelamiento consta de constructos y lenguajes asociados, que indican como debe pensar acerca del sistema que está siendo modelado. En este sentido, el paradigma determina una particular vista del mundo que se debe tener cuando se modela un sistema. Aprender un producto de simulación requiere que primero se aprenda del paradigma del modelamiento usado por el producto. Los paradigmas de modelamiento difieren entre productos de simulación y algunos *software* de simulación tienen una orientación hacia el proceso y están soportados por modelamiento basado en objetos. Conceptualmente, el modelamiento basado en objetos es muy simple, ya que un objeto representa un objeto del mundo real tal como una máquina, un operador, una pieza en un proceso o un cliente. Un objeto es definido en términos de atributo y funcionalidad (comportamiento).

Los atributos son características que son asociadas con un objeto, tales como dimensiones, color, peso, entre otros. Son usados para llevar información acerca del objeto y ser utilizados para tomar decisiones y realizar reportes de salida. El comportamiento define la lógica operacional asociada con el objeto, por ejemplo: los alistamientos de máquinas, decisiones de enrutamiento y movimiento de materiales. Un evento puede activar un comportamiento, es decir, la caída de un inventario en un tanque a un nivel específico o entre otros. Los objetos pueden ser organizados en clases según comportamiento y atributos similares y se aplican conceptos propios de la programación orientada a objetos (OOP) tales como herencia, polimorfismo, encapsulamiento, etc.

3.4.3 Elementos estructurales del modelo

Según Harrell et al. (2004), los objetos de un modelo de simulación representan los elementos estructurales de un sistema, tales como máquinas, personas, ítems de trabajo, áreas de trabajo, entre otros. Por ejemplo Flexsim emplea objetos que son básicos para un modelo de simulación. Estos objetos tienen unos atributos y un comportamiento asociado a ellos. Una descripción más detallada de estos objetos se da a continuación:

- **Flowítem (entidades):** son objetos que se mueven a través del modelo, estos pueden ser productos, piezas, personas o cualquier objeto que pueda ser procesada por el sistema. En Flexsim, los *flowitem* son creados con el objeto denominado *Source*.
- **Source:** este objeto es la fuente de creación del o los *flowitems* y pueden asignárseles propiedades como tipo de entidad o color. Este objeto puede crear *flowitems* por una tasa entre llegadas, por una lista programada de llegadas o desde una secuencia definida de llegadas.
- **Queue:** este objeto representa una cola o área de almacenamiento usada para mantener los *flowitems* cuando un objeto deba de esperar a un proceso posterior.
- **Processor:** este objeto es usado para simular el procesamiento de *flowitems* en un modelo y puede representar una máquina, equipo u otros.
- **Sink:** este objeto es usado para destruir los *flowitems* o darles salida del sistema.

- **Recursos:** agentes usados en el procesamiento de entidades, por ejemplo: personas, montacargas, puente grúas, entre otros. Ayudan al manejo de materiales.
- **Rutas:** caminos de recorrido para los *flowitems* y recursos en el sistema. Y son modelados como red o redes de transporte de los *flowitems*.

3.4.4 Elementos operacionales del modelo

Estos elementos definen el comportamiento e interacción de diferentes elementos físicos del sistema. Entre los principales elementos operacionales a modelar están:

- **Enrutamientos:** define la secuencia y lógica de movimiento que siguen los *flowitems* a través del siguiente proceso. Para modelar apropiadamente el enrutamiento es conveniente tener a la mano el diagrama de flujo de alto nivel.
- **Operaciones con flowitems:** la operación con un *flowitem* define qué le ocurre al *flowitem* cuando entra en un *processor* y lo importante es conocer qué ocurre en términos del tiempo requerido, los recursos empleados y otra lógica que impacta el desempeño del sistema. La decisión de incluir la operación de un *flowitem* en el modelo debe basarse en si la operación impacta el flujo de *flowitems* de alguna manera, es decir, si una actividad de etiquetado es ejecutado en *flowitems* en movimiento sobre una banda, no es necesario modelar esta actividad, al menos que estas se presenten con mucha frecuencia y tenga un impacto significativo en la simulación. Son comunes las siguientes operaciones realizadas con *flowitems*.
 - ✓ **Consolidación de entidades o flowitems:** las entidades a menudo soportan operaciones en las cuales ellas son consolidadas o conectadas física o lógicamente con otras entidades; es decir, la consolidación de entidades son acumuladas (apilamiento) hasta cierta cantidad y luego agrupadas (lotes) en una sola entidad, esta puede ser temporal y posteriormente separarse dado el caso.
 - ✓ **Ensamble de entidades:** las entidades también pueden ser ensambladas en una locación. La diferencia con la consolidación es que las entidades se ensamblan en una entidad base o principal que debe estar presente en la

locación de ensamble, estos ensambles puede ser del caso temporal o permanente.

- ✓ **División de entidades:** en algunos procesos de entidades, una es convertida en dos o más nuevas, por ejemplo, una barra de acero puede ser cortada en varias secciones. Las entidades pueden ser divididas de dos maneras. la entidad original deja de existir o entidades adicionales son simplemente creadas (clonadas) desde la original y esta última sigue existiendo.
- **Arribos de entidades:** está definida como el momento, frecuencia y en las cantidades que ingresan al sistema. Arriban al sistema de producción o de servicios de diferentes maneras (en cada caso pueden arribar individual o en grupos):
 - ✓ **Arribos periódicos:** pueden ocurrir más o menos en el mismo intervalo de tiempo, es a menudo una variable aleatoria. Estos arribos frecuentemente son usados para modelar la salida de un proceso que alimenta al sistema que está siendo simulado. Por ejemplo, los monitores de computador pueden arribar desde una línea de ensamble para ser empacados, en un intervalo que está normalmente distribuido con media de 1.4 minutos y desviación estándar de 0.2 minutos.
 - ✓ **Arribos programados:** este arribo se da en horarios específicos, con alguna posible variación (un porcentaje de arribo temprano o tardío). Para modelarlos, es deseable leer la programación de arribos desde un archivo externo, especialmente cuando el número de arribos es grande. Un ejemplo es la programación de citas médicas.
 - ✓ **Arribos fluctuantes:** las entidades arriban a una tasa que fluctúa en el tiempo. Por ejemplo, la tasa de llegada de los clientes a un banco, restaurante, pasajeros, llamadas telefónicas, entre otros. Estos arribos se modelan generalmente con una tabla de frecuencias, cuya abscisa son las horas del día y la ordenada es la cantidad de arribos.
 - ✓ **Arribos desencadenados por eventos:** este tipo de arribo puede ser modelado mediante lógica de programación embebida en el modelo. Las entidades son introducidas al sistema por algún evento que se desencadena

internamente, tal como la terminación de una operación o la reducción de algún stock de seguridad.

- **Movimiento de entidades y recursos:** las entidades generalmente son movilizadas por recursos, tales como personas, montacargas, banda transportadoras, entre otros, siempre y cuando la lógica de movimiento se requiera en el sistema. El movimiento puede ser considerado de tres maneras en simulación:
 - ✓ Ignorar el movimiento.
 - ✓ Modelar sólo el tiempo que dura el movimiento, lo cual puede ser especificado por velocidad y distancia.
 - ✓ Modelar el movimiento usando una red de transporte.

La decisión de cuál método emplear en el modelo depende del nivel de detalle requerido para validarlo. Las siguientes reglas pueden ser útiles al momento de tomar la decisión:

- ✓ Si el movimiento es significativo comparado con los tiempos de procesamiento o puede ser incluido en estos últimos, puede ser ignorado.
 - ✓ Si los tiempos de movimiento son significativos pero la congestión de tráfico es poca, se puede modelar sólo el tiempo que dura el movimiento.
 - ✓ Si los tiempos de movimiento son significativos y la congestión de tráfico es intensa, se debe modelar una red de transporte.
-
- **Reglas de acceso a locaciones y recursos:** gran parte de la actividad en simulación es gobernada por reglas que regulan el acceso de las entidades a las locaciones y recursos para su procesamiento. Entre estas reglas están:
 - ✓ **Uso de prioridades:** los recursos y locaciones pueden ser requeridos por las entidades según un esquema de prioridades, es decir, estas prioridades se les asigna a la entidad para darle un valor de uso o probabilidades que sea atendida la entidad con respecto al recurso o locación.
 - ✓ **Preempción:** algunas veces es necesario tener un recurso o locación que responda inmediatamente a una tarea, interrumpiendo la actividad que actualmente está

- ejecutando. A esta capacidad de respuesta que tiene las locaciones o recursos se le denomina preempción.
- ✓ **Reglas de selección de tareas:** mientras las prioridades determinan el orden en el cual las entidades usan una locación o recurso, las reglas de selección de tareas determinan a cuáles entidades que están esperando se les da atención de prioridad para ser procesadas por una locación y recurso. Entre estas reglas están en la simulación como:
 - SPT (*shortest processing time*).
 - LPT (*long processing time*).
 - EDD (*earliest due data*).
 -
 - **Programación de recursos:** los recursos frecuentemente tienen tiempos programados durante el no disponible, por ejemplo, periodos fuera del turno, descansos y mantenimientos. Es necesario entonces modelar estas situaciones de disponibilidad y no disponibilidad si se quiere tener un modelo válido, según las siguientes situaciones:
 - ✓ Salir de turno en la mitad de una tarea: en tareas cortas no es problema, ya que simplemente en el modelo se le permite al recurso terminar la tarea y posteriormente luego el programa retoma el control. Para tareas de mayor duración, no es usualmente deseable mantener al recurso hasta que termine la tarea, por lo que hay que considerar las siguientes situaciones en el modelamiento:
 - No comenzar la tarea en primer lugar.
 - Interrumpir la tarea y permitir al recurso salir del turno.
 - Si la tarea está casi terminada, siga adelante y termine la tarea.
 - **Paradas y reparaciones:** Los recursos y locaciones pueden presentar tiempo no disponible debido a razones como fallos en equipos o interrupciones por parte del personal. Entre los principales tipos de paradas que se pueden modelar están:
 - ✓ **Paradas por tiempo total transcurrido:** es decir, los tiempos de descanso de los operarios o los mantenimientos preventivos.

- ✓ **Paradas por tiempo de uso:** este tipo de paradas son programas a las horas de operación del equipo, es decir, un cambio de aceite a una máquina que con frecuencia debe de cambiarse cada 1.200 hrs de uso y otras dadas por el fabricante de la máquina.

- ✓ Paradas por número de veces de uso; esta ocurre según la frecuencia que una locación es usada. También se puede dar por el número de productos o servicios procesados.

- ✓ **Resolución de las paradas:** dependiendo de la naturaleza de la información disponible sobre las paradas, estas pueden ser modeladas de las siguientes maneras:
 - **Ignorar las paradas:** este caso aplica si no hay datos disponibles sobre éstas o no afectan el desempeño del sistema.
 - **Incrementar los tiempos de procesamiento:** consiste en reducir la capacidad de producción de la máquina por un porcentaje, es decir, si una máquina tiene una capacidad efectiva del 100% por hora y el 10% del tiempo sufre paradas, entonces la capacidad efectiva es reducida a 90 piezas por hora. La desventaja de este enfoque es que el ajuste es determinístico y no tiene en cuenta la variabilidad.
 - **MTBF/MTTR:** cuando se modela una parada se definen dos componentes. El primero es el tiempo medio entre fallos y el segundo el tiempo medio para reparar. El uso de distribuciones estadísticas para estos dos componentes debe ser representados por distribuciones como la *Weibull*, *Normal*, *exponencial* y *Lognormal* que reflejan la variabilidad de estos fallos, cuando se determina la distribución para el tiempo éstos, es necesario hacer consideraciones sobre si los eventos por paro ocurren en el tiempo transcurrido o el tiempo de uso del recurso o locación.

- **Uso de lógica de programación:** Dado que no siempre el comportamiento del modelo se puede representar con los constructos provistos por el software de simulación, se hace necesario usar lógica de programación para representar este comportamiento:
 - ✓ Uso de probabilidades para modelar el comportamiento probabilístico.
 - ✓ Uso de atributos para modelar lógica de decisión especial.
 - ✓ Uso de variables globales para modelar estadísticas no estándares.
 - ✓ Uso de variables locales.

- **Temas misceláneos de modelamiento**
 - ✓ **Modelando ocurrencias raras:** el analista de simulación generalmente está interesado en el comportamiento normal del sistema, pero algunas situaciones raras tienen un impacto significativo en la operación del sistema y conllevan un paro completo de la operación. Entonces solo en aquellos casos en que el interés es evaluar los efectos de ocurrencias raras se deben modelar.

 - ✓ **Modelamiento a gran escala:** a veces puede ser deseable modelar un gran sistema tal como una fábrica completa. En ese caso, una buena idea es la partición del modelo en varios sub modelos y enfrentar el problema a una pequeña escala primero, posteriormente cada sub modelo ha sido construido y validado, es posible mezclarlos en uno solo.

 - ✓ **Modelando costos:** En algunos casos es deseable incluir los costos en el modelo para determinar mejor la solución costo beneficio para el problema. Si por ejemplo, dos operarios de una línea de ensamble se desempeñan igual que tres robots, la decisión se basará en costos en lugar desempeño.

3.5 Recolección y análisis de datos

Esta etapa se hace en paralelo con la definición del sistema y la construcción (conceptualización) del modelo. Este proceso implica la recopilación de datos de entrada, el análisis de éstos y el su uso en el modelo de simulación. Dichos datos pueden ser obtenidos a partir de otros de tipo histórico o recogidos en tiempo real. Implica la identificación de la distribución teórica de probabilidad que representan los datos de entrada. Su implementación en el modelo consiste en especificar la distribución teórica en el código del modelo de simulación. Para Chung (2004), en un proyecto de simulación, el uso final de los datos de entrada es conducir la simulación. Una recopilación sobre la recolección y análisis de datos, según Kelton y Law (2007), Harrell et al. (2004) y Chung (2004) comprende:

3.5.1 Directrices a tener en cuenta para la recolección de datos

La recolección de datos debe de ejecutarse de manera sistemática, observando ítems específicos de información que se requieren para construir el modelo. Las siguientes directrices pueden ayudar a mantener el enfoque al analista:

- **Identificar eventos desencadenantes:** definidas las actividades que ocurren en el sistema a ser modelado, es importante identificar las causas o condiciones que desencadenan las actividades. Recomiendan hacer preguntas como: ¿qué causa las paradas de máquinas?, ¿qué causa el movimiento de materiales?, ¿qué desencadena el uso de un particular recurso? Para ser válido, un modelo necesita capturar los eventos desencadenantes correctos que inician las actividades dentro del sistema.
- **Enfocarse en los factores de impacto:** la exclusión de algunos datos que tienen poco o ningún impacto en el desempeño del sistema no es necesario recolectarlos, pero sí de aquellos que son representativos en el sistema y generan un impacto fuerte a este. Por ejemplo, actividades de mantenimiento preventivo que se hacen fuera de turno o paradas de máquinas de rara ocurrencia deben ser ignoradas.

- **Aislar los tiempos reales de la actividad:** al determinar los tiempos de actividades, es importante aislar solo el tiempo que toma hacer la actividad en sí, excluyendo el tiempo de espera por materiales y recursos requeridos para ejecutar la actividad.
- **Buscar agrupaciones comunes:** cuando se trata con cientos de tipos de partes o perfiles de clientes, es de gran ayuda buscar grupos o patrones comunes. Por ejemplo, si se está modelando un proceso en el que se manejan 300 tipos de entidades, puede ser muy difícil obtener información sobre la mezcla exacta de productos toda la variedad de rutas que pueden ocurrir. La solución es reducir los datos a comportamientos y patrones comunes. Una manera de agrupar datos comunes es identificar primero categorías generales dentro de las cuales los datos pueden ser asignados. Entonces el porcentaje de casos que caen dentro de cada categoría es calculado o estimado. A partir de estas categorías es posible recolectar los demás datos (enrutamientos, mezclas de categorías, entre otros).
- **Separar las variables de entrada de las variables de respuesta:** las variables de entrada definen cómo opera el sistema (tiempos de actividad, secuencias de enrutamiento, etc.) y son el foco de atención del analista en esta etapa de recolección. Las variables de respuesta describen cómo responde el sistema a un conjunto dado de variables de entrada (cantidad de WIP, utilización de recursos, etc.).
- **Enfocarse en la esencia:** La definición de un sistema para propósitos de modelamiento debe capturar las relaciones causa – efecto e ignorar los detalles sin sentido. A estos se le denomina la abstracción del sistema y busca definir su comportamiento esencial. Un sistema debe ser abstraído al más alto nivel sin perder la esencia de su operación.

Debe quedar claro que no concierne en el modelamiento la naturaleza de la actividad que se está ejecutando (cortar, torneear, inspeccionar, etc.) sino en el impacto que la actividad tiene en el uso de los recursos y el retraso en el flujo de entidades.

3.5.2 Determinar los requerimientos de datos

Mediante esta tarea se determina qué datos se requieren para conceptualizar el modelo. Esto es principalmente orientado por el alcance del modelo y el nivel de detalle requerido para obtener los objetivos de la simulación, los datos del sistema pueden ser caracterizados como datos estructurales, operacionales y numéricos.

- **Datos estructurales:** involucran datos que describen los objetos del sistema que va a ser modelado. Esto incluye elementos como entidades (productos, clientes, pacientes, pasajeros, etc.), recursos (operadores, equipo de transporte) y locaciones (áreas de espera, estaciones de trabajo, instalaciones, distribución en planta, máquinas). Es importante incluir todos los componentes relevantes que afecta el comportamiento del sistema.
- **Datos operacionales:** estos datos explican cómo opera el sistema, esto es, cuándo, dónde y cómo tienen lugar los eventos y actividades. Consiste en toda la información lógica o funcional del sistema, como rutas de producción, programas de trabajo, comportamiento de paradas y asignación de recursos.
- **Otros datos:** los datos numéricos proveen información cuantitativa acerca del sistema. Ejemplo de datos numérico, entre otros, son, capacidades, tasas de arribo, tiempos de procesamiento, tiempo entre fallas, tiempos de reparación.

3.5.3 Identificar fuentes de datos

Para la construcción del modelo es necesario extraer la información de reportes de algún sistema de información, entrevistas al personal, entre otras. El analista debe aprovechar todas las fuentes de información que pueda encontrar para recolectar los datos de entrada. Estos datos pueden ser históricos, anecdóticos u observados. Raramente toda la información requerida para construir el modelo está disponible desde una sola fuente.

Las fuentes de información que puede contar el analista son:

- Registros históricos

- Especificaciones de los fabricantes de equipos.
- Estimaciones o experiencia de los operadores.
- Estimaciones o experiencia de los gerentes
- Documentación y/o diagramas de proceso del sistema.
- Entrevistas al personal.
- Comparaciones con industrias similares.
- Estimados de diseño.
- Investigaciones publicadas del caso.
- Captura automática de datos.

3.5.4 Recolección de datos

Cuando se recolectan datos, se recomienda ir de lo general a lo particular. Los principales aspectos a considerar en la recolección de datos son:

- **Dispositivos y modos de acopiamiento de datos**
 - ✓ **Dispositivos:** si los datos de entrada se recogen en tiempo real, se puede soportar esta recolección con la ayuda de dispositivos electrónicos. El uso de dispositivos de grabación de video ofrece una posibilidad para la recogida de datos y adicionalmente permite la reproducción indefinida de veces para observar cada detalle de la actividad.
 - ✓ **Modos de recolección:** es necesario definir los intervalos y unidad de tiempo que deben ser usados en la recolección. Un analista novato puede frecuentemente registrar las observaciones sobre arribos de entidades en tiempo absoluto o tiempo reloj. Con este modo de recolección se requiere trabajo extra necesario para convertir el tiempo absoluto a tiempo relativo para permitir que los tiempos entre arribos puedan ser calculados. Otro tema importante es el relacionado a las unidades de tiempo a emplear y estas se realicen en la misma unidad de tiempo o se conviertan en el mismo punto de recolección, es decir, para propósitos de cálculos, si los tiempos se tomaron en

minutos y segundos, los segundos tendrían que ser tomados en fracciones de minutos.

- ✓ **Otras consideraciones en la recolección de datos:** cuando el analista está tomando los tiempos de operación, generalmente algunos operarios pueden sesgar un poco los datos, dado que pueden acelerar su ritmo de trabajo para parecer ser más productivos, otros operarios deliberadamente pueden frenar el ritmo de trabajo para evitar los altos estándares de trabajo que se establecen. Si los datos se sesgan por cualquiera de estas maneras, se producen otros inexactos que no reflejen la realidad del sistema. Así, el mejor curso de acción es proporcionar un breve informe a los trabajadores. En éste, el analista debe explicar los efectos de la recopilación de datos.

- **Definición del flujo de entidades:** la primera tarea en que se debe enfocar la recolección de datos es definir el flujo básico de entidades a través del sistema. Este se define siguiendo el movimiento de entidades a través del sistema. El flujo global de entidades es mejor descrito usando un diagrama de flujo de entidades y en él se describe cualquier ramificación que le pueda ocurrir al flujo, como enrutamientos a centros de trabajo alternativos o ciclos de reproceso. El propósito es documentar el flujo total de entidades en el sistema y proveer una ayuda visual para comunicar esto a otros.
- **Desarrollo de una descripción de la operación:** ésta describe y explica cómo se procesan las entidades a través del sistema. Esta descripción debe ser escrita paso a paso en forma narrativa o puede representarse en forma tabular. La descripción de la operación debe identificar por cada tipo de entidad en cada locación del sistema:
 - ✓ Los requerimientos de tiempo y recursos de la actividad u operación.
 - ✓ Dónde, cuándo y en qué cantidades se enrutan las entidades a la siguiente locación.
 - ✓ Los requerimientos de tiempo y recursos para moverse a la próxima locación.
- **Decidiendo detalles incidentales y refinando los valores de datos:** una vez un modelo básico ha sido construido y probado, pueden añadirse detalles adicionales del

proceso tales como paradas, alistamientos y prioridades de trabajo. La decisión de incluir o no un particular elemento del sistema, por ejemplo una parada de máquina, es más fácil de tomar con un modelo básico corriendo, dado que el potencial impacto del elemento y el tiempo disponible para implementarlo en el modelo es mucho más claro que al principio del proyecto de simulación. En este punto, cualquier valor numérico tal como tiempos de actividad, tasas de arribo y otros, también pueden ser concretados.

3.5.5 Establecer suposiciones de trabajo

No mucho después de que la recolección de datos se ha iniciado, es posible darse cuenta que cierta información no está disponible o tal vez no es confiable. Datos completos, precisos y actualizados rara vez se pueden obtener, especialmente cuando se modela un sistema nuevo sobre el cual se conoce muy poco, es entonces que surge la alternativa de hacer suposiciones acorde al sistema real. Otro enfoque en el tratamiento de los supuestos es la ejecución de tres escenarios diferentes (escenario optimista, medio y pesimista) de esta forma se permitirá al analista realizar una distribución triangular, la cual permite modelar estos tres tipos de escenarios.

3.5.6 Análisis estadístico de los datos de entrada

Para que los datos de entrada sean útiles en el modelo de simulación, es importante analizarlos e interpretarlos para que la operación del sistema sea correctamente representada en el modelo.

Debe determinarse características de los datos tales como independencia (aleatoriedad), homogeneidad (los datos provienen de la misma distribución) y estacionalidad (la distribución de los datos no cambian en el tiempo). Empleando el software de análisis de datos que trae el paquete de simulación Flexsim tales como ExpertFit se facilita el análisis de estos para su utilidad y evaluar su ajuste a una distribución teórica de probabilidad. Y posteriormente someterlas a pruebas estadísticas con el propósito de determinar si los datos son independientes e

idénticamente distribuidos. Las siguientes pruebas estadísticas para el tratamiento de los datos y distribuciones teóricas más comunes:

- Distribuciones de Bernoulli, uniforme, exponencial, triangular, normal, poisson, weibull, Log logistic, entre otros relacionados.
- Histogramas para un enfoque gráfico.
- Gráfico de dispersión
- Gráfico de auto-correlación.
- Pruebas de bondad de ajustes como Chi-cuadrado, Kolmogorov- Smirnov y Anderson – Darling.

Cuando todos los datos relevantes han sido obtenidos, analizados y convertidos a una forma útil para ser usados por el modelo de simulación, es aconsejable documentar los datos empleando tablas, diagramas relacionales y listas de suposiciones, incluyendo las respectivas fuentes de datos.

3.6 Codificación del modelo

El objetivo en esta etapa del modelo es traducir el modelo conceptual en uno informático que pueda ser usado para realizar la etapa de experimentación. Es importante aquí que el analista decida qué *software* de simulación utilizará para codificar el modelo y conozca el lenguaje de programación del software.

3.6.1 Comienzo de la codificación

A veces, la parte más difícil de la programación de la simulación es comenzar. Afortunadamente para este punto ya debe de existir un diagrama de flujo de alto nivel, con el cual es más fácil apoyarse de la representación conceptual de los principales componentes e interacciones del sistema. Debido a que el analista desarrolló el diagrama de flujo de alto nivel desde la perspectiva de las entidades del sistema, existe una correlación directa con el programa real de simulación. La principal diferencia entre los dos, es el grado de detalle. Si el analista tiene

una comprensión conceptual de alto nivel de cómo funciona el sistema, será mucho más fácil desarrollar el código del programa.

3.6.2 Versión administrativa de la codificación

Esta sección se ocupa de cuestiones administrativas para ayudar al analista en la creación y mantenimiento de un sistema de archivos organizado:

- **Subdirectorios del proyecto:** los analistas deberían comenzar todos los proyectos de simulación mediante la creación de un subdirectorio separado para cada proyecto individual. Estos subdirectorios pueden residir tanto en el subdirectorio nativo del *software* de simulación, que contiene archivos del programa ejecutable de simulación. Entre las ventajas de ubicar estos subdirectorios en el subdirectorio nativo están:
 - Los archivos de proyecto son accesibles inmediatamente después de comenzar el programa de simulación.
 - Los archivos de proyecto están en el mismo lugar independientemente del equipo específico que los utiliza.
 - Existen menos problemas de navegación para acceder a las librerías de soporte a la simulación.

Entre las ventajas de utilizar un subdirectorio dedicado al proyecto están:

- El subdirectorio del programa de simulación permanece limpio.
 - Es más fácil hacer copias de seguridad.
 - La instalación o desinstalación del *software* de simulación no afecta a los subdirectorios del proyecto.
-
- **Almacenamiento de los programas de simulación:** los analistas de simulación deben utilizar en lo posible un enfoque de “divide y vencerás” para el desarrollo de programas complejos de simulación, pues, puede ayudar a reducir considerablemente el esfuerzo de desarrollo, con este enfoque el analista comenzará con una simulación relativamente simple. Una vez el programa se ejecuta correctamente, el analista puede

ir añadiendo componentes adicionales e ir guardando en diferentes versiones y de mayor sofisticación en forma continua. Existen dos opciones en términos de gestión en el almacenamiento de las versiones de los archivos:

- **Versión de administración de archivo simple:** esta versión consiste en sobre-escribir el archivo original con el mismo nombre, es decir, que el archivo será siempre el mismo.
 - **Versión de administración de archivos múltiples:** ésta consiste en guardar continuamente cada nueva versión del archivo con un nombre diferente, esto permite al analista que en caso de un daño de algún archivo se pueda regresar a la antepenúltima versión y no se perderá mucho menos trabajo que si lo haría con el método de archivo simple.
-
- Uso de múltiples computadoras en el desarrollo del proyecto:
 - Copia de seguridad de los archivos de la simulación
 - Documentando el programa de simulación.

3.6.3 Organización del programa de simulación

El analista debe mantener el programa de simulación lo más organizado posible. Por lo tanto, debe intentar mantener todos los componentes similares o componentes que están asociados unos con otros en la misma vista. De otra parte el analista debe hacer todos los esfuerzos posibles para el uso contante de convenciones para nombrar (mnemotécnicos): colas, recursos, transportadores, etc. La convección de nombres debe describir la naturaleza exacta de los componentes. De otro lado, la mayoría de los modelos de simulación, por lo general terminan siendo modificados de alguna manera, con el fin de representar los diferentes escenarios de experimentación. Dado que el modelo puede llegar a ser reciclado, casi siempre es de beneficio para el analista tratar de construir el modelo con la mayor flexibilidad razonable y posible, a fin de minimizar el esfuerzo adicional que se requiere para hacer una modificación futura. Este enfoque es conocido en el mundo de la ingeniería industrial como “hacerlo bien la primera vez”.

Finalmente, Harrell et al. (2004) sugiere las dos siguientes estrategias generales a seguir en el proceso de translación del modelo conceptual a un modelo de simulación por computadora:

- **Refinamiento progresivo:** una característica interesante en simulación es que los modelos no tienen que incluir todos los detalles finales antes de que funcionen. Esto permite aplicar una estrategia de perfeccionamiento progresivo que añade componentes al modelo por etapas en lugar de todos a la vez. La complejidad en la construcción de un modelo no debe ser nunca subestimada y siempre es mejor empezar de lo simple e ir añadiéndole complejidad, en lugar de crear todo un modelo complejo de una sola vez.
- **Expansión gradual:** los modelos que tienen un amplio alcance a veces son más fáciles de construir por fases en la que se añaden secciones o componentes paulatinamente. Este método permite que una parte del modelo sea construido, probado y depurado antes de añadir nuevas secciones, logrando que la tarea de construcción sea más manejable.

3.7 Verificación del modelo

Esta etapa se centra en asegurar que el modelo opera según el diseño previsto, es decir, que el modelo de simulación refleje correctamente al modelo conceptual. Si los parámetros de entrada y la estructura lógica del modelo son correctamente representados en el computador, se considera que la verificación ha concluido.

En esencia la verificación es el proceso de depurarlo, intentar detectar errores no deseados en la lógica o datos del modelo y removerlos, estos errores son los llamados *bugs* y son de dos tipos: sintaxis y semántica. Los errores de sintaxis son gramaticales que incluyen adición no intencional u omisión. La omisión de un punto decimal o errores en la expresión de una línea en la programación puede dramáticamente impactar la salida de la simulación. La mayoría de *software* de simulación poseen mecanismos de detección mediante algún mensaje de alerta. Los errores de semántica son asociados con el significado o intención del modelador y son difíciles de detectar. Generalmente son errores lógicos que causan un comportamiento del modelo de simulación, diferente al esperado. Harrell et al. (2004) sugieren las siguientes técnicas que ayudan a la verificación del modelo de simulación.

3.7.1 Medidas preventivas

Lo más importante es concentrarse en lo que se está haciendo en el modelo. Los datos deben chequearse doblemente para descartar que no existan errores de transcripción. Hoover & Perryv (1990) describen cinco principios básicos a seguir cuando se programa estructuralmente:

- **Diseño de arriba-abajo:** el modelo de simulación y su código asociado empiezan en un alto nivel y se van agregando congruentemente áreas de bajo nivel.
- **Modularidad:** el modelo es construido en módulos o divisiones lógicas con el fin de simplificar el desarrollo y depuración del modelo. En lo posible, separar los módulos de datos de los módulos de lógica.
- **Módulos compactos:** los módulos deben mantenerse tan pequeños y simples como sea posible.
- **Refinamiento paso a paso:** el modelo es construido con una complejidad progresiva. En ese proceso se va de lo general a lo específico. En cada paso se hace una prueba de razonabilidad del modelo.
- **Control estructurado:** instrucciones de transferencia de control no estructuradas tal como GOTO debe ser evitadas.

3.7.2 Técnicas de verificación

- **Revisiones de código del modelo:** el propósito es chequear errores e inconsistencias en el modelo de simulación, esta revisión puede ser de dos maneras:
 - ✓ De abajo-arriba: los módulos de más bajo nivel son probados y verificados primero. Seguidamente, las interfaces entre dos o más módulos son probadas. Esto se repite hasta que el modelo pueda ser verificado como un sistema único.
 - ✓ De arriba-abajo: la prueba de verificación empieza con el módulo principal y se mueve hacia abajo gradualmente hasta los módulos de más bajo nivel, se está más interesado en las salidas de los módulos dadas las entradas. Si se presentan discrepancias, debe hacerse un análisis de código de bajo nivel.

- **Chequear que la salida sea razonable:** en cualquier modelo de simulación, hay relaciones operativas y valores cuantitativos que son predecibles durante la simulación. Un ejemplo de relación operativa predecible es el enrutamiento que ocurre cada vez que una operación es terminada. Un valor cuantificable predecible puede ser el número de cajas en una banda transportadora, pues siempre estará entre 0 y la capacidad de la banda. Para modelos simples, una manera que ayuda a determinar la razonabilidad de la salida es reemplazar los tiempos aleatorios y salidas probabilísticas con tiempos y salidas determinísticas en el modelo. Esto permite predecir de manera precisa las salidas y así poder chequear que ésta sea razonable.
- **Viendo la animación:** la animación permite visualmente verificar que la simulación opera de la manera que se espera y detectar errores o comportamientos inconsistentes que son descubiertos en la animación y que de otra manera pasarían inadvertidos. La animación es usualmente más útil en identificar problemas que en describir su causa. Los siguientes son síntomas frecuentes que aparecen con la animación y que revelan problemas subyacentes en el modelo:
 - ✓ la simulación se ejecuta horas e incluso días y se congela de repente.
 - ✓ una simulación con terminación debe tener el sistema vaciado al final de la simulación, pero finaliza con algunas entidades varadas en algunas locaciones.
 - ✓ un recurso se encuentra inactivo habiendo trabajo esperando a ser realizado.
 - ✓ un enrutamiento en particular nunca es ejecutado.
- **Usando facilidades de seguimiento y depuración:** la información de seguimiento y depuración provee información textual detallada de que está ocurriendo durante una simulación. Esto le permite al modelador ver exactamente lo que ocurre subyacente a la animación. Los mensajes de seguimiento describen cronológicamente qué ocurre durante la simulación, evento por evento mediante registros. Estos registros de seguimiento o trazabilidad de los eventos pueden ser mostrados directamente en pantalla o escritos en un archivo plano para un análisis posterior. Otra manera de hacer seguimiento al comportamiento durante una corrida de simulación es a través del uso de un depurador. El depurador de simulación es una herramienta que muestra los pasos que se dan en una lógica determinada del modelo. Usando las características

de trazabilidad y depuración, se pone de relieve la ocurrencia de eventos y las variables de estado puede ser examinadas y comparadas con cálculos hechos a mano para ver si el programa de simulación está operando como debe ser.

3.8 Validación del modelo

Esta etapa consiste en comparar el comportamiento del modelo de simulación versus el comportamiento del sistema real y lo importante es llegar a la precisión del sistema real, en otras palabras, consiste en comparar qué tan fielmente el modelo representa al sistema real, tanto en su estructura como en su operatividad. Existen dos tipos principales de validación. La primera es la validez aparente dada por la animación y significa que el modelo, por lo menos en su aspecto superficial, representa la realidad. La segunda es la validez estadística que implica una comparación cuantitativa entre el rendimiento de salida del sistema actual y el modelo (Law & Kelton, 2007). El analista de simulación debe lograr ambos tipos de validación, a fin de obtener confianza en que el modelo es correcto. Para modelos que tengan lógica compleja de control, la animación gráfica puede ser usada de forma efectiva como una herramienta de validación. Por último, la presentación de resultados debe analizarse para ver si estos parecen razonables. Si las circunstancias lo permiten, el modelo puede ser, incluso, comparado con el actual sistema para ver la forma como se corresponden. Si estos procedimientos se realizan sin tener que afrontar una discrepancia entre el sistema real y el modelo, se dice que éste último es válido. De lo contrario es necesario realizar una validación estadística del modelo.

3.8.1 Validación visual

Con el fin de ejecutar correctamente el proceso de validación visual, el analista debe asegurar de que la animación del modelo tiene suficiente fidelidad visual con respecto al proceso real. Se recomienda efectuar esta validación durante el periodo de desarrollo del modelo. Este tipo de validación se logra normalmente con la ayuda de usuarios del sistema modelado. Un usuario es simplemente un individuo o grupo de individuos que se consideran bien informados sobre el sistema en estudio, ya que brinda credibilidad al modelo.

3.8.2 Validación estadística

Implica un objetivo cuantitativo en términos de la comparación entre el actual sistema y el modelo de simulación. Por tanto, se requiere recopilar datos, tanto del uno, como del otro. Hay un total de cuatro tipos diferentes de pruebas de hipótesis estadísticas para comparar el sistema con el modelo (Johnson et al., 1999). La selección de la prueba de comparación apropiada depende de si los datos se distribuyen normalmente, se aparean naturalmente, o tienen igual varianza. Para este caso del proyecto se podría validar el modelo con la visual y la prueba T que se aplicará en el capítulo 4 y consiste en:

Prueba T: se utiliza cuando los datos se distribuyen normalmente y los conjuntos de datos (sistema y modelo) tienen varianza similar, es decir, que mediante una prueba de hipótesis se podría determinar con un nivel de significancia si existe diferencia estadística entre el modelo y el sistema, para esto es importante calcular las medias y las desviaciones estándar de ambos conjuntos de datos y continuar con el procedimiento siguiente:

1. Se plantean las hipótesis, las medias sería

Ho: las medias de ambos grupos son iguales.

Ha: las medias de ambos grupos no son iguales.

$$H_0 = \mu_1 = \mu_2$$

$$H_a = \mu_1 \neq \mu_2$$

En este caso si se acepta la Ho, se considera válido el modelo de simulación.

2. Seleccionar un nivel de significancia α
3. Determinar el valor crítico de t en el nivel de significancia $\alpha/2$ y el número de grados de libertad menos 2.
4. Se calcula un estadístico t de acuerdo con la siguiente expresión:

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}$$

Dónde:

T: Estadístico a calcular.

$\bar{x}_1 =$ *Media de la primera alternativa*

$\bar{x}_2 =$ *Media de la segunda alternativa*

$s_1^2 =$ *Varianza de la primera alternativa*

$s_2^2 =$ *Varianza de la segunda alternativa*

$n_1 =$ *Tamaño de muestra de la primera alternativa*

$n_2 =$ *Tamaño de muestra de la segunda alternativa*

5. Rechazar la hipótesis nula si el estadístico T es mayor o inferior al valor crítico.

3.9 Diseño experimental

En esta fase se busca respuestas a preguntas sobre el sistema objeto de estudio a través de la información que proporcionan los experimentos ejecutados, buscando respuesta a preguntas de tipo ¿qué pasaría si...? Que mediante a diferentes escenarios se plantea la posibilidad de cambios o modificaciones en el sistema a fin de encontrar valores de respuestas que impacten el sistema real.

Para Harrell et al. (2004), la simulación es básicamente una aplicación del método científico. Se parte de una teoría de por qué ciertas reglas de diseño o estrategias de manejo de sistemas, son mejores que otras. Con base en estas teorías, el diseñador desarrolla hipótesis que pone a prueba a través de la simulación. Basándose en los resultados de la que ésta le ofrece, el analista llega a conclusiones acerca de la validez de su hipótesis. En un experimento de simulación las variables de entrada son independientes, definen el modelo y pueden ser manipuladas o modificadas. Los efectos de esta manipulación en variables dependientes o respuesta, se miden y se correlacionan.

Con respecto a la estimación de las variables de respuesta, se ejecuta repetidas veces (réplicas) el modelo correspondiente a una misma configuración (escenario) con diferentes muestras aleatorias de *input*, lo que proporciona variaciones en los resultados, debido a las variaciones aleatorias en las muestras utilizadas.

Existen dos enfoques, entre otros, para realizar el diseño de experimentos con simulación, por escenarios y por factores. Igualmente una técnica para optimizar lo constituye la metodología de superficies de respuestas.

3.10 Documentación y análisis de salida

En esta etapa la documentación se hace sobre el programa y el progreso del estudio. La documentación del programa permite a otros entender cómo opera. La documentación sobre el progreso del estudio reporta una historia escrita del proyecto de simulación. Este reporte da una idea de la cronología del trabajo hecho y las decisiones tomadas. Se sugieren reportes frecuentes (mensual, al menos) los cuales proveen un comprensivo registro de logros, cambios requeridos, decisiones clave y otros ítems de importancia. El resultado del análisis debe ser reportado clara y concisamente en un informe final, este facilitará a los usuarios del modelo revisar la formulación final, los sistemas alternativos que se abordaron, los criterios por los cuales las alternativas fueron comparadas, los resultados de los experimentos y la solución recomendada al problema.

3.11 Implementación

La implementación depende de que tan bien se realizaran las diez etapas previas y cuán profundamente el analista ha involucrado al usuario final del modelo durante el estudio con simulación.

4 Capítulo IV: simulación y análisis de resultados

Para llevar a cabo este trabajo final basado en simulación fue necesario planear el proyecto, estructurado sobre principios y prácticas de administración de proyectos. La metodología que se llevó a cabo fue la propuesta por Giraldo (2014), adaptada de varios autores y mencionada en el capítulo 3 del presente trabajo. Ver figura 3.1.

4.1 Formulación del problema

¿Cómo impacta el mantenimiento preventivo y correctivo al nivel de producción (envasado) de la empresa Colgas de Occidente en su planta de Manizales?

4.2 Establecimiento de objetivos y plan general del estudio.

4.2.1 Establecimiento de objetivos:

- Identificar los tiempos probabilísticos de procesamiento de los equipos.
- Identificar los tiempos probabilísticos/determinísticos entre fallos/mantenimientos y reparaciones de los equipos críticos en la operación de envasado.
- Determinar los cuellos de botella presentes en el envasado de cilindros y cómo estos pueden ser eliminados para aumentar las unidades envasadas.
- Calcular el *throughput* del sistema de producción del Colgas.
- Estimar cuantos operarios se requieren para cumplir con los niveles de producción planeados.
- Evaluar tasas de utilización de los recursos que intervienen en el proceso de envasado.
- Comparar el desempeño del sistema actual frente a un modelo planteado de simulación en relación a la programación de producción por lotes de referencias con el fin de medir su impacto.

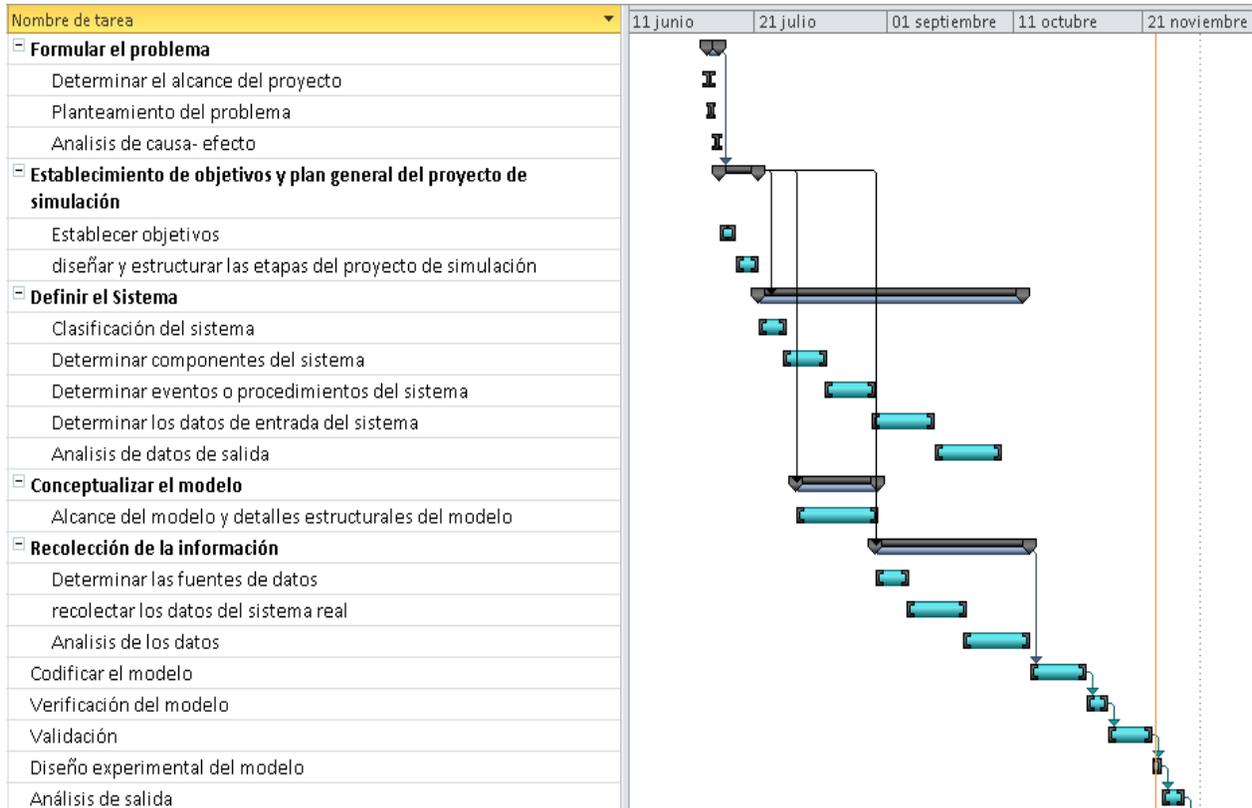
4.2.2 Elaboración del plan general

Como todo proyecto de simulación este debe planearse en base a principios y prácticas de administración de proyectos y para el caso fue importante implementar y desarrollar estos principios. Se diseñó el plan de trabajo con un diagrama de Gantt el cual permitiera observar el desarrollo de la estructura del presente trabajo y los recursos empleados. para este propósito fue necesario emplear el *software* de simulación Flexsim que posee la Universidad Nacional de Colombia, el levantamiento de la información fue suministrada por parte de la empresa Colgas de Occidente S.A. E.S.P objeto de estudio y adicionalmente se cuenta con el apoyo del personal de interés de la empresa.

A continuación se detalla en el diagrama de Gantt la elaboración del presente trabajo.

Figura 4.2-1: Diagrama de Gannt del proyecto.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Formular el problema	4 días	lun 06/07/15	jue 09/07/15	
Determinar el alcance del proyecto	1 día	lun 06/07/15	lun 06/07/15	
Planteamiento del problema	2 días	lun 06/07/15	mar 07/07/15	
Análisis de causa- efecto	2 días	mié 08/07/15	jue 09/07/15	
Establecimiento de objetivos y plan general del proyecto de simulación	8 días	vie 10/07/15	mar 21/07/15	1
Establecer objetivos	3 días	vie 10/07/15	mar 14/07/15	
diseñar y estructurar las etapas del proyecto de simulación	5 días	mié 15/07/15	mar 21/07/15	
Definir el Sistema	60 días	mié 22/07/15	mar 13/10/15	5
Clasificación del sistema	7 días	mié 22/07/15	jue 30/07/15	
Determinar componentes del sistema	10 días	jue 30/07/15	mié 12/08/15	
Determinar eventos o procedimientos del sistema	12 días	mié 12/08/15	jue 27/08/15	
Determinar los datos de entrada del sistema	14 días	jue 27/08/15	mar 15/09/15	
Análisis de datos de salida	15 días	mié 16/09/15	mar 06/10/15	
Conceptualizar el modelo	20 días	lun 03/08/15	vie 28/08/15	5
Alcance del modelo y detalles estructurales del modelo	20 días	lun 03/08/15	vie 28/08/15	
Recolección de la información	35 días	vie 28/08/15	jue 15/10/15	5
Determinar las fuentes de datos	7 días	vie 28/08/15	lun 07/09/15	
recolectar los datos del sistema real	15 días	lun 07/09/15	vie 25/09/15	
Análisis de los datos	15 días	vie 25/09/15	jue 15/10/15	
Codificar el modelo	12 días	vie 16/10/15	lun 02/11/15	16
Verificación del modelo	5 días	mar 03/11/15	lun 09/11/15	20
Validación	10 días	mar 10/11/15	lun 23/11/15	21
Diseño experimental del modelo	3 días	mar 24/11/15	jue 26/11/15	22
Análisis de salida	5 días	vie 27/11/15	jue 03/12/15	23
Implementación	3 días	vie 04/12/15	mar 08/12/15	24



Fuente: Elaboración propia.

4.3 Definición del Sistema de Colgas de Occidente S.A. E.S.P.

Para Chung (2004) la definición del sistema consiste en clasificar el sistema y establecer qué componentes y eventos están relacionados en su estructura operacional, se describe a continuación como está clasificado el sistema a modelar de Colgas de Occidente S.A. E.S.P. Planta Manizales.

4.3.1 Clasificación

Para este caso en particular el sistema a modelar tiene:

- Una alta variabilidad e interdependencia entre sus elementos que lo componen, y estos cambian en puntos separados en el tiempo, por lo que se puede decir que el sistema es un proceso discreto.

- La planta opera dos turnos al día, cada turno se compone de 8 horas en sus dos jornadas laborales diarias (am y pm). La plataforma de llenado inicia y termina el proceso del envasado ese mismo día, es decir, que al culminar la operación del segundo turno por política de la empresa no debe quedar ningún cilindro en espera de ser servido para el día siguiente, por lo que se puede clasificar como un sistema con terminación.

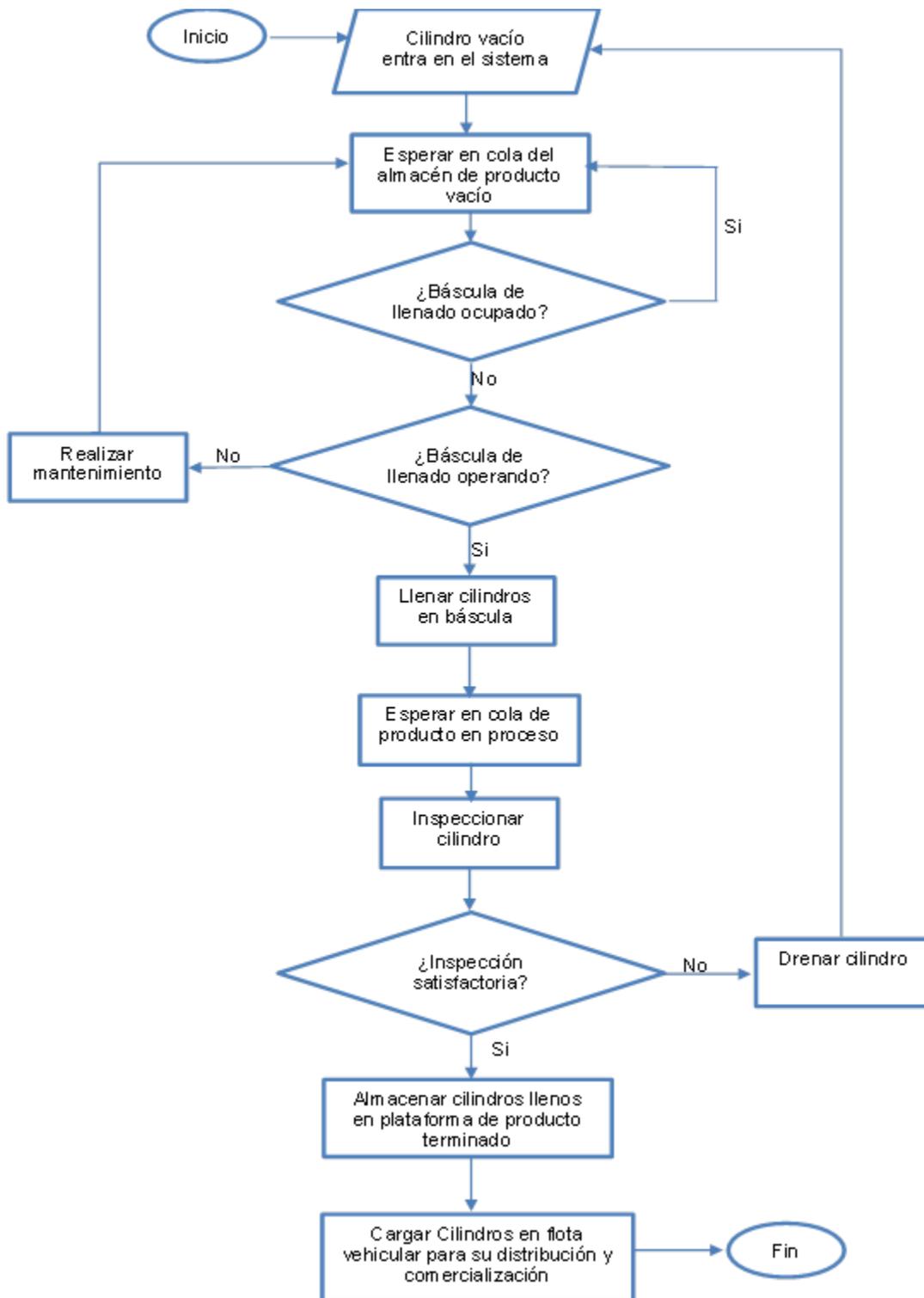
4.3.2 Componentes y eventos

Los componentes y eventos principales del sistema de Colgas de Occidentes S.A.E.S.P son los siguientes:

- **Componentes:** los cilindros de referencia de 20, 33, 40 y 100 libras; dos (2) operarios responsables en el proceso de envasado, cuatro operarios (4) responsables del cargue y descargue de cilindros, un (1) personal operativo responsable de la inspección y dieciséis (16) basculas de llenado.
- **Procesos o eventos:** el sistema de producción de Colgas de Occidente S.A. E.S.P planta Manizales, opera dos (2) turnos diarios de lunes a sábado en horario de 6:00 am hasta las 22:00 horas, distribuidos en jornadas laborales de 8 horas por turno. Los cilindros vacíos ingresan a plataforma de llenado para ser envasado con producto de GLP, cuando los cilindros vacíos arriban a la plataforma, estos son transportados manualmente por dos (2) operarios responsables de llenar los cilindros, si la báscula de llenado (procesador) está ocupada por un cilindro (*Flow ítem*) espera en cola para posteriormente ser procesada para su llenado, cada báscula de llenado la denominaremos en el programa de simulación como procesador, una vez lleno el cilindro en la báscula, este pasa al área de espera de producto en proceso donde son inspeccionados, luego se esta inspección se proceden al almacenar y apilar los cilindros llenos para estos ser cargados en los diferentes vehículos para su distribución y venta.

A continuación se describe mediante el diagrama de flujo del alto nivel propuesta por Chung (2004) para representar la definición del sistema.

Figura 4.3-1: Diagrama de flujo de alto nivel.

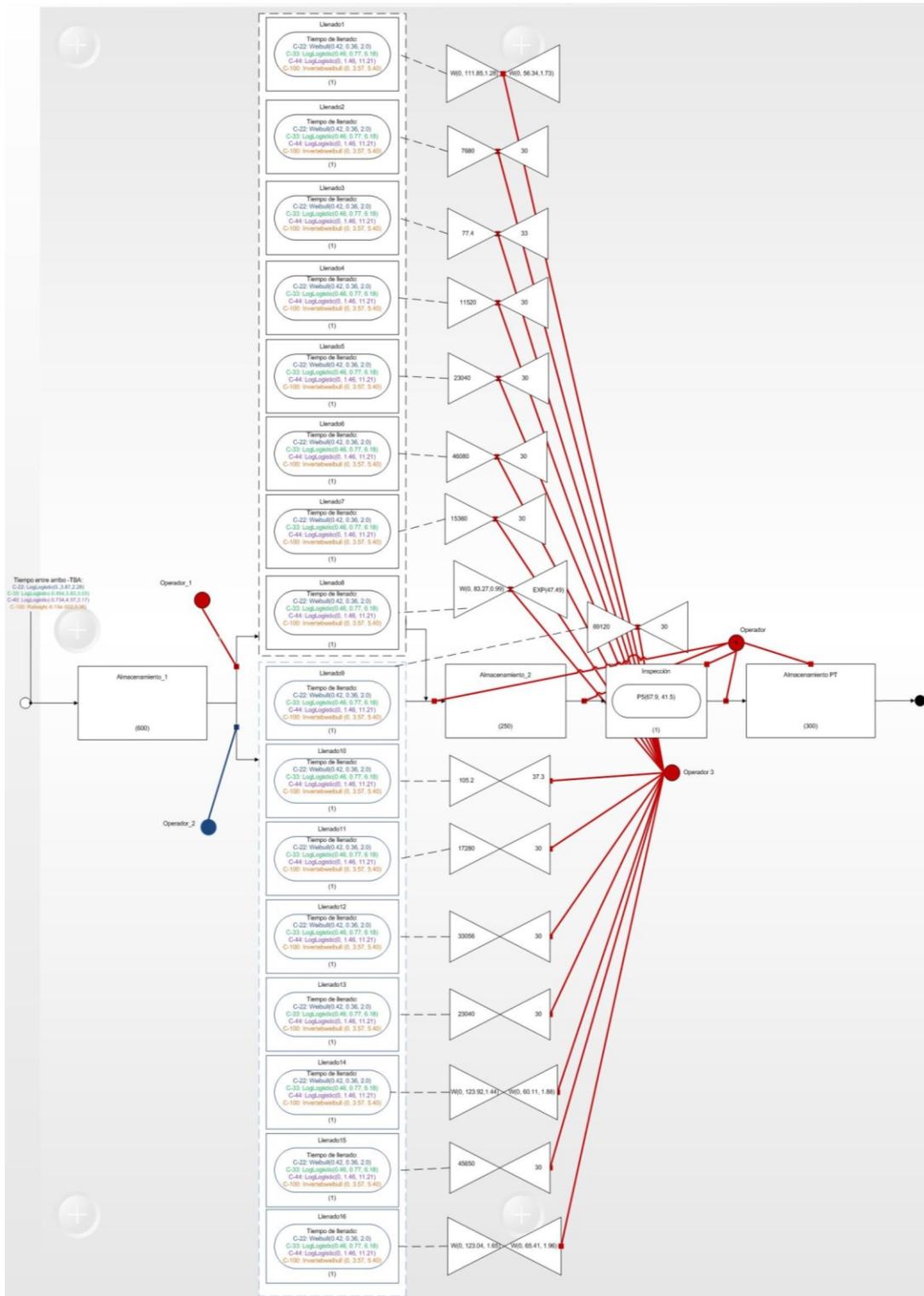


Fuente: Elaboración propia a partir de los aportes de Giraldo (2014).

4.4 Conceptualización del modelo

La conceptualización de este modelo se hará con la simbología propuesta por los autores (Beaverstock, Greenwood, Lavery, & Nordgren, 2012), donde se establecerá el alcance y grado de detalle del modelo de simulación. A continuación en la figura 4.4.1 el modelo Conceptual de Colgas de Occidentes planta Manizales, delimitando el alcance, que será simular el proceso de envasado de cilindros en plataforma con sus diferentes componentes y eventos.

Figura 4.4-1: Modelo conceptual del proceso de envasado de Colgas de Occidente.



Fuente: Elaboración propia.

4.4.1 Elementos estructurales y operacionales del modelo.

- **Los cilindros:** Los cilindros con referencias de 20, 33, 40 y 100 libras son representados y modelados como *Flow items*.
- **Las básculas de llenado:** Las 16 básculas de llenado que componen la estructura operacional del sistema real son modeladas como procesadores mono capacidad. Para este caso se modelarán de 1 a 16 procesadores.
- **Colas, áreas de espera:** En este sistema, las colas se presentan en las áreas espera de almacenamiento temporal de producto vacío, producto en proceso y almacenamiento de producto terminado, para este caso se modelarán tres (3) áreas de espera donde los cilindros se encolan para ser servidos.
- **Servidores y soporte administrativo:** El sistema cuenta con siete (7) operarios; el operario_1 es el responsable de realizar el llenado de los cilindros en las primeras ocho (8) básculas y las ocho (8) restantes por el operario_2, Cuatro operarios (4) responsables del cargue y descargue de cilindros y un (1) operativo responsable de la inspección.
- **Arribos:** para este caso se modelará el arribo de los cilindros con un tiempo entre arribos distribuidos una distribución Triangular con máximo de 0.4694, media de 0.402 y un mínimo de 0.310.
- **Tiempos de fallos (MTBF y MTTR):** Los procesadores tienen tiempos de fallos que fueron creados como *Breakdowns* en el *software* de Flexsim y en el cual son modelados como tiempos medios entre fallos y tiempos medios para reparar, para el caso se utilizaron distribuciones estadísticas que arrojo el programa de ExperFit., los datos para definir las frecuencias y las duraciones de las fallas fueron extraídas de los formatos que cuenta la empresa para registrar cada evento con su respectivo registro de fecha, hora y tiempo de duración que el equipo no estuvo operando.

4.5 Recolección de datos

Para el caso se empleó el programa de *Experimenter*, para el tratamiento de los datos con el fin de encontrar la mejor distribución estadística. La construcción del modelo de simulación se obtuvieron datos del sistema real objeto de estudio, cuyo objetivo fuese que el modelo de simulación represente lo más cercano posible la realidad del proceso de envasado de los diferentes cilindros. En la recolección de los datos para el desarrollo del modelo de simulación se realizó los siguientes pasos: definición de los tipos de datos necesarios para el modelo, recolección de la información, realización del análisis estadístico de los datos, definición de los tipos de datos necesarios para el modelo. El primer paso en la recolección de la información fue determinar los datos necesarios para construir el modelo, estos están relacionados por el alcance del modelo y el nivel de detalle requerido para conseguir los objetivos del estudio del objeto a simular y algunos de ellos son y se obtuvieron de SGC.

Figura 4.5-1: Interfaz gráfica de SGC.



Fuente: <http://www.pegasus.com.mx/>

- Tiempos de servicio: los tiempos de servicio incluye los tiempos de procesamiento y se obtuvieron con datos históricos que arroja el Sistema Global de Control (SGC-Troya) para cada equipo. Estos datos fueron exportados en archivo de Excel para la depuración de la información buscando la mejor distribución probabilística que mejor se ajuste. En la Figura 4.5 1: Interfaz gráfica de SGC. Se muestra la interfaz gráfica del SGC. Este sistema permite extraer información de todo el proceso productivo de llenado en los diferentes cilindros.
- Tiempos de paradas o *Breakdowns*: los datos para determinar estas frecuencias y duraciones se obtuvieron del SGC exportados en archivo Excel y en los formatos de mantenimiento que cuenta la empresa para registrar cada evento que impacto en la no disponibilidad del equipo, con relación a cada fecha y tiempo de duración del fallo. Estos datos se procesaron en el *software* ExperFit para determinar la mejor distribución probabilística que mejor se ajustara a los datos.

Supuestos del modelo:

- El sistema comienza con el almacenamiento del GLP en los tanques, el cual alimenta el sistema de red de tubería conectados a los equipos (Bombas y Básculas) para el trasiego.
- La hora de inicio de labores es a las 6:00 am y finaliza a las 10:00 pm programadas en dos turnos de trabajo por día, es decir, el primer turno comienza de 6:00 am hasta las 2:00 pm; el segundo turno empieza de 2:00 pm hasta las 10:00 pm. Jornadas de trabajo por turno de 8 hrs según lo reglamenta el código sustantivo del trabajo.
- El proceso simula el envasado de cilindros con GLP, con referencias de 20, 33, 40 y 100 libras cada uno.
- Las unidades de tiempo para toda la simulación estarán definidas en minutos y segundos, así como la distancia se establecerá en metros.

- Existe un tiempo de suplemento (30 minutos) en el proceso de producción para cada turno, el cual está destinado para atender las necesidades personales del trabajador con el fin de recuperarse de la fatiga.

Las medidas de desempeño a evaluar con el modelo serán: tiempo de duración del proceso de producción en el envasado de cilindros, tiempo promedio de un cilindro por referencia en el sistema, tiempos de utilización de los recursos productivos, cantidad de cilindros envasados diariamente y por referencia, porcentaje improductivo en los equipos, porcentaje de utilización del personal involucrado al proceso.

4.6 Codificación del modelo

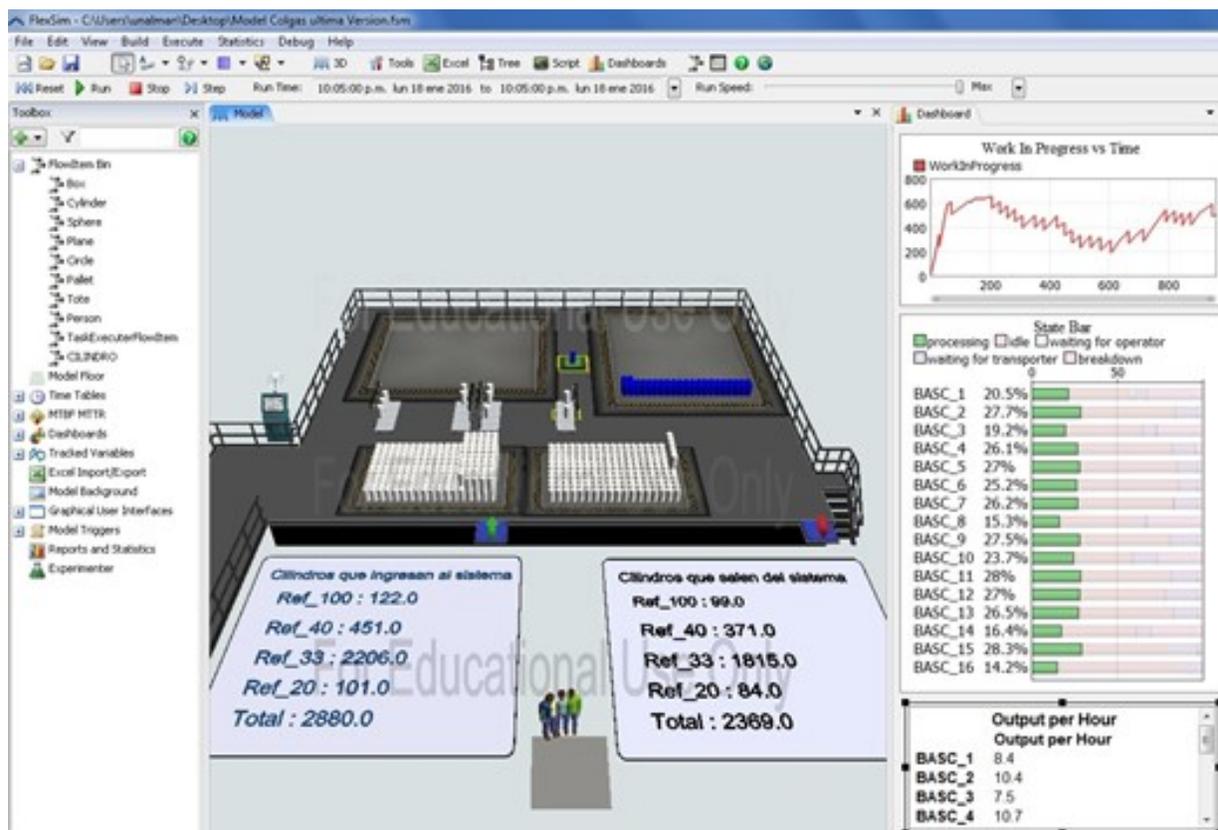
Esta etapa consiste en traducir el modelo conceptual en un modelo informático, es decir, que mediante un software de simulación, que para el caso se utilizó Flexsim versión 7.3.4. Se llevó a cabo la programación computarizada del modelo a fin de que este reproduzca lo más cercano el sistema real. A continuación se detalla la codificación del sistema a modelar. **Ver Anexo F: Modelo Documental codificación**

4.7 Verificación del modelo:

Según Harrel et al. (2006), dentro de esta etapa es importante que el modelo de simulación cumpla con todos los componentes necesarios y comprobar que todos los parámetros usados en la simulación funcionen correctamente, en otras palabras, la verificación del modelo no es otra cosa que determinar si el proceso de simulación refleja correctamente al modelo conceptual.

Para el caso en estudio la verificación del modelo se realizó construyendo el modelo en el software de Flexsim, en esta fase se verificó que todos los componentes tanto estructurales y operacionales estén en el modelo y funcionando correctamente para tener una aproximación cada vez en la lógica definida en el modelo conceptual, para esta verificación se utilizó la animación del mismo software de simulación y se chequearon los datos de salida.

Figura 4.7-1: Verificación del modelo con la Animación.



Fuente: Elaboración propia con Flexsim.

4.8 Validación del modelo

4.8.1 Validación Visual

Esta fase consiste en comparar los resultados del modelo frente a los del sistema real. Para ello se tendrá presente los dos tipos de validación que existen, una de ellas es la validación visual, es decir, viendo la animación cuando se corre el modelo y como este refleja el proceso real. Con este método se utilizaron los datos de salida del sistema de simulación mediante un archivo de Excel, el cual se tendrá en cuenta la información arrojada por el software para compararla con las del sistema real. A continuación se representan los datos en su respectivo análisis y datos reales del Sistema por envasado comprendido entre las 6:00 hrs hasta las 22:00 hrs del día (960 minutos).

Caso 1: Envasado de la báscula No. 1, 8 y 16 del día # 1 con Fallos.

Tabla 3: Reporte SGC de envasado báscula 1

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	3	0	0	0	0	0	2	0	0	0	0	0	0	0	5
C-33	10	5	0	7	9	1	14	2	0	0	8	4	0	6	13	11	1	0	91
C-40	0	8	0	0	0	0	0	0	0	0	0	7	1	6	0	2	0	0	24
C-100	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	3
TOTAL	10	13	0	7	12	1	14	5	0	0	10	11	1	12	13	13	1	0	123

Tabla 4: Reporte SGC de envasado báscula 8

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C-33	9	6	0	7	11	2	11	4	0	0	3	0	0	8	12	10	2	0	85
C-40	0	7	0	0	0	0	0	0	0	0	0	6	1	5	0	1	0	0	20
C-100	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	3
TOTAL	9	13	0	7	11	2	11	6	0	0	3	6	1	13	12	12	2	0	108

Tabla 5: Reporte SGC de envasado báscula 16

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	2	0	0	6
C-33	10	4	0	7	10	2	15	8	0	0	7	15	0	14	16	8	1	0	117
C-40	0	7	0	0	0	0	0	1	0	0	0	3	0	0	0	1	0	0	12
C-100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2
TOTAL	10	11	0	7	12	2	15	9	0	0	9	18	0	14	16	13	1	0	137

Caso 2: Envasado de la báscula No. 1, 8 y 16 del día # 2. Sin fallos**Tabla 6:** Reporte SGC de envasado sin fallos báscula 1.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3
C-33	6	10	8	5	9	12	6	10	5	13	11	17	7	16	17	16	6	12	186
C-40	1	2	0	2	3	0	2	0	5	4	1	0	0	1	0	0	0	6	27
C-100	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3
TOTAL	7	13	8	9	12	12	9	10	10	17	14	17	7	17	17	16	6	18	219

Tabla 7: Reporte SGC de envasado sin fallos báscula 8.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2
C-33	7	12	9	5	9	12	8	10	5	15	1	12	5	10	14	16	7	13	170
C-40	1	0	1	3	3	0	2	0	5	2	3	0	0	0	0	0	0	2	22
C-100	0	0	0	1	0	0	0	0	0	0	0	0	0	3	2	0	0	0	6
TOTAL	8	12	10	9	12	12	10	10	10	17	6	12	5	13	16	16	7	15	200

Tabla 8: Reporte SGC de envasado sin fallos báscula 16.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	3	0	1	1	5	0	0	1	0	3	0	0	0	0	0	0	0	14
C-33	8	11	10	4	11	12	11	11	6	8	8	13	5	13	17	9	5	17	179
C-40	0	1	1	1	2	0	2	0	3	6	0	0	0	2	9	0	0	0	27
C-100	0	0	0	3	0	0	0	0	0	1	2	2	0	2	0	0	1	0	11
TOTAL	8	15	11	9	14	17	13	11	10	15	13	15	5	15	19	18	6	17	231

Caso 3: Envasado de la báscula No. 1, 8 y 14 del día # 3. Con Fallos**Tabla 9:** Reporte SGC de envasado con fallos báscula 1.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C-33	10	19	0	4	8	0	8	0	4	6	0	7	10	17	3	9	0	0	105
C-40	0	1	6	0	0	0	0	1	0	1	0	0	0	2	6	7	0	0	24
C-100	0	0	0	0	4	3	4	8	0	0	3	4	0	0	4	0	0	0	30
TOTAL	10	20	6	4	12	3	12	9	4	7	3	11	10	19	13	16	0	0	159

Tabla 10: Reporte SGC de envasado sin fallos báscula 8.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	0	0	0	2	2	0	0	0	0	0	3	0	0	0	7
C-33	11	19	0	6	12	5	13	0	1	4	4	15	8	16	4	7	0	0	125
C-40	0	0	6	0	0	0	0	10	0	0	0	0	0	3	10	7	0	0	36
C-100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	11	19	6	6	12	5	13	12	3	4	4	15	8	19	17	14	0	0	168

Tabla 11: Reporte SGC de salida de envasado sin fallos báscula 14.

FROM	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	TOTAL
BEFORE	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	
C-20	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
C-33	13	10	5	7	0	3	5	2	5	14	10	16	6	11	15	0	0	0	122
C-40	0	0	1	0	9	1	0	2	0	3	0	0	0	5	1	0	0	0	22
C-100	0	0	0	0	0	3	0	1	0	0	2	0	0	0	0	0	0	0	6
TOTAL	13	10	6	7	9	7	5	5	5	17	13	16	6	16	16	0	0	0	151

Tabla 12: Reporte de salida de envasado de las 16 básculas con Flexsim.

Flexsim Summary Report

Time:,965.000000

Object	Class	stats_contentmax	stats_input	stats_output
BASC_1	Processor	1	111	110
BASC_2	Processor	1	118	118
BASC_3	Processor	1	106	106
BASC_4	Processor	1	114	113
BASC_5	Processor	1	112	111
BASC_6	Processor	1	113	112
BASC_7	Processor	1	117	116
BASC_8	Processor	1	90	89
BASC_9	Processor	1	152	152
BASC_10	Processor	1	125	125
BASC_11	Processor	1	162	162
BASC_12	Processor	1	153	153
BASC_13	Processor	1	155	155
BASC_14	Processor	1	135	135
BASC_15	Processor	1	151	151
BASC_16	Processor	1	122	122
Total			2036	2030

Si compara los reportes del sistema SGC y los de Flexsim, se puede evidenciar que existe una semejanza significativa frente al sistema real y el sistema modelado, ya que en promedio una báscula puede procesar según el fabricante 200 cilindros por día. Y contabilizando el total de las 16 básculas de llenado en el sistema real se envasan entre 2000 y 2400 cilindros por día con excepción de los días festivos ya que este se reduce en un promedio del 50%, porque solo se labora un solo turno. Sin embargo se validó el modelo con prueba estadística para reforzar el caso de estudio.

4.8.2 Validación Prueba estadística: T student.

La segunda validación que se empleó fue aplicar el método estadístico T Student que permitió comparar cuantitativamente los números de cilindros que salen en cada equipo del sistema real y el sistema simulado. A continuación se detalla la prueba T Student para validación del modelo, se emplea la herramienta del *software* SPSS versión 22, que cuenta la Universidad Nacional de Colombia para tal validación. A continuación se detalla en la tabla 4.8.2-1. Para el análisis de esta validación.

Tabla 13: Cilindros envasados del sistema real y simulación.

SISTEMA REAL		SISTEMA SIMULADO	
Equipo	Cilindros envasados	Equipo	Cilindros envasados
Báscula_1	123	Báscula_1	142
Báscula_2	154	Báscula_2	169
Báscula_3	159	Báscula_3	136
Báscula_4	153	Báscula_4	175
Báscula_5	145	Báscula_5	166
Báscula_6	160	Báscula_6	153
Báscula_7	160	Báscula_7	154
Báscula_8	183	Báscula_8	101
Báscula_9	171	Báscula_9	160
Báscula_10	165	Báscula_10	139
Báscula_11	150	Báscula_11	155
Báscula_12	180	Báscula_12	161
Báscula_13	160	Báscula_13	156
Báscula_14	151	Báscula_14	148
Báscula_15	183	Báscula_15	154
Báscula_16	171	Báscula_16	127

Fuente: Elaboración propia.

Para realizar la prueba T student es necesario en primer lugar analizar si los datos de salida tanto del sistema real como en el modelo de simulación se distribuyen normalmente, por lo tanto para verificar el supuesto de normalidad, se utiliza para estos dos casos la prueba de **Kolmogorov- Smirnov**, esta prueba permite contrastar la hipótesis que los datos muestrales provienen de una distribución normal y a continuación se detalla su procedimiento:

Prueba no Paramétrica:

1. Se plantean las hipótesis para el supuesto de normalidad:

H_0 = Los datos provienen de una distribución Normal.

H_1 = Los datos NO provienen de una distribución Normal.

2. Se procede mediante el software estadístico SPSS al análisis de datos mediante la prueba no paramétrica **Kolmogorov- Smirnov**.

Caso 1: Sistema = Real

Tabla 14: Prueba no paramétrica Kolmogorov – Smirnov

Prueba de Kolmogorov-Smirnov para una muestra ^a		
		Envasado
N		16
Parámetros normales ^{b,c}	Media	160,50
	Desviación estándar	15,453
Máximas diferencias extremas	Absoluta	,138
	Positivo	,138
	Negativo	-,123
Estadístico de prueba		,138
Sig. asintótica (bilateral)		,200 ^{d,e}

a. Sistema = Real

b. La distribución de prueba es normal.

c. Se calcula a partir de datos.

d. Corrección de significación de Lilliefors.

e. Esto es un límite inferior de la significación verdadera.

P_1 valor dio como resultado 0,200

Caso 2: Sistema = Simulación.

Prueba de Kolmogorov-Smirnov para una muestra^a

		Envasado
N		16
Parámetros normales ^{b,c}	Media	149,75
	Desviación estándar	18,035
Máximas diferencias extremas	Absoluta	,197
	Positivo	,081
	Negativo	-,197
Estadístico de prueba		,197
Sig. asintótica (bilateral)		,100 ^d

a. Sistema = Simulación

b. La distribución de prueba es normal.

c. Se calcula a partir de datos.

d. Corrección de significación de Lilliefors.

P_2 valor dio como resultado 0.100

3. Para ambos casos el valor de P_1 y P_2 es mayor al de P_0 valor de la significancia de 0.05, es decir que $P_1, P_2 > P_0$. Por lo tanto no se rechaza la hipótesis nula, es decir que los datos provienen de una distribución Normal para ambos casos.
4. Aplicar la regla de decisión: Como los datos se distribuyen normalmente se procede a realizar la Prueba T Student.

Prueba T Student:

Como los datos se distribuyen normalmente, es necesario aplicar la Prueba T student para analizar si los datos tienen o no varianza similar, es decir, que mediante una prueba de hipótesis se podría determinar con un nivel de significancia si existe diferencia estadística entre el modelo y el sistema, es importante calcular las medias y las desviaciones estándar de ambos conjuntos de datos y continuar con el procedimiento siguiente:

1. Se plantean las hipótesis, las medias serán:
 - Ho: Las medias de ambos grupos son iguales.
 - Ha: Las medias de ambos grupos no son iguales.

$$H_0: \mu_1 = \mu_2$$

$$H_a: \mu_1 \neq \mu_2$$

En el caso si se acepta H_0 , se considera válido el modelo de simulación.

Para este caso se utiliza la prueba de Levene para la igualdad de varianzas. Si el p-valor (Significancia Asintótica) es menor que la significancia se rechazará la hipótesis nula, de lo contrario si v-valor es mayor que la significancia no se podrá rechazar la hipótesis nula.

2. Para el caso se seleccionó un nivel de significancia $\alpha = 0.05$
3. Determinar el valor crítico de t en el nivel de significancia $\alpha/2$ y el número de grados de libertad menos 2.
4. Se calcula un estadístico t de acuerdo con la siguiente expresión:

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}$$

Dónde:

T: Estadístico a calcular.

\bar{x}_1 = Media de la primera alternativa

\bar{x}_2 = Media de la segunda alternativa

s_1^2 = Varianza de la primera alternativa

s_2^2 = Varianza de la segunda alternativa

n_1 = Tamaño de muestra de la primera alternativa

n_2 = Tamaño de muestra de la segunda alternativa

5. Mediante el software SPSS versión 22, se procedió al análisis de los datos para poder determinar si las medias entre ambos grupos son iguales o no. evaluadas con un nivel de confianza del 95%.

Prueba T student

Tabla 15: Prueba T student para análisis de varianzas y medias.

Estadísticas de grupo

Sistema	N	Media	Desviación estándar	Media de error estándar
Envasado Real	16	160,50	15,453	3,863
Simulación	16	149,75	18,035	4,509

Prueba de muestras independientes

	Prueba de Levene de calidad de varianzas		prueba t para la igualdad de medias	
	F	Sig.	t	gl
Envasado Se asumen varianzas iguales	,247	,623	1,811	30
No se asumen varianzas iguales			1,811	29,311

La prueba de Levene contrasta las varianzas, el valor F del estadístico de Levene es de 0,247 y su significancia asociada es de 0,623, si se compara con el valor de P valor de significancia de 0,05. Como $0,623 > 0,05$, se traduce que las varianzas son iguales.

Posteriormente se contrasto las medias con las dos poblaciones por medio de la prueba T student, lo que dio como resultado un $t = 1.8111$ con 30 grados de libertad y su significación (bilateral) estadística de 0,080. Para el caso si se compara el T estadístico de significancia asociada al contraste T de 0,05, es decir que $0,08 > 0,05$. No se rechaza la hipótesis nula, concluyendo que no existe diferencia significativa entre ambos grupos.

Prueba de muestras independientes

				prueba t para la igualdad de medias				
				Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
							Inferior	
Envasado	Se asumen varianzas iguales	,080	10,750	5,938	-1,376			
	No se asumen varianzas iguales	,080	10,750	5,938	-1,388			

Prueba de muestras independientes

		prueba t para la igualdad de medias	
		95% de intervalo de confianza de la diferencia	
		Superior	
Envasado	Se asumen varianzas iguales	22,876	
	No se asumen varianzas iguales	22,888	

En conclusión se puede determinar que los datos tanto del sistema real como los datos de salida del sistema simulado, se distribuyen normalmente y las medias para ambos grupos no presentan diferencias significativas (Sistema real y Sistema simulado).

4.9 Diseño experimental

Con el modelo ya construido y validado, se procedió a realizar corridas al mismo, a fin encontrar respuestas a diferentes escenarios y a modificaciones que posiblemente permitan mejorar el desempeño del sistema real, de esta manera se permitirá buscar respuestas a diferentes tipos de preguntas, como por ejemplo: que pasaría con la capacidad de producción si el número de fallos fuese menor o mayor a lo esperado. De acuerdo a los datos arrojados con el *software* de Flexsim se pudo realizar 3 tipos de escenarios con 5 réplicas cada uno. A continuación se detallan los resultados en el capítulo Análisis de datos.

4.10 Análisis de salida

Los resultados generados por el *software* de simulación Flexsim se debieron analizar con el fin de obtener medidas de desempeño del sistema, necesarias para el análisis estadístico y la evaluación conceptual de los resultados. Para tal caso del modelo se plantearon tres tipos de escenarios diferentes y cinco replicas por escenario. Los escenarios permitieron analizar y evaluar la producción total del sistema, teniendo en cuenta los números de operarios que intervienen en el proceso de envasado y los tiempos críticos de cada equipo que se ven afectados por el tiempo que se requiere cuando estos presentan fallos o en tiempos muertos. A continuación se detalla los resultados de cada escenario simulado con el Experimenter del *software* Flexsim en la tabla 4.10-1.

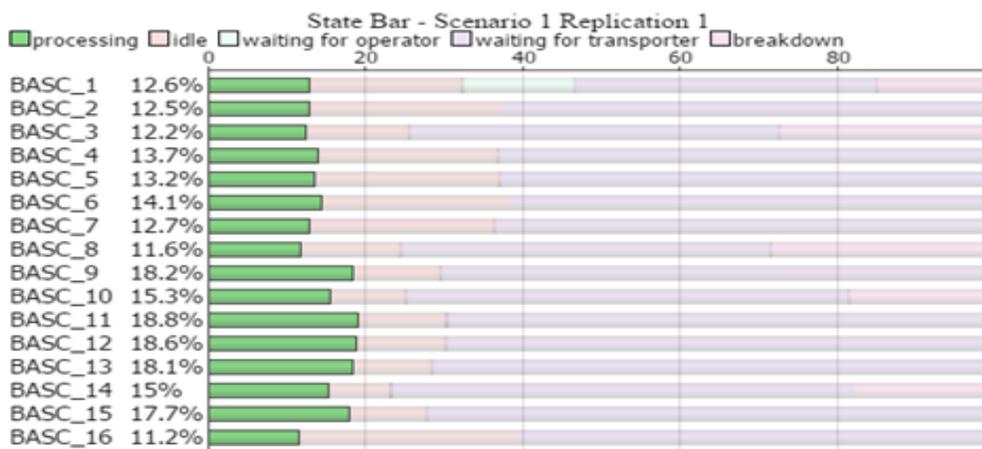
Tabla 16: Número de Replicas por escenario.

Experimentation Report						
Number of Scenarios:		3				
Number of Replications Per Scenario:		5				
Warmup Time:		0				
Producción Total						
	Mean (90% Confidence)	Sample	Std Dev	Min	Max	
Scenario 1	304 < 783 < 1262	502		103	1442	
Scenario 2	1896 < 2328 < 2759	452		1545	2678	
Scenario 3	2040 < 2390 < 2740	367		1854	2678	

Escenario No. 1:

Figura 4.10-1: Resultados del escenario_1

Scenario 1				
Object	idle	processing	waiting for transporter	breakdown
BASC_1	19.7%	12.6%	52.6%	15.1%
BASC_2	24.8%	12.5%	62.7%	0.0%
BASC_3	13.3%	12.2%	47.1%	27.4%
BASC_4	23.1%	13.7%	63.2%	0.0%
BASC_5	23.9%	13.2%	62.9%	0.0%
BASC_6	24.1%	14.1%	61.8%	0.0%
BASC_7	23.6%	12.7%	63.8%	0.0%
BASC_8	12.8%	11.6%	47.0%	28.6%
BASC_9	11.2%	18.2%	70.6%	0.0%
BASC_10	9.8%	15.3%	56.3%	18.6%
BASC_11	11.5%	18.8%	69.8%	0.0%
BASC_12	11.4%	18.6%	69.9%	0.0%
BASC_13	10.4%	18.1%	71.5%	0.0%
BASC_14	8.1%	15.0%	59.0%	17.9%
BASC_15	10.0%	17.7%	72.3%	0.0%
BASC_16	28.3%	11.2%	60.5%	0.0%



Fuente: Elaboración propia a partir del *Experimenter* de Flexsim.

Este escenario representó la operatividad en el proceso del envasado con los siguientes tiempos de fallos aleatorios del sistema, frente a los equipos que representan un impacto fuerte en los indicadores de disponibilidad y confiabilidad en el sistema de producción y el tiempo que toma en reanudar la funcionalidad del equipo.

Name of Machine	BASC_1	BASC_3	BASC_8	BASC_10	BASC_14
Operation availability (min/Day)	960	960	960	960	960

Número de fallos	3	4	6	3	4
TBF (min)	789	571	494	726	737
TTR (min)	171	389	466	234	223
MTBF (min)	263	143	82	242	184
MTTR (min)	57	97	78	78	56
Tasa de fallos	0.38%	0.70%	1.21%	0.41%	0.54%
Tasa de reparación	1.75%	1.03%	1.29%	1.28%	1.79%
Tasa de mantenimiento	21.70%	68.13%	94.25%	32.23%	30.26%
Disponibilidad	82.2%	59.5%	51.5%	75.6%	76.8%

Estos equipos basculas de llenado No. 1, 3, 8, 10 y 14. Representadas en el sistema, requiere una atención para mejorar los índices de disponibilidad de cada uno de los equipos, ya que estas disponibilidades se encuentran entre un 51.5% y 82.2%, impactando los índices de producción de la empresa, otro aspecto a considerar es el tiempo que toma en reanudar la funcionalidad del equipo, es este caso la báscula No. 8 en todo su tiempo de operatividad 960 minutos, está en un tiempo muerto de aproximadamente un 49%, traducido que su tasa de mantenibilidad se encuentra en un 94.25%, lo cual se traduce y se determinó para este escenario, que existe un mínimo de producción de 103 unidades y un máximo de 1.442 unidades envasadas, con un promedio de envasado de 783 unidades.

Experimentation Report

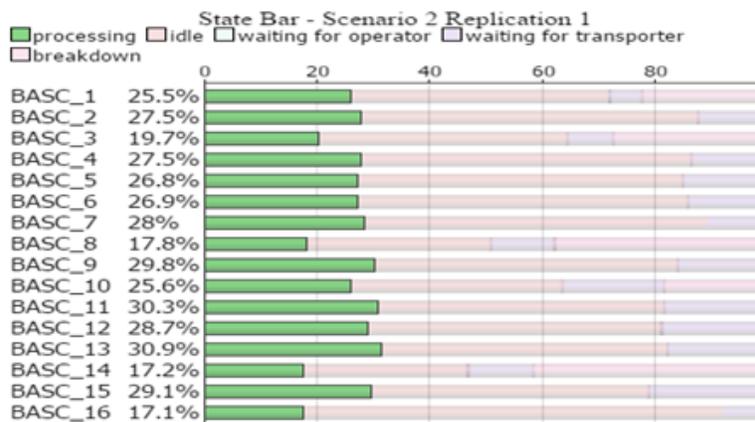
Number of Scenarios:	3
Number of Replications Per Scenario:	5
Warmup Time:	0

Producción Total

	Mean (90% Confidence)	Sample	Std Dev	Min	Max
Scenario 1	304 < 783 < 1262		502	103	1442

Escenario No. 2:**Figura 4.10-2: Resultados del Escenario_2**

Scenario 2				
Object	idle	processing	waiting for transporter	breakdown
BASC_1	46.3%	25.5%		5.8%
BASC_2	60.0%	27.5%		12.5%
BASC_3	44.7%	19.7%		8.2%
BASC_4	58.8%	27.5%		13.7%
BASC_5	58.2%	26.8%		15.1%
BASC_6	58.7%	26.9%		14.4%
BASC_7	61.2%	28.0%		10.8%
BASC_8	32.8%	17.8%		11.4%
BASC_9	54.0%	29.8%		16.2%
BASC_10	37.8%	25.6%		18.0%
BASC_11	51.3%	30.3%		18.4%
BASC_12	52.3%	28.7%		19.0%
BASC_13	51.1%	30.9%		18.0%
BASC_14	29.4%	17.2%		11.6%
BASC_15	49.8%	29.1%		21.1%
BASC_16	74.8%	17.1%		8.1%



Fuente: Elaboración propia a partir del Experimenter de Flexsim

Este escenario representa las básculas de llenado No. 1,2, 8,10 y 14, requieren una mayor atención a los *breakdown* o descomposturas, es decir a la revisión técnica sobre a sus mantenimiento, ya que están representando entre un 18.6% y un 41.7% de fallos en el sistema productivo y por lo tanto afectan la tasa de producción del sistema.

Name of Machine	BASC_1	BASC_2	BASC_8	BASC_10	BASC_14
Operation availability (min/Day)	960	960	960	960	960
Número de fallos	4	3	6	2	5

TBF (min)	741	696	518	775	499
TTR (min)	219	264	442	185	461
MTBF (min)	185	232	86	387	100
MTTR (min)	55	88	74	93	92
Tasa de fallos	0.54%	0.43%	1.16%	0.26%	1.00%
Tasa de reparación	1.83%	1.14%	1.36%	1.08%	1.08%
Tasa de mantenimiento	29.48%	37.91%	85.44%	23.89%	92.46%
Disponibilidad	77.2%	72.5%	53.9%	80.7%	52.0%

También es necesario revisar el estado de procesamiento de los equipos en general, ya la tasa de reparación se encuentra entre un 1.08% y un 1.83%, lo cual impacta de manera negativa los niveles de producción en el envasado del sistema de producción, aun este escenario es mucho mejor que el anterior en el indicador de disponibilidad, los datos reflejan un incremento significativo en los niveles de producción, es decir, que frente a este escenario existe un mínimo de producción de 1.545, una media de 2328 y un máximo de 2.678 unidades producidas por día.

Experimentation Report

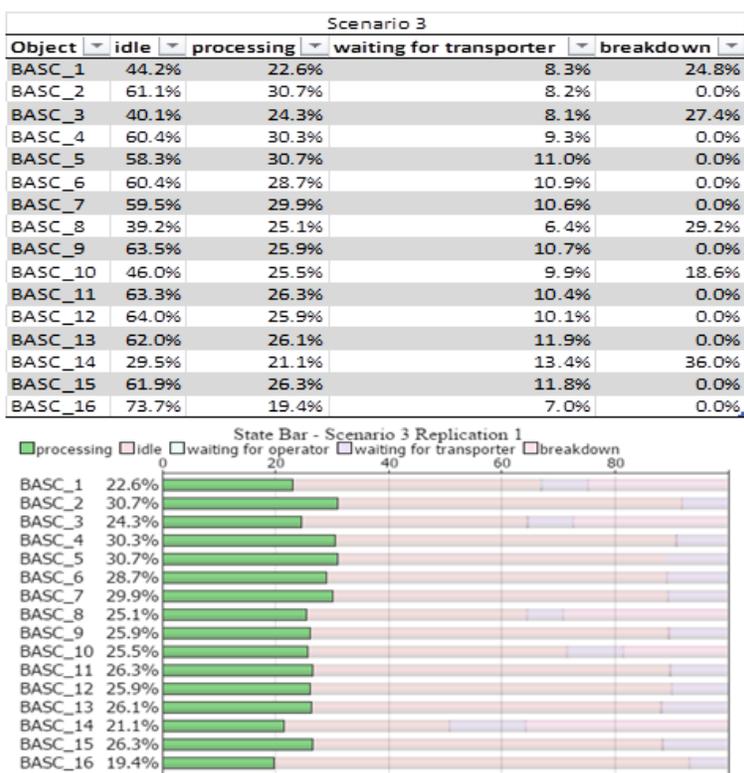
Number of Scenarios:	3
Number of Replications Per Scenario:	5
Warmup Time:	0

Producción Total

	Mean (90% Confidence)	Sample	Std Dev	Min	Max
Scenario 2	1896 < 2328 < 2759		452	1545	2678

Escenario No. 3: Realizando el análisis se observa que en este escenario, el sistema de simulación modelado evidencia que las básculas de llenado No.1, 3, 8,10 y14, continúan presentando índices bajos de confiabilidad dado que los *breakdown* o descomposturas están entre un 18.6% y un 36% de fallos en el sistema productivo y por lo tanto afectan la tasa de producción del sistema.

Figura 4.10-3: Resultados del Escenario_3



Fuente: Elaboración propia a partir del Experimenter de Flexsim.

Para este escenario se determinó que mediante el modelo simulado existe un mínimo de producción de 1.854 unidades y un máximo de 2.678 unidades envasadas, con un promedio de envasado de 2040 cilindros.

Name of Machine	BASC_1	BASC_3	BASC_8	BASC_10	BASC_14
Operation availability (min/Day)	960	960	960	960	960
Número de fallos	3	5	4	2	5

TBF (min)	830	659	644	895	555
TTR (min)	130	301	316	65	405
MTBF (min)	277	132	161	448	111
MTTR (min)	43	60	79	33	81
Tasa de fallos	0.36%	0.76%	0.62%	0.22%	0.90%
Tasa de reparación	2.31%	1.66%	1.27%	3.08%	1.23%
Tasa de mantenimiento	15.63%	45.68%	49.02%	7.26%	72.97%
Disponibilidad	86.5%	68.6%	67.1%	93.2%	57.8%

También es necesario revisar el estado de procesamiento de los equipos en general, ya que estos no están siendo utilizados en un 90% de su capacidad productiva, dado que el indicador de disponibilidad no cumple con la meta establecida por el programa de calidad de la empresa, es decir, que debe ser superior o igual al 95%. Por lo tanto la capacidad productiva está siendo afectada en tiempos que estas permanecen desocupadas por razones de los tiempos de suplementos en la operación, los mantenimientos preventivos realizados en horario de operatividad del equipo y adicionalmente a los fallos presentes.

Experimentation Report

Number of Scenarios:	3
Number of Replications Per Scenario:	5
Warmup Time:	0

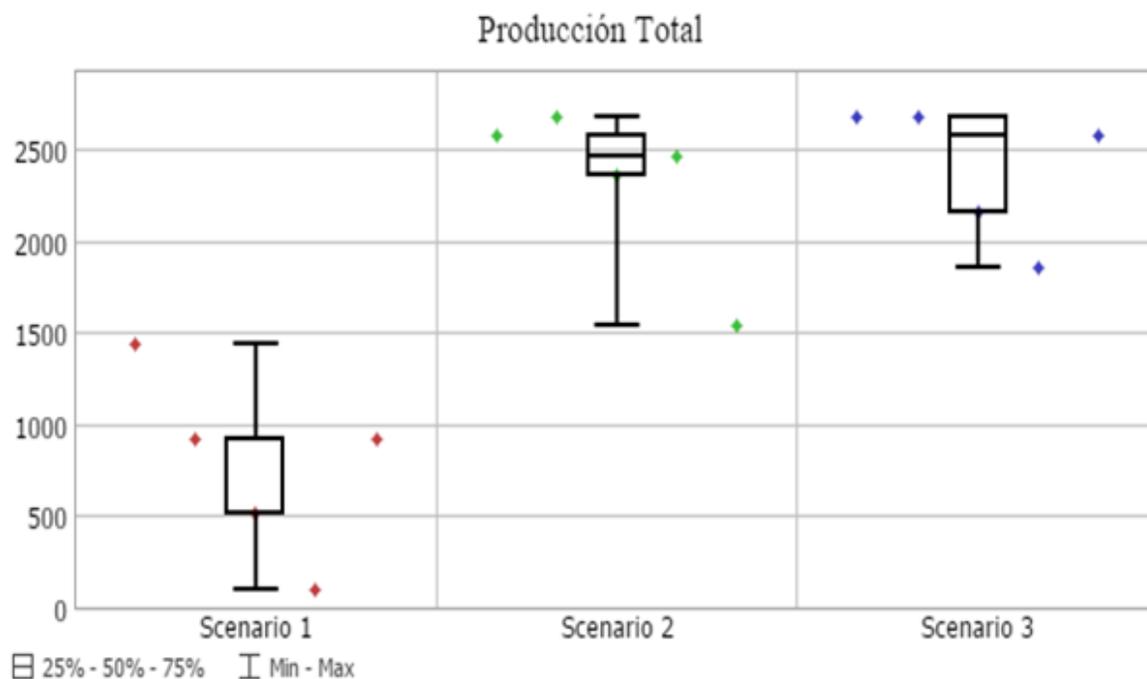
Producción Total

	Mean (90% Confidence)	Sample	Std Dev	Min	Max
Scenario 3	2040 < 2390 < 2740		367	1854	2678

Réplicas: se realizaron 5 corridas por cada escenario simulando del sistema real se determinó que, si se añaden 2 operarios más a la operación del sistema actual real, se podrían tener con más frecuencia un promedio de 2.390 cilindros envasados comparado con el promedio actual que está en 2.328 cilindros. Lo cual se obtiene un incremento en promedio de 62 cilindros con una variación de aproximadamente del 2.6% en la producción diaria del sistema. No es

mucho pero considerando la tasa de producción actual se incrementa el 2.6% en un periodo de tiempo a 26 días laborales al mes, esto es una cantidad significativa que representa 1.612 cilindros adicionales que suman a la disponibilidad para la venta y al igual un aumento sobre el estado de resultados en el ingreso operacional y posiblemente un ajuste a los costos operativos.

Figura 4.10-4: box joking statistic per scenario



Producción Total					
	1	2	3	4	5
Scenario 1	1442	927	515	103	927
Scenario 2	2575	2678	2369	2472	1545
Scenario 3	2678	2678	2163	1854	2575

Fuente: Elaboración propia con Experimenter Flexsim

Las siguientes tablas y gráficas que arrojó el modelo de simulación, refleja los porcentajes de procesamiento y descompostura de los equipos de envasado. Al iniciar el procesamiento de la información se analizó que:

4.11 Plan de implementación.

Para llevar a cabo este proyecto de simulación al sistema real de la empresa Colgas de occidente S.A. E.S.P. fue necesario contar con el interés de los altos directivos de la organización Inversiones del Noreste y al administrador encargado de la planta Manizales de Colgas de Occidente S.A. E.S.P, ya que a la final fueron ellos los que tomaron la decisión de llevar a cabo y de concretar la implementación y ejecución de este proyecto de simulación.

5 Conclusiones y recomendaciones

5.1 Conclusiones

- El desarrollo del modelo de simulación junto con el aporte teórico y práctico, permitió determinar que equipos (Bascula 1, 3,8,10 y 14) impactaban el sistema productivo dado el alcance que se obtuvo para el presente proyecto, logrando cuantificar mediante diferentes escenarios los niveles de producción en el envasado, además de visualizar los tiempos y porcentajes de utilización de cada uno ellos que interviene en el proceso de envasado, al igual que los recursos empleados y requeridos para la operación del sistema productivo.
- Mediante el modelo de simulación versus el modelo real se logró experimentar, de forma rápida y con el mínimo de riesgo, a fin de probar nuevos conceptos al sistema real sin la necesidad de perturbarlo.
- Con el modelo de simulación la alta dirección de la empresa Colgas de Occidente tendrá como herramienta el presente proyecto de simulación con el fin de ser un soporte para la toma de decisiones que contribuyan a desarrollar futuras políticas de operación y empleo de recursos en pro de mejorar el desempeño del sistema.
- El sistema modelado permitió jugar con diferentes escenarios integrando los tiempos medios entre fallos-MTBF y los tiempos medios para reparar – MTTR, con el fin de experimentar y proveer información sobre los resultados del sistema del producción y que recursos son los que sufren impactó en su funcionalidad, afectando de forma significativa los indicadores de disponibilidad, confiabilidad, tasas de fallos, tasas de reparación, tasas de mantenimiento, porcentajes de utilización de los equipos, tiempos de procesamiento y entre otros relacionados, que permitieron proveer resultados de fácil entendimiento y comunicación, ya que la simulación de este proyecto es visualmente atractiva e involucra el interés de los usuarios del sistema.

5.2 Recomendaciones

- Los resultados del análisis realizado presentan un insumo importante para la toma de decisiones que permitan mejoras al sistema productivo de Colgas de Occidente S.A. E.S.P; por lo cual es necesario implementar acciones preventivas que apunten incrementar la disponibilidad y confiabilidad en los equipos a fin de reducir las Tiempos Medios Entre Fallos y los Tiempos Medios de Reparación de las basculas No. 1,3,8,10 y 14, que son las que más impactan en sistema productivo del envasado de la empresa Colgas de Occidente S.A. E.S.P. Planta Manizales.
- Revisar los tiempos en los cuales los equipos (Basculas de llenado) son intervenidos para la realización de actividades de mantenimiento preventivo en un horario especial donde estos no estén en operatividad, por ejemplo, se podrían realizar estas actividades en recesos de operatividad en plataforma de llenado, reuniones y otros espacios considerables a la hora de no estar el personal operacional de los equipos.
- Se recomienda que el personal que realiza la operación de llenado esté involucrado en temas de qué hacer cuando se presente una falla en el equipo y como intervenirla si esta no es muy grave, identificando acciones que puedan eliminar o reducir la probabilidad de que el fallo potencial ocurra. Para esto es necesario que la empresa opte por una política de calidad de mejora continua que permita identificar aquellos factores de riesgo que pueden estar presentes en el proceso y los impactos que estos puedan acarrear en la operatividad diaria del sistema de producción.
- Se recomienda realizar capacitaciones periódicas al personal operativo que manipula directamente los equipos, en relación a la operatividad del mismo. Ya que se ha evidenciado que parte de los fallos que se traducen en tiempos muertos en los equipos se debe a mala manipulación del equipo por parte del operario.

Bibliografía

- Ackoff, U. L. (2002). *El paradigma de Ackoff: Una administración sistemática*. Mexico : LIMUSA.
- Banks, J. (2000). *Principles of simulation*. New York: John Wiley.
- Banks, J. (2009). *Discrete Event System Simulation*. Prentice Hall.
- Bautista Paz, E. (2007). *Breve historia ilustrada de las máquinas*. Madrid: Universidad Politécnica de Madrid.
- Bazovsky, I. (2004). *Reliability Theory and practice*. *Dover Publications Incorporated*, 304.
- Beaverstock, M., Greenwood, A., Lavery, E., & Nordgren, B. (2012). *Applied Simulation Modeling and Analysis Using FlexSim*. Flexsim Software Products, Incorporated.
- Blanchard, B. S. (1995). *A Key To Effective Serviceability And Maintenance Management*. Limusa Wiley.
- Blanco Rivero, L. F. (2003). *Simulación con promodel: casos de producción y logística*. Bogotá D.C: Escuela Colombiana de Ingeniería.
- Campbell, J. D., Duffuaa, S., & Raouf, A. (2010). *Sistemas de Mantenimiento: Planeación y Control*. México: Limusa Wiley.
- Chung, C. (2004). *Simulation Modeling Handbook: A Practical Approach*. Washington D.C.: CRC Press.
- Churchman, C. (1973). *El enfoque de sistemas*. México: Diana.
- Domínguez Machuca, J. (1995). *Dirección de operaciones. Aspectos tácticos y operativos en la producción y los servicios*. Madrid: Mc-Graw Hill.

- Duffuaa, S. (2010). *Sistemas de mantenimiento: Planeación y Control*. México: Limusa Wiley.
- Duffuaa, S. O. (2010). *Planning and control of maintenance System*. New York: Limusa.
- Ebeling, C. E. (2005). *An Introduction to Reliability and Maintainability Engineering*. New York: McGraw-Hill.
- Giraldo García, J. A. (2014). *Simulación de sistemas de producción y de servicios*. Manizales.
- Gutiérrez, P. A. (2003). *Metodología para medir confiabilidad, mantenibilidad y disponibilidad en mantenimiento Mecánica*. Medellín, Antioquia, Colombia: Universidad EAFIT.
- Harrell, C., Ghosh, B., & Bowden, R. (2004). *Simulation using Promodel*. New York: McGraw-Hill.
- Kardec, A. y. (2003). *Manutenção: Função Estratégica*. Ed. Qualitymark.
- Kelton, W., & Law, M. (2007). *Simulation Modeling and Analysis*. Singapore: McGrawHill-Industrial Engineering Series.
- Law, A. (2006). *Simulation Modeling and Analysis, fourth ed*. McGraw-Hill College.
- Miltenburg, J. (1995). *Manufacturing Strategy*. Portland: Productivity Press.
- Mora, L. A. (2010). *Mantenimiento. Planeación, ejecución y Control*. México: Alfaomega.
- Muller, A. M. (2007). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. En *Reliability Engineering and System Safety* (pág. 10.1016).
- Riis, J. O. (1997). A situational Maintenance Model. *International Journal of Quality and Reliability Management*, 349-366.
- Shannon, E. R. (1988). *Simulación de Sistemas*. México: Trillas.
- Swanson, L. (2001). Linking maintenance strategies to performance. *Production Economics*, 70.

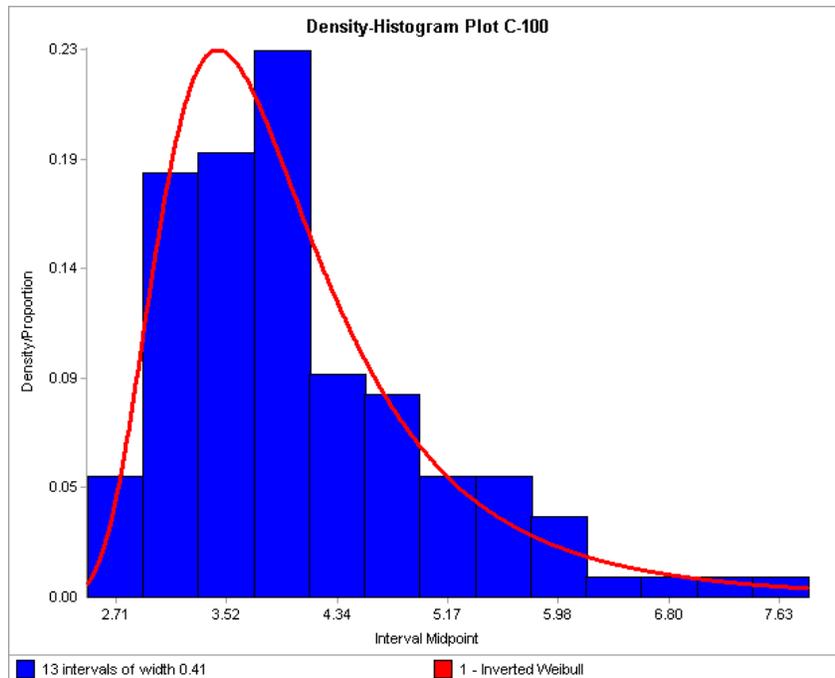
Teixeira, A. (2001). Multicriteria decision on maintenance: spares and contract planning. *European Journal of Operational Research*, 129.

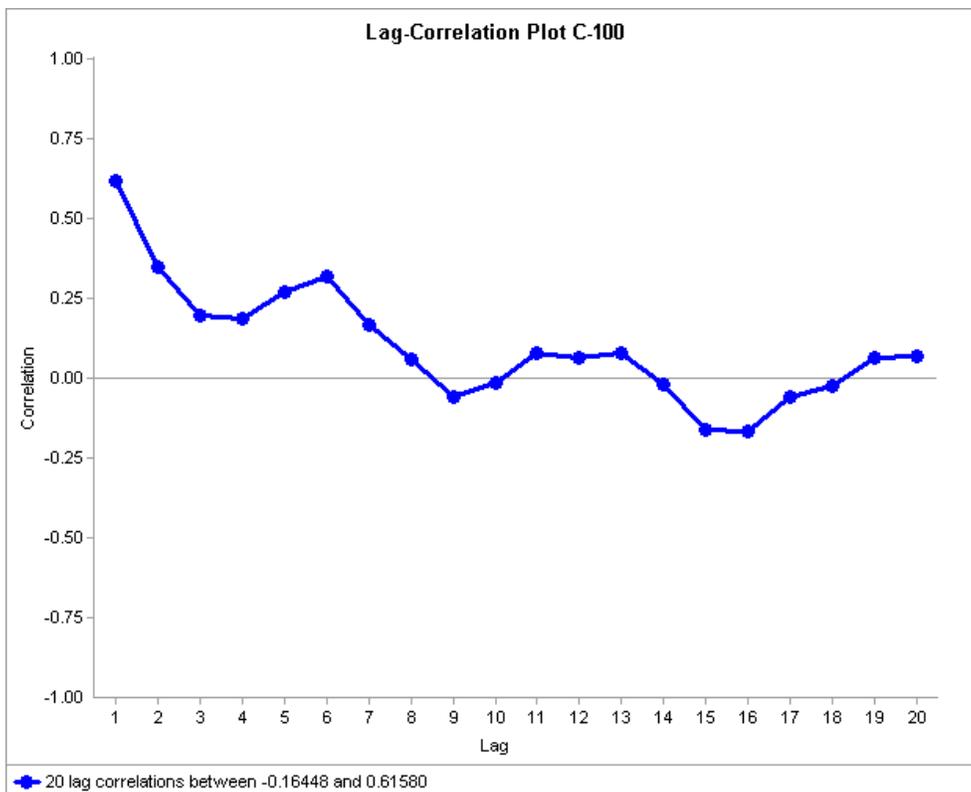
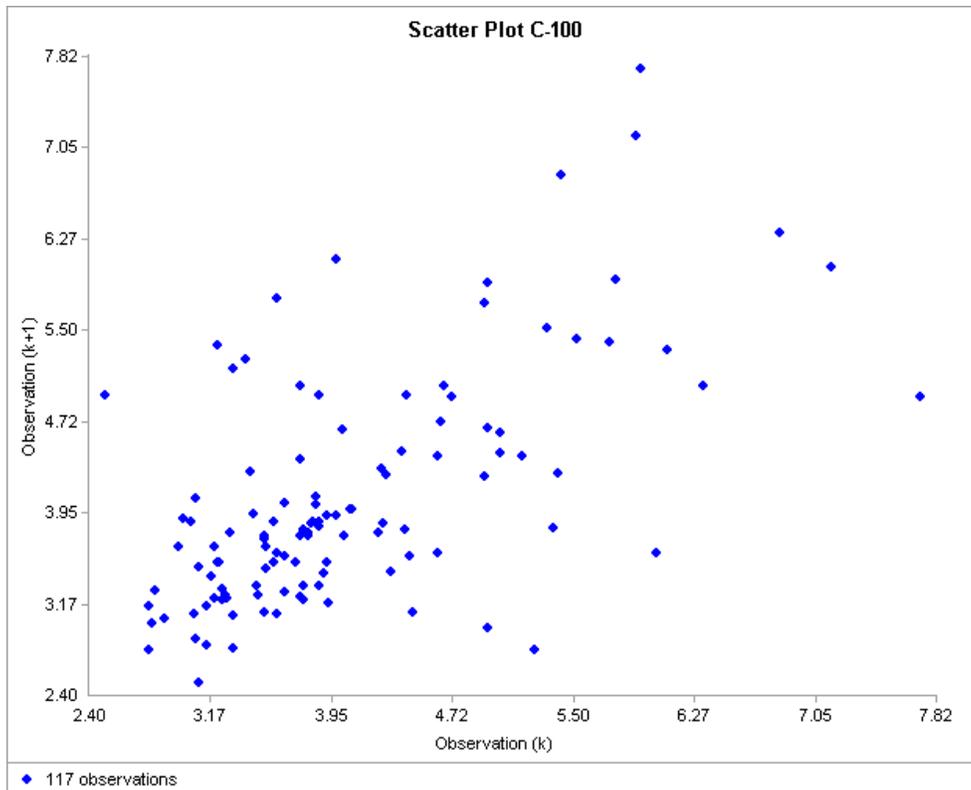
A.Anexo: Estadística tiempo de llenado cilindro 100 libras

Estadística descriptiva - Tiempos de llenado en minutos

	C-100
Media	4.08
Error típico	0.09
Mediana	3.82
Moda	3.87
Desviación estándar	0.99
Varianza de la muestra	0.97
Curtosis	1.52
Coefficiente de asimetría	1.22
Rango	5.22
Mínimo	2.50
Máximo	7.72
Suma	476.87
Cuenta	117
Nivel de confianza(95,0%)	0.18

Interval	Upper Endpoint	Proportion
Below	2.50000	0.00000
1	2.91000	0.05128
2	3.32000	0.17949
3	3.73000	0.18803
4	4.14000	0.23077
5	4.55000	0.09402
6	4.96000	0.08547
7	5.37000	0.05128
8	5.78000	0.05128
9	6.19000	0.03419
10	6.60000	0.00855
11	7.01000	0.00855
12	7.42000	0.00855
13	7.83000	0.00855
Above	infinity	0.00000





Distribuciones estadísticas que mejor se ajustan a los datos de llenado del cilindro de 100 libras depurado con ExperFit

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters
1 - Inverted Weibull	95.97	Location 0.00000
		Scale 3.57766
		Shape 5.40276
2 - Inverted Weibull(E)	94.35	Location 0.00183
		Scale 3.57580
		Shape 5.40004
3 - Pearson Type V(E)	92.74	Location 1.82843
		Scale 11.80498
		Shape 6.21927

32 models are defined with scores between 0.00 and 95.97

Absolute Evaluation of Model 1 - Inverted Weibull

Evaluation: Good

Suggestion: Additional evaluations using Comparisons Tab might be informative.
See Help for more information.

Additional Information about Model 1 - Inverted Weibull

"Error" in the model mean
relative to the sample mean -0.03086 = 0.76%

Flexsim Representation of Model 1 - Inverted Weibull

Use:

When using a picklist option:

Distribution	Inverted Weibull
Location	0.000000
Scale	3.577660
Shape	5.402759

When using code:

`invertedweibull(0.000000, 3.577660, 5.402759, <stream>)`

Test estadístico del cilindro de 100 libras. Distribución Inverted Weibull

Anderson-Darling Test with Model 1 - Inverted Weibull

Sample size 117
Test statistic 0.27893

Note: The following critical values are exact.

Sample Size	Critical Values for Level of Significance (alpha)				
	0.250	0.100	0.050	0.025	0.010
117	0.465	0.625	0.743	0.861	1.019
Reject?	No				

Kolmogorov-Smirnov Test with Model 1 - Inverted Weibull

Sample size 117
Normal test statistic 0.05156
Modified test statistic 0.55766

Note: The following critical values are exact.

Sample Size	Critical Values for Level of Significance (alpha)			
	0.100	0.050	0.025	0.010
50	0.790	0.856	0.922	0.988
infinity	0.803	0.874	0.939	1.007
Reject?	No			

Equal-Probable Chi-Square Test with Model 1 - Inverted Weibull

Number of intervals 23
Expected (model) count 5.08696
Test statistic 22.37607

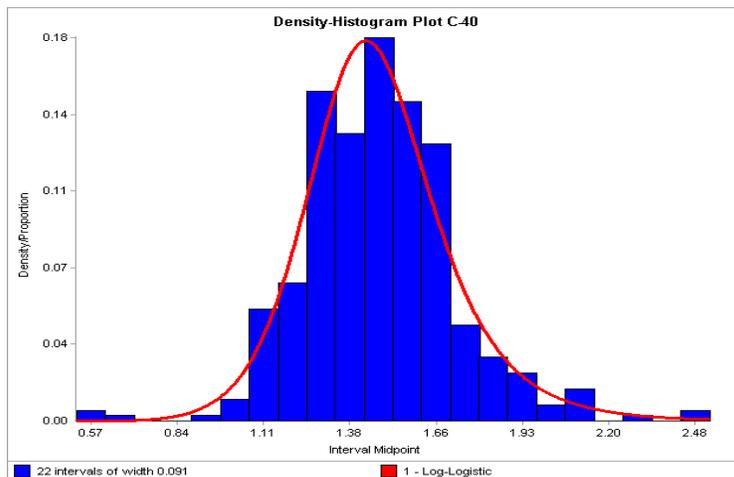
Degrees of Freedom	Observed Level of Significance	Critical Values for Level of Significance (alpha)				
		0.25	0.15	0.10	0.05	0.01
22	0.438	26.039	28.822	30.813	33.924	40.289
	Reject?	No				

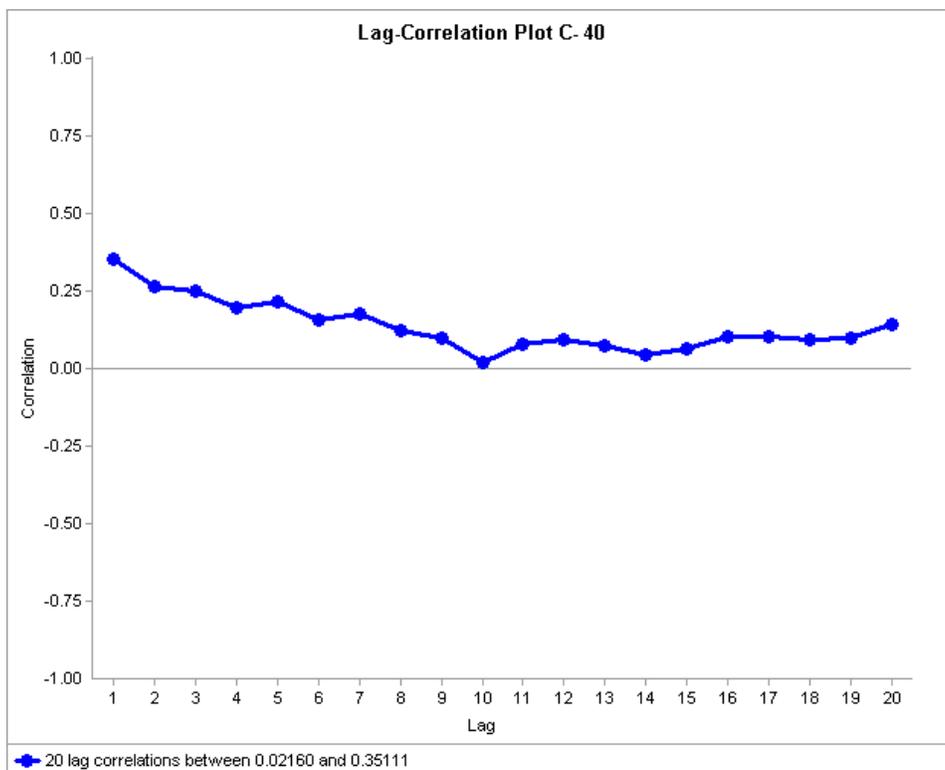
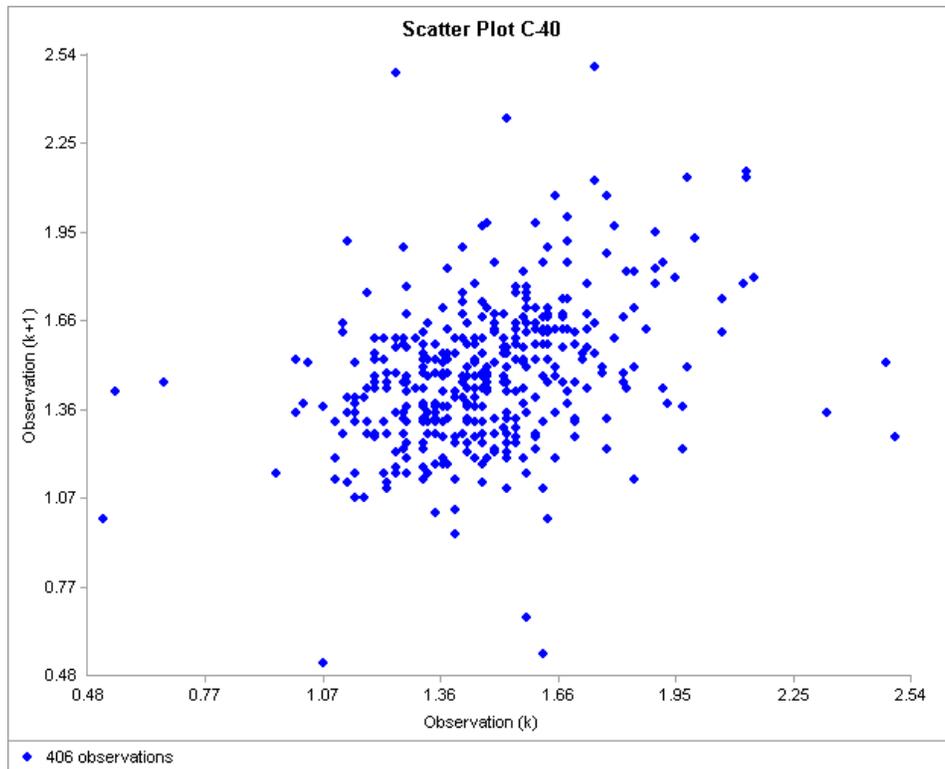
B. Anexo: Estadística tiempo de llenado cilindro 40 libras.

Estadística descriptiva - Tiempos de llenado en minutos

	C-40
Media	1.48
Error típico	0.01
Mediana	1.47
Moda	1.32
Desviación estándar	0.24
Varianza de la muestra	0.06
Curtosis	2.44
Coefficiente de asimetría	0.39
Rango	1.98
Mínimo	0.52
Máximo	2.50
Suma	599.03
Cuenta	406
Nivel de confianza(95,0%)	0.02

Interval	Upper Endpoint	Proportion
Below	0.52	0
1	0.611	0.00493
2	0.702	0.00246
3	0.793	0
4	0.884	0
5	0.975	0.00246
6	1.066	0.00985
7	1.157	0.05172
8	1.248	0.06404
9	1.339	0.15271
10	1.43	0.133
11	1.521	0.17734
12	1.612	0.14778
13	1.703	0.12808
14	1.794	0.04433
15	1.885	0.02956
16	1.976	0.02217
17	2.067	0.00739
18	2.158	0.01478
19	2.249	0
20	2.34	0.00246
21	2.431	0
22	2.522	0.00493
Above	infinity	0





Distribuciones estadísticas que mejor se ajustan a los datos de llenado del cilindro de 40 libras depurado con ExperFit.

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters
1 - Log-Logistic	98.86	Location 0.00000
		Scale 1.45988
		Shape 11.21057
2 - Erlang(E)	89.77	Location 0.09791
		Scale 0.03532
		Shape 39
3 - Gamma	88.64	Location 0.00000
		Scale 0.04081
		Shape 36.14778

23 models are defined with scores between 2.27 and 98.86

Absolute Evaluation of Model 1 - Log-Logistic

Evaluation: Good

Suggestion: Additional evaluations using Comparisons Tab might be informative.
See Help for more information.

Additional Information about Model 1 - Log-Logistic

"Error" in the model mean
relative to the sample mean -0.00379 = 0.26%

Flexsim Representation of Model 1 - Log-Logistic

Use:

When using a picklist option:	
Distribution	Log-Logistic
Location	0.000000
Scale	1.459876
Shape	11.210569
When using code:	
loglogistic{ 0.000000, 1.459876, 11.210569, <stream> }	

Test estadístico del cilindro de 40 libras. Distribución Log-Logistic

Anderson-Darling Test with Model 1 - Log-Logistic

Sample size 406
 Test statistic 0.45725

Note: The following critical values are exact.

Sample Size	Critical Values for Level of Significance (alpha)					
	0.250	0.100	0.050	0.025	0.010	0.005
406	0.426	0.563	0.660	0.769	0.905	1.009
Reject?	Yes	No				

Kolmogorov-Smirnov Test with Model 1 - Log-Logistic

Sample size 406
 Normal test statistic 0.03487
 Modified test statistic 0.70252

Note: The following critical values are exact.

Sample Size	Critical Values for Level of Significance (alpha)			
	0.100	0.050	0.025	0.010
50	0.708	0.770	0.817	0.873
infinity	0.715	0.780	0.827	0.886
Reject?	No			

Equal-Probable Chi-Square Test with Model 1 - Log-Logistic

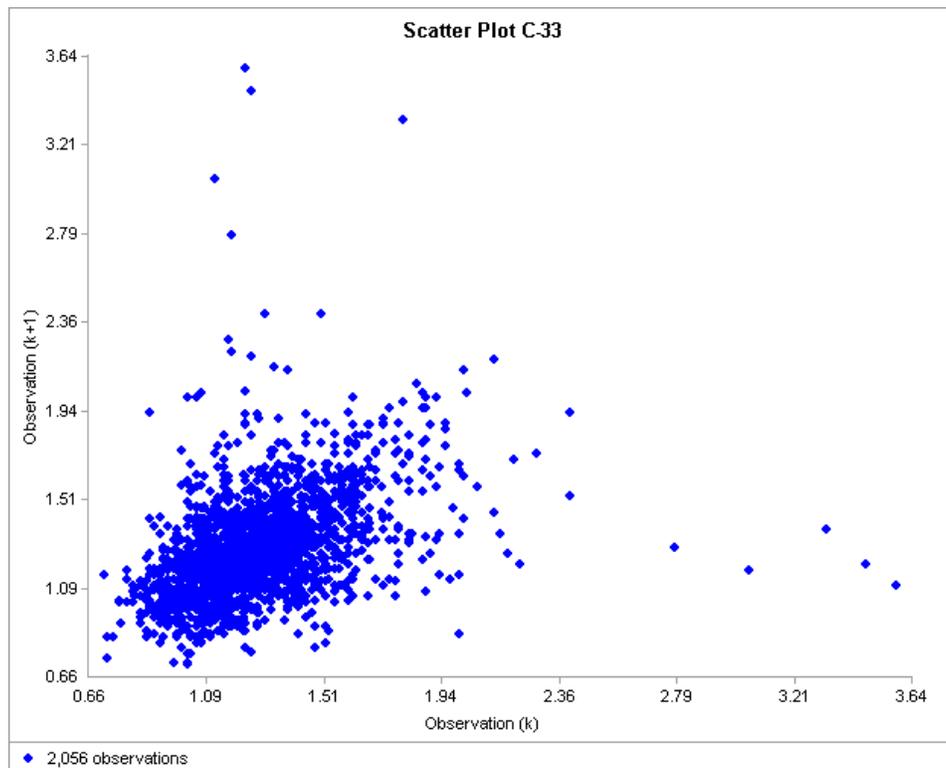
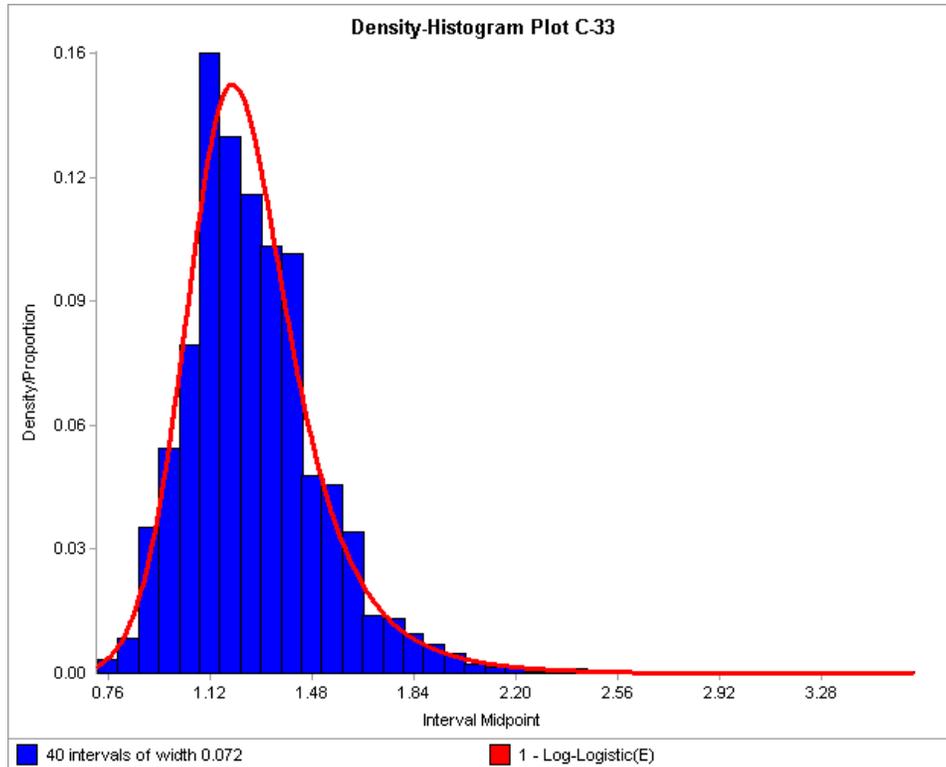
Number of intervals 40
 Expected (model) count 10.15
 Test statistic 110.65025

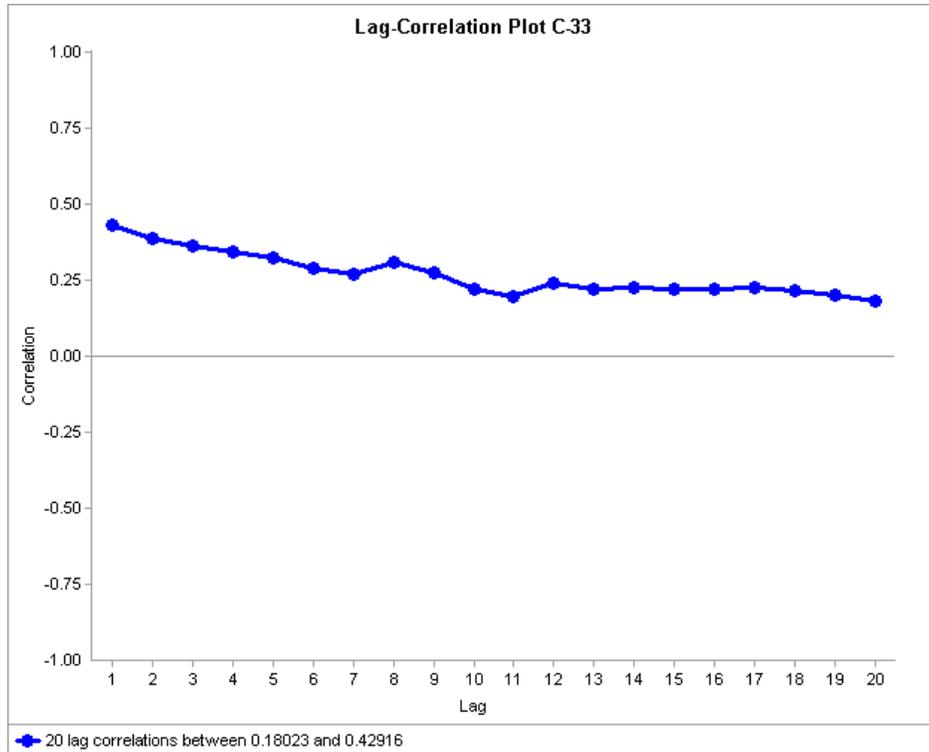
Degrees of Freedom	Observed Level of Significance	Critical Values for Level of Significance (alpha)				
		0.25	0.15	0.10	0.05	0.01
39	0.000	44.539	48.126	50.660	54.572	62.428
	Reject?	Yes				

C.Anexo: Estadística tiempo de llenado cilindro 33 libras.

Estadística descriptiva - Tiempos de llenado en minutos	
C-33	
Media	1.27
Error típico	0.01
Mediana	1.23
Moda	1.18
Desviación estándar	0.25
Varianza de la muestra	0.06
Curtosis	10.66
Coefficiente de asimetría	1.89
Rango	2.87
Mínimo	0.72
Máximo	3.58
Suma	2615.98
Número de observaciones	2,056
Nivel de confianza(95,0%)	0.01

Interval	Upper Endpoint	Proportion
Below	0.72	0
1	0.792	0.0034
2	0.864	0.00875
3	0.936	0.03648
4	1.008	0.05642
5	1.08	0.0822
6	1.152	0.15516
7	1.224	0.13424
8	1.296	0.11965
9	1.368	0.107
10	1.44	0.10506
11	1.512	0.04961
12	1.584	0.04718
13	1.656	0.03551
14	1.728	0.01459
15	1.8	0.01362
16	1.872	0.00973
17	1.944	0.0073
18	2.016	0.00486
19	2.088	0.00243
20	2.16	0.00146
21	2.232	0.00146
22	2.304	0.00048638
23	2.376	0
24	2.448	0.00097276
25	2.52	0
26	2.592	0
27	2.664	0
28	2.736	0
29	2.808	0.00048638
30	2.88	0
31	2.952	0
32	3.024	0
33	3.096	0.00048638
34	3.168	0
35	3.24	0
36	3.312	0
37	3.384	0.00048638
38	3.456	0
39	3.528	0.00048638
40	3.6	0.00048638
Above	infinity	0





Distribuciones estadísticas que mejor se ajustan a los datos de llenado del cilindro de 33 libras depurado con ExperFit.

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters
1 - Log-Logistic(E)	100.00	Location 0.46391
		Scale 0.77451
		Shape 6.18183
2 - Pearson Type V(E)	95.83	Location 0.18443
		Scale 23.67295
		Shape 22.77456
3 - Pearson Type VI(E)	94.17	Location 0.28998
		Scale 0.21142
		Shape #1 99.97892
		Shape #2 22.53125

31 models are defined with scores between 0.83 and 100.00

Absolute Evaluation of Model 1 - Log-Logistic(E)

Evaluation: Indeterminate

Suggestion: Additional evaluations using Comparisons Tab are strongly recommended.
See Help for more information.

Additional Information about Model 1 - Log-Logistic(E)

"Error" in the model mean
relative to the sample mean $-4.3193e-4 = 0.03\%$

Flexsim Representation of Model 1 - Log-Logistic(E)

Use:

When using a picklist option:

Distribution	Log-Logistic
Location	0.463905
Scale	0.774510
Shape	6.181834

When using code:

`loglogistic(0.463905, 0.774510, 6.181834, <stream>)`

Test estadístico del cilindro de 33 libras. Distribución Log-Logistic (E)

Anderson-Darling Test with Model 1 - Log-Logistic(E)

Sample size 2,056
Test statistic 1.06599

Note: No critical values exist for this special case.
The following critical values are for the case where all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)					
	0.250	0.100	0.050	0.025	0.010	0.005
2,056	1.248	1.933	2.492	3.070	3.857	4.500
Reject?	No					

Kolmogorov-Smirnov Test with Model 1 - Log-Logistic(E)

Sample size 2,056
Normal test statistic 0.03230
Modified test statistic 1.46446

Note: No critical values exist for this special case.
The following critical values are for the case where all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)				
	0.150	0.100	0.050	0.025	0.010
2,056	1.135	1.221	1.354	1.476	1.624
Reject?	Yes			No	

Equal-Probable Chi-Square Test with Model 1 - Log-Logistic(E)

Number of intervals 40
Expected (model) count 51.4
Test statistic 587.6965

Warning: The test may not be statistically valid because a method other than maximum likelihood was used to estimate parameters.

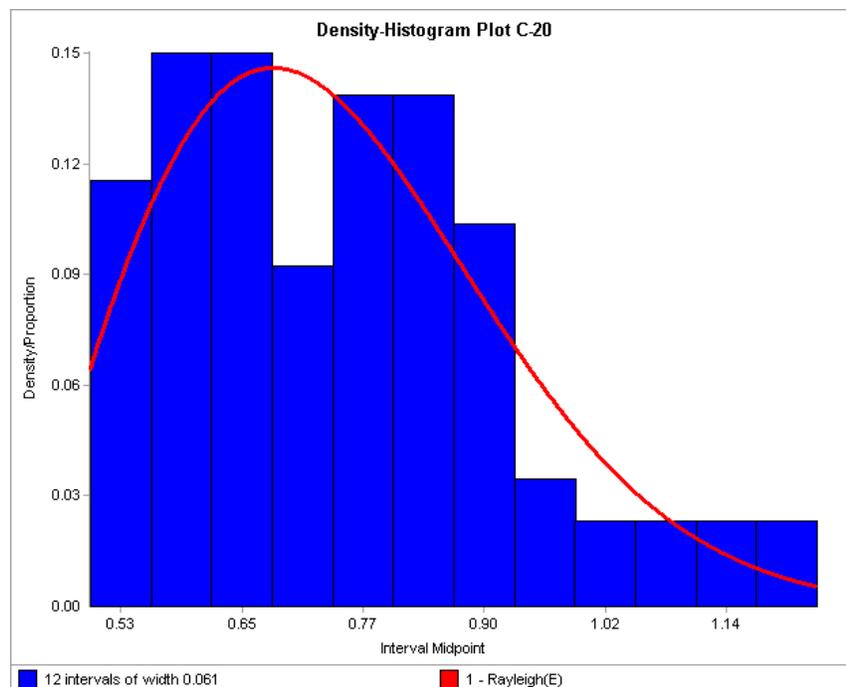
Degrees of Freedom	Observed Level of Significance	Critical Values for Level of Significance (alpha)				
		0.25	0.15	0.10	0.05	0.01
39	0.000	44.539	48.126	50.660	54.572	62.428
	Reject?	Yes				

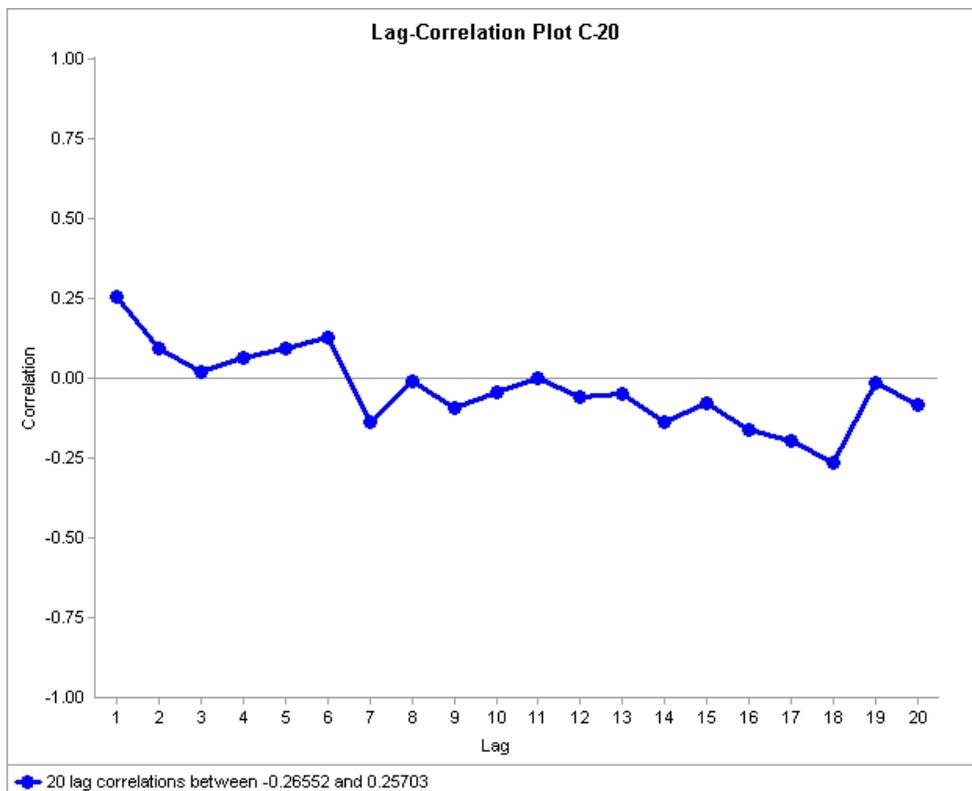
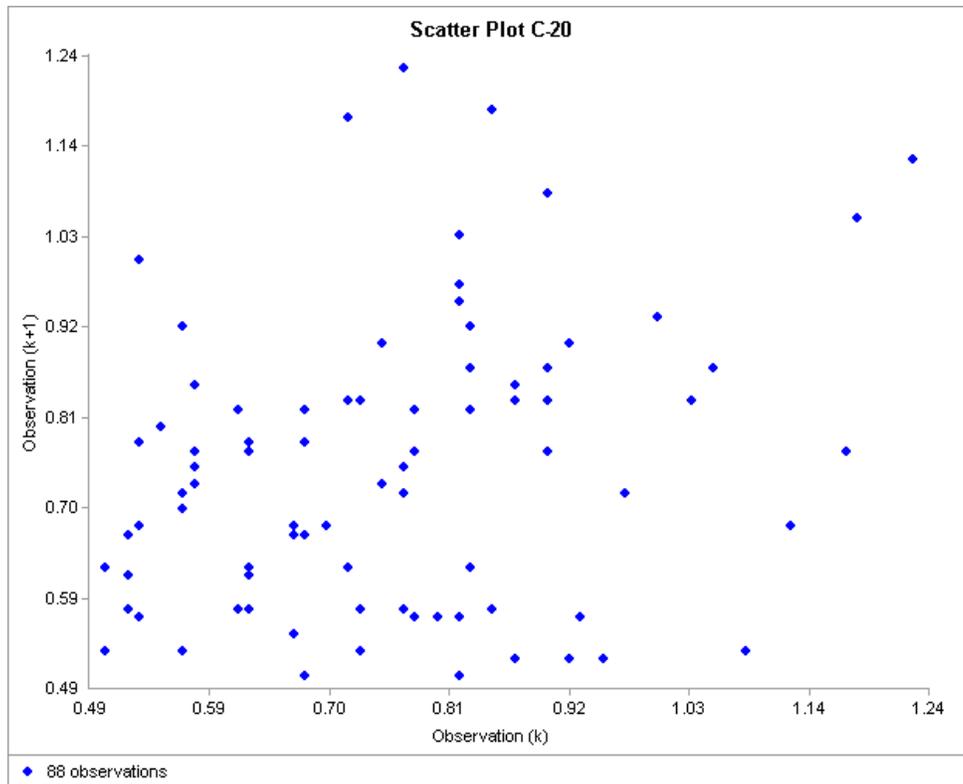
D. Estadística tiempo de llenado cilindro 20 libras.

Estadística descriptiva - Tiempos de llenado en minutos

	<i>C-20</i>
Media	0.75
Error típico	0.02
Mediana	0.74
Moda	0.58
Desviación estándar	0.17
Varianza de la muestra	0.03
Curtosis	0.20
Coficiente de asimetría	0.68
Rango	0.73
Mínimo	0.50
Máximo	1.23
Suma	66.07
Número de observaciones	88
Nivel de confianza(95,0%)	0.04

Interval	Upper Endpoint	Proportion
Below	0.50000	0.00000
1	0.56100	0.11364
2	0.62200	0.14773
3	0.68300	0.14773
4	0.74400	0.09091
5	0.80500	0.13636
6	0.86600	0.13636
7	0.92700	0.10227
8	0.98800	0.03409
9	1.04900	0.02273
10	1.11000	0.02273
11	1.17100	0.02273
12	1.23200	0.02273
Above	infinity	0.00000





Distribuciones estadísticas que mejor se ajustan a los datos de llenado del cilindro de 20 libras depurado con ExperFit.

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters
1 - Rayleigh(E)	97.41	Location 0.42822
		Scale 0.36357
2 - Beta	95.69	Lower endpoint 0.47155
		Upper endpoint 1.44455
		Shape #1 1.72341
		Shape #2 4.26621
3 - Weibull(E)	91.38	Location 0.44018
		Scale 0.35120
		Shape 1.95481

30 models are defined with scores between 0.86 and 97.41

Absolute Evaluation of Model 1 - Rayleigh(E)

Evaluation: Good

Suggestion: Additional evaluations using Comparisons Tab might be informative.

See Help for more information.

Additional Information about Model 1 - Rayleigh(E)

"Error" in the model mean

relative to the sample mean $3.7691e-4 = 0.05\%$

Flexsim Representation of Model 1 - Rayleigh(E)

Use:

When using a picklist option:

Distribution	Weibull
Location	0.428216
Scale	0.363567
Shape	2.000000

When using code:

```
weibull( 0.428216, 0.363567, 2.000000, <stream>)
```

Test estadístico del cilindro de 20 libras. Distribución Rayleigh(E)

Anderson-Darling Test with Model 1 - Rayleigh(E)

Sample size 88
Test statistic 0.40773

Note: No critical values exist for this special case.
The following critical values are for the case where all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)					
	0.250	0.100	0.050	0.025	0.010	0.005
88	1.248	1.933	2.492	3.070	3.857	4.500
Reject?	No					

Kolmogorov-Smirnov Test with Model 1 - Rayleigh(E)

Sample size 88
Normal test statistic 0.07869
Modified test statistic 0.73814

Note: No critical values exist for this special case.
The following critical values are for the case where all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)				
	0.150	0.100	0.050	0.025	0.010
88	1.122	1.207	1.339	1.460	1.605
Reject?	No				

Equal-Probable Chi-Square Test with Model 1 - Rayleigh(E)

Number of intervals 17
Expected (model) count 5.17647
Test statistic 39.5

Warning: The test may not be statistically valid because a method other than maximum likelihood was used to estimate parameters.

Degrees of Freedom	Observed Level of Significance	Critical Values for Level of Significance (alpha)				
		0.25	0.15	0.10	0.05	0.01
16	0.001	19.369	21.793	23.542	26.296	32.000
	Reject?	Yes				

E. MTBF y MTTR.

Datos del sistema real de los equipos más representativos.

BASCULA # 1		
EVENTO	TBFmin.	TTRmin.
E1	111.7	95.3
E2	89.1	59.9
E3	59.2	25.8
E4	26.2	156.9
E5	94.4	38.6
E6	1.2	24.8
E7	181	47.9
E8	152.1	44.8
E9	260.2	37.8
E10	216.2	40.7
E11	224.3	92.6
E12	54.3	52.8
E13	135.4	58.7
E14	87.2	23
E15	102.3	57.6
E16	11	36.6
E17	23.4	11.8
E18	85.4	57.9
E19	194	20.8
E20	106.2	12.1
E21	22.1	43.9
E22	33.2	64.7
E23	77	42.8
E24	16.2	
E25	59.3	
E26	190.2	
E27	202	
Total	2,814.80	1,147.80

$$MTBF_1 = \frac{\sum_{j=1}^m DT_j}{n} = \frac{2,814.8}{27} = 104.3$$

$$MTTR_1 = \frac{\sum_{i=1}^n UT_i}{m} = \frac{1,147.8}{23} = 49.90$$

$$D_1 = \frac{MTBF}{MTBF + MTTR} = \frac{104.3}{104.3 + 49.90} \approx 68\%$$

$$\sum_{i=1}^n UT_i = 2,814.8$$

$$n = 27$$

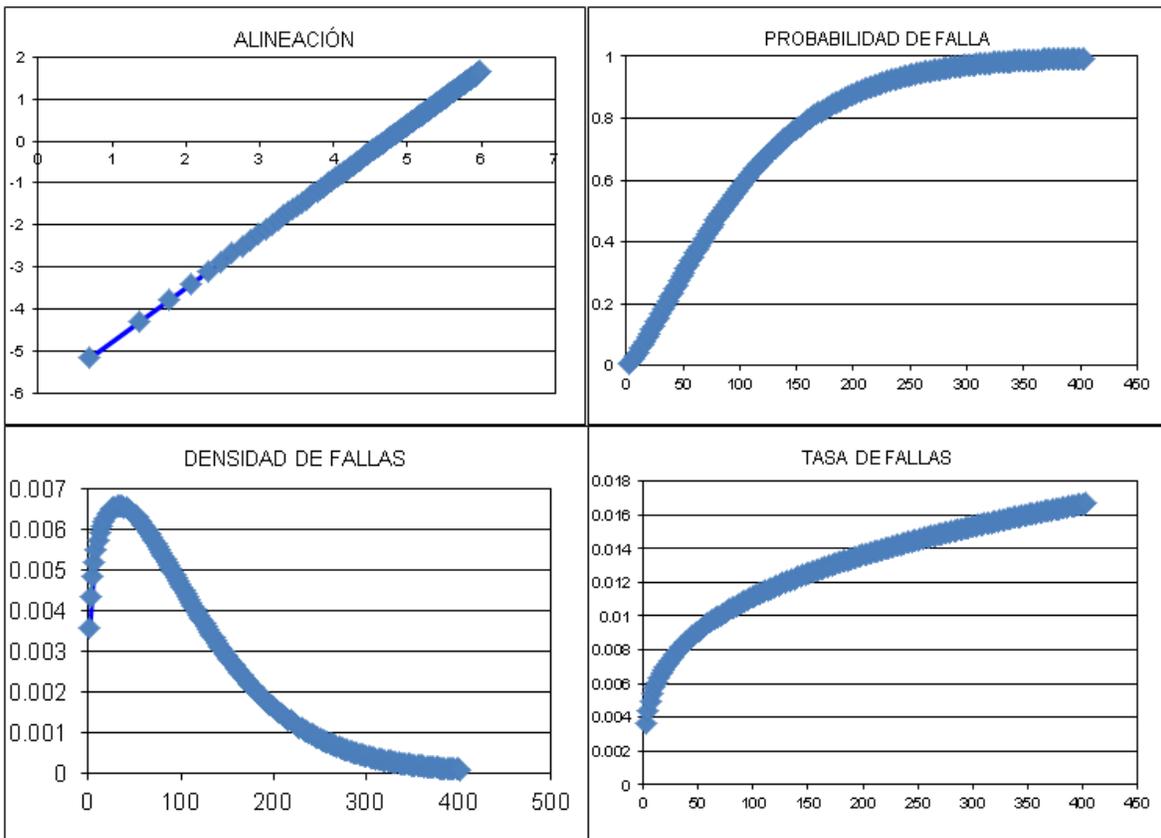
$$\sum_{j=1}^m DT_j = 1,147.8$$

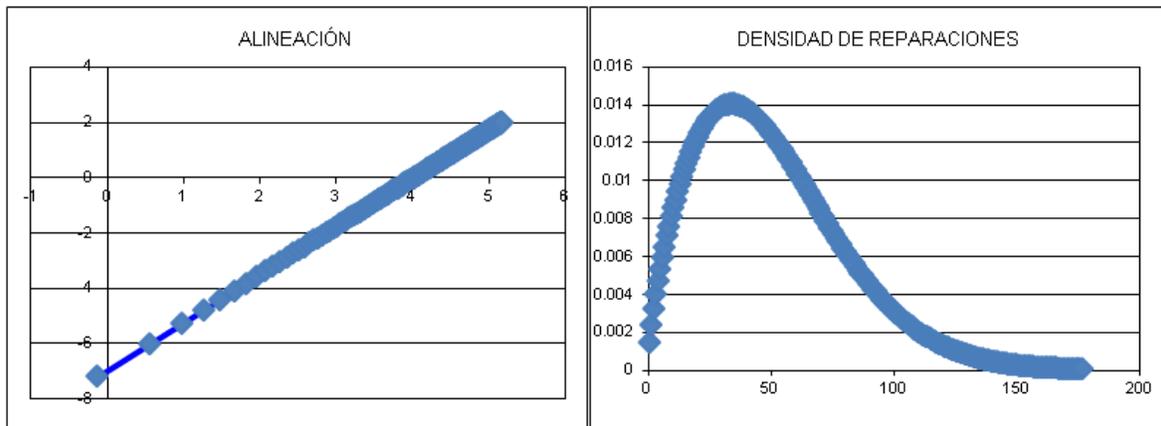
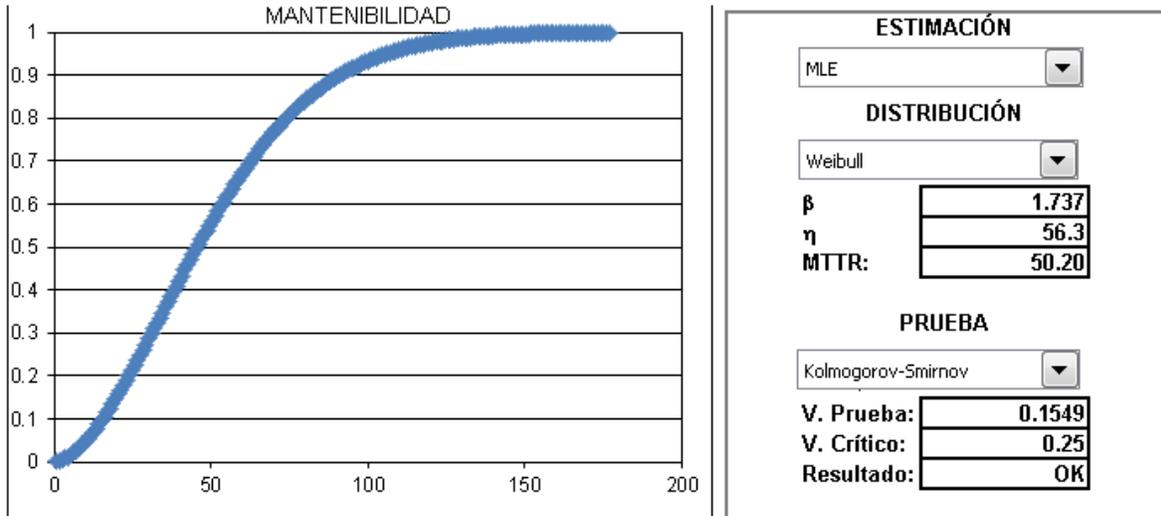
$$m = 23$$

Flexsim Representation of Model 3 - Weibull

Use:

When using a picklist option:	
Distribution	Weibull
Location	0.000000
Scale	111.851255
Shape	1.289139
When using code:	
<code>weibull(0.000000, 111.851255, 1.289139, <stream>)</code>	





Flexsim Representation of Model 19 - Weibull

Use:

When using a picklist option:	
Distribution	Weibull
Location	0.000000
Scale	56.340208
Shape	1.736865
When using code:	
<code>weibull(0.000000, 56.340208, 1.736865, <stream>)</code>	

MTBF	103.48
MTTR	50.20
DISP	67.33%

BASCULA # 3		
EVENTO	TBFmin.	TTRmin.
E1	141.2	55.8
E2	39.2	18.8
E3	67.6	17.4
E4	24.7	31.3
E5	48.2	17.8
E6	4.3	58.8
E7	130.9	20.1
E8	39.2	43.9
E9	201	
Total	696.30	263.90

$$MTBF_3 = \frac{\sum_{j=1}^m DT_j}{n} = \frac{696.3}{9} = 77.4$$

$$MTTR_3 = \frac{\sum_{i=1}^n UT_i}{m} = \frac{1,147.8}{8} = 32.99$$

$$D_3 = \frac{MTBF}{MTBF + MTTR} = \frac{77.4}{77.4 + 32.99} \approx 70\%$$

$$\sum_{i=1}^n UT_i = 696.3$$

n = 9

$$\sum_{j=1}^m DT_j = 263.9$$

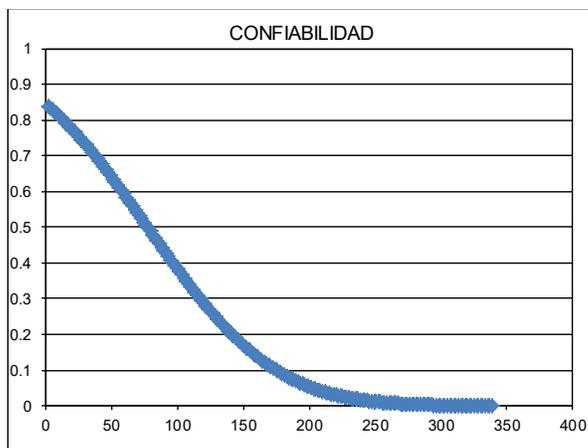
m = 8

Número de datos	8	Tiempos de Operación entre fallas	141.2	Tiempos de Mantenimiento	55.8	MÉTODO DE ESTIMACIÓN
Número de censurados	0		39.2		18.8	<input checked="" type="checkbox"/> BENARD
Pvalue	0.1		67.6		17.4	<input checked="" type="checkbox"/> KAPLAN MEYER
			24.7		31.3	<input checked="" type="checkbox"/> 1/(N+1)
			48.2		17.8	<input checked="" type="checkbox"/> MLE
Número de datos confiabilidad	9		4.3		58.8	DISTRIBUCIONES A USAR
Número de censurados	0		130.9		20.1	<input checked="" type="checkbox"/> WEIBULL
			39.2		43.9	<input checked="" type="checkbox"/> EXPONENCIAL
Número de datos mantenibilidad	8		201			<input type="checkbox"/> EXPONENCIAL2
Número de censurados	0					<input checked="" type="checkbox"/> NORMAL
						<input checked="" type="checkbox"/> LOGNORMAL
						PRUEBAS
						<input checked="" type="checkbox"/> Kolmogorov - Smirnov
						<input checked="" type="checkbox"/> Kolmogorov - Smirnov 2
						<input checked="" type="checkbox"/> Anderson-Darling
						<input checked="" type="checkbox"/> Anderson - Darling mod.
						<input checked="" type="checkbox"/> Chi cuadrado

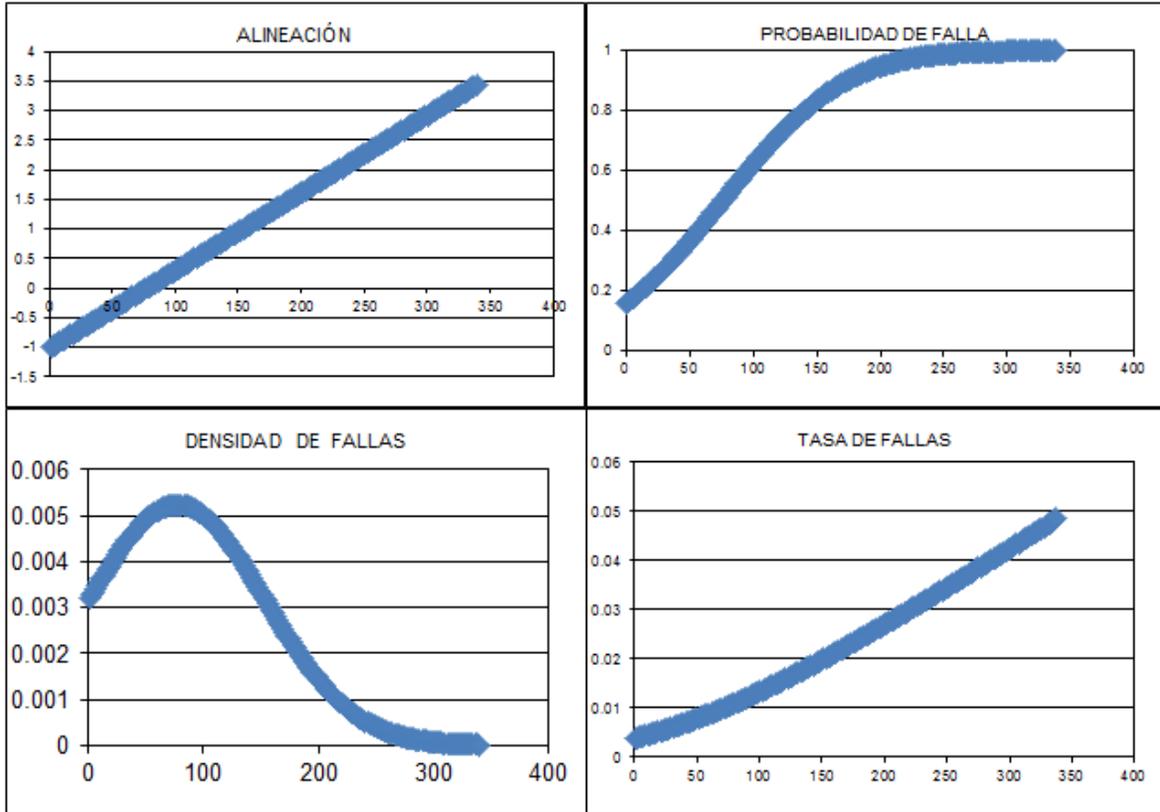
C

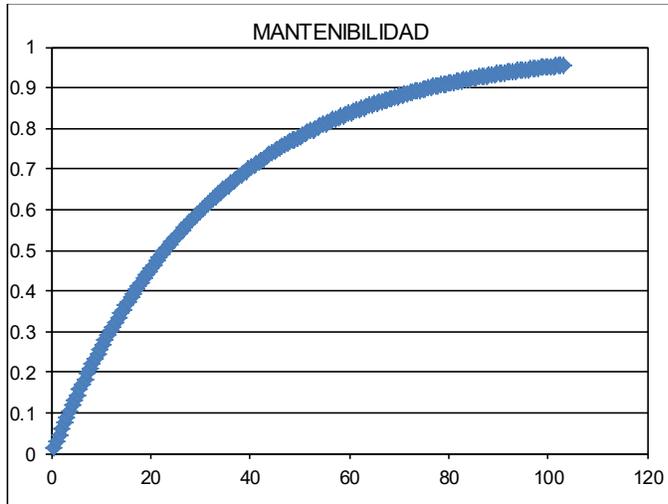
M

D

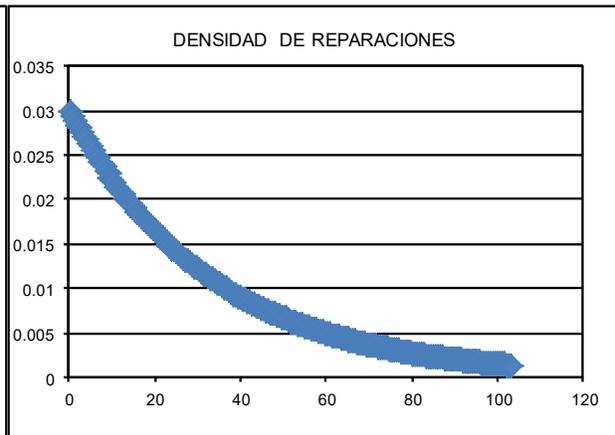
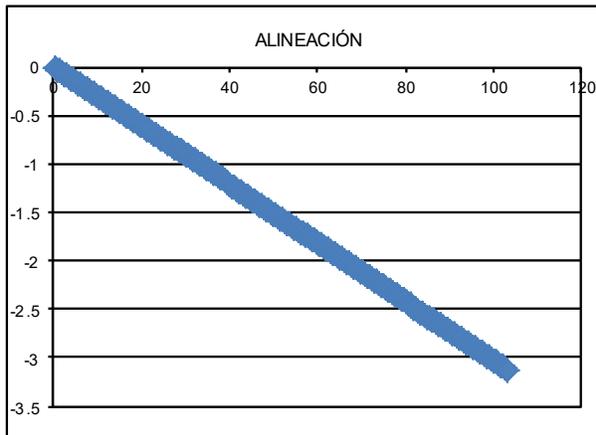


ESTIMACIÓN	
BENARD	▼
DISTRIBUCIÓN	
Normal	▼
μ	77.367
σ	76.111
MTBF:	77.37
PRUEBA	
Kolmogorov-Smirnov	▼
V. Prueba:	0.2177
V. Crítico:	0.39
Resultado:	OK





ESTIMACIÓN	
MLE	▼
DISTRIBUCIÓN	
Exponencial	▼
λ	0.030
MTTR:	32.99
PRUEBA	
Kolmogorov-Smirnov	▼
V. Prueba:	0.2849
V. Crítico:	0.41
Resultado:	OK



MTBF	77.37
MTTR	32.99
DISP	70.11%

BASCULA # 8		
EVENTO	TBFmin.	TTRmin.
E1	109.4	98.6
E2	86.3	60.7
E3	50.4	34.6
E4	28.3	173.7
E5	14.1	35.9
E6	24.3	43.8
E7	2.5	18.5
E8	183.4	10.9
E9	86.1	46.5
E10	60.6	42.8
E11	258.3	16.5
E12	140.6	10.6
E13	33.4	36.6
E14	18.4	39.5
E15	223.6	78.6
E16	52.4	52.8
E17	135.4	62
E18	101.2	22.7
E19	99	14.5
E20	12.3	60.8
E21	1.5	36.7
E22	3.3	
E23	84.3	
E24	195	
Total	2,004.10	997.30

$$MTBF_8 = \frac{\sum_{j=1}^m DT_j}{n} = \frac{2,004.1}{24} = 83.5$$

$$MTTR_8 = \frac{\sum_{i=1}^n UT_i}{m} = \frac{997.3}{21} = 47.49$$

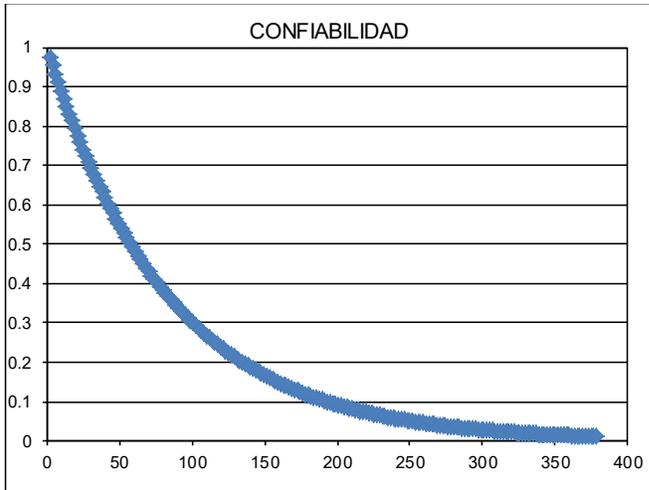
$$D_8 = \frac{MTBF}{MTBF + MTTR} = \frac{83.5}{83.5 + 47.49} \approx 64\%$$

$$\sum_{i=1}^n UT_i = 2,004.1$$

$$n = 24$$

$$\sum_{j=1}^m DT_j = 997.3$$

$$m = 21$$



ESTIMACIÓN	
MLE	▼
DISTRIBUCIÓN	
Weibull	▼
β	0.993
η	83.276
MTBF:	83.53
PRUEBA	
Kolmogorov-Smirnov	▼
V. Prueba:	0.0949
V. Crítico:	0.24
Resultado:	OK

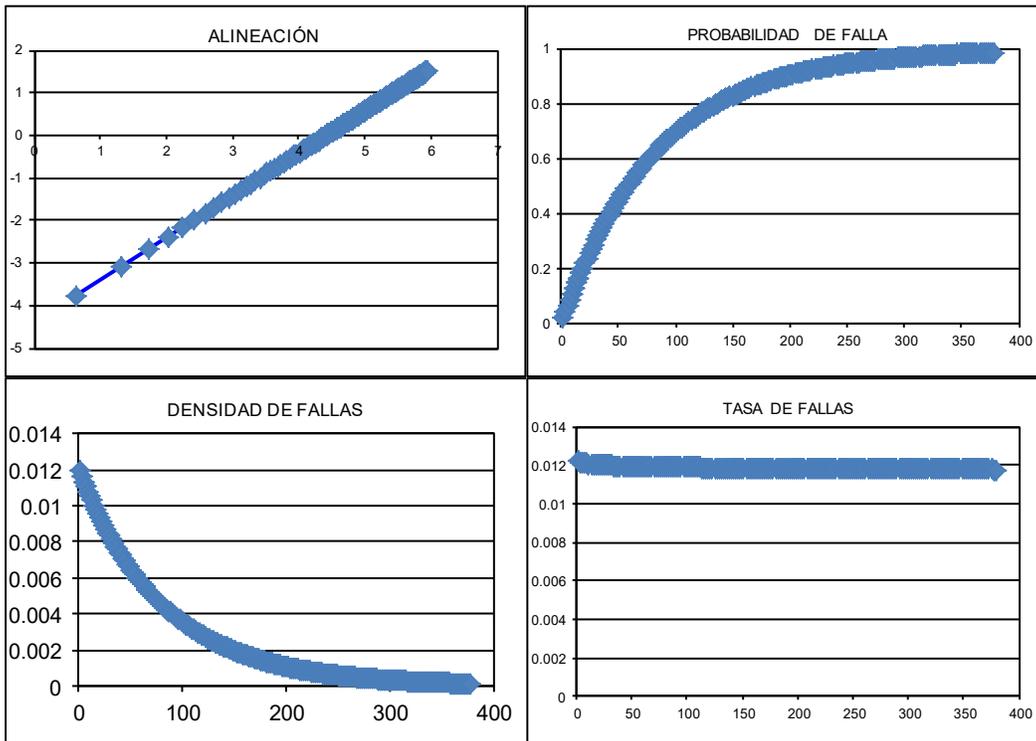
Flexsim Representation of Model 6 - Weibull

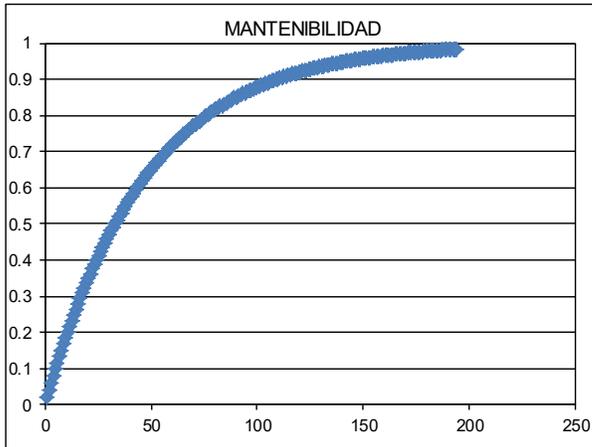
Use:

When using a picklist option:

Distribution	Weibull
Location	0.000000
Scale	83.276165
Shape	0.992903

When using code:
`weibull(0.000000, 83.276165, 0.992903, <stream>)`





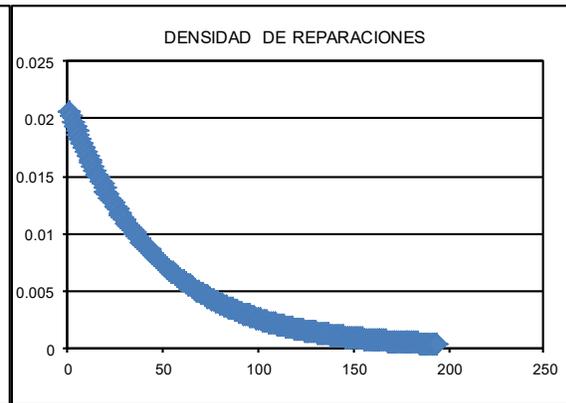
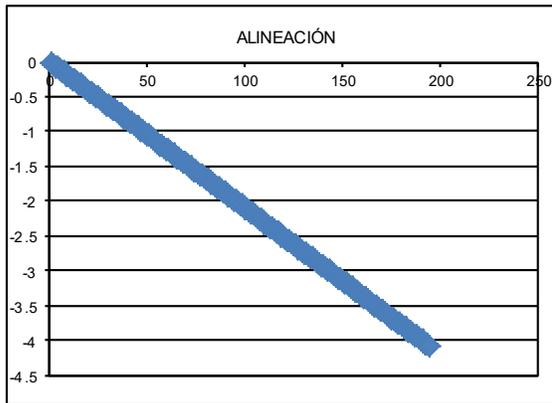
ESTIMACIÓN	
MLE	▼
DISTRIBUCIÓN	
Exponencial	▼
λ	0.021
MTTR:	47.49
PRUEBA	
Kolmogorov-Smirnov	▼
V. Prueba:	0.1840
V. Crítico:	0.26
Resultado:	OK

Flexsim Representation of Model 23 - Exponential

Use:

When using a picklist option:
Distribution **Exponential**
Location **0.000000**
Scale **47.490476**

When using code:
exponential(0.000000, 47.490476, <stream>)



MTBF	83.53
MTTR	47.49
DISP	63.75%

BASCULA # 10		
EVENTO	TBFmin.	TTRmin.
E1	140.4	57.6
E2	39.9	12.2
E3	108.3	32.7
E4	53.6	13.4
E5	1.6	65.4
E6	190.4	42.6
E7	202	
Total	736.20	223.90

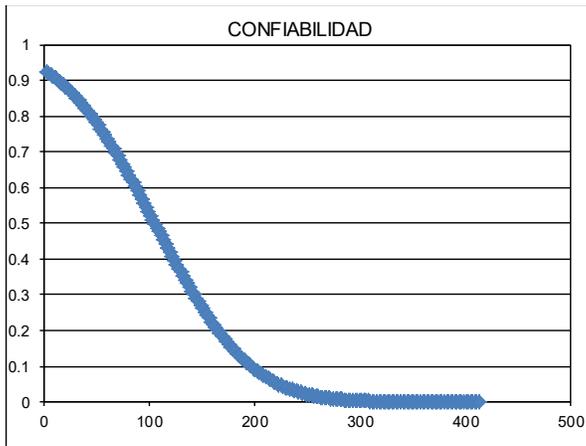
$$MTBF_{10} = \frac{\sum_{j=1}^m DT_j}{n} = \frac{736.2}{7} = 105.2$$

$$\sum_{i=1}^n UT_i = 2,004.1 \quad n = 7$$

$$MTTR_{10} = \frac{\sum_{i=1}^n UT_i}{m} = \frac{223.9}{6} = 37.32$$

$$\sum_{j=1}^m DT_j = 223.9 \quad m = 6$$

$$D_{10} = \frac{MTBF}{MTBF + MTTR} = \frac{105.2}{105.2 + 37.32} \approx 74\%$$



ESTIMACIÓN

MLE

DISTRIBUCIÓN

Normal

μ: 105.171

σ: 71.252

MTBF: 105.17

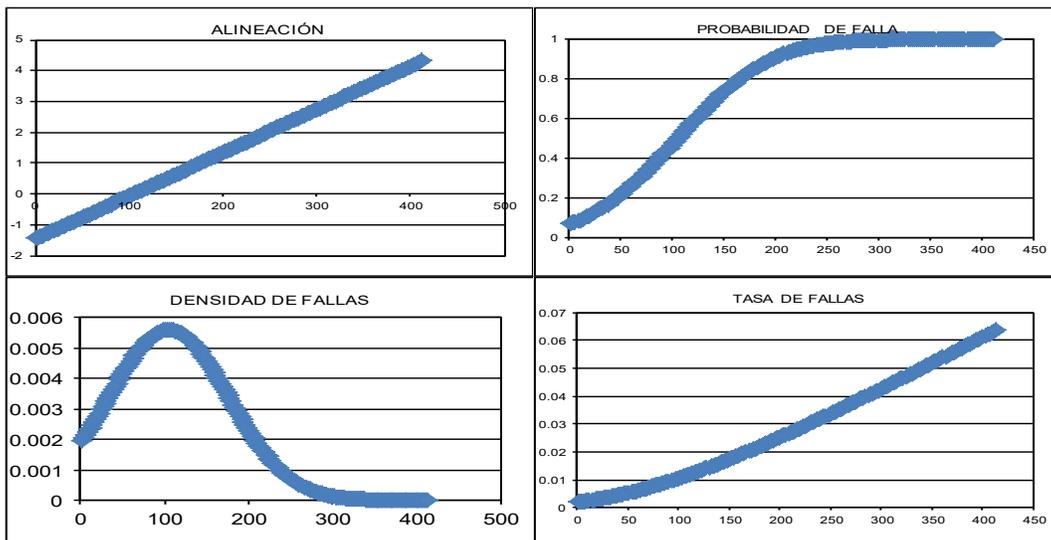
PRUEBA

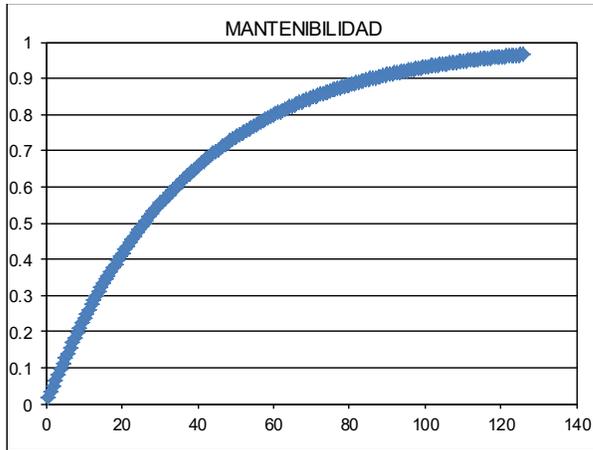
Kolmogorov-Smirnov

V. Prueba: 0.1940

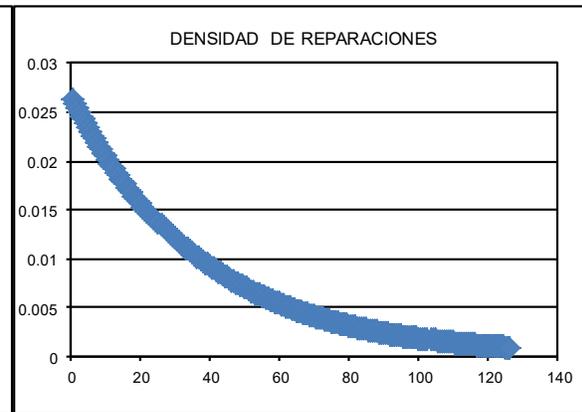
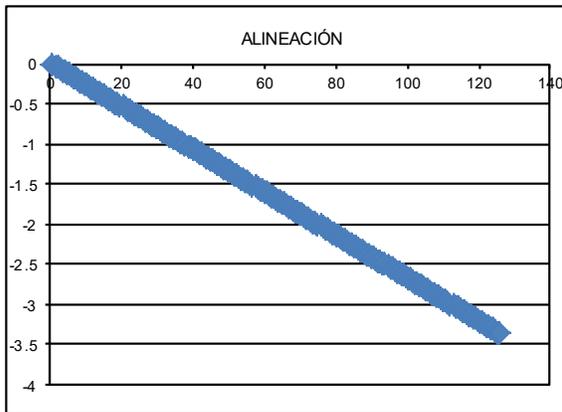
V. Crítico: 0.44

Resultado: OK





ESTIMACIÓN	
MLE	▼
DISTRIBUCIÓN	
Exponencial	▼
λ	0.027
MTTR:	37.32
PRUEBA	
Kolmogorov-Smirnov	▼
V. Prueba:	0.1733
V. Crítico:	0.47
Resultado:	OK



MTBF	105.17
MTTR	37.32
DISP	73.81%

BASCULA # 14		
EVENTO	TBFmin.	TTRmin.
E1	106.3	100.7
E2	88.1	59.9
E3	66.1	17.9
E4	29.6	152.4
E5	97	67
E6	179.1	40.8
E7	158.2	44
E8	262.1	50.1
E9	210.9	41
E10	217	77.9
E11	55.1	53.9
E12	136.1	60.9
E13	100.2	23.2
E14	99.1	54.6
E15	9.8	38.8
E16	27.4	56
E17	84.2	15.4
E18	194	37
E19	141.1	16.5
E20	38.6	64.9
E21	106	42.7
E22	48.5	
E23	1.2	
E24	190.3	
E25	202	
Total	2,848.00	1,115.60

$$MTBF_1 = \frac{\sum_{j=1}^m DT_j}{n} = \frac{2,848}{25} = 113.9$$

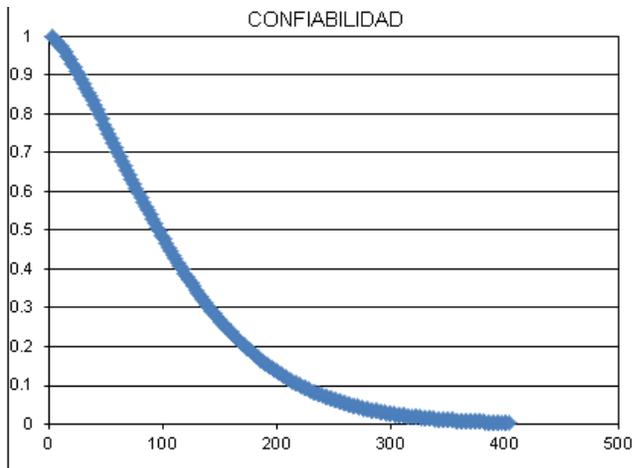
$$MTTR_1 = \frac{\sum_{i=1}^n UT_i}{m} = \frac{1,115.6}{21} = 53.12$$

$$D_1 = \frac{MTBF}{MTBF + MTTR} = \frac{113.9}{113.9 + 53.12} \approx 68\%$$

$$\sum_{i=1}^n UT_i = 2,848 \qquad \sum_{j=1}^m DT_j = 1,115.6$$

n = 25

m = 21

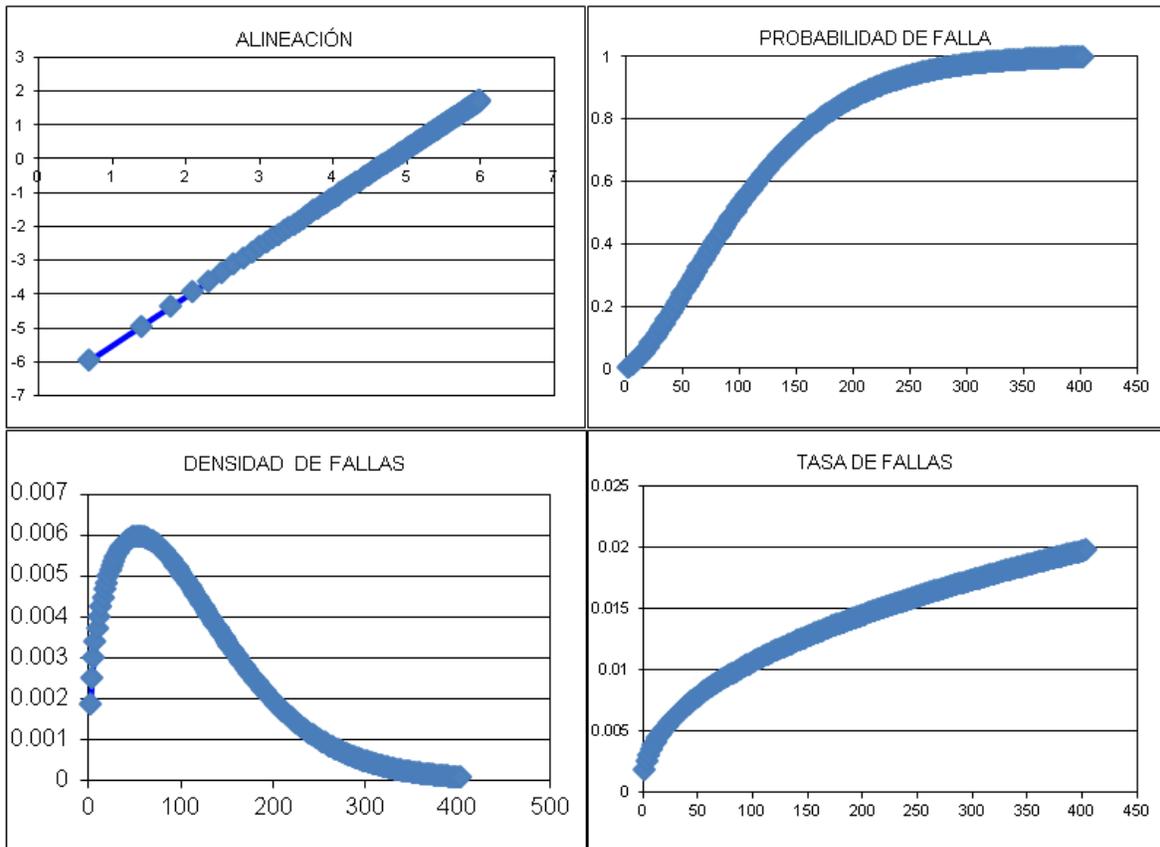


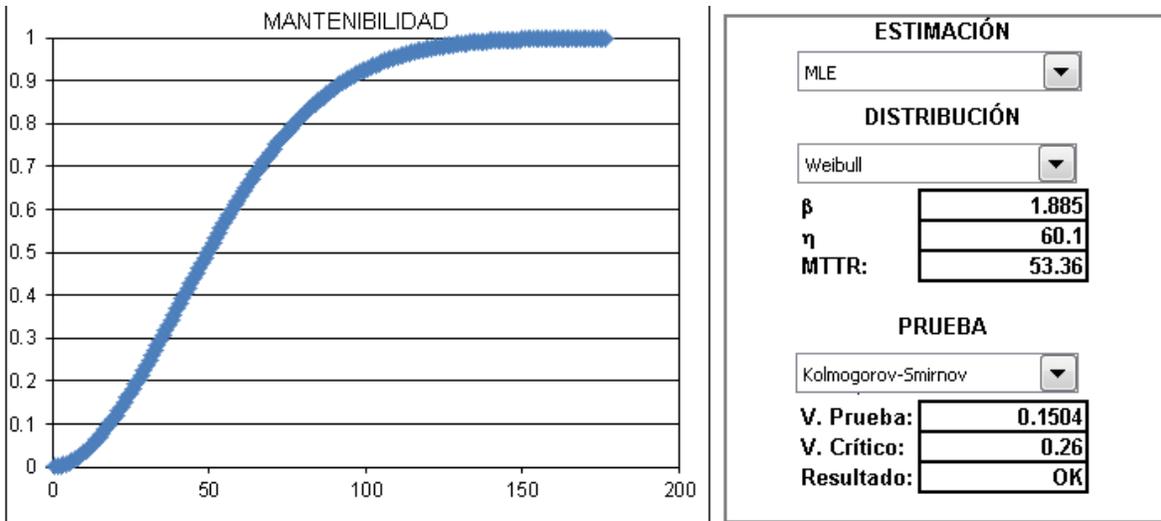
ESTIMACIÓN	
MLE	
DISTRIBUCIÓN	
Weibull	
β	1.446
η	123.923
MTBF:	112.41
PRUEBA	
Kolmogorov-Smirnov	
V. Prueba:	0.0755
V. Crítico:	0.24
Resultado:	OK

Flexsim Representation of Model 3 - Weibull

Use:

When using a picklist option:	
Distribution	Weibull
Location	0.000000
Scale	123.922686
Shape	1.446076
When using code:	
weibull(0.000000, 123.922686, 1.446076, <stream>)	





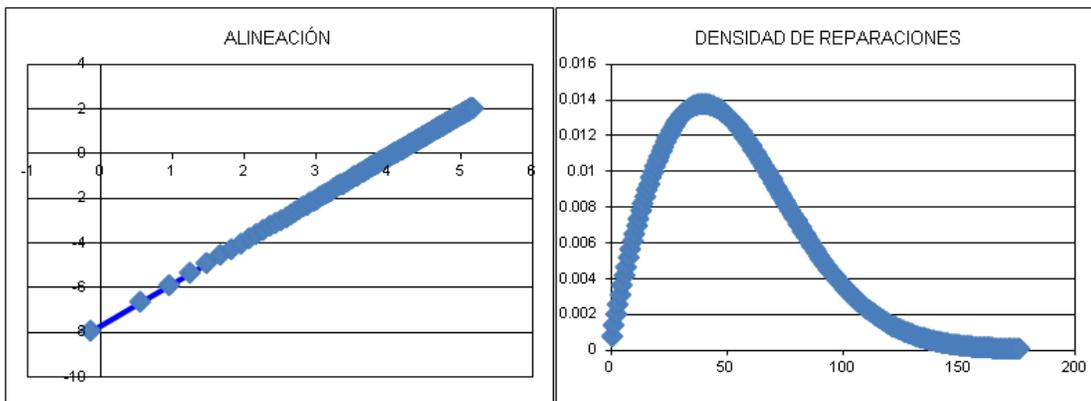
Flexsim Representation of Model 15 - Weibull

Use:

When using a picklist option:

Distribution	Weibull
Location	0.000000
Scale	60.113657
Shape	1.884766

When using code:
`weibull(0.000000, 60.113657, 1.884766, <stream>)`



MTBF	112.41
MTTR	53.36
DISP	67.81%

BASCULA # 16		
EVENTO	TBFmin.	TTRmin.
E1	107.5	99.5
E2	89.1	58.9
E3	64.2	17.8
E4	34.2	149.8
E5	97.4	66.7
E6	178.3	42.8
E7	157.2	43.8
E8	261.2	40.6
E9	217.4	39.7
E10	220.3	75.7
E11	56	52.8
E12	136.3	62.4
E13	101.2	24.2
E14	100.6	54.9
E15	6.8	35.7
E16	28.1	
E17	86.3	
E18	195	
Total	2,137.10	865.30

$$MTBF_{16} = \frac{\sum_{j=1}^m DT_j}{n} = \frac{2,137.1}{18} = 118.7$$

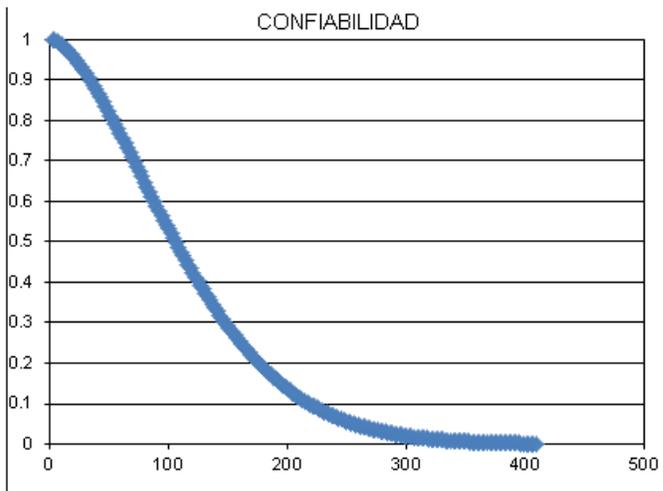
$$MTTR_{16} = \frac{\sum_{i=1}^n UT_i}{m} = \frac{865.3}{15} = 57.69$$

$$D_{16} = \frac{MTBF}{MTBF + MTTR} = \frac{118.7}{118.7 + 57.69} \approx 67\%$$

$$\sum_{i=1}^n UT_i = 2,137.1 \qquad \sum_{j=1}^m DT_j = 865.3$$

n = 18

m = 15

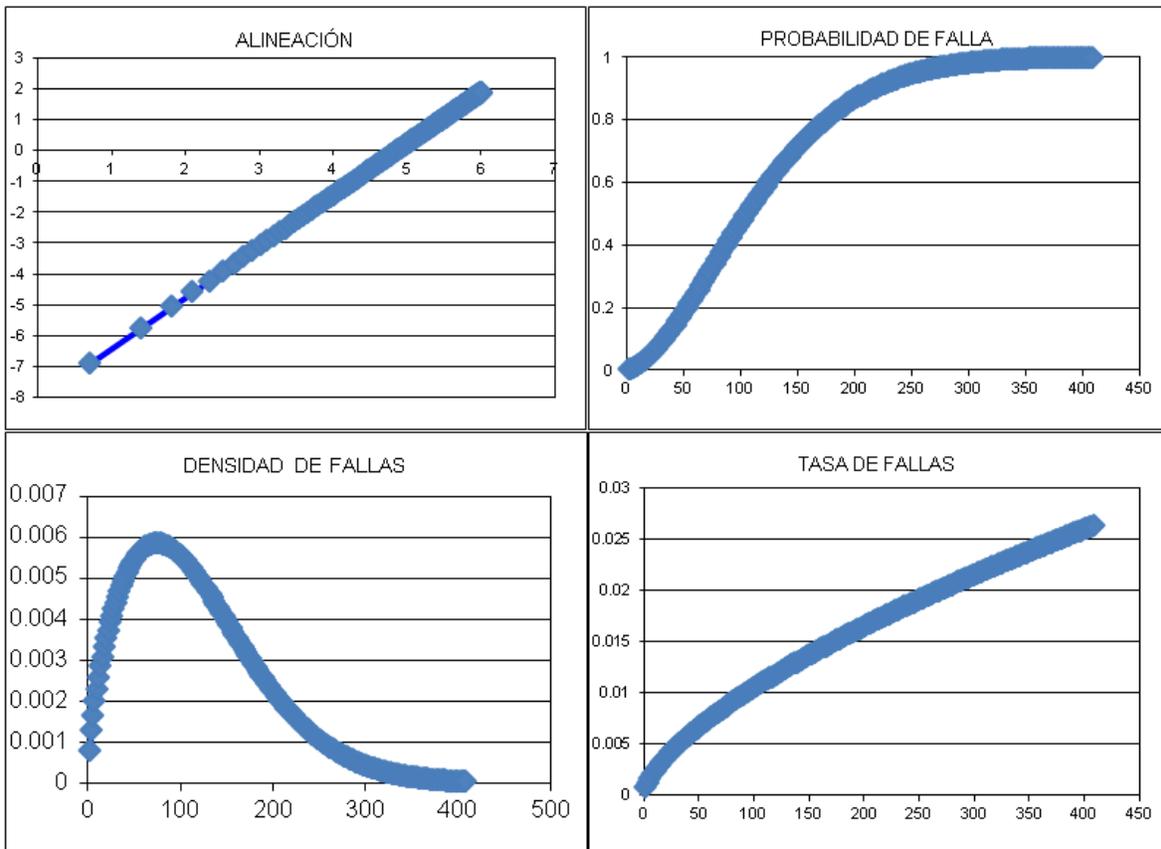


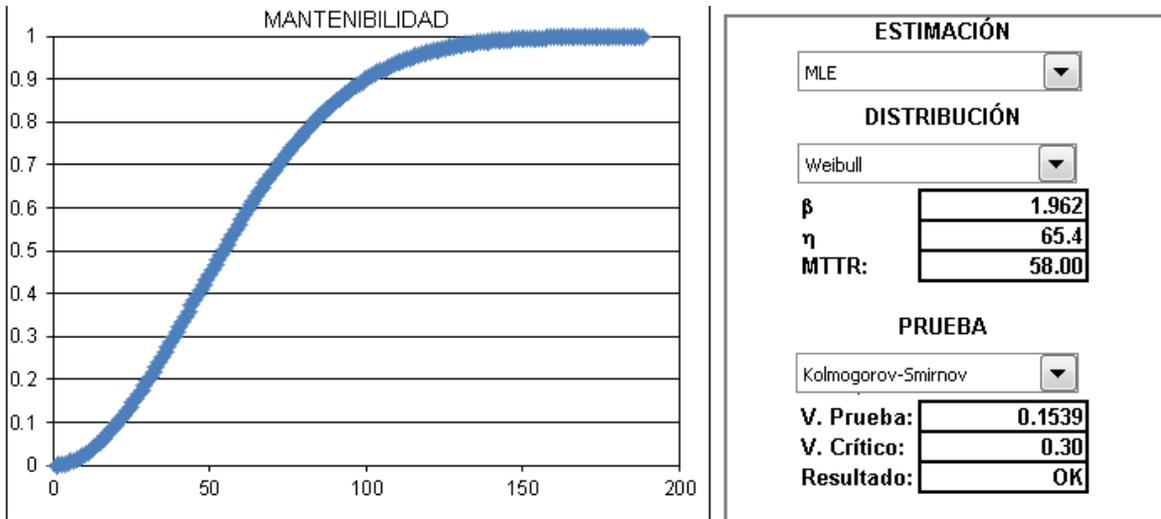
ESTIMACIÓN	
MLE	
DISTRIBUCIÓN	
Weibull	
β	1.656
η	132.050
MTBF:	118.05
PRUEBA	
Kolmogorov-Smirnov	
V. Prueba:	0.1021
V. Crítico:	0.28
Resultado:	OK

Flexsim Representation of Model 3 - Weibull

Use:

When using a picklist option:	
Distribution	Weibull
Location	0.000000
Scale	132.049695
Shape	1.656194
When using code:	
weibull(0.000000, 132.049695, 1.656194, <stream>)	





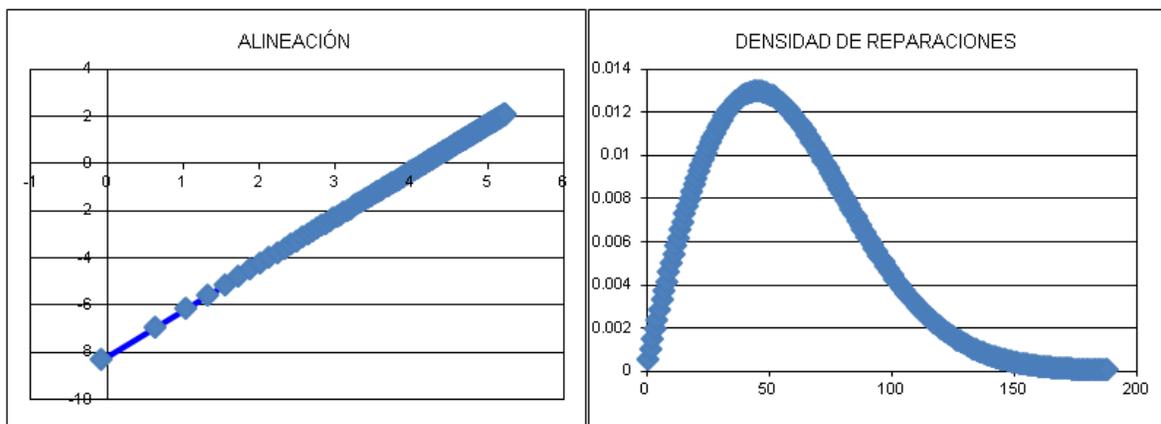
Flexsim Representation of Model 21 - Weibull

Use:

When using a picklist option:

Distribution	Weibull
Location	0.000000
Scale	65.418837
Shape	1.961598

When using code:
`weibull(0.000000, 65.418837, 1.961598, <stream>)`



MTBF	118.05
MTTR	58.00
DISP	67.05%

F. Modelo Documental codificación por Objetos

Source:

```

treenode current = ownerobject(c);
/**popup:PeriodicRates*/
/**Hourly Rates Repeating Weekly*/
// To use custom types not supported by the GUI, remove the popup:PeriodicRates tag
above

// HOUR_BY_WEEK: a table with 24 rows and 7 columns. Each cell is one hour.
//           the first column represents Monday, and the algorithm will use the
//           model start time to figure out where in the table to start
#define HOUR_BY_WEEK 1

// HOUR_BY_N_DAYS: a table with 24 rows and a customizable number of columns.
//           the first column represents Day 1 of the simulation (not necessarily
//           Monday). The algorithm uses the model's start time of day to figure
//           out what row to start on. It will always start the simulation at column 1
#define HOUR_BY_N_DAYS 2

// HOUR_BY_N_COLS: Not implemented in the GUI yet (as of v7.0). This is a table with
custom
//           # rows and custom # columns. Each cell in the table represents one hour.
//           Row 1 Column 1 represents Hour 1 Day 1 of the simulation (not Monday at
//           12 am). The repeat time is nRows * nCols * oneHour.
#define HOUR_BY_N_COLS 3

// DAY_BY_N_WEEKS: a table with 7 rows and customizable columns. Each cell
//           represents one day. Row 1-7 represent Monday - Sunday respectively.
//           The algorithm uses the model start time to figure out where in the
//           table it should start at.
#define DAY_BY_N_WEEKS 4

// DAY_BY_N_COLS: Not implemented in the GUI yet (as of v7.0). This is a table
//           with customizable # rows and # columns. Each cell is one day. Row 1
Column 1
//           represents Day 1, Week 1 of the simulation (not Monday of Week 1). The
//           algorithm will thus start on row 1 column 1.
#define DAY_BY_N_COLS 5

// CUSTOM: Not implemented in the GUI yet (as of v7.0). This table allows full
customization
//           of the rate table. Here you first define how much time (in seconds) one
cell in the
//           table represents. Then you define the start column and row, i.e. the
//           column and row that is associated with time of the simulation. The fill
in the table
//           with the correct values, and there you go.
#define CUSTOM 6

int periodType = /**\nPeriod Type:
****tag:periodType****/HOUR_BY_WEEK/**list:HOUR_BY_WEEK~HOUR_BY_N_DAYS~HOUR_BY_N_CO
LS~DAY_BY_N_WEEKS~CUSTOM*/;
/**\nZero-Rate Cross Option: ****tag:zeroCrossOption****No*/
double oneDay = days(1);

```

```

double oneHour = hours(1);
double oneMinute = minutes(1);
double oneSecond = oneMinute / 60;

// Figure out
double periodInSeconds;
switch (periodType) {
    case HOUR_BY_WEEK:
    case HOUR_BY_N_DAYS:
    case HOUR_BY_N_COLS:
        periodInSeconds = 3600; break;
    case DAY_BY_N_WEEKS:
    case DAY_BY_N_COLS:
        periodInSeconds = 86400; break;
    default: periodInSeconds = /**\nPeriod Seconds (For Custom Only):
    ****tag:customPeriod****/3600/**/;
}
// periodTime: the amount of model time represented by one cell in the table
double periodTime = periodInSeconds / getmodelunitnum(TIME_MULTIPLE);

// get when the start day, hour, minute and second of the model are
int startDay = getmodelunitnum(START_DAYOFWEEK);
int startHour = getmodelunitnum(START_HOUR);
int startMinute = getmodelunitnum(START_MINUTE);
int startSecond = getmodelunitnum(START_SECOND);

// If you choose Locally_Defined_Table, then it will use a sub-node of the code
node named PeriodicRates.
// otherwise it will look at a table you define.
treenode Locally_Defined_Table = 0;
treenode table = /**\nTable:
****tag:table****/Locally_Defined_Table/**list:Locally_Defined_Table~reftable(
"GlobalTableName")*/;
if (table == 0)
    table = node("PeriodicRates", c);

// get various stuff I'll need later on
int periodsPerColumn = gettablerows(table);
int numCols = gettablecols(table);
if (numCols == 0 || periodsPerColumn == 0)
    return 1000000000000;

int totalNumPeriods = periodsPerColumn * numCols;
double repeatTime = periodTime * totalNumPeriods;
int startRow = 0;
int startCol = 0;
double timeIntoStartCell = 0;

// figure out the row and column associated with simulation time 0
switch (periodType) {
    case HOUR_BY_N_COLS:
    case HOUR_BY_WEEK:
    case HOUR_BY_N_DAYS:
        startCol = (periodType == HOUR_BY_WEEK ? startDay : 1);
        startRow = (periodType == HOUR_BY_N_COLS ? 1 : startHour + 1);
        timeIntoStartCell = startMinute * oneMinute + startSecond *
oneSecond;
        break;
    case DAY_BY_N_COLS:
    case DAY_BY_N_WEEKS:
        startCol = 1;
        startRow = (periodType == DAY_BY_N_COLS ? 1 : startDay);

```

```

timeIntoStartCell += startHour * oneHour + startMinute * oneMinute + startSecond
* oneSecond;
    break;
    case CUSTOM:
        startCol = /**\nStart Col (For Custom
Only)*/****tag:customStartCol****/1/**/;
        startRow = /**\nRow Col (For Custom
Only)*/****tag:customStartCol****/1/**/;
        break;
}
// now figure out how many cells into the model I am right at this moment
// as I jump columns and rows, I'm going to decrement cellsIntoModel
// to keep track of how far I still need to "jump"
double cellsIntoModel = (time() + timeIntoStartCell) / periodTime;
// set columnOffset to the number of columns to offset from the start column
int columnOffset = max(0, cellsIntoModel) / periodsPerColumn;
int col = startCol + columnOffset;
cellsIntoModel -= columnOffset * periodsPerColumn;

// set rowOffset to the number of rows to offset from the start row
int rowOffset = max(0, cellsIntoModel);
int row = startRow + rowOffset;
// now, if my row overran the total number of rows then I need to
// pare it back and increment my column
if (row > periodsPerColumn) {
    int overflow = (row - 1) / periodsPerColumn;
    col += overflow;
    row -= overflow * periodsPerColumn;
}
cellsIntoModel -= rowOffset;

// now if my columns overran the total columns in the table, then I
// need to pare it back by "looping around" on the table.
if (col > numCols) {
    int overflow = (col - 1) / numCols;
    col -= overflow * numCols;
}

// set a bunch of initial values
// Now I can set timeIntoCurPeriod, which is the time into the current period
cell
// associated with the current time
double timeIntoCurPeriod = cellsIntoModel * periodTime;
// percTimeLeft: the percentage (0-1) of a "part" or "inter-arrival" that I still
have
// to fill out
double percTimeLeft = 1;
// totalTime: the accumulated time so far
double totalTime = 0;
double curTimeInterval = 0;
// other stuff set by the user
int randStream = /**\nRandom Stream: */****tag:stream****/0/**/;
int numZeroRatePeriods = /**\nMax Zero-Rate Periods:
****tag:maxZeroPeriods****/0/**/;
int numZeroRatePeriodsLeft = numZeroRatePeriods;
double timeIntervalOverflow = 0;
double lastNonZeroTotalTime = -1;
double deviationRatio = /**\nDeviation: */****tag:deviation****/0.20/**/;
while (1) {
    // get the mean number of parts for the current period
    double meanPartsPerPeriod = gettablenum(table, row, col);
    // if it's non-zero, then evaluate
    if (meanPartsPerPeriod > 0) {

```

```

figure out the mean time interval base on that rate.
    double meanTimeInterval = periodTime / meanPartsPerPeriod;
    // get a user-defined deviation
    double deviation = deviationRatio * meanTimeInterval;
    // and get the actual time interval that I should use
    curTimeInterval = percTimeLeft * /**\nTime Interval:
****tag:timeInterval****/exponential(0, meanTimeInterval, randStream)/**/;
    // timeIntervalOverflow tells me if that time interval will take me past
the end of the current time period
    timeIntervalOverflow = curTimeInterval + timeIntoCurPeriod - periodTime;
    // if I'm still within the current time period
    if (timeIntervalOverflow <= 0.00001 * meanPartsPerPeriod) {
        // then return my accumulated time plus the current time interval
        return totalTime + curTimeInterval;
    } else {
        // if I go past the current time period then increment my total
time
        totalTime += periodTime - timeIntoCurPeriod;
        // decrement the percentage of a "part" that I still have left to
do
        percTimeLeft *= (timeIntervalOverflow / curTimeInterval);
        // and save off a value for zero-rate usage later
        lastNonZeroTotalTime = totalTime;
    }
} else {
    // if the current cell is a zero-rate cell, then decrement the number of
    // consecutive zero-rate cells I have left
    numZeroRatePeriodsLeft--;
    // if I'm at 0, then I need to break out of the loop and return
    if (numZeroRatePeriodsLeft == 0 && lastNonZeroTotalTime != -1) {
        #define TRUNCATE_TIME 1
        #define EXTEND_PREVIOUS_RATE 2
        int zeroRateRule = /**\nZero-Rate Rule:
****tag:zeroCrossRule****/EXTEND_PREVIOUS_RATE/**/;
        switch (zeroRateRule) {
            // If I'm supposed to truncate to the end of the last cell,
then return lastNonZeroTotalTime
            case TRUNCATE_TIME: return lastNonZeroTotalTime;
            // if I'm supposed to extend the last non-zero rate into
this time, then return
            // the lastNonZeroTotalTime plus the overflow time
            case EXTEND_PREVIOUS_RATE: return lastNonZeroTotalTime +
timeIntervalOverflow;
        }
    } else // This is infinite loop avoidance, in case the user put 0's in
all the cells
        if (numZeroRatePeriodsLeft < -2 * totalNumPeriods) // break out of
an infinite loop
            return 1000000000000; // if everything's zero, then return a
really big number
        else // if I'm just doing the regular rule or I've still got zero-rate
cells to go
            // then increment my totalTime by the time left in my current
period.
            totalTime += periodTime - timeIntoCurPeriod;
        }
    // and increment values to the next cell in the table
    row++;
    timeIntoCurPeriod = 0;
    if (row > periodsPerColumn) {
        row = 1;
        col++;
    }
}

```

```

if (col > numCols)
    col = 1;
}
}

Labels:

treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

{ /******* PickOption Start *****\\
/**popUp:ItemTypePercentage:valuestr=ItemType*/
/**Set Item Type by Percentage*/
int stream = /**\nStream number: /**tag:stream/**/0/**/;
double randomnum = uniform(0.0, 100.0, stream);
double total = 0.0;
int foundmatch = 0;
/**tagex:data*/
total += /**\nPercent: /**/4.4/**/;
if (!foundmatch && randomnum <= total) {
    setitemtype(item, /** ItemType: /**/1/**/);
    foundmatch = 1;
}
total += /**\nPercent: /**/15.2/**/;
if (!foundmatch && randomnum <= total) {
    setitemtype(item, /** ItemType: /**/2/**/);
    foundmatch = 1;
}
total += /**\nPercent: /**/77.1/**/;
if (!foundmatch && randomnum <= total) {
    setitemtype(item, /** ItemType: /**/3/**/);
    foundmatch = 1;
}
total += /**\nPercent: /**/3.3/**/;
if (!foundmatch && randomnum <= total) {
    setitemtype(item, /** ItemType: /**/4/**/);
    foundmatch = 1;
}/***/
} /******* PickOption End *****\\

/**Custom Code*/
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

if (getitemtype(item) == 1) {
    int value = inc(label(current, "C_100"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

if (getitemtype(item) == 2) {
    int value = inc(label(current, "C_40"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

if (getitemtype(item) == 3) {
    int value = inc(label(current, "C_33"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

```

```

if (getitemtype(item) == 4) {
    int value = inc(label(current, "C_20"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

{ //***** PickOption Start *****\\
/**popup:IncrementValue*/
/**Increment Value*/
treenode thenode = /** \nNode: */ /**tag:node*//**/label(current,
"Total_Entrada")/**/;
double value = /** \nIncrement By: */ /**tag:value*//**/1/**/;
inc(thenode,value);
} //***** PickOption End *****\\

```

Almacén_1:

```

treenode item = parnode(1);
treenode current = ownerobject(c);
/**Random Available Port*/
/**If no ports are available, send to the first port to become available.*/

double arraysize = nrop(current);
doublearray openports = makearray(arraysize);
int nrofportsopen = 0;

for ( int index = 1; index <= arraysize; index++) {
    if (opavailable(current,index)) {
        nrofportsopen++;
        openports[ nrofportsopen ] = index;
    }
}

double returnvalue;
if (nrofportsopen > 0) {
    int returnindex = duniform(1, nrofportsopen);
    returnvalue = openports[returnindex];
} else
    returnvalue = 0;

return returnvalue;

```

Basc_1:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
*//**tag:ValueFunc*//**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;

```

```

case /**\nCase: */**/3/**/: return /** Time: */**/loglogistic(0.0, 1.459876,
11.210569, 0)/**/;
    case /**\nCase: */**/4/**/: return /** Time: */**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**//**/ /**/
    default: return /** \nDefault Time: */**tag:default**//0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\nCase 1: return 10;*/
return 0;

```

Basc_2:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
*/**tag:ValueFunc**//getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: */**/1/**/: return /** Time: */**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: */**/2/**/: return /** Time: */**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: */**/3/**/: return /** Time: */**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: */**/4/**/: return /** Time: */**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**//**/ /**/
    default: return /** \nDefault Time: */**tag:default**//0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\nCase 1: return 10;*/
return 0;

```

Basc_3:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
*/**tag:ValueFunc**//getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: */**/1/**/: return /** Time: */**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: */**/2/**/: return /** Time: */**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: */**/3/**/: return /** Time: */**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: */**/4/**/: return /** Time: */**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**//**/ /**/
    default: return /** \nDefault Time: */**tag:default**//0/**/;
}

```

```

Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_4:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \ncase Function:
*//**tag:ValueFunc*//**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\ncase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\ncase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\ncase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\ncase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**//**/ /**/
    default: return /** \ndefault Time: *//**tag:default**//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_5:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \ncase Function:
*//**tag:ValueFunc*//**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\ncase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\ncase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\ncase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\ncase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**//**/ /**/
    default: return /** \ndefault Time: *//**tag:default**//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_6:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
****tag:ValueFunc****/getitemtype(item)****/;
/** \nCases:\n*/
switch (case_val) {
****tagex:data*/
    case /**\nCase: ****/1****/: return /** Time: ****/weibull(0.428216,
0.363567, 2, 0)****/;
    case /**\nCase: ****/2****/: return /** Time: ****/loglogistic(0.463905,
0.774510, 6.181834, 0)****/;
    case /**\nCase: ****/3****/: return /** Time: ****/loglogistic(0.0,
1.459876, 11.210569, 0)****/;
    case /**\nCase: ****/4****/: return /** Time: ****/invertedweibull(0.0,
3.577660, 5.402759, 0)****/****/****/ ****/
    default: return /** \nDefault Time: ****tag:default****/0****/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_7:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
****tag:ValueFunc****/getitemtype(item)****/;
/** \nCases:\n*/
switch (case_val) {
****tagex:data*/
    case /**\nCase: ****/1****/: return /** Time: ****/weibull(0.428216,
0.363567, 2, 0)****/;
    case /**\nCase: ****/2****/: return /** Time: ****/loglogistic(0.463905,
0.774510, 6.181834, 0)****/;
    case /**\nCase: ****/3****/: return /** Time: ****/loglogistic(0.0,
1.459876, 11.210569, 0)****/;
    case /**\nCase: ****/4****/: return /** Time: ****/invertedweibull(0.0,
3.577660, 5.402759, 0)****/****/****/ ****/
    default: return /** \nDefault Time: ****tag:default****/0****/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_8:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_9:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_10:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_11:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_12:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_13:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
/**tag:ValueFunc/**/getitemtype(item)/**/;
/** \nCases:\n*/
switch (case_val) {
/**tagex:data*/
    case /**\nCase: *//**/1/**/: return /** Time: *//**/weibull(0.428216,
0.363567, 2, 0)/**/;
    case /**\nCase: *//**/2/**/: return /** Time: *//**/loglogistic(0.463905,
0.774510, 6.181834, 0)/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/loglogistic(0.0,
1.459876, 11.210569, 0)/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/invertedweibull(0.0,
3.577660, 5.402759, 0)/**/;/**/**/ /**/
    default: return /** \nDefault Time: *//**tag:default*//**/0/**/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_14:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
****tag:ValueFunc****/getitemtype(item)****/;
/** \nCases:\n*/
switch (case_val) {
****tagex:data*/
    case /**\nCase: ****/1****/: return /** Time: ****/weibull(0.428216,
0.363567, 2, 0)****/;
    case /**\nCase: ****/2****/: return /** Time: ****/loglogistic(0.463905,
0.774510, 6.181834, 0)****/;
    case /**\nCase: ****/3****/: return /** Time: ****/loglogistic(0.0,
1.459876, 11.210569, 0)****/;
    case /**\nCase: ****/4****/: return /** Time: ****/invertedweibull(0.0,
3.577660, 5.402759, 0)****/****/****/ ****/
    default: return /** \nDefault Time: ****tag:default****/0****/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_15:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
****tag:ValueFunc****/getitemtype(item)****/;
/** \nCases:\n*/
switch (case_val) {
****tagex:data*/
    case /**\nCase: ****/1****/: return /** Time: ****/weibull(0.428216,
0.363567, 2, 0)****/;
    case /**\nCase: ****/2****/: return /** Time: ****/loglogistic(0.463905,
0.774510, 6.181834, 0)****/;
    case /**\nCase: ****/3****/: return /** Time: ****/loglogistic(0.0,
1.459876, 11.210569, 0)****/;
    case /**\nCase: ****/4****/: return /** Time: ****/invertedweibull(0.0,
3.577660, 5.402759, 0)****/****/****/ ****/
    default: return /** \nDefault Time: ****tag:default****/0****/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

Basc_16:

```

treenode current = ownerobject(c);
treenode item = parnode(1);
/**popup:ValuesByCase:hasitem=1:valustr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function:
****tag:ValueFunc****/getitemtype(item)****/;
/** \nCases:\n*/
switch (case_val) {
****tagex:data*/
    case /**\nCase: ****/1****/: return /** Time: ****/weibull(0.428216,
0.363567, 2, 0)****/;
    case /**\nCase: ****/2****/: return /** Time: ****/loglogistic(0.463905,
0.774510, 6.181834, 0)****/;
    case /**\nCase: ****/3****/: return /** Time: ****/loglogistic(0.0,
1.459876, 11.210569, 0)****/;
    case /**\nCase: ****/4****/: return /** Time: ****/invertedweibull(0.0,
3.577660, 5.402759, 0)****/****/****/ ****/
    default: return /** \nDefault Time: ****tag:default****/0****/;
}
/** \nNote:
Case values must be positive integers.
Additional cases may be added as needed.
Use Format:\ncase 1: return 10;*/
return 0;

```

MTBFMTTR_1:

```

treenode current = ownerobject(c);
treenode involved = parnode(1);
/**popup:StatisticalDistribution*/
return
****tag:distribution****/weibull****/****/(****/
****tag:par1****/0****/
****tag:par2****/, 111.851255****/
****tag:par3****/, 1.289139****/
****tag:par4****/, 0****/
****tag:par5****/****/
****)/****/;

treenode current = ownerobject(c);
treenode involved = parnode(1);
/**popup:StatisticalDistribution*/
return
****tag:distribution****/weibull****/****/(****/
****tag:par1****/0****/
****tag:par2****/, 56.3402****/
****tag:par3****/, 1.7368****/
****tag:par4****/, 0****/
****tag:par5****/****/
****)/****/;

```

MTBFMTTR_2:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/11520 /**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_3:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/77.4/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/32.9/**direct*/;
```

MTBFMTTR_4:

```
treenode current = ownerobject(c);
treenode involved = param(1);
return /**/11520/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_5:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/23040/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_6:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/46080/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_7:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/15360/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_8:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
/**popup:StatisticalDistribution*/
return
/**tag:distribution***/weibull/**/**/(/**/
/**tag:par1***/0.0/**/
/**tag:par2***/ , 83.276165/**/
/**tag:par3***/ , 0.992903/**/
/**tag:par4***/ , 0/**/
/**tag:par5***//**/
/**)/**/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
/**popup:StatisticalDistribution*/
return
/**tag:distribution***/exponential/**/**/(/**/
/**tag:par1***/0.0/**/
/**tag:par2***/ , 47.490476/**/
/**tag:par3***/ , 0/**/
/**tag:par4***//**/
/**tag:par5***//**/
/**)/**/;
```

MTBFMTTR_9:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/69120/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_10:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/105.2/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/37.3/**direct*/;
```

MTBFMTTR_11:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/17280/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_12:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/33056/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_13:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/23040/**direct*/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
return /**/30/**direct*/;
```

MTBFMTTR_14:

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
/**/popup:StatisticalDistribution*/
return
/**/tag:distribution/**/weibull/**/****/(/**/
/**/tag:par1/**/****/0.0/**/
/**/tag:par2/**/****/, 123.922686/**/
/**/tag:par3/**/****/, 1.446076/**/
/**/tag:par4/**/****/, 0/**/
/**/tag:par5/**/****/**/
/**/)/**/;
```

```
treenode current = ownerobject(c);
treenode involved = parnode(1);
/**/popup:StatisticalDistribution*/
return
/**/tag:distribution/**/weibull/**/****/(/**/
/**/tag:par1/**/****/0.0/**/
/**/tag:par2/**/****/, 60.113557/**/
/**/tag:par3/**/****/, 1.884766/**/
/**/tag:par4/**/****/, 0/**/
/**/tag:par5/**/****/**/
/**/)/**/;
```



```

treenode timetable = var_s(current, "table");
treenode destobject = node(objname, model());

treenode ts = createemptytasksequence(downobject, priority, preempt);
if (objectexists(destobject)) inserttask(ts, TASKTYPE_TRAVEL, destobject, NULL, -
1);
inserttask(ts, TASKTYPE_DELAY, NULL, NULL, duration, state);
dispatchtasksequence(ts);

return 0;

treenode current = ownerobject(c);
treenode members = getvarnode(current, "members");
treenode downobject = parnode(1);
int state = parval(2);
int tablerow = parval(3);
int stateprofile = parval(4);
/**popup:ResumeObject*/
/**Resume Object*/
/**The resumeobject() command is class dependent.
It basically tells the object it can resume doing
whatever it was doing before stopobject() was called.
It's important to note that a complimentary Resume
Function must be defined based on the Down Function defined.
(matching ids)*/
int id = /** \nResume ID: *//**tag:id*//**/1/**/;
resumeobject(downobject, id, stateprofile);
if (stateprofile != 0)
    resumeobject(downobject, id);
return 0;

treenode activets = parnode(1);
treenode current = ownerobject(c);
/**popup:BreakToSame*/
/**New Tasksequences Only*/
/** Only break to task sequences that have not been started already.*/
/** Object queuing the tasksequences: */
treenode theobject = /**tag:dispatcher*//**/current/**/;
treenode tsqueue = gettasksequencequeue( theobject );

treenode returnts = NULL;
for (int index = 1; index <= content(tsqueue) && ! returnts; index++) {
    treenode curts = rank(tsqueue, index);
    if (gettotalnroftasks(curts) == getnroftasks(curts))
        returnts = curts;
}
return (returnts);

```

Sink:

```
/**Custom Code2*/
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

if (getitemtype(item) == 1) {
    int value = inc(label(current, "C 100"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

if (getitemtype(item) == 2) {
    int value = inc(label(current, "C 40"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

if (getitemtype(item) == 3) {
    int value = inc(label(current, "C 33"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

if (getitemtype(item) == 4) {
    int value = inc(label(current, "C 20"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

{ /******* PickOption Start *****\\
  /**popup:IncrementValue*/
  /**Increment Value*/
  treenode thenode = /** \nNode: */ /**tag:node*//**/label(current,
  "Total_Salida")/**/;
  double value = /** \nIncrement By: */ /**tag:value*//**/1/**/;
  inc(thenode,value);
} /******* PickOption End *****\\
```