



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE MECÁNICA**  
**CARRERA MANTENIMIENTO INDUSTRIAL**

**“DESARROLLO DE HERRAMIENTAS PARA LA GESTIÓN DE  
MANTENIMIENTO DE LAS UNIDADES MÓVILES DE BOMBEO  
(M.T.U) DE LA EMPRESA TEAM S.A.”**

**Trabajo de Integración Curricular**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN MANTENIMIENTO INDUSTRIAL**

**AUTOR:**

**DARÍO JAVIER JÁCOME IZA**

Riobamba – Ecuador

2023



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE MECÁNICA**  
**CARRERA MANTENIMIENTO INDUSTRIAL**

**“DESARROLLO DE HERRAMIENTAS PARA LA GESTIÓN DE  
MANTENIMIENTO DE LAS UNIDADES MÓVILES DE BOMBEO  
(M.T.U) DE LA EMPRESA TEAM S.A.”**

**Trabajo de Integración Curricular**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN MANTENIMIENTO INDUSTRIAL**

**AUTOR: DARÍO JAVIER JÁCOME IZA**

**DIRECTOR: ING. CÉSAR MARCELO GALLEGOS LONDOÑO**

Riobamba – Ecuador

2023

© 2023, Darío Javier Jácome Iza

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, DARÍO JAVIER JÁCOME IZA, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 31 de mayo de 2023



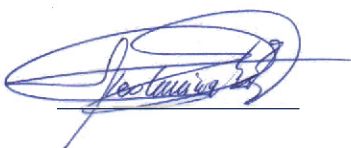


**Darío Javier Jácome Iza**

**050280441-2**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE MECÁNICA**  
**CARRERA MANTENIMIENTO INDUSTRIAL**

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, “**DESARROLLO DE HERRAMIENTAS PARA LA GESTIÓN DE MANTENIMIENTO DE LAS UNIDADES MÓVILES DE BOMBEO (M.T.U) DE LA EMPRESA TEAM S.A.**”, realizado por el señor: **DARÍO JAVIER JÁCOME IZA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	<b>FIRMA</b>	<b>FECHA</b>
Ing. Marco Antonio Ordóñez Viñán <b>PRESIDENTE DEL TRIBUNAL</b>		2023-05-31
Ing. César Marcelo Gallegos Londoño <b>DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR</b>		2023-05-31
Ing. Alex Giovanni Tenicota García <b>ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR</b>		2023-05-31

## **DEDICATORIA**

A mis padres y hermanos.

Darío

## **AGRADECIMIENTO**

Mi agradecimiento eterno los docente y personal de la Escuela Superior Politécnica de Chimborazo, por brindarme su conocimiento y atención como estudiante, en especial a mi director del presente trabajo Ing. César Gallegos y a mi asesor Ing. Alex Tenicota por la paciencia y el tiempo dedicado a lo largo de este trabajo de titulación.

A el personal de mantenimiento de la empresa TEAM S.A. y a todas la personal que me apoyaron en el desarrollo del presente trabajo

Darío

## ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE ILUSTRACIONES.....	xii
ÍNDICE DE ANEXOS .....	xiv
ÍNDICE DE ABREVIATURAS.....	xv
RESUMEN.....	xvi
SUMMARY .....	xvii
INTRODUCCIÓN .....	1
CAPÍTULO I.....	2
1. <b>DIAGNÓSTICO DEL PROBLEMA</b> .....	2
1.1. <b>Antecedentes</b> .....	2
1.2. <b>Planteamiento del problema</b> .....	2
1.3. <b>Justificación y actualidad</b> .....	3
1.3.1. <i>Consideraciones técnicas y tecnológicas</i> .....	4
1.4. <b>Objetivos</b> .....	4
1.4.1. <i>Objetivo general</i> .....	4
1.4.2. <i>Objetivos específicos</i> .....	4
CAPÍTULO II .....	5
2. <b>FUNDAMENTOS TEÓRICOS</b> .....	5
2.1. <b>Definición y funcionamiento, aplicaciones de las MTU</b> .....	5
2.2. <b>Descripción de los sistemas y componentes</b> .....	5
2.2.1. <i>Válvulas de cuatro vías</i> .....	5
2.2.2. <i>Recipiente de acumulación y amortiguamiento</i> .....	6
2.2.3. <i>Desarenador ciclónico</i> .....	6
2.2.4. <i>Separador de acondicionamiento</i> .....	6
2.2.5. <i>Tanques de almacenamiento de fluido motriz</i> .....	6
2.2.6. <i>Las bombas múltiplex</i> .....	6
2.2.7. <i>Motor a diésel en las MTU</i> .....	7
2.2.8. <i>Accesorios y tuberías</i> .....	7
2.3. <b>Gestión de mantenimiento en el sector petrolero</b> .....	8



<b>2.4.</b>	<b>Gestión de mantenimiento en la empresa TEAMS.....</b>	<b>8</b>
<b>2.5.</b>	<b>La norma ISO 14224:2016 .....</b>	<b>8</b>
<b>2.6.</b>	<b>Análisis de confiabilidad .....</b>	<b>8</b>
<b>2.6.1.</b>	<b><i>Tiempo medio entre fallos.....</i></b>	<b>9</b>
<b>2.6.2.</b>	<b>Ajuste de distribución .....</b>	<b>11</b>
<b>2.6.3.</b>	<b><i>Pruebas de bondad de ajuste.....</i></b>	<b>11</b>
<b>2.6.4.</b>	<b><i>Confiabilidad de sistemas en serie, paralelo y mixto .....</i></b>	<b>15</b>
2.6.4.1.	<i>Teoría de probabilidad.....</i>	15
2.6.4.2.	<i>Simulación de Montecarlo.....</i>	16
<b>2.6.5.</b>	<b><i>Modelos de Markov.....</i></b>	<b>16</b>
<b>2.7.</b>	<b>Planificación del mantenimiento. ....</b>	<b>20</b>
<b>2.7.1.</b>	<b><i>Optimización de tiempos para mantenimiento.....</i></b>	<b>20</b>
2.7.1.1.	<i>Método de la ruta crítica .....</i>	20
2.7.1.2.	<i>Técnica de Revisión y Evaluación de Programas (PERT) .....</i>	21
<b>2.7.2.</b>	<b><i>Distribución de horas de mantenimiento: .....</i></b>	<b>22</b>
2.7.2.1.	<i>Control de tareas de mantenimiento.....</i>	22
2.7.2.2.	<i>Número de inspecciones .....</i>	23
<b>2.7.3.</b>	<b><i>Gestión de logística y recursos.....</i></b>	<b>23</b>
2.7.3.1.	<i>Stock de seguridad.....</i>	23
2.7.3.2.	<i>Estimación de repuestos .....</i>	24
<b>2.7.4.</b>	<b><i>Análisis de fallas .....</i></b>	<b>24</b>
2.7.4.1.	<i>Pareto .....</i>	24
2.7.4.2.	<i>Análisis de correlación.....</i>	25
2.7.4.3.	<i>Diagrama causa y efectos.....</i>	26
2.7.4.4.	<i>Árbol de fallas (FTA).....</i>	27
<b>2.7.5.</b>	<b><i>Estimación de costos de mantenimiento: .....</i></b>	<b>28</b>
2.7.5.1.	<i>Costos de mano de obra: .....</i>	28
2.7.5.2.	<i>Costos por acciones de mantenimiento .....</i>	29
2.7.5.3.	<i>Costos de repuestos .....</i>	31
2.7.5.4.	<i>Costos de los equipos .....</i>	32
<b>2.8.</b>	<b>Informes para mantenimiento .....</b>	<b>32</b>
<b>2.8.1.</b>	<b><i>Documentos en Rmarkdown.....</i></b>	<b>32</b>
<b>2.8.2.</b>	<b><i>Librerías para la creación de documentos .....</i></b>	<b>34</b>
<b>2.9.</b>	<b>Acerca de organizadores de tareas: .....</b>	<b>34</b>
<b>CAPÍTULO III.....</b>		<b>35</b>

<b>3.</b>	<b>HERRAMIENTAS PARA LA GESTIÓN DE LAS M.T.U.....</b>	<b>35</b>
1.1.	Análisis de las necesidades de la empresa.....	35
1.2.	Restricciones.....	35
3.1.	Disposición de los módulos.....	35
3.2.	Desarrollo de las herramientas informáticas para la gestión, .....	37
3.2.1.	<i>Plantilla para los diferentes aplicativos.....</i>	38
3.2.2.	Infraestructura utilizada para el despliegue de las herramientas.....	39
3.2.3.	Mantenimiento del software .....	39
3.2.4.	<i>Inconvenientes a futuro .....</i>	40
3.3.	Bombas MTU de la empresa TEAM S.A.....	41
3.3.1.	<i>Herramientas desarrolladas para la gestión de las M.T.U. ....</i>	41
3.3.2.	<i>Modelos utilizados para la gestión de mantenimiento .....</i>	44
3.3.3.	<i>Desarrollo del documento de ayuda .....</i>	44
3.4.	Análisis de fiabilidad, probabilidad de fallo y MTBF .....	45
3.5.	Fórmulas reescritas utilizadas el análisis de confiabilidad .....	48
3.5.1.	<i>Alternativa a compuertas de confiabilidad.....</i>	49
3.5.2.	<i>Tareas de mantenimiento.....</i>	52
3.5.3.	<i>Gestión de repuestos.....</i>	52
3.5.4.	<i>Estimación de costos. ....</i>	54
3.5.5.	<i>Markov tiempo continuo.....</i>	54
3.5.6.	<i>Recolección de la información .....</i>	55
3.5.6.1.	<i>Limitaciones acerca del uso de bases de datos .....</i>	57
3.5.6.2.	<i>Uso de vistas en las bases de datos .....</i>	58
3.5.7.	<i>Gestión de tareas de mantenimiento.....</i>	59
	<b>CAPÍTULO IV .....</b>	<b>62</b>
<b>4.</b>	<b>RESULTADOS .....</b>	<b>62</b>
4.1.	Análisis de los aplicativos para la gestión de mantenimiento de las M.T.U. ....	62
4.1.1.	<i>Debilidad de las herramientas. ....</i>	62
4.1.2.	<i>Beneficios de utilizar la interfaz del aplicativo .....</i>	62
4.1.3.	Uso del aplicativo en la empresa.....	62
4.1.4.	<i>Ajuste de distribuciones .....</i>	63
4.1.5.	<i>Cálculo fiabilidad, tasa entre fallos y MTBF.....</i>	64
4.1.6.	<i>Cronograma anual.....</i>	64

4.1.7.	<i>Markov tiempo continuo</i> .....	65
4.1.8.	<i>Método de la ruta crítica en R</i> .....	66
4.1.9.	<i>Número de inspecciones óptimo</i> .....	66
4.1.10.	<i>Stock de seguridad y factor de seguridad</i> .....	67
4.1.11.	<i>Análisis de correlación aplicado a datos de fallas.</i> .....	68
4.1.12.	<i>Diagrama causa – efecto</i> .....	69
4.1.13.	<i>Árbol de fallas</i> .....	69
4.1.14.	<i>Costos de mantenimiento</i> .....	70
4.1.15.	<i>Calculadora y registros de código</i> .....	71
4.2.	<b>Costos de infraestructura del aplicativo</b> .....	72
4.2.1.	<b>Escalabilidad de la aplicación</b> .....	73
4.2.1.1.	<b>Tiempo de respuesta de la aplicación</b> .....	73
4.3.	<b>Capacitación del manejo de la aplicación al personal de la empresa</b> .....	73
	<b>CONCLUSIONES</b> .....	75
	<b>RECOMENDACIONES</b> .....	76

**GLOSARIO**

**BIBLIOGRAFÍA**

**ANEXOS**

## ÍNDICE DE TABLAS

<b>Tabla 2-1:</b>	Comparación de tiempo para falla y reparaciones.....	9
<b>Tabla 2-2:</b>	Modelos de distribución disponibles en R.....	13
<b>Tabla 2-3:</b>	Diagramas de bloques de confiabilidad.....	15
<b>Tabla 2-4:</b>	Resultado obtenido con la librería <i>KableExtra</i> . ....	33
<b>Tabla 2-5:</b>	Librerías disponibles en R necesarias para el presente proyecto.....	34
<b>Tabla 3-1:</b>	Clasificación de los modelos matemáticos implementados en los módulos. ....	36
<b>Tabla 3-2:</b>	Modos de fallo de las MTUs .....	42
<b>Tabla 3-3:</b>	Librerías para el análisis de probabilidad de fallo, confiabilidad y MTBR.....	46
<b>Tabla 3-4:</b>	Librerías para el análisis de probabilidad de fallo, confiabilidad y MTBR.....	52
<b>Tabla 3-5:</b>	Librerías para la gestión de repuestos .....	53
<b>Tabla 3-6:</b>	Librerías para la estimación de costos.....	54
<b>Tabla 4-1:</b>	Descripción de los inconvenientes en el desarrollo de las herramientas.....	62
<b>Tabla 4-2:</b>	Aplicaciones disponibles .....	63
<b>Tabla 4-3:</b>	Costo de alquiler de la infraestructura del despliegue de la aplicación.....	68
<b>Tabla 4-4:</b>	Costo de alquiler de la infraestructura del despliegue de la aplicación.....	72

## ÍNDICE DE ILUSTRACIONES

<b>Ilustración 1-2:</b>	Unidad móvil de bombeo de la empresa TEAM. S.A.....	5
<b>Ilustración 2-2:</b>	Unidad móvil de bombeo de la empresa TEAM. S.A.....	7
<b>Ilustración 3-2:</b>	Secuencia a considerar para la simulación de Montecarlo.....	17
<b>Ilustración 4-2:</b>	Modelos de Markov de tiempo continuo (fórmulas en el anexo B).....	18
<b>Ilustración 5-2:</b>	Modelos de Markov de tiempo continuo (fórmulas en el anexo B).....	18
<b>Ilustración 6-2:</b>	Modelos de Markov de tiempo continuo (fórmulas en el anexo B).....	19
<b>Ilustración 7-2:</b>	Modelos de Markov de tiempo continuo (fórmulas en el anexo B).....	20
<b>Ilustración 8-2:</b>	Diagrama del método de la ruta crítica, mediante la librería <i>criticalpath</i> ...	21
<b>Ilustración 9-2:</b>	Distribución de horas hombre para mantenimiento por semanas.....	22
<b>Ilustración 10-2:</b>	Diagrama de Pareto aplicado al número de fallas.....	25
<b>Ilustración 11-2:</b>	Correlación de 4 tipos de fallas.....	26
<b>Ilustración 12-2:</b>	Diagrama causa-efecto.....	26
<b>Ilustración 13-2:</b>	Árbol de fallas desarrollado con la librería <i>faultree</i> .....	27
<b>Ilustración 14-2:</b>	Generación de números aleatorios en R.....	32
<b>Ilustración 15-2:</b>	Generación de gráficos con <i>Latex</i> en <i>Rmarkdown</i> .....	33
<b>Ilustración 16-2:</b>	Salida de gráficos con <i>Latex</i> en <i>Rmarkdown</i> .....	33
<b>Ilustración 17-2:</b>	Tabla de ejemplo en R .....	33
<b>Ilustración 1-3:</b>	Diseño de la portada de la aplicación web .....	38
<b>Ilustración 2-3:</b>	Diseño general para cada aplicación .....	38
<b>Ilustración 3-3:</b>	Servicios utilizados para el despliegue de la aplicación web.....	39
<b>Ilustración 4-3:</b>	Actualización del sistema operativo Ubuntu.....	39
<b>Ilustración 5-3:</b>	Asignación de memoria swap primera parte .....	40
<b>Ilustración 6-3:</b>	Asignación de memoria segunda parte.....	40
<b>Ilustración 7-3:</b>	Bombas M.T.U. de la empresa TEAM S.A. ....	41
<b>Ilustración 8-3:</b>	Interfaz del usuario.....	42
<b>Ilustración 9-3:</b>	Manual de ayuda .....	45
<b>Ilustración 10-3:</b>	Extracto análisis de fiabilidad, disponibilidad y MTBF (global.R).....	46
<b>Ilustración 11-3:</b>	Extracto análisis de fiabilidad, disponibilidad y MTBF (server.R) .....	47
<b>Ilustración 12-3:</b>	Extracto análisis de fiabilidad, disponibilidad y mantenibilidad (ui.R).....	48
<b>Ilustración 13-3:</b>	Ejemplo gráfico de las compuertas de confiabilidad .....	49
<b>Ilustración 14-3:</b>	Compuertas para el cálculo de la confiabilidad .....	50
<b>Ilustración 15-3:</b>	Combinación de compuertas de confiabilidad.....	51
<b>Ilustración 16-3:</b>	Compuertas adicionales para el análisis de confiabilidad.....	51

<b>Ilustración 17-3:</b>	Código par la compuerta binomial.....	52
<b>Ilustración 18-3:</b>	Código utilizado para el cálculo de factor de seguridad .....	53
<b>Ilustración 19-3:</b>	Diagrama de Markov de tiempo continuo .....	54
<b>Ilustración 20-3:</b>	Código utilizado para la resolución de diagramas de Markov .....	55
<b>Ilustración 21-3:</b>	Codificación de equipos y sistemas .....	56
<b>Ilustración 22-3:</b>	Lista de herramientas .....	56
<b>Ilustración 23-3:</b>	Lista de herramientas .....	57
<b>Ilustración 24-3:</b>	Lista de materiales y repuestos .....	57
<b>Ilustración 25-3:</b>	Lista de equipos de protección.....	58
<b>Ilustración 26-3:</b>	Galería de la base de datos de repuestos y equipos.....	58
<b>Ilustración 27-3:</b>	Cronograma de mantenimiento.....	59
<b>Ilustración 28-3:</b>	Registro de tareas de mantenimiento .....	60
<b>Ilustración 29-3:</b>	Historial de fallos y rutinas diarias .....	61
<b>Ilustración 1-4:</b>	Ajuste de distribuciones .....	63
<b>Ilustración 2-4:</b>	Cálculo fiabilidad, tasa entre fallos y MTBF .....	64
<b>Ilustración 3-4:</b>	Cronograma de mantenimiento .....	64
<b>Ilustración 4-4:</b>	Diagrama de Markov.....	65
<b>Ilustración 5-4:</b>	Markov tiempo continuo .....	65
<b>Ilustración 6-4:</b>	Método de la ruta crítica.....	66
<b>Ilustración 7-4:</b>	Tiempo óptimo para la inspección .....	66
<b>Ilustración 8-4:</b>	Punto de pedido y stock de seguridad .....	67
<b>Ilustración 9-4:</b>	Factor de seguridad .....	67
<b>Ilustración 10-4:</b>	Análisis de correlación.....	68
<b>Ilustración 11-4:</b>	Diagrama causa - efecto.....	69
<b>Ilustración 12-4:</b>	Árbol de fallas.....	70
<b>Ilustración 13-4:</b>	Costos de mantenimiento Valor presente .....	70
<b>Ilustración 14-4:</b>	Calculadora simple en R.....	71
<b>Ilustración 15-4:</b>	Anotaciones .....	72
<b>Ilustración 16-4:</b>	Tiempo necesario para la apertura de las aplicaciones .....	73

## ÍNDICE DE ANEXOS

**ANEXO A:** DESPLIEGUE DE APLICACIONES EN R

**ANEXO B:** RELACIÓN ENTRE SEIS FUNCIONES QUE DESCRIBEN LA VIDA ÚTIL ESTOCÁSTICA

**ANEXO C:** LISTADO DE CÓDIGOS

## **ÍNDICE DE ABREVIATURAS**

<b>AWS</b>	SERVICIOS WEB DE AMAZON
<b>ISO</b>	ORGANIZACIÓN INTERNACIONAL DE NORMALIZACIÓN
<b>UNE</b>	UNA NORMA ESPAÑOLA
<b>MTU</b>	UNIDADES MÓVILES DE BOMBEO
<b>PERL</b>	TÉCNICA DE GESTIÓN Y REVISIÓN DE PROYECTOS
<b>CPM</b>	MÉTODO DE LA RUTA CRÍTICA




## RESUMEN

A fin de obtener información precisa para la gestión de recursos para el mantenimiento de las unidades móviles de prueba M.T.U. de la empresa TEAM S.A. El presente trabajo de integración tuvo como objetivo desarrollar diferentes herramientas para la gestión de estos equipos. Dado los requerimientos de la empresa se utilizaron los modelos matemáticos: distribuciones de confiabilidad, bloques de confiabilidad, los diagramas de Markov de tiempo continuo, procesos no homogéneos de Poisson, análisis de correlación y el costo anual equivalente (CAE). Estos modelos se aplicaron para estimar la confiabilidad, la disponibilidad y los costos de mantenimiento aplicables en la gestión de las unidades. Los modelos se implementaron en el lenguaje R, se desplegaron como aplicaciones web en la plataforma de cómputo Amazon EC2 y se incorporaron en un gestor de tareas para su uso. Como resultado, las herramientas informáticas desarrolladas permitirán procesar información referente a confiabilidad, tiempos para mantenimiento y costos. En conclusión, se obtuvieron herramientas informáticas para la gestión de las M.T.U que son aplicables a otros equipos, a razón de que los modelos matemáticos aplicados son de uso general. Se recomienda considerar la implementación de modelos adicionales, como el análisis de Weibull, el diagrama de eventos, diagrama Jack-Knife, Crow-AMSAA, para ampliar aún más la capacidad de análisis. Además, se recomienda crear herramientas para el análisis de sistemas reparables.

**Palabras clave:** <GESTIÓN DEL MANTENIMIENTO> <UNIDADES MÓVILES DE PRUEBA> <DISTRIBUCIONES DE CONFIABILIDAD> <ANÁLISIS DE CORRELACIÓN> <APLICACIONES SHINY>.

1313-DBRA-UPT-2023

ESPOCH - DBRAI  
PROCESOS TÉCNICOS Y ANÁLISIS  
BIBLIOGRÁFICO Y DOCUMENTAL  
 28 JUN 2023  
REVISIÓN DE RESUMEN Y BIBLIOGRAFÍA  
Por: P. Ortiz Hora: 16:28

## SUMMARY

To obtain accurate information for resource management for the maintenance of the Mobile Testing Units (MTUs) at TEAM S.A., this integration project aimed to develop various tools for managing this equipment. Given the company's requirements, mathematical models such as reliability distributions, reliability block diagrams, continuous-time Markov diagrams, non-homogeneous Poisson processes, correlation analysis, and Equivalent Annual Cost (EAC) were utilized. These models were applied to estimate reliability, availability, and maintenance costs applicable to the management of the units. The models were implemented in the R programming language, deployed as web applications on the Amazon EC2 computing platform, and incorporated into a task manager for practical use. As a result, advanced computer tools enable information processing regarding reliability, maintenance times, and costs. In conclusion, computer tools for managing MTUs were obtained, which can also apply to other equipment, as the applied mathematical models are of general use. It is recommended to consider implementing additional models, such as Weibull analysis, event diagrams, Jack-Knife diagrams, and Crow-AMSAA, to expand the analysis capability further. Additionally, it is advisable to create tools to analyze repairable systems.

**Keywords:** <MAINTENANCE MANAGEMENT> <MOBILE TESTING UNITS> <LIFE TIME DISTRIBUTIONS> <SYSTEM RELIABILITY> <SHINY APPLICATIONS>



Lic. Sandra Leticia Guijarro Paguay

C.I.: 0603366113

## INTRODUCCIÓN

Según, la norma EN-15341 indica que la gestión de mantenimiento es la implantación, control y mejora de las actividades que determinan y dan cumplimiento a requisitos, objetivos, estrategias y responsabilidades. (UNE-EN 15341 2018)

Según los objetivos de mantenimiento, estrategias y requerimientos de la empresa se pueden seleccionar modelos estadísticos y matemáticos para estimar el tiempo de cambio de componentes, la cantidad de repuestos y costos a fin de ser desarrolladas e implementadas en servicios web.

Los modelos estadísticos permiten modelar el comportamiento de un activo en función del tiempo a fin de obtener indicadores como la confiabilidad a partir de historiales de fallo mediante la distribución exponencial. Por otra parte, PERT (técnica de gestión y revisión de proyectos) y CPM (método de ruta crítica) son herramientas de gestión probabilísticas y deterministas respectivamente que permiten estimar el tiempo de las actividades en el proceso de planificación y ejecución.

La utilización del lenguaje de programación R, facilita el análisis estadístico por motivo de que esta posee un gran número de librerías enfocadas a estadística como lo son *dplyr* para la manipulación de datos, *fitdistrplus* permite realizar ajuste de distribuciones paramétricas, y *shiny* que permite elaborar aplicativos webs para ser desplegados en servicios en la nube como AWS (servicios web de Amazon) utilizando el programa *Shiny server*.

La empresa TEAM S.A. dedicada a la prestación de servicios petroleros utiliza hojas de cálculo para el registro de datos, así como análisis y presentación de informes como parte de la gestión de mantenimiento de la M.T.U. (Unidades móviles de bombeo y prueba).

En el presente trabajo se desarrollarán herramientas para la gestión de las unidades MTU de la empresa TEAMS S.A. mediante una selección de modelos matemáticos para realizar análisis de confiabilidad, número de repuestos y costos, utilizando el lenguaje de programación R y servicios en la nube.

## CAPÍTULO I

### 1. DIAGNÓSTICO DEL PROBLEMA

#### 1.1. Antecedentes

La empresa TEAM S.A. *Team Tools Equipment And Maintenance* presta servicios de alquiler de equipos y repuestos en la industria petrolera. Se encuentra en la ciudad de Quito y cuenta con una sucursal ubicada en la provincia de Orellana.

Uno de los equipos utilizados en la empresa son las unidades móviles de bombeo M.T.U, estas proporcionan la energía necesaria para el desplazamiento y recuperación de la Bomba *Jet Claw*, y proporcionan información para el levantamiento artificial de pozos petroleros. Están conformadas por: un motor, una bomba de alta presión, un separador bifásico/trifásico, dispositivos de medición de gas, líquidos y un panel de control.

Con el propósito de aumentar las formas de gestionar el mantenimiento de las M.T.U se ha dado la necesidad de incorporar funciones extras a su software.

#### 1.2. Planteamiento del problema

El cálculo de la disponibilidad, estimación de tiempos, determinación del número de personal y costo, son necesarios para llevar un control de recursos y tiempos en la planificación de mantenimiento. En el caso de la empresa en cuestión, estas herramientas son necesarias para realizar la gestión de mantenimiento de las M.T.U.

La empresa TEAM S.A. ha presentado dificultades para estimar los recursos para la gestión y los tiempos para mantenimiento, por lo cual, los datos observados y calculados difieren significativamente.

Para la gestión de mantenimiento se debe considerar: la planificación, programación y ejecución de las tareas de mantenimiento, histórico de fallas, indicadores de tiempo medio entre fallas (MTBF) y tiempo medio de reparación (MTTR), recursos financieros asignados al mantenimiento, impacto económico o en producción (consecuencia de falla del equipo) por parada no programada de la planta (sistema) o subsistema. (Viveros et al. 2013, p. 20)

Por otra parte, las aplicaciones de código cerrado no proporcionan el código fuente de los modelos matemáticos implementados, por lo cual se dificulta su aprendizaje y su implementación cuando se trata de costos.

Los lenguajes de código abierto como R nos proporcionan librerías para análisis estadísticos y matemáticos avanzados que pueden utilizarse para desarrollar aplicativos online para la gestión de mantenimiento.

### **1.3. Justificación y actualidad**

La implantación herramientas de software que dé soporte a la gestión del mantenimiento tiene las siguientes ventajas:(Dhillon 2002, p. 46)

- Disponibilidad de información para la toma de decisiones: control de costos, equipos críticos, repuestos, proveedores, personal o sobre cualquier otro parámetro relevante.
- Gestión de recursos, planificación y control de la ejecución del mantenimiento.
- Posibilita la captura y utilización de una gran cantidad de datos y parámetros

El desarrollo de herramientas informáticas complementa las funciones del software para la gestión de mantenimiento utilizado en la empresa TEAM S.A.

A su vez, si se utiliza el lenguaje R para el desarrollo de herramientas para la gestión de mantenimiento su aprendizaje y reducen costos serán bajos al no pagar licenciamiento.

La metodología del proyecto promueve la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica respecto a la detección de fallos, cálculo de la disponibilidad y estimación de costos, por lo tanto, proporcionan un mayor control en las pérdidas en las máquinas a fin de evitar las averías, así, potencian el uso de los recursos de manera eficiente siguiendo el objetivo 5 del plan de desarrollo nacional y enmarcada en las políticas 5.2.

Con el software propuesto y desarrollado indirectamente fortalece la eficiencia de las empresas públicas al proporcionar herramientas para el mantenimiento que disminuyan los fallos en las bombas de uso en la industrial siendo posible su uso en la industria petrolera ecuatoriana que puede fortalecer e incrementar la eficiencia de las empresas al mejorar la disponibilidad de sus equipos industriales dispuesto en las políticas 5.2 y 5.10

### **1.3.1. Consideraciones técnicas y tecnológicas**

Existen una gran variedad de modelos matemáticos para la gestión de mantenimiento, como Markov para la estimación de la disponibilidad (Dhillon 2002, p. 76; Aisha, p. 88), así como modelos para los tiempos inspección (Pham 2003, p. 382), entre otros. Además de contar con lenguajes de programación que nos permiten realizar análisis avanzados, como lo es R, y lenguajes de composición tipográfica para generar reportes como es el caso de Latex.(Antonov et al. 2018, p. 12)

Por otra parte, la reducción de costos debido al avance de la computación en la nube permite a los usuarios disponer de una máquina virtual para instalar diferentes tipos de aplicaciones y así poder acceder a la aplicación de forma online.

Las aplicaciones para la gestión de tareas como *Notion* permiten a los usuarios crear bases de datos y almacenar todo tipo de datos en una misma hoja.

El software *shiny-server* puede ser implementado en una máquina virtual a fin de obtener aplicaciones el lenguaje R que a su vez se pueden incorporar en la aplicación *Notion*.

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

Desarrollar herramientas para la gestión de mantenimiento para las M.T.U. de la empresa TEAM S.A.

### **1.4.2. Objetivos específicos**

Seleccionar modelos matemáticos para la gestión de mantenimiento.

Incorporar los modelos matemáticos en una aplicación web utilizando lenguaje R.

Generar documentos para mantenimiento de las unidades móviles MTU de la empresa TEAM S.A. referente a confiabilidad, disponibilidad, tiempos de mantenimiento, repuestos y costos.

## CAPÍTULO II

### 2. FUNDAMENTOS TEÓRICOS

#### 2.1. Definición y funcionamiento, aplicaciones de las MTU

La Unidad Móvil de evaluación y producción (MTU) es un equipo versátil, que dispone de un motor, bomba de alta presión y un separador bifásico/trifásico. Este equipo impulsa la energía requerida para el funcionamiento de la Bomba Jet en el fondo del pozo con lo cual estimula la producción del pozo. Ver ilustración 1-2

Sus aplicaciones son: evaluación y/o producción de pozos, inyección de agua de formación, transferencia de crudo, inyección de solventes y prueba de tuberías..(«Tripoint - Servicios» 2021)



**Ilustración 1-2:** Unidad móvil de bombeo de la empresa TEAM. S.A.  
Realizado por: Jácome Darío, 2022

#### 2.2. Descripción de los sistemas y componentes

##### 2.2.1. Válvulas de cuatro vías

Estas válvulas permiten revertir el flujo de los fluidos al introducir o retirarla bomba de subsuelo, las mismas que se accionan mediante un brazo de acción positiva tipo carreta. En la conexión de la línea de fluido motriz se encuentra instalada una válvula de bola para altas presiones (5,000

psi), mientras que par las líneas de flujo y de retorno del pozo, están presentes válvulas de bola de presiones medias (1,500 psi)(«Tripoint - Servicios» 2021)

### **2.2.2. Recipiente de acumulación y amortiguamiento**

Este recipiente es un separador vertical que permite almacenar el fluido proveniente del pozo, evita que un flujo excesivo de gas ingrese al filtro ciclónico de arena, y sirve como una cámara de compensación en caso de existir una corriente con sobrepresión proveniente del pozo, evitando así que ésta se transmita al filtro ciclónico y recipiente de acondicionamiento.

### **2.2.3. Desarenador ciclónico.**

El filtro ciclónico para arena retiene los sólidos éste no debe dejar que más de un 5% del total de sólidos en suspensión en el fluido motriz pase hasta la siguiente fase de separación, y que, ninguna de estas partículas debe tener más de 25 micras de diámetro.

### **2.2.4. Separador de acondicionamiento**

Este recipiente no permite que los sólidos en suspensión se asienten hasta el fondo, por lo que el diseño del filtro ciclónico debe ser el adecuado. Éste, es un separador trifásico, y la descarga de los fluidos se controla mediante diferenciales de presión entre el recipiente y la línea de flujo. En este recipiente se definirá el fluido motriz para la bomba de subsuelo, pues posee una salida alta para descargar petróleo y una salida baja para descargar agua; por lo que es necesario mantener un nivel lo suficientemente alto para abastecer a la bomba múltiplex. («Tripoint - Servicios» 2021)

### **2.2.5. Tanques de almacenamiento de fluido motriz**

El volumen de fluido motriz que se inyectará hacia la bomba jet debe ser el adecuado para evitar posteriores problemas operacionales. Los tanques del almacenamiento son utilizados en caso de que el volumen de fluido motriz saliente del separador de acondicionamiento no abastezca directamente el requerimiento del sistema de bombeo.

### **2.2.6. Las bombas múltiplex**

Son las encargadas de enviar el fluido motriz hasta la entrada de la bomba jet en subsuelo, y usualmente, («Tripoint - Servicios» 2021)



### 2.2.7. Motor a diésel en las MTU

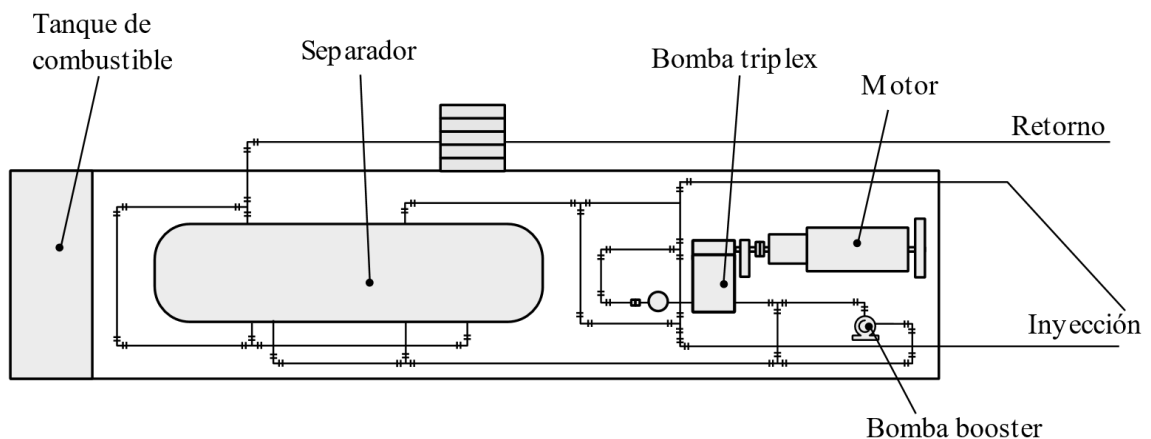
El motor que impulsa la bomba múltiplex puede ser eléctrico, a diésel o a gas. La bomba se encuentra conectada al motor mediante un reductor de velocidad, el cual se compone de un conjunto de engranajes y bandas que permiten disminuir las RPM del motor. Esto, debido a que la acción recíproca de la bomba se vería afectada moviéndose a altas velocidades.

### 2.2.8. Accesorios y tuberías

Líneas de alta presión y baja presión se componen de tuberías que transportan el fluido motriz desde la descarga de las bombas múltiplex hasta el cabezal del pozo. Dichas tuberías soportan presiones de hasta 5.000 psi y se encuentran diseñadas bajo especificaciones de la norma API 5L.

En cambio, las de baja presión son tuberías utilizadas para transportar los fluidos desde el cabezal del pozo hasta los separadores y posteriormente, el fluido motriz hasta los tanques de almacenamiento y bombas. Al igual que las líneas de alta presión, su construcción se encuentra bajo especificaciones de la norma API 5L. («Tripoint - Servicios» 2021)

Por otra parte, las juntas giratorias son conexiones con pivotes de rodamiento de bola que permiten unir las líneas de flujo en diferentes configuraciones de acuerdo con el espacio disponible. Estas juntas son esenciales en la configuración de las líneas de flujo de alta y baja presión.



**Ilustración 2-2:** Unidad móvil de bombeo de la empresa TEAM. S.A.  
Realizado por: Jácome Darío, 2022

### **2.3. Gestión de mantenimiento en el sector petrolero**

La gestión de mantenimiento se define como:

*Todas las actividades de la gestión que determinan los requisitos, los objetivos, las estrategias y las responsabilidades del mantenimiento y la implantación de dichas actividades por medios tales como la planificación del mantenimiento, el control de este y la mejora de las actividades de mantenimiento y las cuestiones económicas. (UNE-EN 15341 2018, p. 7)*

A fin de realizar desarrollar herramientas para la gestión de las MTU, en los siguientes temas se muestran varias formas de calcular la confiabilidad, tasa de fallas, tiempo medio entre fallas, así como la mantenibilidad, tasa de mantenibilidad, el tiempo medio para reparar y disponibilidad.

### **2.4. Gestión de mantenimiento en la empresa TEAMS.**

Se destaca las siguientes características

- El mantenimiento se lleva a cabo en base a un cronograma de mantenimiento, bitácoras y uso de hojas de cálculo.
- La empresa cuenta con dos técnicos en el departamento de mantenimiento y se dispone actualmente de dos M.T.U. es estado operativo.
- Además, se realizan pruebas de ultrasonido de forma externa.
- La empresa cuenta con dos unidades móviles de bombeo.
- El mantenimiento depende del tipo de fluido.

### **2.5. La norma ISO 14224:2016**

La norma ISO 14224:2016 proporciona información para la recopilación de datos de fiabilidad y mantenimiento en un formato estándar para todas las y operaciones de petróleo, as natural e industrial petroquímicas durante el ciclo de vida operativo de los equipos.

### **2.6. Análisis de confiabilidad**

Esta técnica cuantifica confiabilidad y mantenibilidad de un sistema complejo, utiliza información de las fallas de los componentes de los sistemas, a fin de generar distribuciones de probabilidad que representa su funcionamiento a través del tiempo.

**Tabla 2-1:** Comparación de tiempo para falla y reparaciones

Fallas	Reparaciones
<p><math>f(x)</math> La función de densidad <math>f</math> de la variable aleatoria <math>T</math> se define como la probabilidad que tiene un componente de fallar en un instante <math>t</math>.</p>	<p><math>m(x)</math> Considérese la variable aleatoria TTR que representa el tiempo de ejecución de las tareas de mantenimiento, ya sea preventivo o correctivo. Esta variable toma valores en el intervalo</p>
<p><b>Confiabilidad.</b> Probabilidad de que una unidad o componente realice la función para la cual fue diseñado, bajo ciertas condiciones de uso especificadas, por un periodo de tiempo determinado</p> $R(x) = 1 - \int_0^x f(x)dx \quad (1)$	<p><b>Mantenibilidad.</b> Probabilidad de devolver el equipo a condiciones operativas</p> $M(t) = \int_0^t m(x)dx \quad (2)$
<p><b>Tasa de riesgo.</b> Esta función representa la probabilidad instantánea, por unidad de tiempo, que tiene un componente de fallar en un instante <math>t</math>, dado que había funcionado hasta el instante anterior.</p> $h(t) = \frac{f(t)}{1 - F(t)} \quad (3)$	<p><b>Tasa de mantenibilidad.</b> Probabilidad instantánea de devolver un equipo a condiciones operativas.</p> $\gamma(t) = \frac{m(t)}{1 - M(t)} \quad (4)$
<p><b>Tasa entre fallos. (TBF)</b> Representa la distribución de tiempo transcurrido entre una falla y la otra</p> $TTF = \int_0^{\infty} R(x)dt \quad (5)$	<p><b>Tiempo para reparar.</b> Representa la distribución de tiempos que existe entre una reparación y otra</p> $TTR = \int_0^{\infty} M(t)dt \quad (6)$

Fuente: (Ye et al. 2019)

### 2.6.1. Tiempo medio entre fallos

L1, L2, L3 y L4 representan la vida, en años, del componente sujeto a falla. Utilizando los datos aplicables recopilados por una gran empresa petroquímica en la década de 1980, se calcularon los tiempos medios entre fallas y los valores estimados para una bomba mejorada de servicio mediano. Se forma inversa se tiene:

$$1/MTBF = \left[ (1/L_1)^2 + (1/L_2)^2 + (1/L_3)^2 + (1/L_4)^2 \right]^{0.5} \quad (7)$$

Como ejemplo, una bomba ANSI B73.1 de construcción estándar con un MTBF de sello mecánico de 1.2, un MTBF de rodamiento de 3.0, un MTBF de acoplamiento de 4.0 y un valor de MTBF del eje de 15.0, dio como resultado un MTBF total de la bomba de 1.07 años. Al actualizar el sello y los cojinetes, el MTBF total de la bomba se puede mejorar en un 80% a 1,93.

La disponibilidad intrínseca ( $A$ ) se obtiene una vez calculado los tiempos medio para falla y mantenimiento con la siguiente fórmula:

$$A = \frac{\text{MTTF}}{\text{MTTRes} + \text{MTTF}} \quad (8)$$

Donde:

MTTRes: Es el tiempo medio para restaurar, estimado utilizando el tiempo activo de mantenimiento observado en el terreno, MTTF es el tiempo medio transcurrido hasta la falla.

La disponibilidad operativa ( $A_o$ ) se expresa de la siguiente forma:

$$A_o = \frac{\text{MUD}}{\text{MDT} + \text{MUD}} \quad (9)$$

Donde:

MUD: Es el tiempo activo medio, estimado utilizando el tiempo activo real observado en terreno;

MDT: Es el tiempo de parada medio, estimado utilizando el tiempo activo y tiempo de parada real observado en terreno.

Si sólo se conoce una de estas funciones  $f$ ,  $h$ ,  $F$  y  $R$ , se pueden deducir las demás relaciones y equivalencias detalladas en el anexo B. En la tabla 2-2-2 se muestra la función de densidad de probabilidad y la función de distribución acumulada en lenguaje  $R$ .

La disponibilidad instantánea  $A(t)$ , es la probabilidad de un ítem de cumplir una función requerida bajo condiciones determinadas en un estado específico de tiempo suponiendo que todos los recursos externos requeridos están presentes. Operativamente, la función de disponibilidad instantánea dependiente del tiempo de un sistema se sintetiza mediante valores puntuales, por ejemplo:

Para unidades o sistemas en mantenimiento correctivo, la disponibilidad límite o de estado estacionario se calcula como el límite matemático de la función de disponibilidad instantánea en el tiempo a medida que esta última crece hasta el infinito. Para sistemas en mantenimiento periódico, se introduce como indicador de desempeño la disponibilidad, (ISO 14224)

La disponibilidad en mantenimiento correctivo representa la probabilidad de que el sistema esté funcionando en un momento arbitrario de tiempo, después de la transitorio de los procesos de falla y reparación se han estabilizado. Obviamente no está definido para sistemas en mantenimiento periódico, para los cuales el límite no existe, En mantenimiento periódico representa la proporción esperada de tiempo que el sistema está operando en el período considerado de tiempo. (Zio 2013, p. 10).

### **2.6.2. Ajuste de distribución**

El histórico de fallas y mantenimiento correctivo proveen los datos para el cálculo de la disponibilidad.

### **2.6.3. Pruebas de bondad de ajuste**

Son pruebas de hipótesis utilizadas para verificar si los datos observados en una prueba aleatoria se ajustan con algún nivel de significancia de alguna distribución de probabilidad.

Los estadísticos utilizados son: *Kolmogorov-Smirnov*, *Cramer-von Mises*, *Anderson-Darling*, de los cuales un valor cercano a cero indica un mejor ajuste. Por otra parte, los indicadores de información AIC (*Akaike's Information Criterion*), y BIC (*Bayesian Information Criterion*) son muy utilizados, éstos no se basan en la calidad del ajuste más bien en la complejidad como un número.

El estadístico se halla de la resta de la función de distribución empírica y la de referencia obteniendo el máximo valor, para luego y se contrasta con la distribución de prueba.

$$D = \max \{ (F_0 - F) \} \quad (10)$$

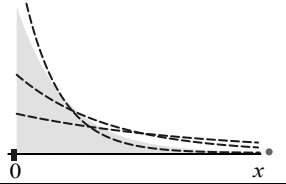
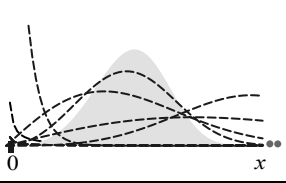
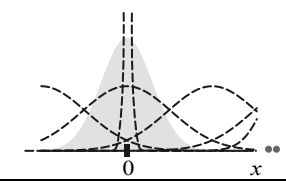
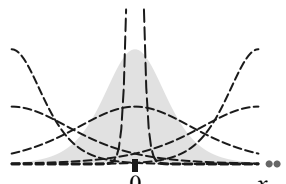
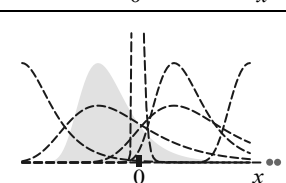
Los estadísticos indicados anteriormente, están disponible en R mediante la librería *fitdistrplus*. En cuanto a las distribuciones éstas se detallan en la tabla 2-2-2.

Dentro de esta variedad de pruebas existentes, la prueba de *Kolmogorov-Smirnov* es una de las más clásicas en el estudio de la normalidad, se basa en el concepto de la función de distribución empírica y sus propiedades como aproximación de la función de distribución teórica cuando se trabaja sobre variables continuas y se conocen todos los parámetros muestrales. Así, esta prueba compara la función de distribución teórica con la empírica y calcula un valor de discrepancia máxima entre ambas distribuciones, proporcionando un valor  $p$ , asociado a la probabilidad de obtener una distribución que discrepe tanto como la observada si verdaderamente se hubiera obtenido una muestra aleatoria, de tamaño  $n$ , de una distribución normal (Chakravarti, Laha, & Roy, 1967).

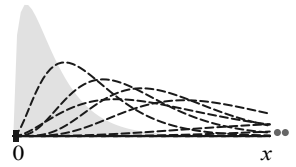
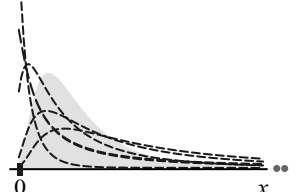
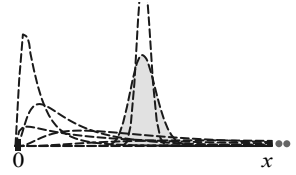
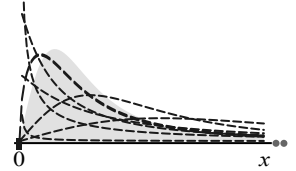
Sin embargo, la prueba mencionada anteriormente cuenta con ciertas limitaciones que restringen su aplicación, entre las que destacan el hecho de que si los parámetros de posición, escala y forma de la distribución se calculan a partir de los datos, la región crítica de la prueba no es válida, por lo que estos deben determinarse mediante simulación. Además la prueba muestra una mayor sensibilidad en el centro de la distribución que en las colas (Thadewald & Buning, 2007). Por otra parte, a estas dos limitaciones hay que añadir su tendencia conservadora, provocando que la hipótesis nula se acepte en un número excesivamente elevado de ocasiones (Shahabuddin, Ibrahim, & Jemain, 2009; Steinskog, et al., 2007).

Finalmente, la prueba de Anderson-Darling supone una modificación del test de Cramervon Mises, que se basa en la diferencia de cuadrados entre las distribuciones pero, en su caso, otorga una mayor relevancia a los datos existentes en las colas de la distribución (Farrel & Rogers-Stewart, 2006).

**Tabla 2-2:** Modelos de distribución disponibles en R

Nombre	Gráfica	Parámetros	Función de densidad de probabilidad	Funciones en r
Exponencial		$\lambda$	$f(x) = \exp^{-\lambda x}$ (11)	<i>pnorm</i> <i>dnorm</i> <i>rnorm</i>
Weibull		$a, \sigma$	$f(x) = \left(\frac{a}{\sigma}\right) \left(\frac{x}{\sigma}\right)^{a-1} \exp\left[-\left(\frac{x}{\sigma}\right)^a\right]$ (12)	<i>pweibull</i> <i>dweibul</i> <i>rweibull</i>
Normal		$\mu, \sigma$	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$ (13)	<i>pnormal</i> <i>dnormal</i> <i>rnormal</i>
Logística		$\mu, s$	$f(x) = \frac{1}{\sigma} \frac{e^{(x-\mu)/\sigma}}{(1 + e^{(x-\mu)/\sigma})^2}$ (14)	<i>plogis</i> <i>dlogis</i> <i>rlogis</i>
Gumbel		$a, b$	$f(x) = \left(\frac{1}{b}\right) \exp\left(-\frac{x-a}{b}\right) \exp\left[-\exp\left(-\frac{x-a}{b}\right)\right]$ (15)	<i>pgumbel</i> <i>dgumbel</i> <i>rgumbel</i>

(continúa)

Nombre	Gráfica	Parámetros	Función de densidad de probabilidad	Funciones en r
Gamma		$\alpha, \beta$	$f(x) = \frac{1}{\sigma^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\sigma} \quad (16)$	<i>pgamma</i> <i>dgamma</i> <i>rgamma</i>
Inversa gaussiana		$\mu, \varphi$	$f(x) = \sqrt{\frac{1}{2\pi\varphi x^3}} \exp\left(-\frac{(x-\mu)^2}{2\mu^2\varphi x}\right) \quad (17)$	<i>pinvgauss</i> <i>dinvgauss</i> <i>rinvgauss</i>
Log-normal		$\mu, \sigma$	$f(x) = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left(-\frac{(\log(x)-\mu)^2}{2\sigma^2}\right) \quad (18)$	<i>plnormal</i> <i>dlnormal</i> <i>rlnormal</i>
Log-logística		$a, s$	$f(x) = \frac{a(x/s)^a}{x[1+(x/s)^a]^2} \quad (19)$	<i>pllogis</i> <i>dlogis</i> <i>rllogis</i>

Fuente: («Life Data Analysis Reference Book - ReliaWiki» 2021; Zio 2007, p. 12)



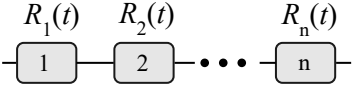
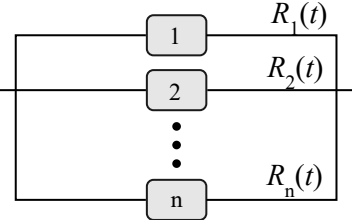
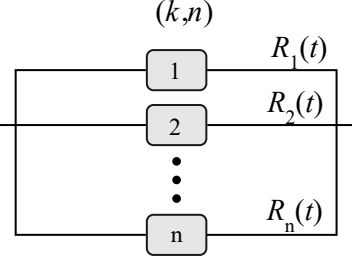
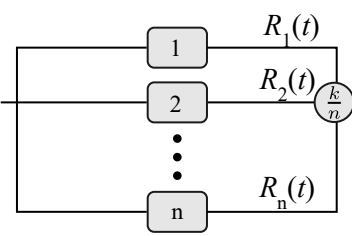
## 2.6.4. Confiabilidad de sistemas en serie, paralelo y mixto

### 2.6.4.1. Teoría de probabilidad

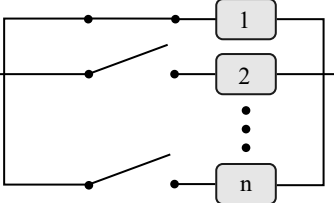
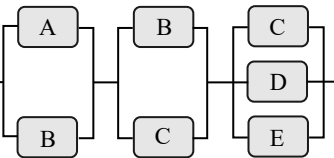
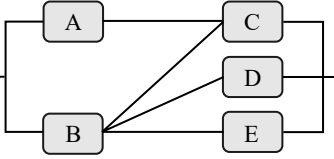
Si dos o más eventos pueden ocurrir simultáneamente, y se conocen sus probabilidades individuales de ocurrir, entonces la probabilidad de eventos simultáneos es el producto de las probabilidades individuales (Zio 2007, p. 76)

De igual forma se presenta en la tabla 3.2 diagramas derivados de la teoría de la probabilidad, en donde se muestra los modelos de confiabilidad y sus respectivas fórmulas.

**Tabla 2-3:** Diagramas de bloques de confiabilidad

Gráfico	Tipos de diagrama de bloques
	<p><b>Serie.</b> Todas la unidades deben operar para que el sistema funcione. No es necesario que las unidades estén conectadas físicamente en serie.</p> $R = \prod_{i=1}^n p_i \quad (20)$
	<p><b>Redundancia.</b> Una o varias unidades pueden fallar y el sistema aún puede funcionar. El sistema falla cuando todos los elementos han fallado.</p> $R = 1 - \prod_{i=1}^n (1 - p_i) \quad (21)$
	<p><b>Sistema k en n.</b> Almenos k elementos deben estar operativos para que el sistema sea funcional. Cada elemento tiene la misma probabilidad de estar operativo.</p> $R(k, n) = \sum_{i=k}^n \binom{n}{i} p^i \cdot q^{n-i} \quad (22)$
	<p><b>Sistema k en n.</b> Almenos k elementos deben estar operativos para que el sistema sea funcional. Cada elemento tiene la misma probabilidad de estar operativo.</p> $R(k, n) = \sum_{i=k}^n (-1)^{i-k} \binom{i-1}{k-1} \sum_{j_1 < j_2 < \dots < j_n} \prod_{l=1}^i p_{j_l} \quad (23)$

(continúa)

Gráfico	Tipos de diagrama de bloques
	<p><b>Reserva (Stand-by).</b> Consta de una unidad activa y una o más unidades en espera. Cuando las unidades activas y en espera son iguales y constantes, la conmutación y la detección son perfectas la ecuación de la confiabilidad es:</p> $R_s(t) = e^{-\lambda t} \left( 1 + \lambda t + \frac{\lambda^2 t^2}{2!} + \dots + \frac{\lambda^{(n-1)} t^{(n-1)}}{(n-1)!} \right) \quad (24)$
	<p><b>Mixto.</b> Esta estructura que puede agrupar en subestructuras serie y paralelo</p> $R_s = (1 - (1 - p_A)(1 - p_B)) \times (1 - (1 - p_B)(1 - p_C)) \times (1 - (1 - p_C)(1 - p_D)(1 - p_E)) \quad (25)$
	<p><b>Complejo.</b> No se pueden agrupar en subestructuras, serie y paralelo. Para el esquema a la derecha la fórmula resulta:</p> $R_s = p_A p_C + p_B p_C + p_B p_D + p_B p_E - p_A p_B p_C - p_B p_D p_E - p_B p_C p_D - p_B p_C p_E \quad (26)$

Fuente: (IEEE-1413.1 2002, p. 67-72; Jin et al. 2018, p. 47-55; Zio 2007, p. 83-84)

#### 2.6.4.2. Simulación de Montecarlo

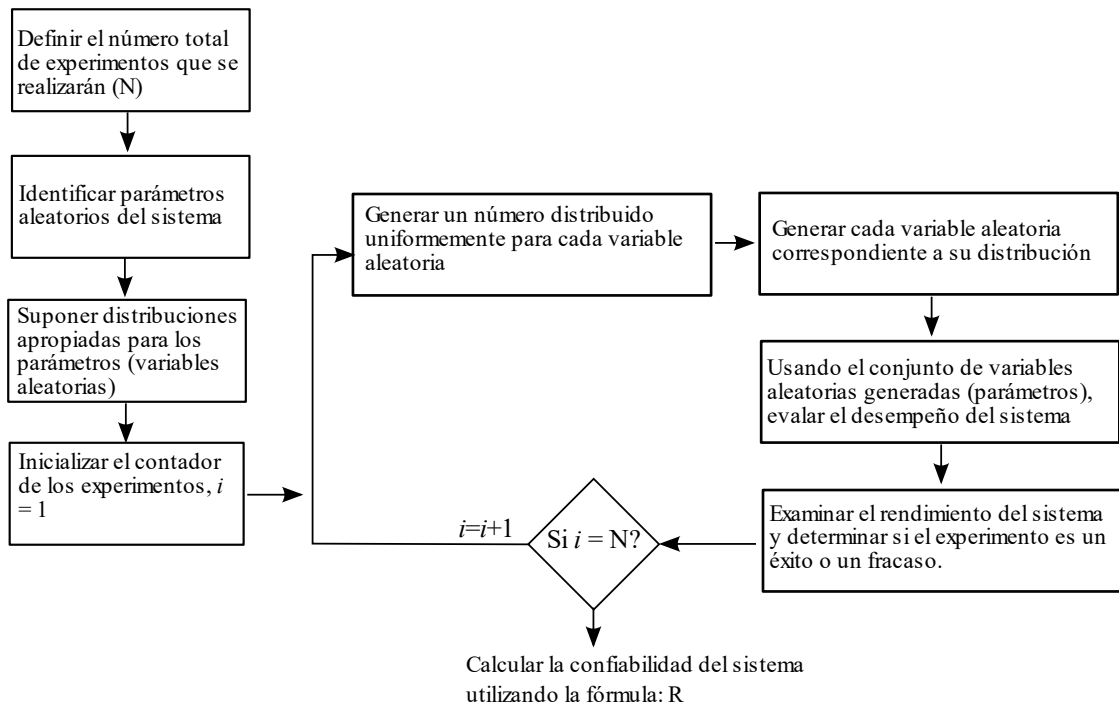
Conociendo la distribución de fallas de los componentes y una ecuación de validación que puede ser la confiabilidad se realiza los cálculos generando números aleatorios. En la ilustración 3-2 se muestra el procedimiento.

$$R = \frac{\text{Número de experimentos exitosos}}{\text{Número total de experimentos}} \quad (27)$$

#### 2.6.5. Modelos de Markov

Algunas familias de procesos aleatorios como Poisson, procesos gaussianos, cadenas de Markov, tienen propiedades que permiten estudiar y comprender cada uno de ellos.

Las cadenas de Markov tiene una propiedad que se denomina “sin memoria”, explicando, la probabilidad de que ocurra un evento depende solamente del evento anterior.

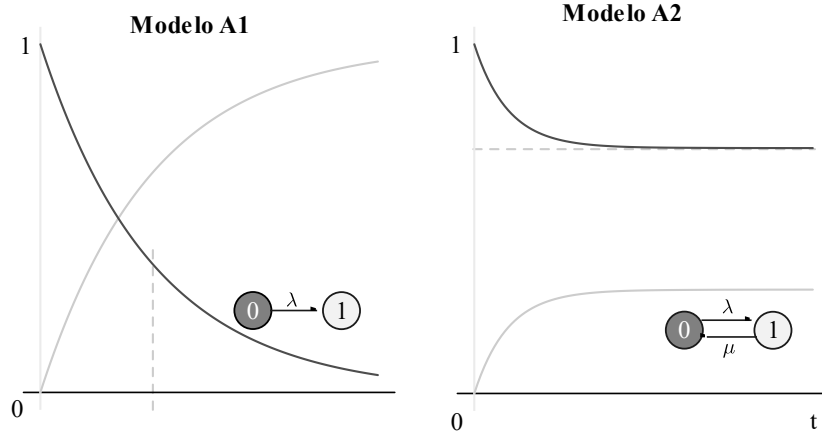


**Ilustración 3-2:** Secuencia a considerar para la simulación de Montecarlo

Realizado por: Jácome Darío, 2022

El uso tradicional del análisis de Markov, con respecto a sistemas redundantes, es fundamentalmente defectuoso en el sentido de que los tiempos de reparación o inactividad no son un proceso aleatorio y dependen de eventos anteriores al inmediatamente anterior a la falla en cuestión. La reparación de la segunda falla depende del progreso de la primera en el sentido de que un solo equipo de reparación continuará ocupándose de esa primera falla. Sin embargo, hasta el punto en que ocurra la segunda falla, la suposición de Markov, dado que hay múltiples equipos de reparación, se mantiene bien. (Zio 2007, p. 337)

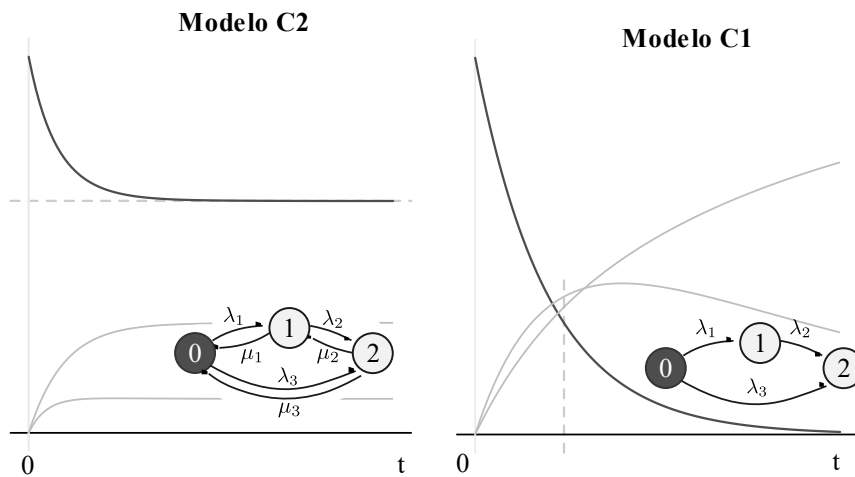
Los modelos A1, B1, C1, D1, de la ilustración 3-2, muestra un sistema de dos a cuatro estados, en donde consta de una tasa de falla “lambda”, y en los modelos A2, B2, C2, D2 se añade una tasa de reparación “mu” en donde la disponibilidad disminuye hasta un valor en específico. Cada modelo posee diferentes curvas la cual representan la probabilidad de que el proceso este un estado 1, 2, 3 o 4, a lo largo del tiempo, sumado la probabilidad de estar curvas resulta la unidad. El valor horizontal representa la disponibilidad del sistema.



**Ilustración 4-2:** Modelos de Markov de tiempo continuo (fórmulas en el anexo B)  
 Realizado por: Jácome Darío, 2022

$$\begin{aligned} \frac{dP_0(t)}{dt} + \lambda P_0(t) &= \mu P_1(t) \\ \frac{dP_1(t)}{dt} + \mu P_1(t) &= \lambda P_0(t) \end{aligned} \quad (28)$$

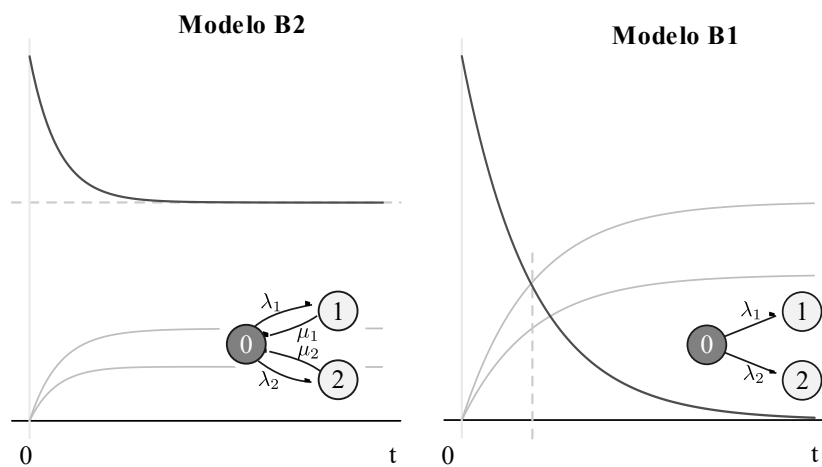
Modelo A2 muestra un sistema con dos unidades idénticas, cada una con tasa de falla y tasa de reparación (recíproca del tiempo medio de inactividad). El sistema puede estar en cada uno de los tres estados posibles.



**Ilustración 5-2:** Modelos de Markov de tiempo continuo (fórmulas en el anexo B)  
 Realizado por: Jácome Darío, 2022

$$\begin{aligned}
\frac{dP_0(t)}{dt} + (\lambda_1 + \lambda_2)P_0(t) &= \mu_1 P_1(t) + \mu_2 P_2(t) \\
\frac{dP_1(t)}{dt} + (\mu_1 + \lambda_3)P_1(t) &= \mu_3 P_2(t) + \lambda_1 P_0(t) \\
\frac{dP_2(t)}{dt} + (\mu_1 + \mu_3)P_2(t) &= \lambda_3 P_1(t) + \lambda_2 P_0(t)
\end{aligned}
\tag{29}$$

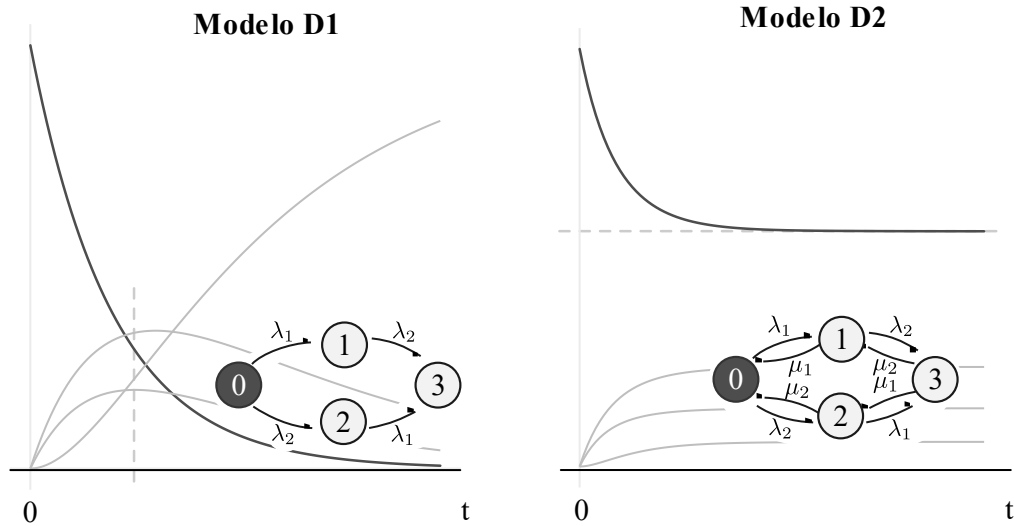
Los modelos de A pueden ocuparse para modelar un tipo de falla, el B se puede ocupar para dos tipos de falla, C un tipo de falla con degradación, y el D para un sistema redundantes de dos elementos. En donde los estados serían: 0) los dos componentes fallan, 1) falla solo el primer equipo, 2) falla en segundo equipo, 3) los dos elementos han fallado.



**Ilustración 6-2:** Modelos de Markov de tiempo continuo (fórmulas en el anexo B)  
Realizado por: Jácome Iza, Darío, 2022

$$\begin{aligned}
\frac{dP_0(t)}{dt} + (\lambda_1 + \lambda_2)P_0(t) &= \mu_1 P_2(t) + \mu_2 P_1(t) \\
\frac{dP_1(t)}{dt} + \mu_1 P_1(t) &= \lambda_2 P_0(t) \\
\frac{dP_2(t)}{dt} + \mu_1 P_2(t) &= \lambda_1 P_0(t)
\end{aligned}
\tag{30}$$

No solamente se pueden modelar fallas con este sistema, también se puede evaluar los errores cometidos por el personal de mantenimiento o el trabajador, por ejemplo: 0) el trabajador no comente ningún error, 1) el trabajador cometió un error crítico, 2) el trabajador comete un error no crítico. Ver Anexo C



**Ilustración 7-2:** Modelos de Markov de tiempo continuo (fórmulas en el anexo B)  
**Realizado por:** Jácome Darío, 2022

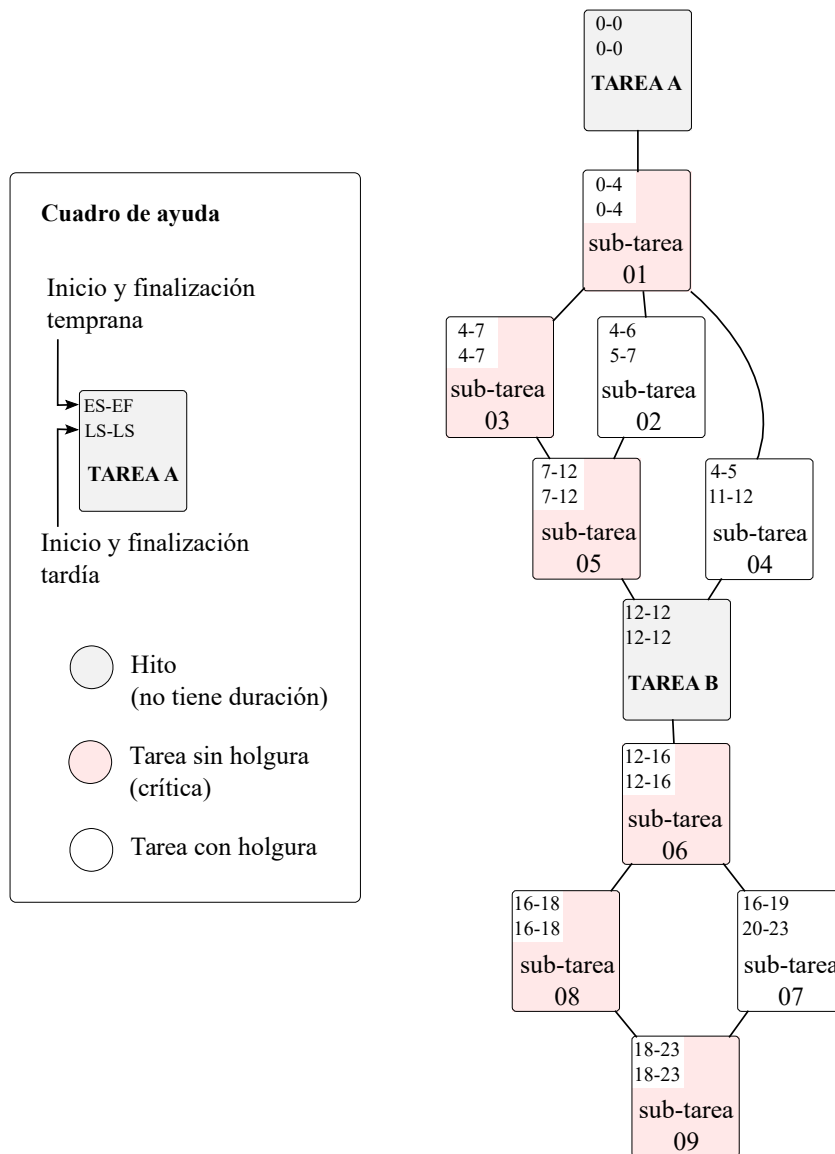
$$\begin{aligned}
 \frac{dP_0(t)}{dt} + (\lambda_1 + \lambda_2)P_0(t) &= \mu_2P_1(t) + \mu_1P_3(t) + \mu_3P_2(t) \\
 \frac{dP_1(t)}{dt} + (\lambda_1 + \lambda_2 + \mu_2)P_1(t) &= \lambda_1P_0(t) + \lambda_3P_3(t) \\
 \frac{dP_2(t)}{dt} + \mu_1P_2(t) &= \lambda_1P_0(t) + \mu_2P_1(t) \\
 \frac{dP_3(t)}{dt} + (\mu_2 + \mu_1)P_3(t) &= \lambda_1P_1(t)
 \end{aligned}
 \tag{31}$$

## 2.7. Planificación del mantenimiento.

### 2.7.1. Optimización de tiempos para mantenimiento

#### 2.7.1.1. Método de la ruta crítica

Es una herramienta utilizada para calcular el tiempo o fecha máxima de un proyecto desde el inicio de sus actividades hasta su culminación. En el siguiente ejemplo se muestra el uso de la librería *criticalpath* del lenguaje R para calcular fechas tempranas de inicio, finalización y holguras de tiempo. Este método permite identificar la secuencia de actividades más larga y sus limitaciones en el tiempo.



**Ilustración 8-2:** Diagrama del método de la ruta crítica, mediante la librería *criticalpath*  
 Realizado por: Jácome Darío, 2022

### 2.7.1.2. Técnica de Revisión y Evaluación de Programas (PERT)

Estima el tiempo de la actividad esperada basada en tiempos probabilísticos, tiene como argumentos de entrada el tiempo mínimo, máximo y tiempo más probable. La ecuación se basa en la distribución Beta. (Dhillon 2002, p. 43)

$$T_a = \frac{OT + 4MT + PT}{6} \quad (32)$$

$T_a$ : Tiempo de duración esperado de la actividad,

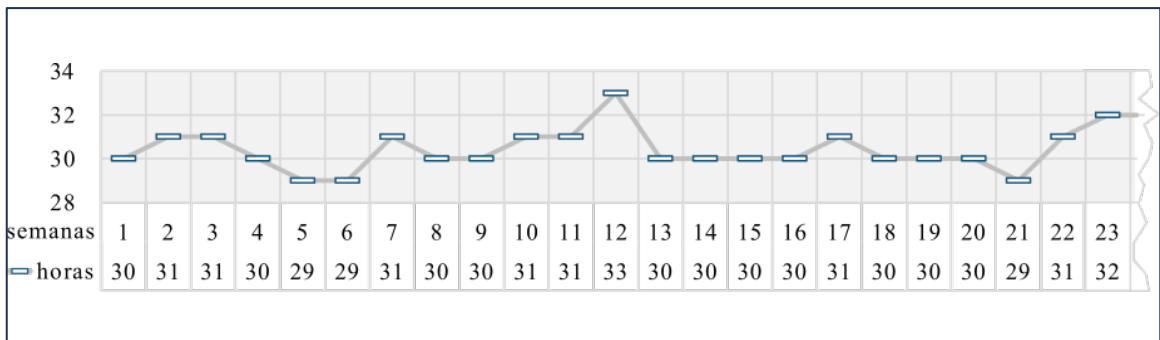
OT: Tiempo optimista o mínimo que requerirá una actividad para completarse,

- PT: Pesimista o tiempo máximo que requerirá una actividad para completarse,  
 MT: Tiempo más probable que una actividad requiera para completarse. Este es el tiempo utilizado para actividades de CPM.

**2.7.2. Distribución de horas de mantenimiento:**

El tiempo que se ocupa para realizar cada una de las tareas de mantenimiento preventivo se distribuye a lo largo de cada semana, usualmente se ocupa 52 semanas para el año. Dado que cada tarea suma su tiempo al tiempo total de cada semana, es necesario que cada semana se ocupe un número de horas similar. Una de las formas para optimizar estos tiempos es ordenarlo según el número de tareas y frecuencias, y luego recorrer las tareas de una semana a la siguiente.

Por lo general el tiempo utilizado para mantenimiento a la semana es del 60%



**Ilustración 9-2:** Distribución de horas hombre para mantenimiento por semanas  
 Realizado por: Jácome Darío, 2022

**2.7.2.1. Control de tareas de mantenimiento**

Los gráficos de control de calidad se pueden utilizar para establecer límites en los defectos relacionados con el mantenimiento. Límites de control con la distribución son:  $UCL = C \pm 3\sigma$ . En cambio la media de la distribución de Poisson y la distribución estándar se expresa por:

$$C = \frac{MD}{N} \quad \sigma = \sqrt{C} \tag{33}$$

- C: La media de la distribución de Poisson  
 MD: Número total de defectos relacionados con el mantenimiento,  
 N: Número total de equipos.



### 2.7.2.2. *Número de inspecciones*

A mayor número de inspecciones en tiempo por inactividad por este hecho aumenta, en cambio el tiempo por inactividad se reduciría, bajo estos supuestos el tiempo total de inspección TDT se representa por:

$$\text{TDT} = y T_i + \frac{c T_b}{y} \quad (34)$$

Se halla en mínimo de la ecuación 27 para obtener el valor óptimo de la inspección

$$y^* = \left( c \frac{T_b}{T_i} \right)^{1/2} \quad (35)$$

donde:

- $T_i$  : Tiempo de inactividad por inspección,
- $T_b$  : Tiempo de inactividad por avería o falla,
- $c$  : Una constante asociada con una instalación en particular,
- $y$  : Número de inspecciones por unidad de tiempo.

### 2.7.3. *Gestión de logística y recursos*

#### 2.7.3.1. *Stock de seguridad*

Dado la demanda de un repuesto a utilizar en el tiempo, a un nivel de confiabilidad dado, se puede calcular las existencias en bodega con la siguiente fórmula: (Dhillon 2002, p. 119).

$$\text{ST} = z\sigma \quad (36)$$

- $\text{ST}$  : Existencias de seguridad necesarias para obtener el nivel de servicio deseado,
- $\sigma$  : Desviación estándar de la demanda durante el tiempo de entrega,
- $z$  : Número de desviaciones estándar del valor medio requerido para obtener el nivel de servicio deseado.

$$\text{ODP} = \mu + z\sigma \quad (37)$$

ODP: Punto de pedido,

$\mu$ : Demanda media durante el tiempo de entrega.

### 2.7.3.2. Estimación de repuestos

La probabilidad de tener una pieza cuando se requiera según el número de repuestos en bodega y el número de repuestos a utilizar se representa por la siguiente ecuación:

$$P_s = \sum_{j=0}^n \left\{ \left[ (-1) \ln e^{-q\lambda t} \right]^j e^{-q\lambda t} \right\} / j! \quad (38)$$

$P_s$ : Probabilidad de tener una pieza de repuesto disponible cuando sea necesario,

$\lambda$ : Tasa de falla de la pieza,

$t$ : Tiempo,

$n$ : Número de repuestos almacenados,

$q$ : Número de piezas de un tipo específico utilizadas.

La ecuación se basa en una distribución exponencial, e indica el nivel de protección deseado en la estimación de repuestos (Dhillon 2002, p. 121)

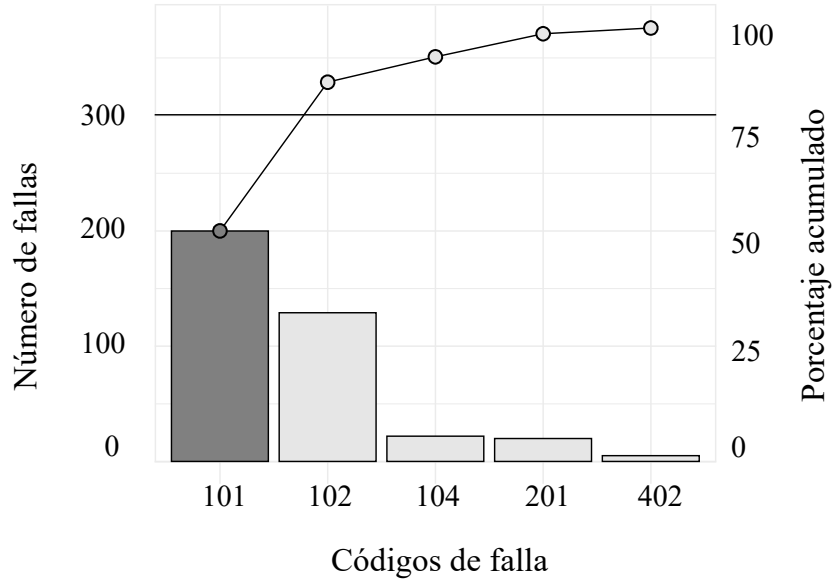
### 2.7.4. Análisis de fallas

#### 2.7.4.1. Pareto

También conocida como la ley 80-20, Pareto, basándose en conocimiento empírico y estudios sobre la riqueza afirma que el 80% de las consecuencias son debido al 20% de las causas. Estos porcentajes suelen variar, por ejemplo, se puede establecer una distribución de 70-20-10. (Gano 2011, p. 41)

El diagrama se puede utilizar para obtener cuales son las fallas más relevantes en un sistema, también para determinar cuáles acciones de mantenimiento son las que se desarrollan con una mayor cantidad de tiempo.

Para construir este diagrama es necesario ordenar el campo a evaluar y añadir un gráfico de barras y en el otro eje se grafica el porcentaje acumulado del campo por un gráfico de líneas. Se termina trazando una línea al 80% del porcentaje y seleccionando los campos que estén dentro del porcentaje.



**Ilustración 10-2:** Diagrama de Pareto aplicado al número de fallas  
Realizado por: Jácome, Darío, 2021

#### 2.7.4.2. Análisis de correlación

Sirve para conocer si existen relaciones entre fallas según el coeficiente de correlación. Este análisis sirve para otros datos que no sean fallas.

Los datos de entrada son los tiempos entre falla para la siguiente descripción:

$x$ : Tiempo entre cada modo de falla 1

$y$ : Tiempo entre cada modo de falla 2

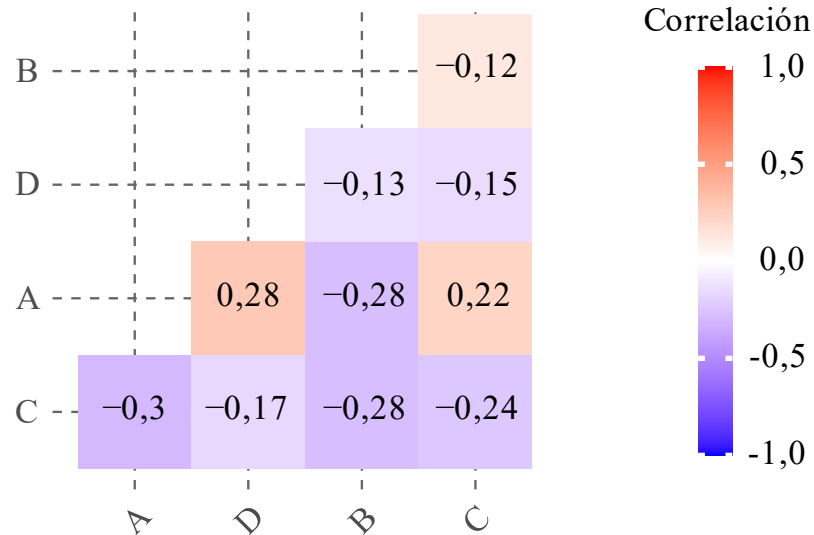
$\hat{\rho}$ : 1, correlación positiva,

$\hat{\rho}$ : -1, correlación negativa

$\hat{\rho}$ : 0, no existe correlación.

$$\hat{\rho} = \frac{\sum (x_i - \bar{x})^2 (y_i - \bar{y})^2}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} \quad (39)$$

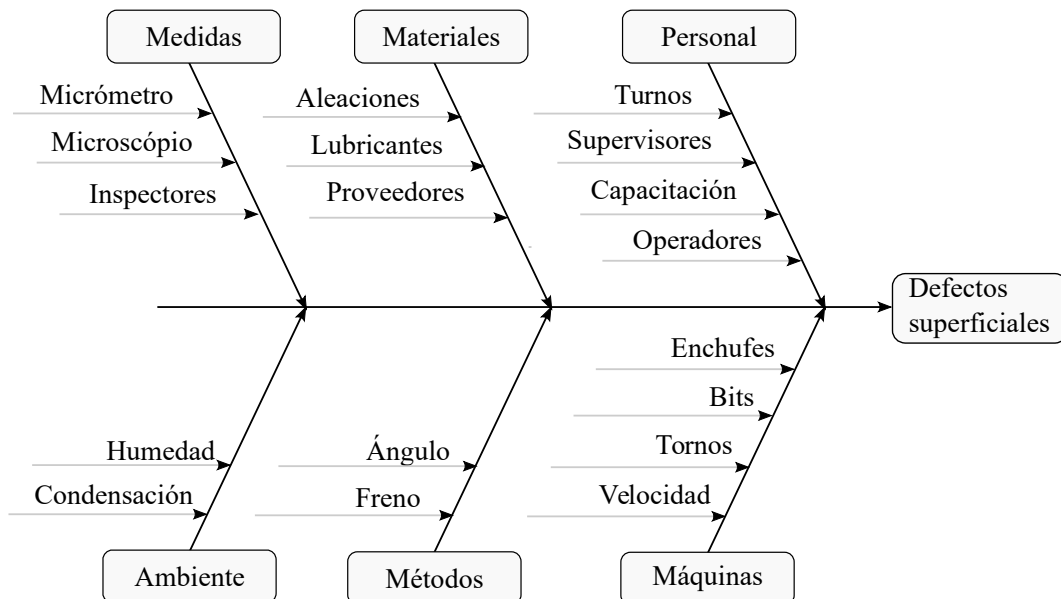
Una correlación positiva implica que los elementos se relacionan de manera proporcional directa, en cambio una correlación negativa lo hace de manera inversa. Es decir, si una variable aumenta la otra disminuye.



**Ilustración 11-2:** Correlación de 4 tipos de fallas  
Realizado por: Jácome Darío, 2022

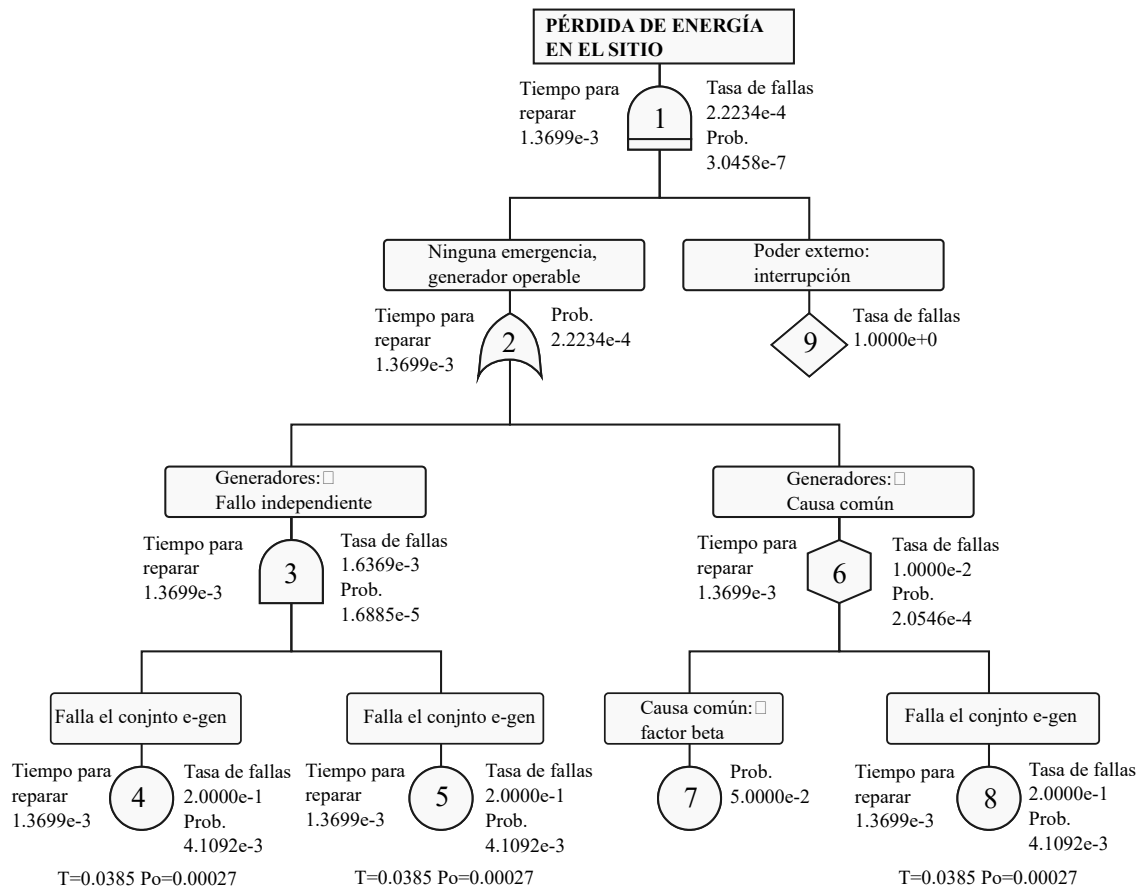
### 2.7.4.3. Diagrama causa y efectos

Los métodos de análisis causal son procesos para identificar las causas principales dentro de un problema, éstas a su vez se pueden desarrollar por medio del método de las 5M como se muestra en la ilustración 8.2.



**Ilustración 12-2:** Diagrama causa-efecto  
Realizado por: Jácome Darío, 2022

### 2.7.4.4. Árbol de fallas (FTA)



**Ilustración 13-2:** Árbol de fallas desarrollado con la librería *faultree*  
 Realizado por: Jácome Darío, 2022

El diagrama muestra jerárquicamente todas las fallas de un sistema, subsistema, ensamblaje y módulo hasta un componente individual. A menudo, el FTA comienza con una consideración de los efectos de falla del sistema, denominados "eventos principales", y el análisis procede determinando cómo estos pueden ser causados por fallas individuales o combinadas o eventos de subsistemas o componentes de nivel inferior. Los símbolos lógicos estándar se utilizan para construir un diagrama FTA y para capturar las conexiones lógicas entre eventos o componentes individuales. (Jin et al. 2018, p. 120)

En R se puede utilizar la librería *FaultTree* para obtener la probabilidad de falla y el tiempo medio entre fallas de un componente individual de nivel superior. En donde se pueden modelar eventos de componentes latentes, equipos reparables y no reparables.

### 2.7.5. Estimación de costos de mantenimiento:

Los siguientes modelos presentan alternativas para estimar el costo de acciones correctivas, material, equipos, y el valor presente del equipo, del cual se describe la notación utilizada. (Dhillon 2002, p. 154-163)

#### 2.7.5.1. Costos de mano de obra:

El costo total de la mano de obra viene dado por:

$$C_{em} = LR(1 + BR)TAH \quad (40)$$

- $C_{em}$ : Costo por empleado  
LR: Tarifa de mano de obra por hora,  
BR: Proporción de beneficios,  
TAH: Número total de horas anuales.

Coste laboral de mantenimiento

$$TLC = C_{em} N \quad (41)$$

$$C_{sh} = \frac{PD \times FBF}{PHP + EPHP} \quad (42)$$

- $C_{sh}$ : Costo por hora estándar producida,  
PD: Dólares de nómina por período,  
PHP: Horas planificadas producidas,  
EPHP: Horas planificadas equivalentes producidas,  
FBF: Factor de beneficio adicional.

Costo de reparación de mano de obra

$$RC = \alpha(1 - RSF)C_{um} \quad (43)$$

- RC: Costo de reparación respecto a mano de obra,

- $\alpha$  : Número de unidades reparables que fallan durante la vida útil del sistema,  
 RSF: Factor de contracción reparable por pérdida, daño, etc. Sus valores están tabulados en la y varían de 0 a 0,1375.  
 $C_{um}$ : Costo unitario de reparación respecto a la mano de obra,

El número de unidades reparables que fallan durante la vida útil del sistema viene dado por:

$$\alpha = \lambda n_r L_s H_o \quad (44)$$

- $\alpha$  : Número de unidades reparables que fallan durante la vida útil del sistema,  
 $\lambda$ : Tasa de falla constante del elemento,  
 $n_r$ : Número total de elementos reparables,  
 $L_s$ : Vida del sistema; (años),  
 $H_o$ : Horas de funcionamiento por año.

Costo unitario de reparación con respecto a la mano de obra se expresa por:

$$C_{um} = MCH \times AMHPR \times MUF \quad (45)$$

- $C_{um}$ : Costo unitario de reparación respecto a la mano de obra,  
 MCH: Costo de mano de obra por hora, incluidos los gastos generales,  
 AMHPR: Número medio de horas-hombre por acción de reparación,  
 MUF: Factor de uso de mano de obra y sus valores se tabulados. Los valores tabulados varían de 3 a 1,04.

#### 2.7.5.2. Costos por acciones de mantenimiento

El modelo de estimación de costos de construcción según Dillon:

$$BC = NCF \times ICPCF \quad (46)$$

- NCF: Número de pies cúbicos necesarios para la infraestructura de mantenimiento  
 ICPCF: Costo inicial por pie cúbico,

La ecuación 47 permite una tasa de inflación del 1% por mes del costo de compra mientras el artículo está en inventario, y una asignación del 10% para toda la vida útil del artículo para tomar en consideración el deterioro, deterioro, obsolescencia y robo.

$$PDC = w \times PP \times (1 \times L_s \times L_u) - SM \quad (47)$$

- $w$ : Peso u otra unidad de cantidad de material utilizado,  
 $PP$ : Precio de compra de material por unidad; más específicamente, el precio de entrega,  
 $L_s$ : Pérdidas por desechos, esqueletos, virutas, etc.,  
 $L_u$ : Pérdidas generadas por el stock no utilizado devuelto al inventario que está demasiado pequeño para uso futuro,  
 $SM$ : Precio unitario del material recuperado.

$$WI = PDC \times (1 + i)^n \quad (48)$$

- $i$ : Tasa de interés para un período específico,  
 $n$ : Número de períodos de interés,

$$IC = \frac{FSC \times B}{K \times R} \quad (49)$$

- $FSC$ : Costo anual de espacio de piso por pie cuadrado,  
 $B$ : Tamaño del contenedor expresado en pies cuadrados,  
 $R$ : Recíproco de los años que el artículo normalmente pasa en inventario,  
 $K$ : Número promedio de artículos almacenados en el contenedor.

Coste laboral de mantenimiento.

$$TCS = PDC + IC + (WI - PDC) + (0.01 \times T \times PDC) + (0.1 \times PDC) \quad (50)$$

- $TCS$ : Costo total de existencias o almacenamiento en el momento de la reparación,  
 $PDC$ : Valor presente del costo del artículo de inventario, incluido el precio de compra y el costo de entrega.  
 $WI$ : Valor del artículo de inventario después de  $n$  períodos,  
 $IC$ : Costo de inventario por artículo,  
 $T$ : Tiempo, expresado en meses, el artículo de stock está en inventario.



Coste laboral de mantenimiento correctivo.

$$CM_{al} = \frac{SOH \times LCH \times MTTR}{MTBF} \quad (51)$$

$CM_{al}$ : Costo anual de mano de obra de mantenimiento correctivo,

SOH: Horario de funcionamiento anual programado.

LCH: Costo de mano de obra de mantenimiento correctivo por hora,

Instalación de producción modelo de estimación de costos de tiempo inactivo

$$PFDC = S_i + C_r + RC_r + RL + C_c + C_p \quad (52)$$

$S_i$ : Salario del operador inactivo,

$C_c$ : Costos tangibles e intangibles asociados con la insatisfacción del cliente, pérdida de buena voluntad, costos ocultos, etc.,

$RC_r$ : Costo de reemplazo del producto arruinado,

RL: Pérdida de ingresos, menos costos recuperables como materiales,

$C_r$ : Costo de alquiler de la unidad de reemplazo (si corresponde),

$C_p$ : Costos asociados con multas tardías, etc.

### 2.7.5.3. Costos de repuestos

Estimación del costo del material de mantenimiento.

$$TCS = WI + IC + \frac{(T \times PDC) + (10 \times PDC)}{100} \quad (53)$$

Modelo de estimación de costos del equipo de mantenimiento

$$CME = RDC + \theta(UPC) \quad (54)$$

CME: Costo del equipo de mantenimiento,

RDC: Costo de investigación y desarrollo asociado con el equipo de mantenimiento,

$\theta$ : Número total de equipos de mantenimiento,

UPC: Costo de adquisición de la unidad de equipo de mantenimiento.

#### 2.7.5.4. Costos de los equipos

Cálculo del valor presente

$$PV = \frac{AM}{(1+i)^k} \quad (55)$$

PV: Valor actual,

AM: Cantidad única,

$k$ : Número de períodos de interés o conversión (normalmente tomados como años),

$i$ : Tasa de interés por período,

Ciclo de propiedad del equipo estimación de costos de mantenimiento.

$$PV = MC \left[ \frac{1 - (1+i)^{-k}}{i} \right] \quad (56)$$

MC: Costo de mantenimiento que se produce al final de cada período de conversión (de intereses).

## 2.8. Informes para mantenimiento

### 2.8.1. Documentos en Rmarkdown

Los archivos markdown está bsado en markdown el cual agrega formato a documentos con texto plano, además se integra codigó R para permitir al usuario generar documentos científicos. A continuación se muestr un ejemplo de su utilización.

Para la generación de 10 números aleatorios de distribución normal con media de 7 y desviación estandar de 3.2 en Lenguaje R utilizamo los siguientes código:

```
rnorm(10, mean=7, sd=3.2) #R
## [1] 9.6771939 0.7334424 10.1791317 7.8341956 7.7098050 9.6911053
## [7] 10.1463457 4.5098319 7.6074404 11.2297619
```

**Ilustración 14-2:** Generación de números aleatorios en R

Realizado por: Jácome, Darío, 2022

Para realizar diferentes tipos de diagramas en *Rmarkdown* se puede utilizar librerías de latex como lo es *tikz* no obstante, los gráficos serán de tipo estático. Se recomienda crear una función que cree todo el gráfico y asu vez los valores que se desee modificar esté como entradas de la función. En el siguiente ejemplo muestra el proceso descrito.

```
latex_fun <-
"\begin{equation}
y = \sin(x) + \frac{1}{\cos(x)}
\end{equation}
" %>% stringr::str_replace_all("x", toString(round(rnorm(1), 2)))
```

**Ilustración 15-2:** Generación de gráficos con *Latex* en *Rmarkdown*

Realizado por: Jácome, Darío, 2022

$$y = \sin(1,28) + \frac{1}{\cos(1,28)}$$

**Ilustración 16-2:** Salida de gráficos con *Latex* en *Rmarkdown*

Realizado por: Jácome, Darío, 2022

También es posible utilizar librerías de R que compongan tablas utilizando latex. En la tabla 5-2 se muestra un ejemplo de tabla con la librería *KableExtra*.

```
dt <- mtcars[1:2, 1:6]
kbl(dt, caption = "Tabla de demostración", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
```

**Ilustración 17-2:** Tabla de ejemplo en R

Realizado por: Jácome Darío, 2022

La tabla 2-4 presenta una salida del código de la ilustración 17-2 en donde se muestran los siguientes campos: mpg (millas por galón), cyl (cilindros), disp(desplazamiento) hp (potencia), drat(relación del eje posterior), wt (peso)

**Tabla 2-4:** Resultado obtenido con la librería *KableExtra*.

	<i>mpg</i>	<i>cyl</i>	<i>disp</i>	<i>hp</i>	<i>drat</i>	<i>wt</i>
Mazda RX4	21	6	160	110	3.9	2.620
Mazda RX4 <i>wag</i>	21	6	160	110	3.9	2.875

Realizado por: Jácome Darío, 2022

## 2.8.2. Librerías para la creación de documentos

La tabla 2-5 muestra algunas librerías de R disponibles en la versión 4.1.1 (2021-08-10) llamada "Kick Things", éstas se pueden instalar en ©Rstudio 2021.09.0. Note que en la tabla las librerías de Python están marcadas con asterisco (\*), Para más detalles ver el anexo B

## 2.9. Acerca de organizadores de tareas:

En la web existen diferentes aplicaciones como lo es *Notion* el cuál es es un organizador de tareas multiplataforma que permite que la información escrita esté disponible fuera de línea.

Con esta aplicación se pueden realizar diferentes plantillas que constan de texto, bases de datos, guardar imágenes, archivos e incorporar páginas web externas.

**Tabla 2-5:** Librerías disponibles en R necesarias para el presente proyecto.

Utilidad	Librerías
Creación de la aplicación y documento	<i>shiny, shinydashboard, shinydashboardPlus, flexdashboard, Rmarkdown, knitr, pander</i>
Ampliación de funciones de R	<i>forcats, purrr, magrittr, tidyverse</i>
Creación de tablas	<i>DT, data.table, formattable</i>
Manipulación de datos de tablas	<i>Dplyr</i>
Composición de gráficos y elemento visuales	<i>ggplot2, ggridges, ggQC, plotly, timevis</i>
Lectura y escritura en formato xlsx	<i>readxl</i>
Operaciones con fechas	<i>lubridate</i>
Manipulación de líneas de texto	<i>stringr</i>
Ajuste de funciones de distribución	<i>MASS, actuar, fitdistrplus</i>
Creación de nomograma	<i>Nomogramer*</i>
Interprete de Python	<i>Reticulate</i>
Cálculo simbólico	<i>SciPy*</i>
Cálculo numérico	<i>NumPy*</i>

Fuente: (Crawley 2007, p. 769)

## CAPÍTULO III

### 3. HERRAMIENTAS PARA LA GESTIÓN DE LAS M.T.U.

#### 1.1. Análisis de las necesidades de la empresa

Los algoritmos a desarrollar para cada herramienta para la gestión dependerán de los requerimientos de la empresa, así como de la gestión misma para las MTU. Esto puede realizarse mediante un sondeo o una encuesta al departamento de mantenimiento.

Una vez seleccionado las necesidades de la empresa, se elegirán modelos matemáticos que den solución a estas necesidades, para luego ampliar las aplicaciones de cada modelo, esto es así, porque un modelo puede ser aplicado en diferentes ámbitos, por ejemplo: el método de Markov puede ser aplicado para calcular la disponibilidad de un componente considerando dos estados, funciona correctamente o no funciona, con tasas de mantenimiento correctivo y tasa de fallas, del mismo modo se puede utilizar este método para calcular.

En la ilustración 1-3 se muestra un diagrama del proceso para el desarrollo de las herramientas informáticas, de donde el proceso inicial con el establecimiento de los requerimientos de la empresa acerca de la gestión, posteriormente se seleccionan los modelos matemáticos, se transcribe a lenguaje r y se incorpora en una máquina virtual en la nube.

#### 1.2. Restricciones

- El software requiere de una conexión a internet
- Funcional en navegadores
- Se puede modificar las bases de datos
- Disponible para varios usuarios

#### 3.1. Disposición de los módulos

Los modelos basados en distribuciones como la distribución de Weibull, permite modelar la probabilidad de fallo, fiabilidad, tasa de fallos y tiempo medio entre fallas con los datos obtenidos de los elementos MTU.

Por otra parte, para modelar la disponibilidad es necesario obtener los datos de los tiempos medios entre fallas y el tiempo medio para reparar, también se puede utilizar el método de Markov para calcular la disponibilidad, pero disponiendo las tasas de falla y de reparación.

En la tabla 3-1 se puede observar los modelos seleccionados para su aplicación en la gestión de mantenimiento.

**Tabla 3-1:** Clasificación de los modelos matemáticos implementados en los módulos.

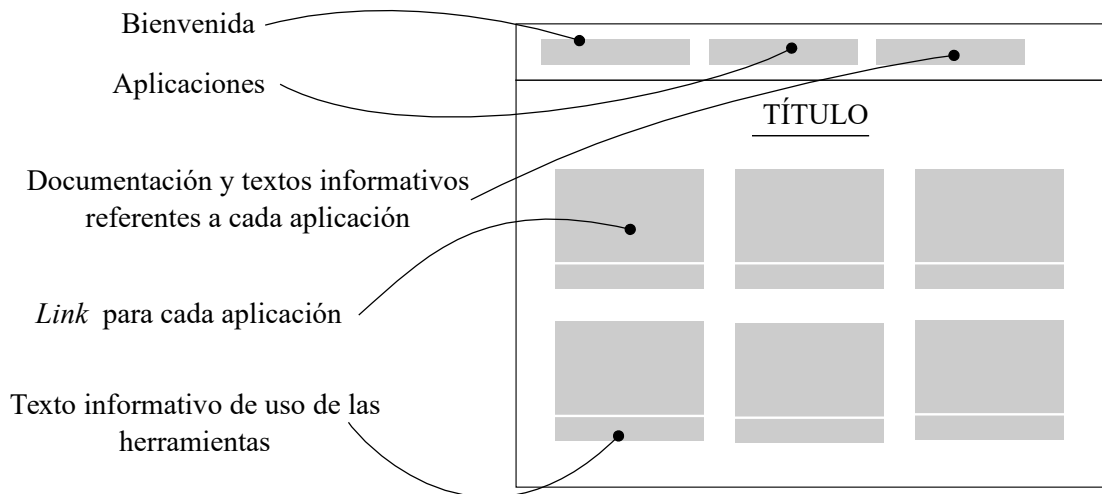
<b>Modelos matemáticos</b>	<b>Aplicación</b>
<i>Ajuste de distribución</i>	
Diagrama de bloques de confiabilidad: serie y paralelo	Confiabilidad total del sistema con componentes en serie y en paralelo
Sistema k-n	Confiabilidad total a partir de la confiabilidad de los elementos
Diagrama de bloques de confiabilidad: complejo	Confiabilidad total del sistema por medio de varios diseños de estructura de bloques
Árbol de fallas	Probabilidad de fallo del sistema, dadas las probabilidades de falla
Montecarlo	Probabilidad del de fallo del sistema, asignando distribuciones de probabilidad y generando números aleatorios.
<i>Cálculo de la disponibilidad</i>	
Markov: Un solo tipo de falla	Disponibilidad de un largo periodo de tiempo dado la tasa de fallo y la tasa de mantenimiento correctivo
Markov: Dos tipos de falla	Disponibilidad de un largo periodo de tiempo dado la tasa de fallo y la tasa de mantenimiento correctivo para cada modo de fallo
Markov: Un solo tipo de falla y modo de degradación	Disponibilidad según: tasa de fallo y tasa de reparación para cada estado: el sistema opera normalmente el sistema opera en modo de degradación el sistema falló
Markov: Sistema redundante (paralelo) dos unidades idénticas.	Disponibilidad según la tasa de fallo y la tasa de reparación
Markov: Parada para mantenimiento preventivo	Disponibilidad según las tasas de falla y las tasas para mantenimiento preventivo
<i>Optimización de tareas y tiempos de mantenimiento</i>	
Método de la ruta crítica (CMP)	Tiempo más temprano de inicio y finalización. Tiempo más tardío de inicio y finalización. Tareas críticas (tareas sin holgura de tiempo)
Técnica de Revisión y Evaluación de Programas (PERT)	Tiempo de la actividad esperada basada en tiempos probabilísticos, tiene como argumentos de entrada el tiempo mínimo, máximo y más probable.
Distribución de horas de mantenimiento	Número de horas hombre a la semana según la frecuencia y la duración de la tarea

<b>Modelos matemáticos</b>	<b>Aplicación</b>
Límites de control basado en la distribución de Poisson	Número de tareas de mantenimiento máximo y mínimo para un nivel de servicio Límites para el número de defectos relacionados con el mantenimiento
Optimización para tiempo para la inspección	Tiempo de la inspección según el tiempo de inactividad, el tiempo de inspección y número de inspecciones
<b>Gestión de repuestos</b>	
Stock de seguridad	Número de repuestos para un nivel de servicio
Estimación de repuestos	Probabilidad de tener un repuesto disponible
<b>Análisis de fallas</b>	
Pareto	Tareas más importantes según el número de fallas y el tiempo utilizado para mantenimiento correctivo
Análisis de correlación	Conocer si existe relaciones entre fallas según el coeficiente de correlación. Los datos de entrada son los tiempos entre falla
<b>Estimación de costos</b>	
Coste laboral de mantenimiento	Costo por empleado basado en la tarifa de mano de obra, horas anuales y la proporción de beneficios
Costo total de la mano de obra	Costo por hora estándar según el costo por periodo, horas planificadas producidas y equivalentes producidas, factor de beneficio
Costo de la reparación en relación a la mano de obra	Costo según el costo unitario para la reparación y factor de contracción reparable por pérdida
Coste laboral por acción correctiva	En base al costo de mano de obra y el horario programado anual
Costo por material	En base a el costo de almacenamiento, inventario, valor del artículo a cierto porcentaje de inflación
Costos por construcción	Calculado a partir del costo inicial y el costo de la infraestructura
Costo del equipo de mantenimiento	En base al costo de investigación y desarrollo y el costo de adquisición de la unidad de mantenimiento
Modelo de estimación de costos de tiempo inactivo	Salario del operador inactivo
Ciclo de vida del activo	Valor actual del activo en base a una tasa de interés

Realizado por: Jácome Darío, 2022

### 3.2. Desarrollo de las herramientas informáticas para la gestión,

Los modelos matemáticos se agrupan en cada uno de los módulos. El software de *back-end* shiny server carga las librerías a la memoria RAM cada vez que se inicializa cada una de las aplicaciones shiny, resulta idóneo agrupar en secciones en lugar cargar todas las librerías en una sola aplicación como se muestra en la ilustración 1-3

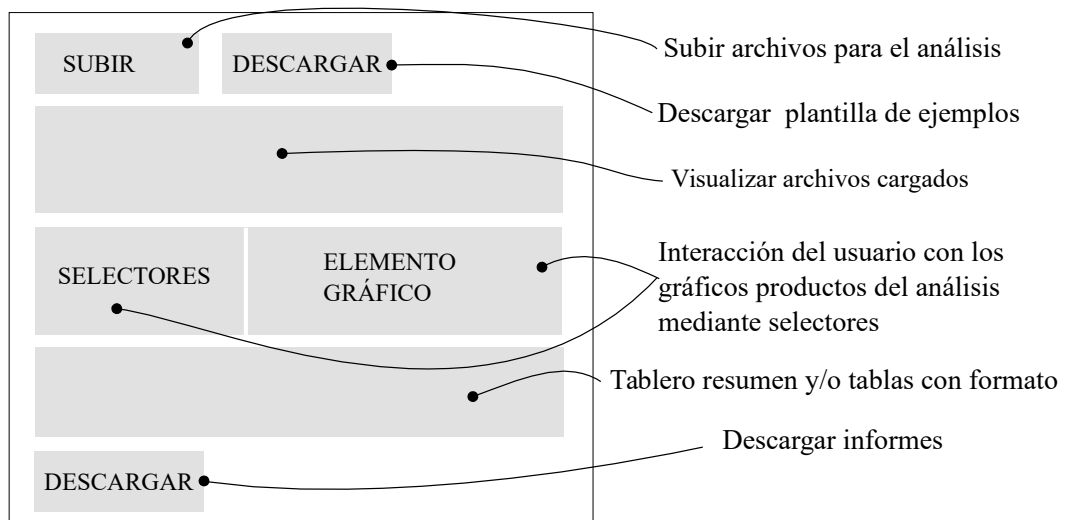


**Ilustración 1-3:** Diseño de la portada de la aplicación web

Realizado por: Jácome Darío, 2022

### 3.2.1. *Plantilla para los diferentes aplicativos*

Los diferentes tipos de datos pueden ser estructurados o semiestructurados, por ejemplo, una tabla o una lista son ejemplo de aquello, en la ilustración 2-3 se describe el formato para ingresar los datos.



**Ilustración 2-3:** Diseño general para cada aplicación

Realizado por: Jácome Darío, 2022

Shiny app no está optimizada para la web no obstante nos permite realizar una gran variedad de análisis, así como la generación de reportes



Entre los procesos que más recursos consumen en cuanto a uso de memoria cpu y ram se refiere está es proceso de generación de pdf y la renderización de gráficas en la página web.

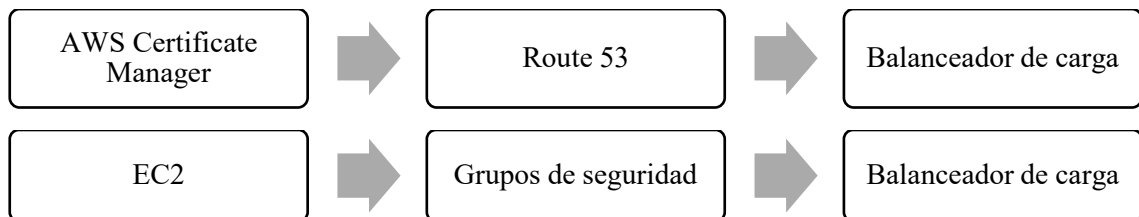
### 3.2.2. Infraestructura utilizada para el despliegue de las herramientas

Se puede crear aplicaciones en lenguaje R, con las librerías *shiny*, así como documentos dinámicos. La aplicación se conoce como *shiny-server* el cual se puede instalar en una distribución de Linux la cual puede ser Ubuntu.

La administración de la infraestructura se lo hace en máquinas virtuales de AWS llamado Amazon EC2, de igual forma se dispone de servicios para adquirir un dominio (nombre de la página web) y el certificado SSL que nos servirá para establecer una conexión segura, mostrado en la ilustración 3-3.

Al adquirir una máquina virtual se nos concederá un IP la cual cambia una vez se reinicie el servidor. Para evitar este inconveniente, en el mismo servicio permite crear una dirección IP elástica, además de asignarle un balanceador de carga.

El certificado SSL viene con registros del tipo CNAME, el cual deberá ser registrado en un administrador de dominios DNS, y a su vez registrar un alias al balanceador de carga.



**Ilustración 3-3:** Servicios utilizados para el despliegue de la aplicación web

Realizado por: Jácome Darío, 2022

### 3.2.3. Mantenimiento del software

Cada tres meses en necesario comprobar las actualizaciones, esto se las realiza con las siguientes líneas de código (ver ilustración 4-3)

```
sudo apt-get update  
sudo apt-upgrade
```

**Ilustración 4-3:** Actualización del sistema operativo Ubuntu

Realizado por: Jácome Darío, 2021

La infraestructura para el despliegue del software consta de 1giga de memoria RAM, en caso de que muchos usuarios se conecten, esta memoria se llenaría, de ser necesario se puede utilizar memoria de intercambio SWAP (ver ilustración 5-3)

```
sudo dd if=/dev/zero of=/swapfile bs=128M count=32
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
sudo swapon -s
sudo nano /etc/fstab
```

**Ilustración 5-3:** Asignación de memoria swap primera parte

Fuente: AWS, 2022

Una vez abierto el archivo “/etc/fstab”, se añade al final la siguiente línea de código (ver ilustración 6-3):

```
/swapfile swap swap defaults 0 0
```

**Ilustración 6-3:** Asignación de memoria segunda parte

Fuente: AWS, 2022

### 3.2.4. *Inconvenientes a futuro*

Las aplicaciones *shiny* puede presentar los siguientes inconvenientes:

- Error por utilizar librerías externas
- Lentitud al abrir las aplicaciones por utilizar un exceso de librerías
- En aplicaciones que permita tener paginas embebidas, los *widgets* pueden no funcionar impidiendo imprimir reportes y archivos.
- Es necesario utilizar aplicaciones externas para automatizar procesos, como la generación de documentos para la gestión de mantenimiento.
- Las versiones gratuitas de *shiny-server* no posee un control de usuarios por defecto.

Toda la codificación necesaria para el desarrollo de las herramientas se muestra en el anexo A, en donde se detalla, la utilización del lenguaje R, el desarrollo y despliegue de aplicaciones.

En cambio, en el Anexo B, se muestran las relaciones de ecuaciones aplicadas a confiabilidad

### 3.3. Bombas MTU de la empresa TEAM S.A.



**Ilustración 7-3:** Bombas M.T.U. de la empresa TEAM S.A.

Fuente: TEAM S.A.

En la ilustración 7-3 se observa que las unidades móviles, constan de un motor diésel *Kumms*, una bomba *triplex*, un contenedor *mainfloid* separados, que tiene las tres líneas alimentación, gas y producción, dispuestos en un separador trifásico.

#### 3.3.1. Herramientas desarrolladas para la gestión de las M.T.U.

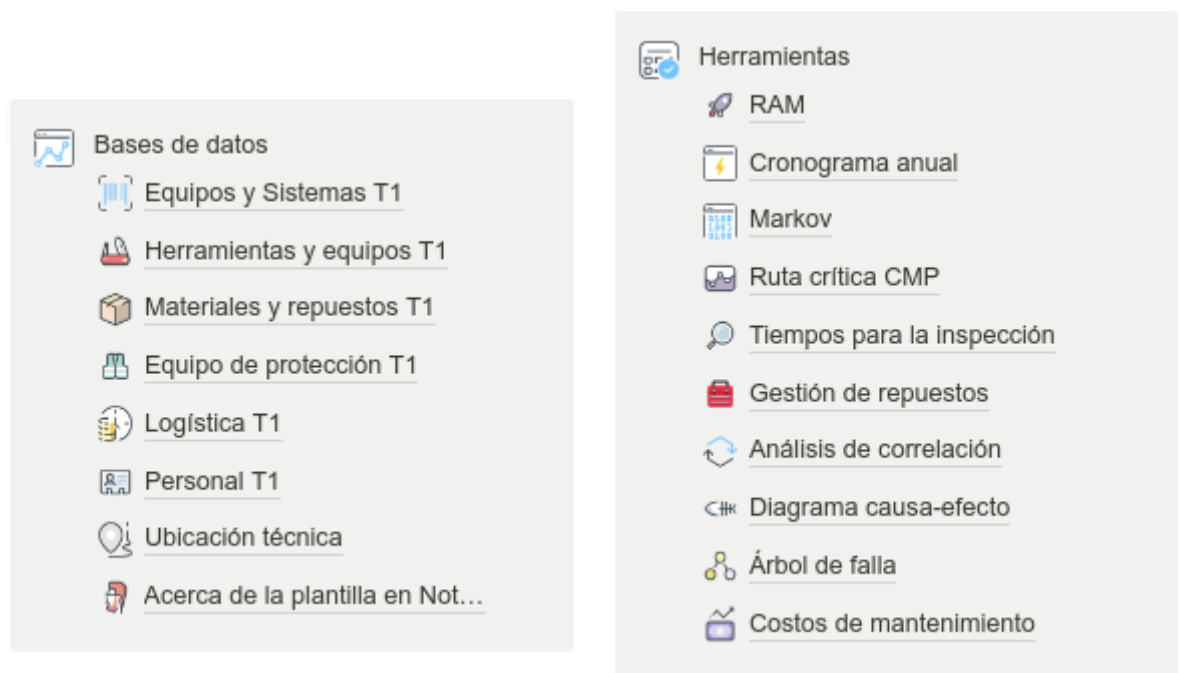
El lenguaje R permite crear aplicativos web llamadas aplicaciones “shiny”.

Primero se realiza una encuesta para verificar que herramientas de gestión son necesarias y generan valor para la empresa.

Una vez seleccionado los modelos matemáticos que cumplan las necesidades de la empresa, se procede a transcribir estos modelos a lenguaje R, este procedimiento se muestran en el anexo A, del mismo modo se muestran las fórmulas utilizadas.

Luego se prueban estos algoritmos con una serie de ejemplos y se puede comprobar que la codificación está correcta comparándolo con otro método.

Se despliega la aplicación mediante el software de *backend shiny server*, este programa permite ejecutar los programas desarrollados



**Ilustración 8-3:** Interfaz del usuario

Realizado por: Jácome Darío, 2022.

Luego se realizan las aplicaciones utilizando las funciones “tags”, que son etiquetas en lenguaje R que permite utilizar el lenguaje de marcado *html*, esto se lo hace con el fin de dar formato a cada aplicación. Las aplicaciones se muestran en la ilustración 8-3

**Tabla 3-2:** Modos de fallo de las MTUs

Descripción	Código
Lectura anormal en instrumento	AIR
Parada	BRD
Comunicación entre línea de control y pozo	CLW
Falla de control/señal	CSF
Operación retrasada	DOP
Fuga externa - combustible	ELF
Fuga externa – medio del proceso	ELP
Fuga externa – medio de suministro	ELU
Producción errática	ERO
Falla de conexión	FCO
Falla de corte	FCU
Falla en sistema de protección contra rayos	FLP
Error en frecuencia de salida	FOF
Error en voltaje de salida	FOV
Falla de rotación	FRO
No cierra bajo demanda	FTC
Falla de desconexión	FTD
Falla de funcionamiento bajo demanda	FTF

<b>Descripción</b>	<b>Código</b>
Falla de funcionamiento según lo requerido	FTI
Falla de bloqueo/desbloqueo	FTL
No abre bajo demanda	FTO
Falla de regulación	FTR
Falla de arranque bajo demanda	FTS
Falla durante funcionamiento	FWR
Alta producción	HIO
Falla de calefacción	HTF
Transferencia de calor insuficiente	IHT
Fuga interna – medio del proceso	ILP
Fuga interna – medio de suministro	ILU
Fuga interna	INL
Baja presión del suministro de aceite	LBP
Fuga en posición cerrada	LCP
Reducción de carga	LOA
Pérdida de flotación	LOB
Baja producción	LOO
Falla de anclaje	MOF
Ruido	NOI
Sin producción	NOO
Sobrecalentamiento	OHE
Otros	OTH
Cierre prematuro	PCL
Desviación de parámetro	PDE
Taponamiento/atascamiento	PLU
Pérdida de funciones en ambas cápsulas	POD
Potencia insuficiente	POW
Falla de transmisión de potencia/señal	PTF
Acumulación de lodo	SBU
Problemas menores en servicio	SER
Falla de ajuste/recuperación	SET
Alarma espuria de alto nivel	SHH
Alarma espuria de bajo nivel	SLL
Deslizamiento	SLP
Operación espuria	SPO
Deficiencia estructural	STD
No se detiene bajo demanda	STP
Atascamiento	STU
Deformación general	UBU
Desconocido	UNK
Parada espuria	UST
Vibración	VIB
Muy baja producción	VLO
Comunicación entre línea de control y pozo	WCL

**Realizado por:** Jácome Darío, 2021

### **3.3.2. Modelos utilizados para la gestión de mantenimiento**

El flujo de datos necesarios como: combinación de fallos, tiempo de fallos, tiempo para reparación, estos pasan por un proceso de filtrado y agrupamiento a fin de obtener solo los datos necesarios para realizar los siguientes análisis: bondad de ajuste, gráficos de Pareto, correlación, gráficos de distribución.

Muchas de estas salidas pueden ser reutilizadas, como en el caso del ajuste de distribución en donde se ingresa el TBF (Tiempo Entre Fallos) y el TTR (Tiempo Para Reparar).

El cálculo de la disponibilidad se lo realiza a mediante la obtención de la media de las distribuciones que más se ajusta a los tiempos de falla y los tiempos de reparación. Adicionalmente se grafica las funciones de densidad de probabilidad y la distribución acumuladas.

El gráfico de Pareto se lo realiza con los datos de falla y tiempos para reparación, posteriormente se muestran en un análisis de correlación.

Las tareas de mantenimiento, frecuencia, tiempo para la ejecución de tareas son analizados con la metodología de la ruta crítica, y gráficos de control de calidad.

El diagrama de control de calidad es útil también para marcar límites en los defectos en las tareas de mantenimiento.

La combinación de varios tiempos de falla puede ser analizados con diferentes árboles de fallas. No obstante, esta metodología puede utilizarse en otros ámbitos

### **3.3.3. Desarrollo del documento de ayuda**

La documentación necesaria para la correcta utilización de cada herramienta está disponible en la aplicación *notion*, como se muestra en la ilustración 9-3.

El manual de ayuda muestra la definición de los conceptos y fórmulas utilizadas para el desarrollo de las funciones utilizadas en el proyecto, de esta forma, si el usuario puede hacer uso de las fórmulas de manera individual.

En la ilustración 9-3 se muestra la documentación de las herramientas desarrolladas que consta de ejemplos, fórmulas y sugerencias para su aplicación. Además, el usuario podrá utilizar los códigos utilizados en el software.

### Distribución de horas de mantenimiento

```

crono <- function(s, f, length=52){
  n <- length(s)
  x <- 0
  y <- numeric(n)
  fs <- f-s
  for(i in seq(n)){
    val <- sign(!seq(fs[i]+1, fs[i]+length)%%f[i])
    x <- x + val
    y[i] <- sum(val)
  }
  return(list(sumx = x, sumy = y))
}

uno <- crono(round(rnorm(1000, mean = 10)), round(rnorm(1000, mean = 10)), 52)
eje_y <- uno$sumx
plot(eje_y, type = "o", xlab = 'tiempo [días]', ylab = 'Número de tareas')
abline(h = mean(uno$sumx))

```

### Límites de control basado en la distribución de Poisson

En trabajos de mantenimiento, este cuadro se puede utilizar para controlar la aparición de una serie de defectos del equipo relacionados con el mantenimiento. La distribución de Poisson se utiliza para obtener expresiones para los límites de control superior e inferior del gráfico. La media C de la distribución de Poisson se expresa mediante

$$\bar{C} = \frac{MD}{N}, \quad \sigma = \sqrt{\bar{C}} \text{ donde}$$

MD: número total de defectos relacionados con el mantenimiento, N: número total de equipos.  
La desviación estándar es:

$$UCL = C + 3\sigma, \quad LCL = C - 3\sigma$$

**Ilustración 9-3:** Manual de ayuda

Realizado por: Jácome Darío, 2022

**3.4. Análisis de fiabilidad, probabilidad de fallo y MTBF**

La creación de aplicaciones para el análisis de fiabilidad, disponibilidad y MTBF consta de las siguientes librerías como se muestra en la tabla 3-3.

**Tabla 3-3:** Librerías para el análisis de probabilidad de fallo, confiabilidad y MTBR

Utilidad	Librerías
Creación de aplicaciones web	<i>shiny</i>
Creación gráficos	<i>ggplot2</i>
Proporciona distribuciones de probabilidad adicionales	<i>actuar</i>
Renderiza ecuaciones en los gráficos, es decir $\lambda$ equivale a $\lambda$	<i>Latex2exp</i>

Realizado por: Jácome Darío, 2022

A continuación, se muestran partes de código utilizados para el desarrollo de las herramientas:

```

library(shiny)
library(ggplot2)
library(actuar)
library(latex2exp)

# Distribución exponencial
Rexp <- function(x,rate) 1-pexp(x,rate=rate)
texp <- function(x,rate) rate
mtbfexp <- function(rate) 1/rate

# Distribución de Weibull
Rweibull <- function(x, shape, scale) 1 - pweibull(x, shape = shape, scale =
scale)
tweibull <- function(x,shape,scale) (shape/scale)*(x/scale)^(shape-1)
mtbfweibull <- function(shape,scale) scale*gamma(1+1/shape)

# Distribución gamma
Rgamma <- function(x, rate, shape) 1 -pgamma(x, rate=rate, shape=shape)
tgamma <- function(x, rate, shape) dgamma(x, rate=rate, shape=shape)/(1-pgamma(x,
rate=rate, shape=shape))
mtbfgamma <- function(rate,shape) shape/rate

# Distribución normal
Rnorm <- function(x, mean, sd) 1 - pnorm(x, mean = mean, sd = sd)
tnorm <- function(x, mean, sd) dnorm(x, mean=mean, sd=sd)/(1-pnorm(x, mean=mean,
sd=sd))
mtbfnorm <- function(mean) mean

# Distribución log-normal
Rlnorm <-function(x,meanlog,sdlog) 1 - plnorm(x, meanlog = meanlog, sdlog =
sdlog)
tlnorm <- function(x, meanlog, sdlog) dlnorm(x, meanlog = meanlog, sdlog =
sdlog)/(1-plnorm(x, meanlog = meanlog, sdlog = sdlog))
mtbflnorm <- function(meanlog, sdlog) exp(meanlog+(1/2)*sdlog^2)

# Distribución inversa gaussiana
Rinvgauss <- function(x, mean, shape) 1 - pinvgauss(x, mean = mean, shape = shape)
tinvgauss <- function(x, mean, shape) dinvgauss(x, mean=mean, shape=shape)/(1-
pinvgauss(x, mean=mean, shape=shape))
mtbfinvgauss <- function(mean) mean

```

**Ilustración 10-3:** Extracto análisis de fiabilidad, disponibilidad y MTBF (global.R)

Realizado por: Jácome Darío, 2022



```

shinyServer(function(input, output) {

  output$distPlot <- renderPlot({
    if (is.na(input$ylim) || is.na(input$xdato)
    || is.na(input$ndatos) || is.na(input$adatos)){
      ggplot+geom_blank()+theme_classic()
    } else{
      adatos <- input$adatos
      ndatos <- input$ndatos
      ylim <- input$ylim
      xdato <- input$xdato

      if (input$dis=='exponencial'){
        if (is.na(input$lambda)){ggplot+geom_blank()+theme_classic()} else{
          ggplot(data.frame(x = adatos:ndatos), aes(x))+
            geom_vline(xintercept=mtbfexp(input$lambda),linetype="dotted")+
            geom_function(fun = dexp, args = list(rate =
            input$lambda),linetype="longdash")+
            geom_function(fun = Rexp, args = list(rate = input$lambda))+
            geom_function(fun = texp, args = list(rate =
            input$lambda),linetype="dotted",color="red")+
            labs(caption = TeX(paste("$\\lambda = ",input$lambda,"$")),
              x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
          }
        }

      else if (input$dis=='logística'){
        if (is.na(input$mdata) || is.na(input$sdata)){ggplot+geom_blank()+theme_classic()}
        else{
          ggplot(data.frame(x = adatos:ndatos), aes(x))+
            geom_vline(xintercept=mtbflogis(location = input$mdata),linetype="dotted")+
            geom_function(fun = dlogis, args = list(location = input$mdata, scale =
            input$sdata),linetype="longdash")+
            geom_function(fun = Rlogis, args = list(location = input$mdata, scale =
            input$sdata))+
            geom_function(fun = tlogis, args = list(location = input$mdata, scale =
            input$sdata),linetype="dotted",color="red")+
            labs(caption = TeX(paste("$\\mu = ",input$mdata,"$, $s =
            ",input$sdata,"$")),
              x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
          }
        }
      }
    }
  })
}

```

**Ilustración 11-3:** Extracto análisis de fiabilidad, disponibilidad y MTBF (server.R)

Realizado por: Jácome Darío, 2022

Ajuste de funciones para los datos del tiempo medio entre fallas y de reparación  
 Cálculo de la disponibilidad según las configuraciones en serie, paralelo, mixto, complejo,  
 sistemas k en n, sistemas con redundancia en espera.

Gráficos para análisis de fallas como Pareto, análisis de correlación, diagrama causa y efecto,  
 cálculo de tasas efectivas para sistemas redundantes.

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      selectInput("dis", label = "Distribución", choices =
c('exponencial', 'weibull', 'gamma', 'normal', 'log-normal', 'logística', 'log-
logística', 'gumbel')),
      conditionalPanel(condition = "input.dis == 'exponencial'",
        numericInput("lambda", "lambda:", min=0, value = 0.9)
      ),
      conditionalPanel(condition = "input.dis == 'logística'",
        numericInput("mdata", "m:", min=0, value = 7),
        numericInput("sdata", "s:", min=0, value = 2)
      ),
      conditionalPanel(condition = "input.dis == 'log-logística'",
        numericInput("allogis", "a:", min=0, value = 1.1),
        numericInput("sllogis", "s:", min=0, value = 0.5)
      ),
      conditionalPanel(condition = "input.dis == 'gumbel'",
        numericInput("adata", "alpha:", min=0, value = 4),
        numericInput("bdata", "scale:", min=0, value = 2)
      ),
    ),
  ),
)

```

**Ilustración 12-3:** Extracto análisis de fiabilidad, disponibilidad y mantenibilidad (ui.R)

Realizado por: Jácome Darío, 2022

### 3.5. Fórmulas reescritas utilizadas el análisis de confiabilidad

Dado que las fórmulas pueden ser expresadas con diferentes variables, se reescriben algunas de ellas para ser transcritas a código R con mayor facilidad.

Media de una distribución

$$m = \int_0^{\infty} R(t) \quad (57)$$

Confiabilidad acumulada hasta un tiempo T

$$m_T = \int_0^T R(t) \quad (58)$$

Tasa de fallos acumulada

$$H = -\ln(R) \quad (59)$$

Confiabilidad acumulada hasta un tiempo T

$$R_{(k,n)} = \sum_{i=k}^n (-1)^{i-k} \binom{-i-1}{k-1} \sum_{j_1 < j_2 < \dots < j_i} \prod_{l=1}^i p_{j_l} \quad (60)$$

Confiabilidad acumulada para para un sistema k-n con la misma confiabilidad

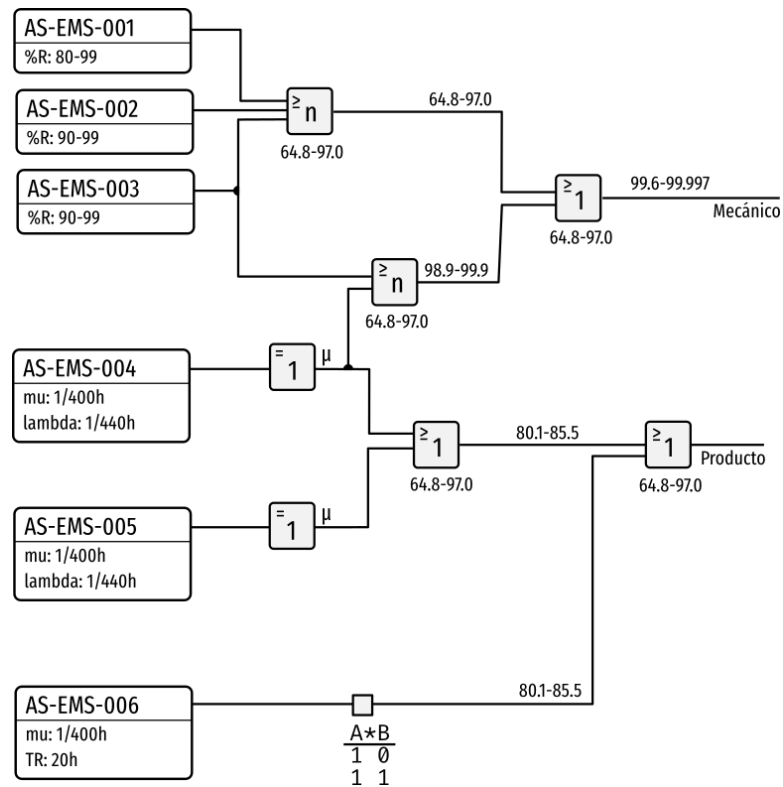
$$R_s = \text{pbeta}(1 - R, k, n - k + 1) \quad (61)$$

Disponibilidad en base a diagramas de Markov de dos estados

$$R_s = \prod_{i=1}^n \frac{\mu_i}{\mu_i + \lambda_i} \quad (62)$$

### 3.5.1. Alternativa a compuertas de confiabilidad

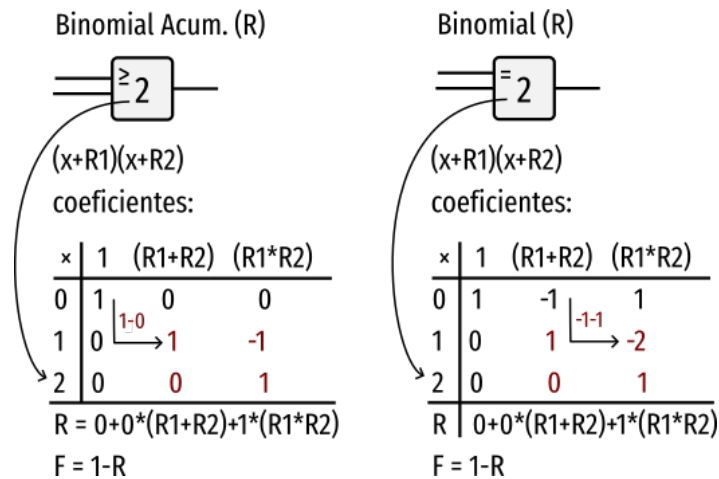
En un sistema de componentes en donde se requiera conocer la confiabilidad resultante se sugiere graficarlos de la siguiente manera:



**Ilustración 13-3:** Ejemplo gráfico de las compuertas de confiabilidad

Realizado por: Jácome Darío, 2022

Comúnmente se utiliza compuertas de manera análoga a los circuitos eléctricos, es decir con compuertas, *and*, *or*, *not*, etc. Sin embargo, en este caso se utiliza conceptos de la estadística para obtener los cálculos requeridos. En este caso, se sugiere utilizar la distribución binomial a fin de obtener las diferentes combinaciones para el cálculo de la confiabilidad y no confiabilidad.



**Ilustración 14-3:** Compuertas para el cálculo de la confiabilidad

Realizado por: Jácome Darío, 2022

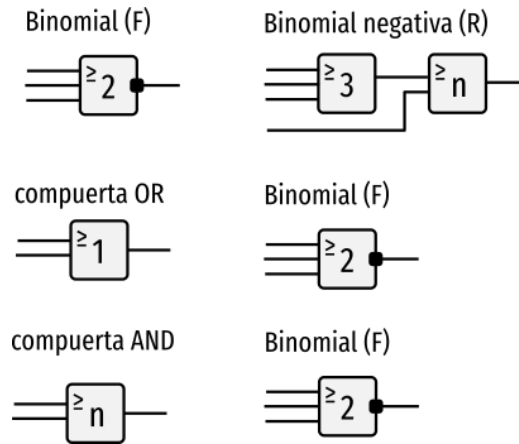
En la ilustración 14-3 se muestra la compuerta binomial acumulada, de la cual se puede obtener la confiabilidad resultante de un sistema “k en n”, es decir:

- Si k es igual a n, entonces se trata de un sistema en serie porque requiere que todos los componentes para funcionar.
- Si k es igual a 1, se trata de un sistema en paralelo.

El procedimiento para efectuar el cálculo es el siguiente:

- Se forma la expresión a partir de las confiabilidades individuales
- Se extraen los coeficientes desde el monomio de mayor grado.
- Se forma una columna y una fila con ceros, pero el primer elemento es uno. En cambio, para los demás datos de la columnas y filas se resta el primer elemento del segundo de la columna anterior y se escribe en la segunda fila. Como se muestran los datos marcado de rojo de la ilustración 14-3.
- Luego se escribe al lado derecho de la matriz realizada la serie empezando desde cero hasta el número de elementos, lo que corresponde a las alternativas para la variable k.
- Luego se multiplica los coeficientes con la fila en donde está k y se suman estos resultados

El resultado representa que la probabilidad de que k elementos.



**Ilustración 15-3:** Combinación de compuertas de confiabilidad

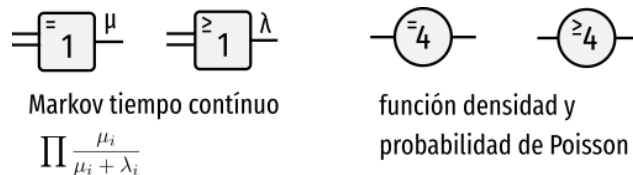
Realizado por: Jácome Darío, 2022

Para un sistema con probabilidad de estar en buen funcionamiento  $mu$  y en falla  $lambda$  en un diagrama de Makov se obtiene que la disponibilidad es:

$$A_s = \frac{u}{u + \lambda} \tag{63}$$

De igual forma para un diagrama de Markov de tiempo continuó con tasa de reparación constante y tasa de fallo constante  $mu$  y  $lambda$ .

Por otra parte, para calcular el nivel de servicio de un sistema se utiliza modelos basados en la distribución de Poisson, la cual se puede representar con un círculo.



**Ilustración 16-3:** Compuertas adicionales para el análisis de confiabilidad

Realizado por: Jácome Darío, 2022

El desarrollo del código se muestra en la ilustración 17-3 en donde la función *imult* representa los coeficientes para un monomio:

$$(x + p_1)(x + p_2)(x + p_3) \dots (x + p_n) \tag{64}$$

La variable *ind* muestra la matriz inicial

La función *gcomb* devuelve la confiabilidad para un sistema “k en n” con diferentes tipos de confiabilidad.

```
# Productoria para compuerta binomial

imult <- function(x){
  z <- c(1, x[1])
  n <- length(x)
  if (n==1) return(z)
  for (i in 2:n) z <- c(z,0)+c(0,z*x[i])
  z
}

# Compuerta binomial

gcomb <- function(x,k,a=1){
  n <- length(x)
  ind <- choose(a-(k+1),(-k):(n-k))
  sum(imult(x)*t(ind))
}
```

**Ilustración 17-3:** Código para la compuerta binomial

Realizado por: Jácome, Darío, 2022

**3.5.2. Tareas de mantenimiento**

**Tabla 3-4:** Librerías para el análisis de probabilidad de fallo, confiabilidad y MTBR

Utilidad	Librerías
Creación de aplicaciones web	<i>shiny</i>
Creación de tablas <i>html</i> responsivas	<i>DT</i>
Lee archivos de Excel de extensión <i>xlsx</i>	<i>readxl</i>
Manipulación de <i>dataframe</i> (tablas), permite formar grupos, filtro, incluir nuevos campos (columnas)	<i>dplyR</i>
Gráficos de Gantt	<i>plotrix</i>

Realizado por: Jácome Darío, 2022

Para la gestión de tareas de mantenimiento se describen las técnicas de revisión y evaluación de programas, método de la ruta crítica, distribución de horas de mantenimiento y control de defecto en las tareas de mantenimiento y tiempos para la inspección.

**3.5.3. Gestión de repuestos**

En la documentación de gestión de repuestos se muestra el cálculo del stock de seguridad, según el nivel de servicio requerido y según el número de repuestos necesarios en bodega.

**Tabla 3-5:** Librerías para la gestión de repuestos

Utilidad	Librerías
Creación de aplicaciones web	<i>Shiny</i>

Realizado por: Jácome Darío, 2022

El factor de seguridad se calcula como:

$$P_s = \sum_{j=0}^n \left\{ \left[ (-1) \ln e^{-q\lambda t} \right]^j e^{-q\lambda t} \right\} / j! \quad (64)$$

Reescrita de otra forma

$$P_s = \sum_{j=0}^n \frac{[-\ln R]^j R}{j!}$$

y su código en R se puede observar en la ilustración 13-3

```
Ps <- function(x,q,lambda,n){
  res <- numeric()
  for (i in x){
    E <- exp(-q*lambda*i)
    res <- c(res, E*sum((( -log(E))^^(0:n))/factorial(0:n)))
  }
  res
}
```

**Ilustración 18-3:** Código utilizado para el cálculo de factor de seguridad

Realizado por: Jácome Darío, 2022

La fórmula anterior basa en una distribución de Poisson con lambda igual a la tasa de fallo acumulada. De la cual se puede obtener otra fórmula simplificada basada en la función gama de la siguiente manera:

$$R_s = \frac{\Gamma(n+1, \lambda)}{\Gamma(n+1)} \quad (65)$$

donde  $\Gamma(x, y)$  es la función gamma incompleta y  $\Gamma(x)$  es la función gama. Si se generaliza este concepto a otro tipo de distribuciones se obtiene la siguiente ecuación:

$$R_s = \frac{\Gamma(n+1, -\ln(R))}{\Gamma(n+1)} \quad (66)$$

### 3.5.4. Estimación de costos.

**Tabla 3-6:** Librerías para la estimación de costos

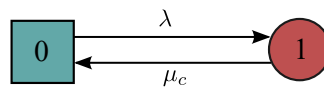
Utilidad	Librerías
Creación de aplicaciones web	Shiny

Realizado por: Jácome Darío, 2022

Los modelos están basados en análisis estadístico para la estimación del costo de mano de obra, equipos, material, construcción y costo de propiedad de los equipos. Estos modelos están descritos en la tabla 3-6 y son estimaciones simples que se utilizan para la proyección de los costos de mantenimiento.

Las fórmulas utilizadas corresponden al valor presente, valor presente de una anualidad y el costo anual equivalente

### 3.5.5. Markov tiempo continuo



**Ilustración 19-3:** Diagrama de Markov de tiempo continuo

Realizado por: Jácome Darío, 2022

$$\begin{aligned} \frac{dP_0(t)}{dt} &= -\lambda P_0(t) + \mu P_1(t) \\ \frac{dP_1(t)}{dt} &= -\mu P_1(t) + \lambda P_0(t) \end{aligned} \quad (67)$$

Donde:

$P_0$  Es la probabilidad de que el sistema esté operativo

$P_1$  Es la probabilidad de que el sistema falle:

Aplicando la transformada de Laplace

$$\begin{aligned} sP_0(s) - P_0(0) &= -\lambda P_0(s) + \mu P_1(s) \\ sP_1(s) - P_1(0) &= -\mu P_1(s) + \lambda P_0(s) \end{aligned} \quad (68)$$



Revolviendo el sistema

$$P_0(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t} \quad (69)$$

Como alternativa se puede utilizar el método de Runge Kutta para resolver las ecuaciones diferenciales. Este método se encuentra disponible en la función *rk4sys* de la librería *pracma*.

En la ilustración 20-3 se muestra el cálculo para un sistema de Markov con dos estados con condiciones iniciales 1 y 0 en un intervalo de tiempo de 0 a 20.

```
u <- 700
l <- 500

f <- function(t, p) {
  A <- c(-l, u, l, -u)
  s <- matrix(A, 2, byrow = T)%%c(p[1], p[2])
  return(c(s[1], s[2]))
}

sol <- rk4sys(f, 0, 0.01, c(1,0), 20)
```

**Ilustración 20-3:** Código utilizado para la resolución de diagramas de Markov

Realizado por: Jácome Darío, 2022

### 3.5.6. *Recolección de la información*

Se utiliza el software “Notion” para acumular la información acerca de la gestión de mantenimiento, la cual, almacena información en base de datos a fin de realizar sus respectivas relaciones entre cada equipo.

La plantilla consta de los siguientes bases de datos:

- Equipos y Sistemas (ver ilustración 21-3)
- Herramientas y equipos (ver ilustración 22-3)
- Materiales y repuestos (ver ilustración 23-3)
- Equipo de protección (ver ilustración 24-3)

### Equipos y Sistemas T1

Datos Sistemas Equipos + Filter Sort Q ... New

Código	Descripción	Nivel	Ubicación	Ruta de in...	Imagen	Qr	+ ...
T1-P01-MTU01	Unidad Móvil de bombeo y prueba	Sistema	00AA001	R-12			
T1-P01-MTU01-MMC01	Motor de Combustión	Equipo	00AA001	R-A R-B			
T1-P01-MTU01-MBB01	Bomba desplazamiento positivo	Equipo	00AA001	R-A			
T1-P01-MTU01-EBB02	Tablero de control	Equipo	00AA001	R-A R-B			
T1-P01-MTU01-MBB03	Bomba de Inyección de Químico	Equipo	00AA001	R-A			
T1-P01-MTU01-MTR01	Transmisión	Equipo	00AA001	R-A			
T1-P01-MTU01-MSP01	Separador	Equipo	00AA001	R-B			
T1-P01-MTU01-MPL01	Plataforma	Equipo	00AA001	B-C			
T1-P01-MTU02	Unidad Móvil de bombeo y prueba	Sistema	00AA001	R-14			
T1-P01-MTU02-MMC01	Motor de Combustión	Equipo	00AA001	R-B			
T1-P01-MTU02-MBB01	Bomba desplazamiento positivo	Equipo	00AA001	R-A			
T1-P01-MTU02-EBB02	Tablero de control	Equipo	00AA001	R-B			

### Ilustración 21-3: Codificación de equipos y sistemas

Realizado por: Jácome Darío, 2022

Cada base de datos consta de los siguientes ítems:

- Nombre: Se refiere al código del equipo o al nombre, este dato deberá ser único
- Imagen: Incluye una o diferentes imágenes del equipo
- Tipo: Se propone una clasificación al activo, para segmentar la información
- Link: Se refiere al enlace en donde puedo adquirir el activo

### Herramientas y equipos T1

Datos Galeria + Filter Sort Q ... New

Name	tipo	costo inicial	imagen	link	+ ...
Esmeriladora amoladora de 4 1/2 pulgadas motor 850 watts 11000 rpm 110 voltios makita	herramientas eléctricas	\$9.64		<a href="https://www.pintulac.com.ec/m...">https://www.pintulac.com.ec/m...</a>	
Llave de impacto inalambrica con mango de 1/2 pulgada bateria de 18 voltios torque de 1000nm makita	herramientas eléctricas	\$564.48		<a href="https://www.pintulac.com.ec/la...">https://www.pintulac.com.ec/la...</a>	
Lavadora A Presion Porten 2hp - 1950psi	herramientas eléctricas	\$130.78		<a href="https://www.pintulac.com.ec/lav...">https://www.pintulac.com.ec/lav...</a>	
Escalera	andamios y estructuras	\$118.65		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Morza 6" Ingco base giratoria con yunque Ingco hbv086	herramientas de sujeción	\$201.32		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Taladro de banco 13mm ingco 350w 580-2650 RPM	herramientas eléctricas	\$194.17		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Destornillador phillips	herramientas de mano	\$1.34		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Estilete	herramientas de mano	\$1.11		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Pinza ojalillo sacabocado	herramientas de mano	\$11.06		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Set organizador +29 herramientas básicas para el hogar ingco hktv01h291	set de herramientas	\$41.85		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Llave combinada boca y estriada	herramientas de mano	\$2.00		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	
Valija maletín metálico 147 herramientas con ruedas INGCO hktv021473	set de herramientas	\$294.00		<a href="https://www.ingcotools.com.uy/">https://www.ingcotools.com.uy/</a>	

Calculate v sum \$36,432.33

### Ilustración 22-3: Lista de herramientas

Realizado por: Jácome Darío, 2022

### 3.5.6.1. Limitaciones acerca del uso de bases de datos

Cada campo puede ser creado para almacenar información referente a números, texto, archivos, además permite relacionar tablas con otras tablas con la opción “*Relation*” y traer esa información con “*Rollup*” (ilustración 23-3), para luego aplicar funciones de agregación (suma, media, etc). No obstante, solo se puede traer a la tabla un solo campo, es decir si se quiere traer a la tabla de Logística el costo de “1 galón de aceite de lubricación” solo se puede obtener el campo precio y no dos campos la unidad y el precio.



#### Ilustración 23-3: Lista de herramientas

Realizado por: Jácome Darío, 2022

En la ilustración 24-3 se muestra la siguiente notación Aceite 15w40 [1 gl] con un costo de 21,50 USD. Si una tarea de mantenimiento requiere de 1/10 de ese recurso se registraría como 15w40 [0.1 gl] con un costo de 2.15 USD.

El inconveniente explicado anteriormente hace que se tengan que duplicar los registros según las porciones de material que deseamos utilizar.

Descripción	Costo p...	Unidades	Costo	Tipo	Imagen	Link
Aceite 15W40 [1 gl]	\$21.50	1	\$21.50	lubricación		
Aceite 15W40 [0.1 gl]	\$21.50	0.1	\$2.15	lubricación		
Aceite sintético para frenos Prestone DOT 3 [1 gal]	\$9.20	1	\$9.20	lubricación		<a href="https://www.amazon.com/-/es/">https://www.amazon.com/-/es/</a>
Aceite 20W50 [1 gl]	\$17.50	1	\$17.50	lubricación		
Banda de transmisión V [1 unidad]	\$25.00	1	\$25.00	repuestos		
Stanadyne Bomba de inyección Re-518164 John Deere 6068 núcleo inyector de combustible diesel JD [1 unidad]	\$499.99	1	\$499.99	repuestos		<a href="https://www.ebay.com/itm/2345">https://www.ebay.com/itm/2345</a>
Bomba de transferencia [1 unidad]	\$165.00	1	\$165.00	repuestos		
Portaelectrodo [1 unidad]	\$2.86	1	\$2.86	uso eléctrico		<a href="https://www.bpecuador.com/pr">https://www.bpecuador.com/pr</a>
Cables de batería [1 m]	\$20.00	1	\$20.00	uso eléctrico		<a href="https://www.amazon.com/-/es/t">https://www.amazon.com/-/es/t</a>
Contactador de 50A [1 unidades]	\$39.00	1	\$39.00	repuestos		<a href="https://www.plc-city.com/shop/e">https://www.plc-city.com/shop/e</a>
Contactador trifásico 50a 120v ( ref: v-0113 )	\$24.83	1	\$24.83	repuestos		<a href="https://inselec.com.ec/store/inlk">https://inselec.com.ec/store/inlk</a>

#### Ilustración 24-3: Lista de materiales y repuestos

Realizado por: Jácome Darío, 2022

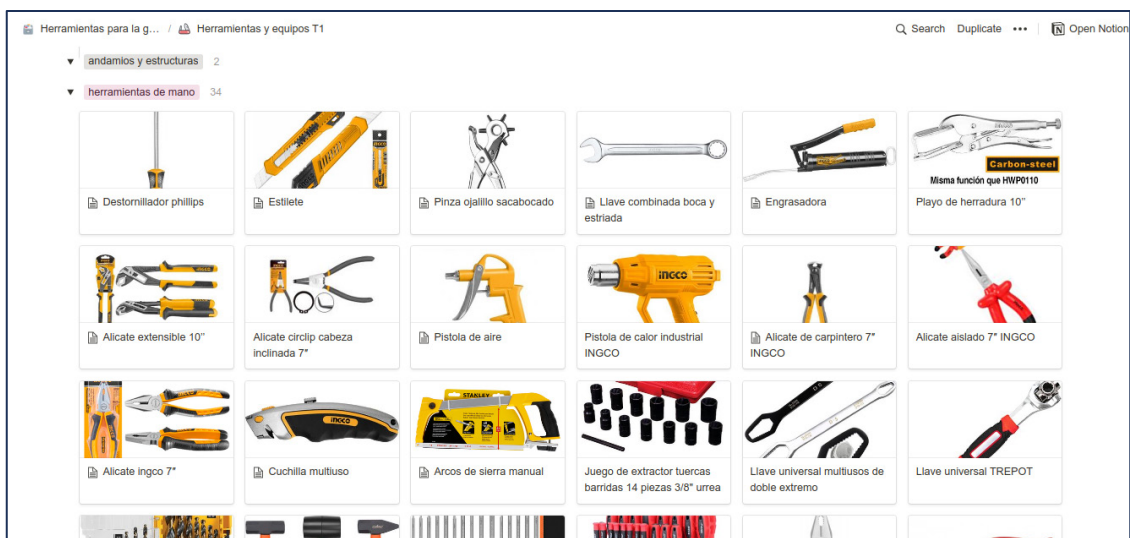
Datos		Galería		
Name	Imágenes	Costo	link	Tipo
Zapato de trabajo		\$86.40	<a href="https://tiendamia.com/ec/producto?amz=B01BZ4D2AO&amp;">https://tiendamia.com/ec/producto?amz=B01BZ4D2AO&amp;</a>	calzado
Arnes de seguridad tipo paracaidas		\$75.64	<a href="https://www.ingcotools.com.uy/catalogo/herramientas/arnes-de-seguridad-tipo-paracaidas/">https://www.ingcotools.com.uy/catalogo/herramientas/arnes-de-seguridad-tipo-paracaidas/</a>	contra caídas
Respirador		\$7.99	<a href="https://www.elferretero.com.ec/products/356?pr_prod_sti">https://www.elferretero.com.ec/products/356?pr_prod_sti</a>	respiratoria
Gafa policarbonato		\$1.99	<a href="https://www.elferretero.com.ec/products/201?pr_prod_sti">https://www.elferretero.com.ec/products/201?pr_prod_sti</a>	visual
Guante mecánico nitrilo		\$4.92	<a href="https://www.pintulac.com.ec/guante-mecanico-nitrilo-esp">https://www.pintulac.com.ec/guante-mecanico-nitrilo-esp</a>	manos
Guante de cuero corto		\$9.98	<a href="https://www.degso.com/?product=guante-de-cuero-corto">https://www.degso.com/?product=guante-de-cuero-corto</a>	manos
Guante bañado en nitrilo		\$10.51	<a href="https://www.degso.com/?product=guantes-de-doble-cap">https://www.degso.com/?product=guantes-de-doble-cap</a>	manos
Guante de algodón con pupos		\$0.91	<a href="https://www.degso.com/?product=guante-de-algodon-cor">https://www.degso.com/?product=guante-de-algodon-cor</a>	manos
Guantes desechables de nitrilo		\$14.00	<a href="https://www.amazon.com/-/es/desechables-seguridad-me">https://www.amazon.com/-/es/desechables-seguridad-me</a>	manos
Guante para las industrias petroquímica		\$12.67	<a href="https://www.degso.com/?product=guantes-de-nitrilo-19-p">https://www.degso.com/?product=guantes-de-nitrilo-19-p</a>	manos
Protección facial		\$15.00	<a href="https://www.amazon.com/-/es/Safe-Handler-Protector-Pr">https://www.amazon.com/-/es/Safe-Handler-Protector-Pr</a>	facial
Cinta de seguridad		\$14.70	<a href="https://kywitiendaenlinea.com/product/cinta-seguridad-ar">https://kywitiendaenlinea.com/product/cinta-seguridad-ar</a>	Señalización
Bota negra venus 39		\$11.60	<a href="https://kywitiendaenlinea.com/product/bota-negra-rainfai">https://kywitiendaenlinea.com/product/bota-negra-rainfai</a>	calzado

**Ilustración 25-3:** Lista de equipos de protección

Realizado por: Jácome Darío, 2022

### 3.5.6.2. *Uso de vistas en las bases de datos*

Cada base de datos se puede presentar de diferentes vistas como puede ser, calendarios, galerías, líneas de tiempo y listas. En la ilustración 26-3 se muestra la vista de “galería” para la base de datos de repuestos y equipos.



**Ilustración 26-3:** Galería de la base de datos de repuestos y equipos

Realizado por: Jácome Darío, 2022

### 3.5.7. Gestión de tareas de mantenimiento

Según la norma 14224 acerca de la recolección de datos para mantenibilidad y confiabilidad, los registros de mantenimiento correctivo y preventivo deben ser similares

Los campos para las tareas de mantenimiento son:

- Estado (iniciado, en progreso y completado)
- Fecha (puede ser una fecha única o un rango en el cual se desempeña la tarea de mantenimiento)
- Personal (Correo electrónico de la persona asignada)
- Tipo de mantenimiento (Preventivo, correctivo, a condición, programado, inspecciones)
- Tiempo de ejecución en minutos (tiempo real en la que se ejecutó la tarea)
- Tiempo de parada en minutos (tiempo real en el que estuvo parado el equipo)
- Equipos (Código del sistema o equipo)

En la ilustración 27-3 se muestra el campo de estado, que describe si la tarea está siendo ejecutada o no, y si se ha culminado con dicha tarea.

Las fechas pueden ser un día en específico o un rango a la cual se le ha asignado un intervalo de tiempo para desarrollarla, en la parte inferior de la plantilla se describe las fechas de planificación, a fin de que las tareas puedan ser reprogramadas, sin perder la información.

Cronograma							
[Datos] [Diario] [Semanal] [Mensual]							
Código	Estado	Fecha	# T. Ejec...	# T. de p...	Personal	Tipo de...	Equipos
FT-34421	No iniciado	2022/08/03 12:00 AM → 2022/08/08 12:00 AM	45.3	23.5	DARIO JA	Inspección	T1-P01-MTU01-MTR01 T1-P01-MTU01-MPL01
FT-2311	No iniciado	2022/08/24	24.5	26.3	DARIO JA	Correctivo	T1-P01-MTU01-MBB03
FT-2311	En progreso	2022/09/13					
FT-2311	No iniciado	2022/08/01					
FT-0021	Completado	2022/07/18					
FT-0012	No iniciado	2022/09/01					
FT-2131	Completado	2022/08/02					
FT-0121	Completado	2022/07/09					
FT-0001	En progreso	2022/08/19					
FT-0021	Completado	2022/09/05					
FT-0011	En progreso	2022/07/07					

**Ilustración 27-3:** Cronograma de mantenimiento

Realizado por: Jácome Darío, 2022

## FT-34421

Estado	No iniciado
Fecha	2022/08/03 12:00 AM → 2022/08/08 12:00 AM
Personal	DARIO JAVIER JACOME IZA
Tipo de manteni...	Inspección
T. Ejecución [min]	45.3
T. de parada [min]	23.5
Equipos	T1-P01-MTU01-MTR01 T1-P01-MTU01-MPL01

Fecha planificada:	Tiem. de ejecución ocupado [min]:	Tiempo de parada estimado [min]:	Personal:	Tipo de tarea:
2022/06/06	20	20	Electricista	A condición

### Observaciones generales:

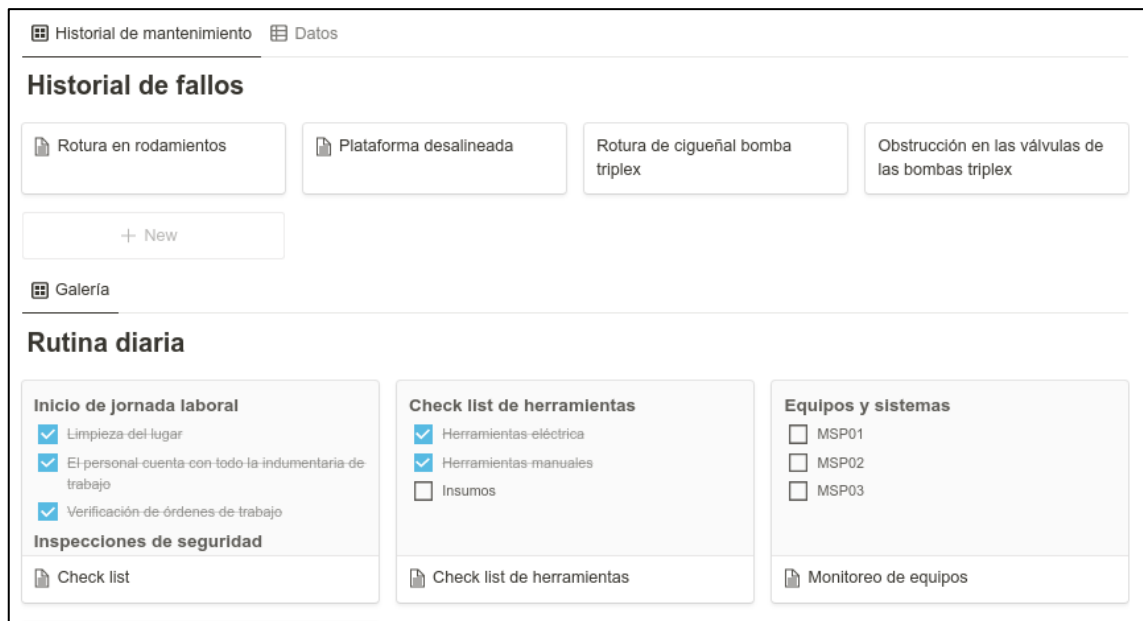
### Observaciones de seguridad:

### Ilustración 28-3: Registro de tareas de mantenimiento

Realizado por: Jácome Darío, 2022

En la parte de observaciones generales y de seguridad, el encargado de la tarea deberá describir si ha existido algún inconveniente en estas áreas.

El campo equipo, sirve para asignar las tareas a múltiples equipos a fin de evitar repetir la información, en el presente trabajo de integración curricular se repites los equipos MTU's. Si estos activos presentan un contexto operacional similar se pueden establecer las mismas tareas.



**Ilustración 29-3:** Historial de fallos y rutinas diarias

Realizado por: Jácome Darío, 2022

En la ilustración 29-3 se muestra una vista en forma de galería para los datos de fallos ocurridos en las unidades móviles de bombeo, así como las rutinas diarias.

Las rutinas diarias, consta del registro del personal, verificación de herramientas y registro de datos de operación.

## CAPÍTULO IV

### 4. RESULTADOS

#### 4.1. Análisis de los aplicativos para la gestión de mantenimiento de las M.T.U.

##### 4.1.1. Debilidad de las herramientas.

Tabla 4-1: Descripción de los inconvenientes en el desarrollo de las herramientas

Herramientas utilizadas	Descripción
Montecarlo	Requiere de un mayor tiempo para su ejecución
Modelos de Markov	Aumenta la complejidad de la resolución dado que se forma un sistema de ecuaciones diferenciales
Distribuciones de probabilidad	Presenta dificultades para modelar tasas de fallas en conjunto con las tasas de reparación
Diagrama causa-efecto	Mediante las librerías de R solo se puede graficar un solo nivel de causas
Árbol de fallas	Necesita de una estructura al momento de ingresar los datos. Además de capacitación para su utilización

Realizado por: Jácome Darío, 2022

##### 4.1.2. Beneficios de utilizar la interfaz del aplicativo

La interfaz gráfica permite ordenar cada uno de los módulos, a fin del usuario pueda elegir el método a analizar, además de que al momento de ingresar solo se carguen los datos solicitados.

En cada uno de los módulos se dispone de una plantilla en Excel para ingresar los datos, esto facilita la compilación y el ingreso de los datos en el software. A su vez al contar con un esquema de datos estructurados como es el caso de una tabla, se puede utilizar los códigos desarrollados para la gestión de mantenimiento en otra plataforma, por ejemplo, en el uso de API's, es donde es necesario contar con este tipo de estructura de datos.

##### 4.1.3. Uso del aplicativo en la empresa

En sí las herramientas para la gestión fueron desarrolladas en la aplicación “*Notion*” y su plantilla se encuentra disponible en la página web [mantenimientoactivo.com](http://mantenimientoactivo.com), en cambio el procesamiento de datos se lo realiza con el software “*shiny-server*” cuyas aplicaciones web se encuentran embebidas en la plantilla anteriormente mencionada.



Una vez habilitada la aplicación en la web la empresa TEAM S.A. contará con los siguientes módulos para realizar su gestión de las M.T.U. ver tabla 4-2

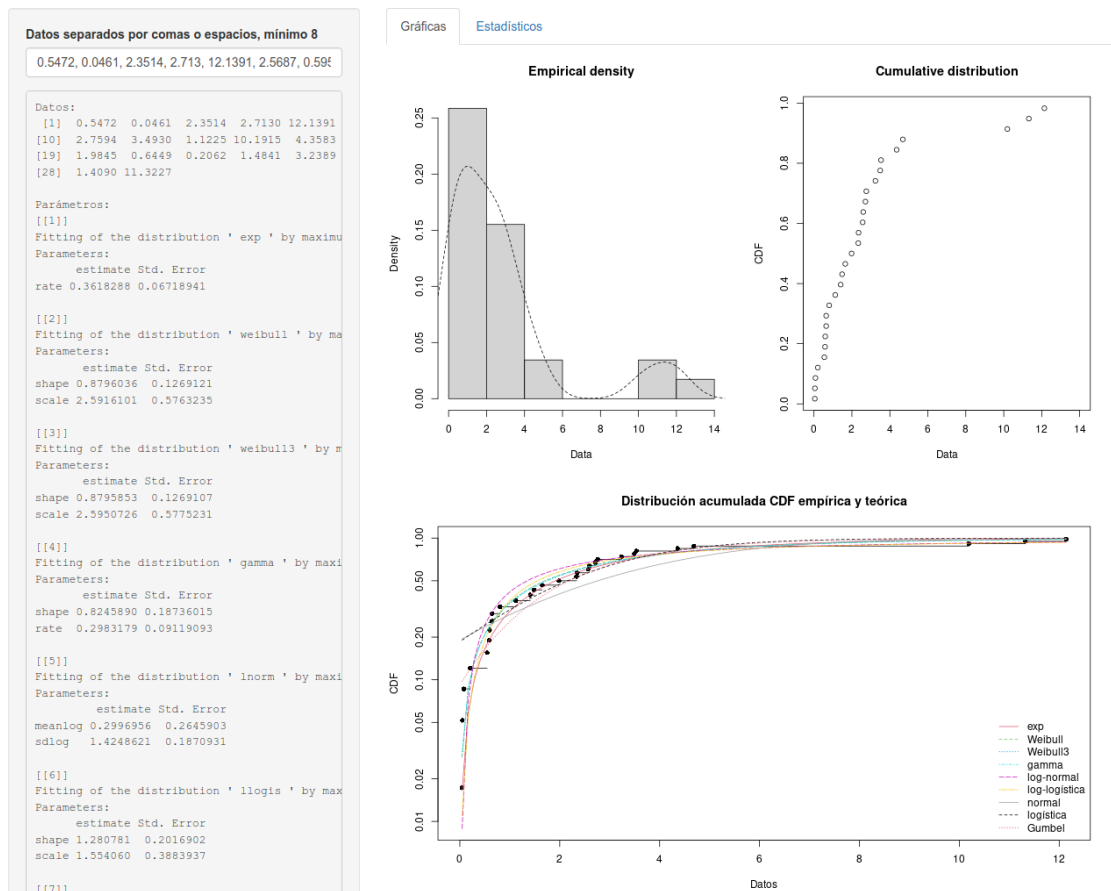
**Tabla 4-2:** Aplicaciones disponibles

Modelo	Aplicaciones
Cálculo de la fiabilidad	Cálculo de la fiabilidad
Cronograma anual	Determina que día de la semana realiza el mantenimiento, de acuerdo de acuerdo a la carga horaria, tiempo y frecuencia
Análisis de fallas	Estimación del tiempo necesario para la reparación
Método de la ruta crítica	Establecer secuencia de tareas sin holgura de tiempo
Gestión de repuestos	Se determina el número necesario de repuestos
Módulo de costos	Proyección de los costos de mantenimiento

Realizado por: Jácome Darío, 2022

#### 4.1.4. Ajuste de distribuciones

La aplicación presenta diferentes distribuciones de probabilidad como: exponencial, Weibull de dos y res parámetros, gamma, normal, log-normal, logística, log-logística, Gumbel.



**Ilustración 1-4:** Ajuste de distribuciones

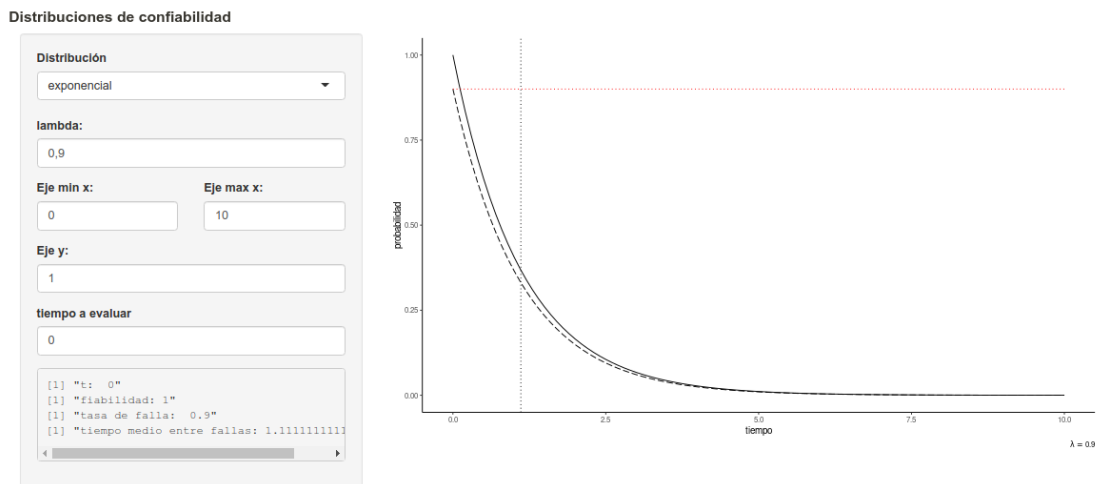
Realizado por: Jácome Darío, 2022

Los estadísticos que contrastan el ajuste a las distribuciones teóricas, son: Kolmogorov-Smirnov, Cramer- Von Mises, Anderson-Darling, IAC y BIC

#### 4.1.5. Cálculo fiabilidad, tasa entre fallos y MTBF

Dada los parámetros de distintas funciones de probabilidad, la aplicación calcula la fiabilidad, tasa de falla y tiempo medio entre fallas. En la ilustración 2-4 se muestra la distribución exponencial con  $\lambda=0.9$

#### Entradas

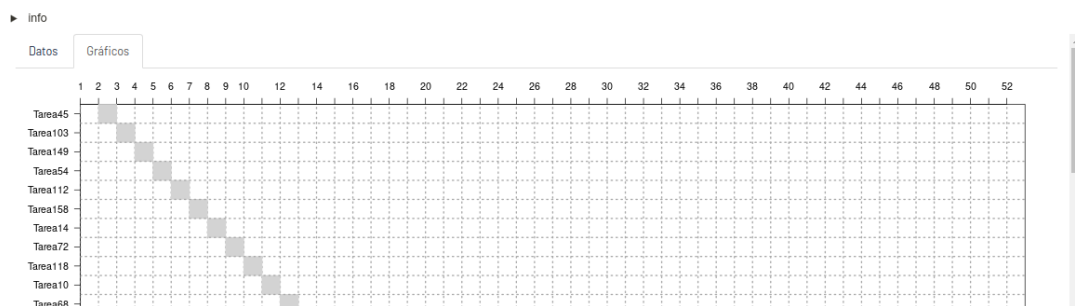


**Ilustración 2-4:** Cálculo fiabilidad, tasa entre fallos y MTBF

Realizado por: Jácome, Darío, 2021

#### 4.1.6. Cronograma anual

Dadas el tiempo y la frecuencia para cada tarea, el aplicativo sugiere la semana en la que se debe realizar dichas tareas, esto basado en que las tareas que se realizan continuamente y las que tienen mayor duración tienen prioridad.

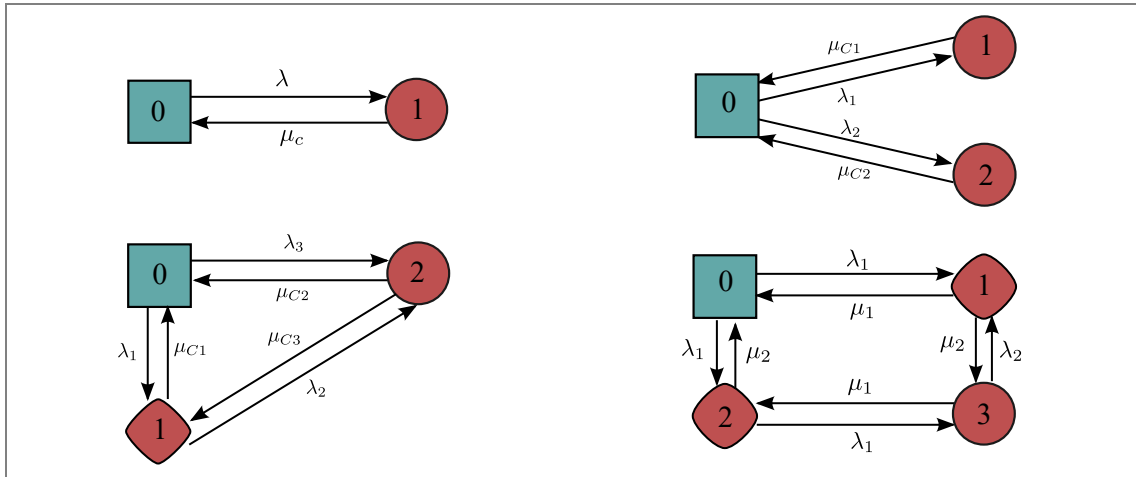


**Ilustración 3-4:** Cronograma de mantenimiento

Realizado por: Jácome Darío, 2022

#### 4.1.7. Markov tiempo continuo

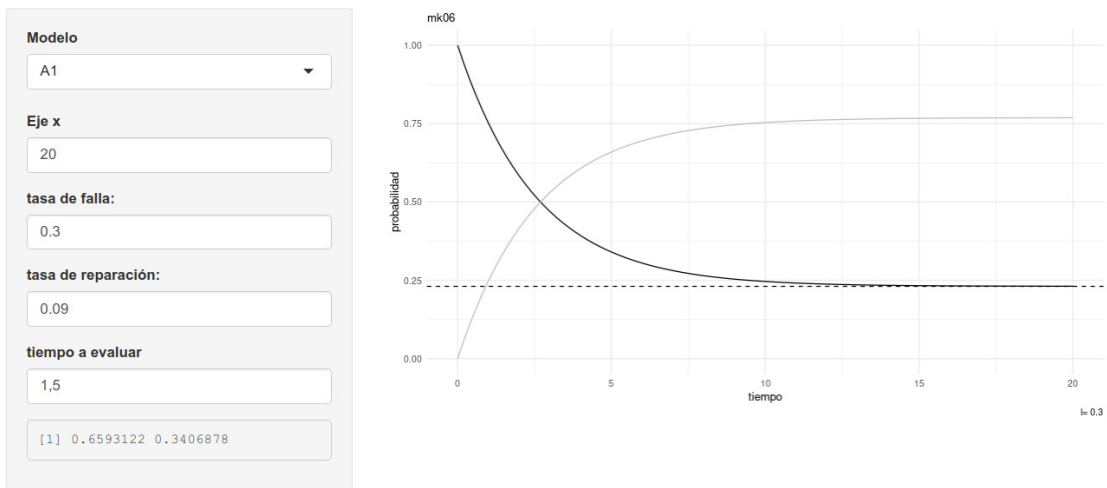
Según las tasas de reparación y de falla, calcula la disponibilidad del equipo, para un sistema redundante.



**Ilustración 4-4:** Diagrama de Markov

Realizado por: Jácome Darío, 2022

Las tasas de falla se representan con la letra griega “*lambda*” y la de reparación con la letra griega “*mu*”, en la ilustración 5-4 se muestra una tasa de falla de 0,03 y 0,02 en cambio para tasas reparación: 0.1 y 0.4

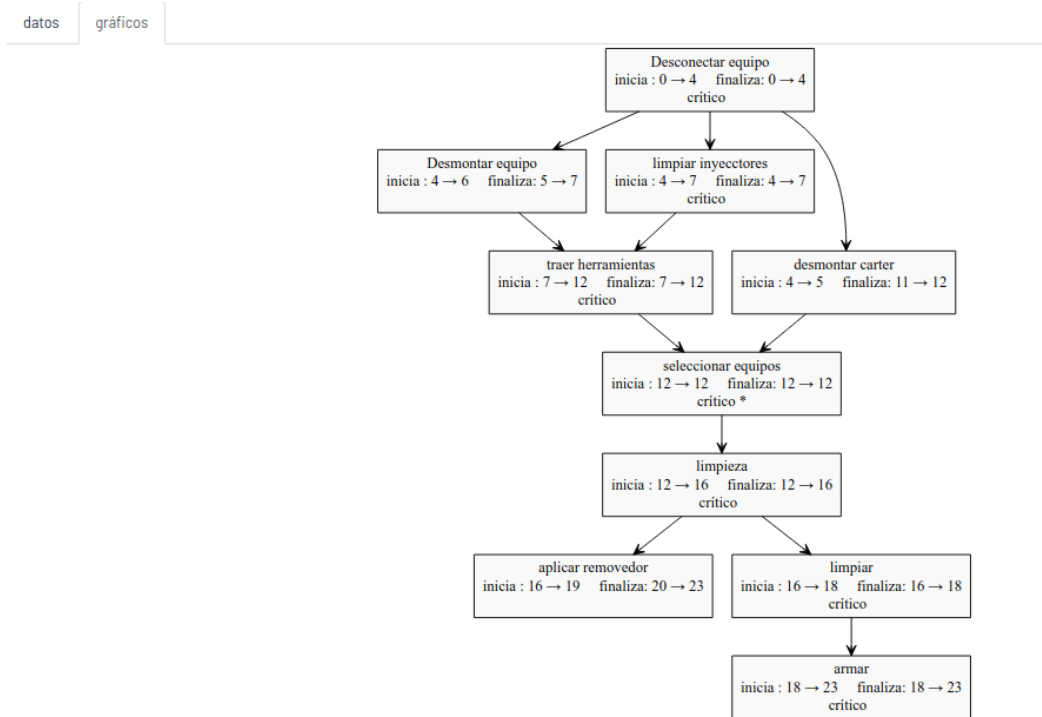


**Ilustración 5-4:** Markov tiempo continuo

Realizado por: Jácome Darío 2022

#### 4.1.8. Método de la ruta crítica en R

Se ingresan los valores de tiempo en valores enteros, y para colocar su posición los datos deben estar ordenados como grafos.



**Ilustración 6-4:** Método de la ruta crítica

Realizado por: Jácome Darío, 2022

#### 4.1.9. Número de inspecciones óptimo

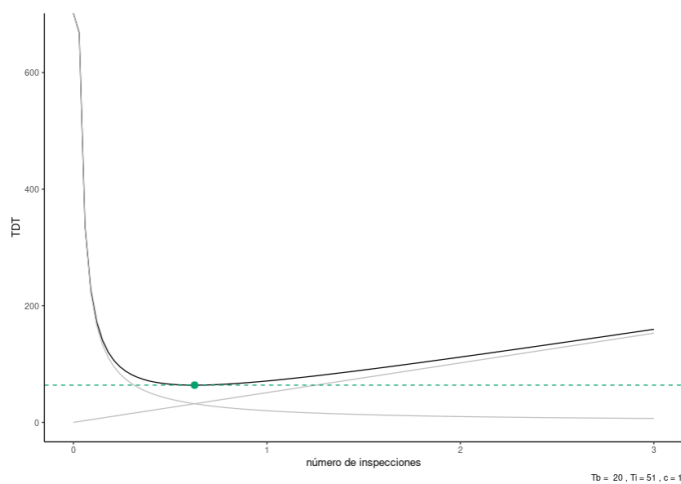
constante c:

n. inspecciones :

tb:  ti:

```

    [1] "n. inspecciones 0.626224291085
    [1] "tiempo empleado 63.87487769068
  
```



**Ilustración 7-4:** Tiempo óptimo para la inspección

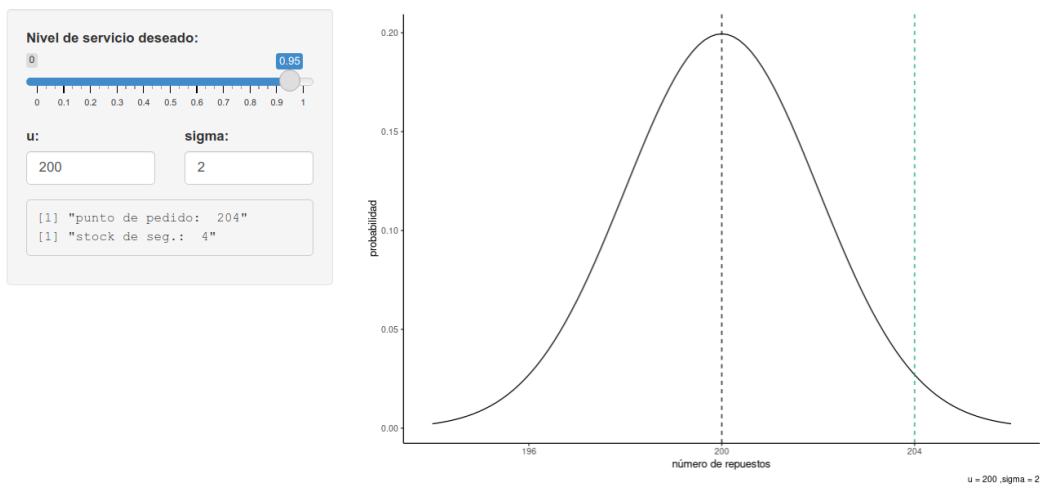
Realizado por: Jácome Darío, 2022

Se ingresa el tiempo de parada “tb” y el tiempo de inspección “ti” para calcular el número óptimo de inspecciones. La constante “c”, establece la calidad de la inspección.

Dado un número de inspecciones de 3 al mes, con un tiempo para inspección de 51 min y tiempo de parada de 20 el número de inspecciones necesarias corresponde a 0.62 aproximando se n

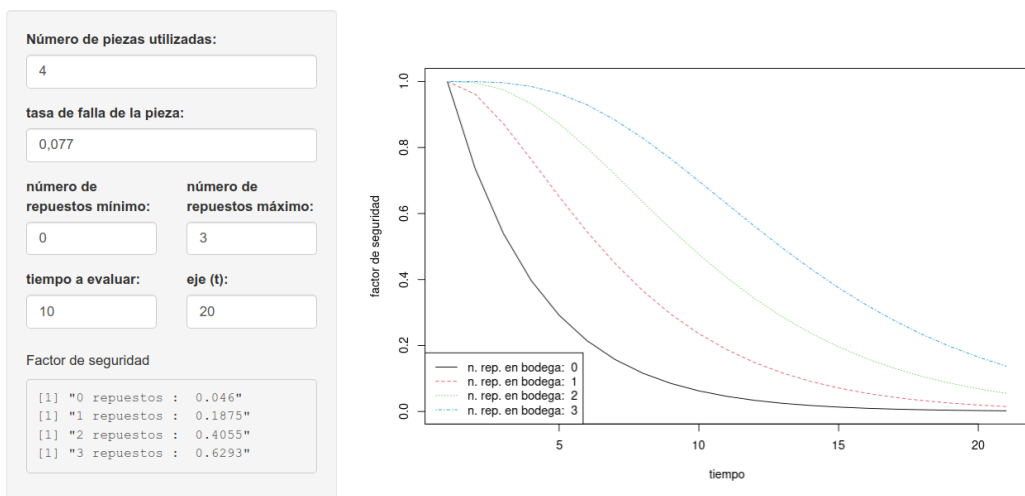
#### 4.1.10. Stock de seguridad y factor de seguridad

La ilustración 7-4 muestra que el stock de seguridad es de 4 unidades para el cambio de empaquetaduras de pistones con una media de 200 unidades y una desviación estándar de 2 unidades con una probabilidad de 95% que se encuentre disponible.



**Ilustración 8-4:** Punto de pedido y stock de seguridad

Realizado por: Jácome Darío, 2022



**Ilustración 9-4:** Factor de seguridad

Realizado por: Jácome Darío, 2022

Si se requiere de un conjunto de repuestos para que la bomba tríples funcione correctamente, y si se dispone varios de estos elementos en bodega, se puede obtener el factor de seguridad de la pieza en cuestión, ver ilustración 9-4.

Dada la tasa de fallas de 0.077 en una bomba que necesita 4 repuestos y que se tenga 3 de reserva, la probabilidad de que esté disponible es del 0.63.

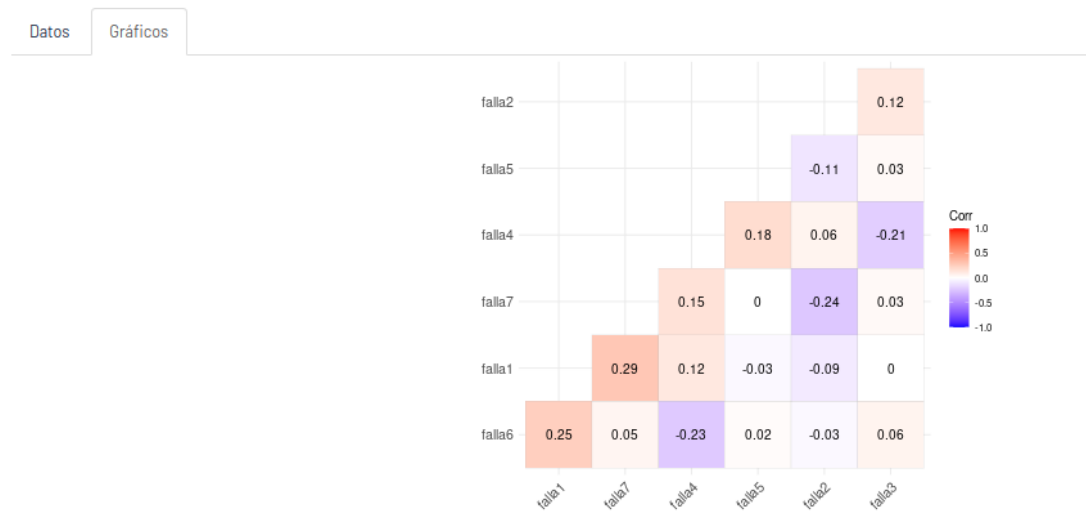
#### 4.1.11. Análisis de correlación aplicado a datos de fallas.

Se ingresa valores en una hoja de cálculo las fallas ordenas en columnas, de esta forma se obtiene el valor de la correlación para cada uno de los datos.

**Tabla 4-3:** Costo de alquiler de la infraestructura del despliegue de la aplicación

Falla 1	Falla 2	Falla 3	Falla 4	Falla 5	Falla 6	Falla 7
201	30	244	102	201	12	120
102	34	192	202	403	42	253
302	32	105	103	39	22	210
120	51	195	402	22	41	512

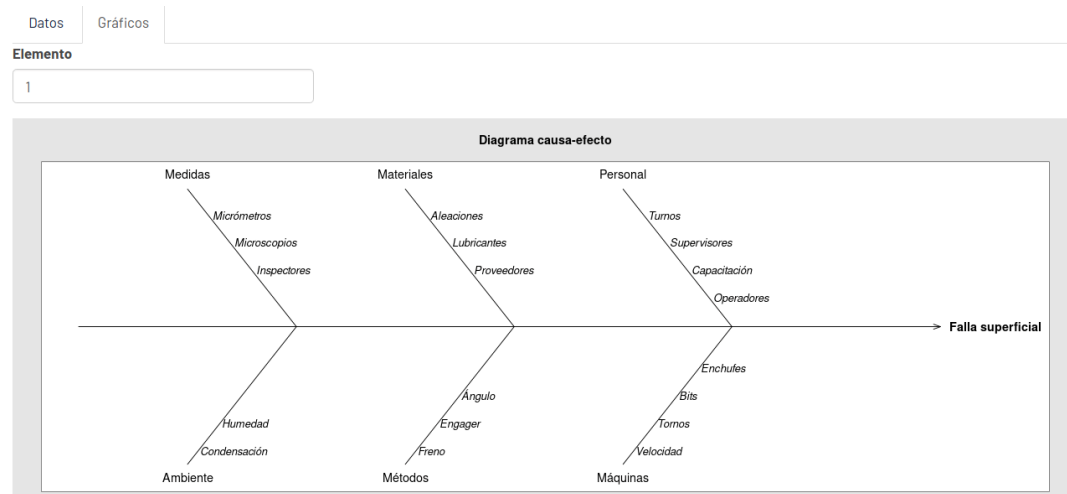
Realizado por: Jácome Darío, 2022



**Ilustración 10-4:** Análisis de correlación

Realizado por: Jácome Darío, 2022

#### 4.1.12. Diagrama causa – efecto



**Ilustración 11-4:** Diagrama causa - efecto

Realizado por: Jácome Darío, 2022

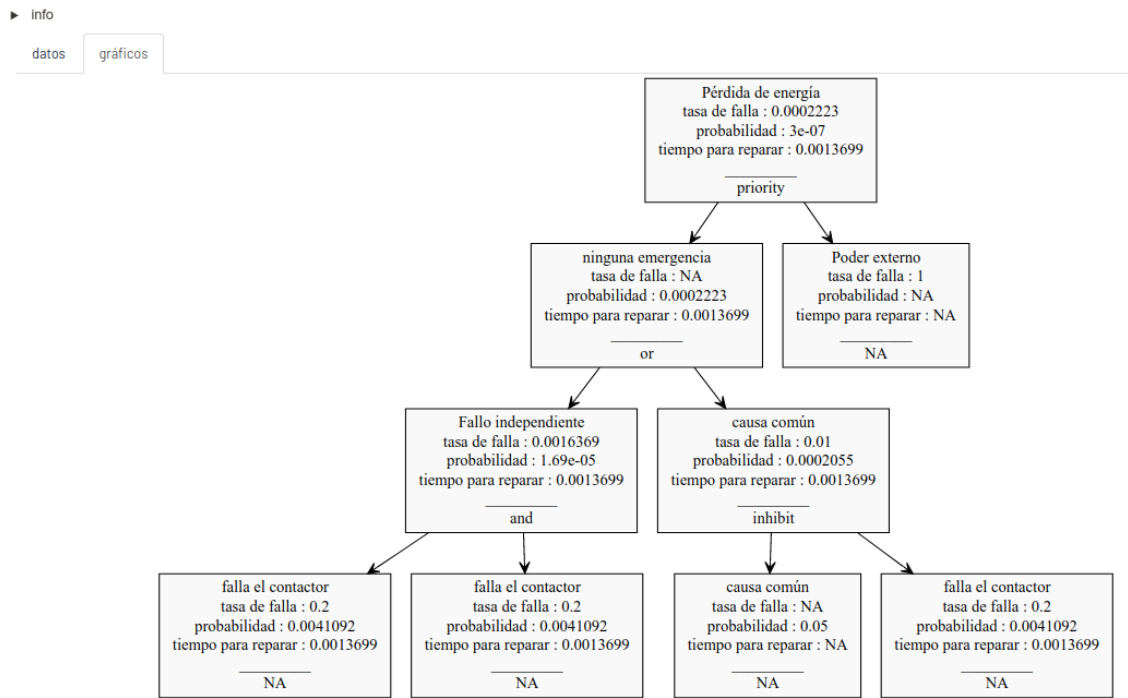
Se ingresa tres filas de campos causa, efecto y un campo adicional para clasificar las causas y se obtiene una gráfica como en la ilustración 11-4.

#### 4.1.13. Árbol de fallas

Se ingresa los valores de tiempo medio entre fallo y tiempo medio para reparar a fin de calcular la probabilidad de falla de un componente.

Las fallas deben estar organizadas como grafos.

Consta de tres compuertas, AND, OR e INHIBIT, esta última es un condicionante, si se supera cierto valor se tomará en valor de probabilidad de la otra falla.



**Ilustración 12-4:** Árbol de fallas

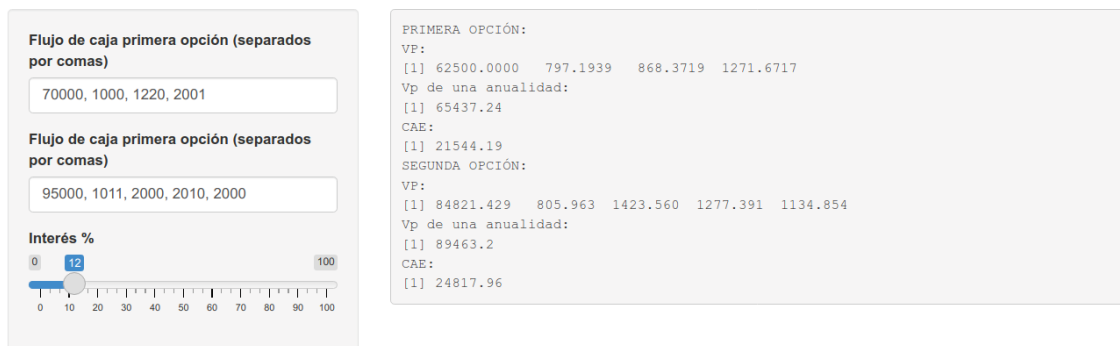
Realizado por: Jácome Darío, 2022

#### 4.1.14. Costos de mantenimiento

Esta aplicación calcula el valor presente, el valor presente de una anualidad y el costo anual equivalente a partir del flujo de caja considerando un 8% como tasa de descuento.

Esto permite comparar entre dos opciones y determinar cuál inversión es mejor o no.

En la ilustración 12-4 se muestra la comparación de dos flujos de caja por la compra y mantenimiento de una bomba tríplex por medio del costo anual equivalente.



**Ilustración 13-4:** Costos de mantenimiento Valor presente

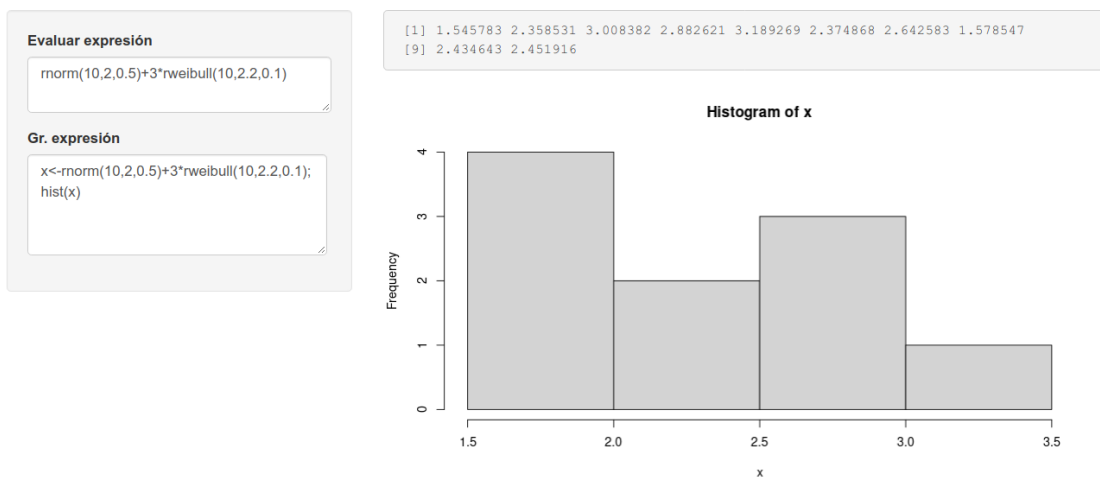
Realizado por: Jácome Darío 2022



En la ilustración 12-4 se muestra el flujo de caja para dos bombas tríples de las MTU, con un costo de 70000 y otra de 95000, con sus respectivos costos de mantenimiento, de la cual el CAE de la primera es menor, por este motivo se seleccionaría la primera bomba.

#### 4.1.15. Calculadora y registros de código

A fin de realizar los siguientes cálculos para la gestión de mantenimiento de las M.T.U se ha habilitado una aplicación en la cual se puede ejecutar código en R, como se muestra en la ilustración 14-4.



#### Ilustración 14-4: Calculadora simple en R

Realizado por: Jácome Darío, 2022

Los listados de código se pueden almacenar en distintos bloques de texto y de código como se muestra en la ilustración 15-4. De este modo, se puede registrar el uso de diferentes cálculos para utilizarlos posteriormente.

Esto ayuda al entendimiento de cada una de las entradas de cada fórmula utiliza, además se muestra cada proceso sin el código de la interfaz del usuario a fin de que se pueda ejecutar el código en R y no sea un proceso de *caja negra*, es decir que el proceso no sea oculto.

Toda la documentación se agregó de esta forma en la aplicación notionl o que dá la posibilidad de mejorar dicho texto.

La aplicación utilizada facilita el procesamiento de texto, de esta forma se podrá analizar grandes volúmenes de texto.

**Tasa de fallas efectivas aproximadas para sistemas redundantes con mantenimiento correctivo**

Ecuaciones en código Latex

$$\lambda_{se} = \frac{\lambda_1 \lambda_2 [(\mu_1 + \mu_2) + (\lambda_1 + \lambda_2)]}{\mu_1 \mu_2 + (\mu_1 + \mu_2) (\lambda_1 + \lambda_2)}$$

Compuesto por dos unidades independientes y no idénticas en paralelo

$\lambda_{se}$  = tasa de falla efectiva del sistema paralelo de dos unidades,  
 $\lambda_i$  = tasa de falla de la unidad i, para i = 1, 2,  
 $\mu_i$  = unidad i tasa de mantenimiento correctivo, para i = 1, 2.

Listado de código en R

```

l1 <- 0.004; u1 <- 0.005; l2 <- 0.002; u2 <- 0.003
lse <- l1*l2*((u1+u2)+(l1+l2))/(u1*u2+(u1+u2)*(l1+l2))
lse
#-----
plot(c(lse, l1, l2))

```

**Ilustración 15-4:** Anotaciones

Realizado por: Jácome Darío, 2022

**4.2. Costos de infraestructura del aplicativo**

Las herramientas desarrolladas en su mayoría en lenguaje R, se puede desplegar en un servidores con el software *shiny server*, que nos permite visualizar las aplicaciones por medio de la web. A su vez, está necesita de un dominio, es decir el nombre de la página y un certificado SSL. Estos servicios se los puede adquirir en los servicios de AWS. Ver anexo A

**Tabla 4-4:** Costo de alquiler de la infraestructura del despliegue de la aplicación

Descripción	Costo mensual (USD)	Costo anual (USD)
DOMINIO	1,00	36,00
1 RAM ACPU	3,19	114,84
ALMACENAMIENTO 24G	2,4	86,40
TOTAL (USD)	6,60	234,40

Fuente: AWS

Realizado por: Jácome Darío, 2022

Por motivo de que las herramientas desarrolladas para la gestión no recopilan datos y funciona como un servicio, es posible implementarlo en la web, en donde no se comprometen ningún tipo de información.

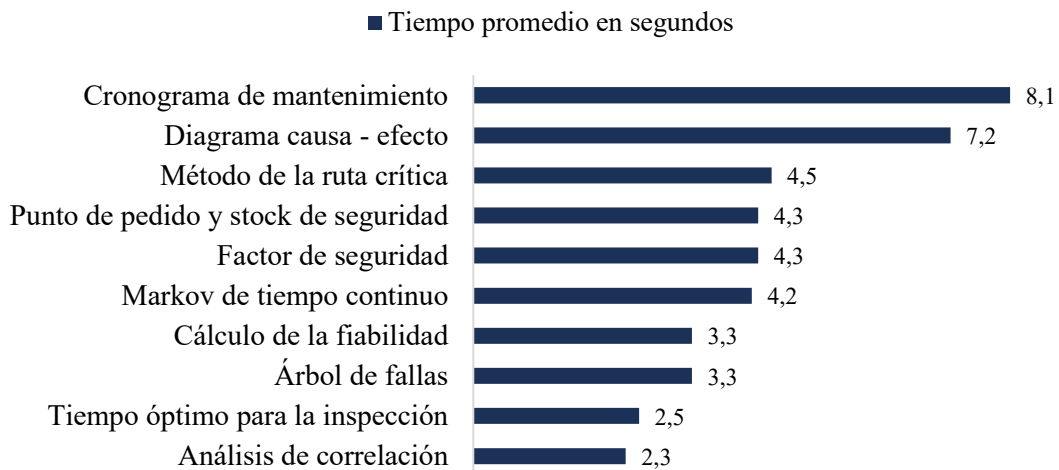
#### 4.2.1. Escalabilidad de la aplicación

La plataforma utilizada para el desarrollo Amazon web Service se elige por su escalabilidad, es decir que si se necesitan más recursos como más número de núcleos del procesador se puede pasar a otra instancia sin problemas.

Los archivos necesarios para el despliegue de la aplicación se encuentran en la página web [www.mantenimientoactivo.com](http://www.mantenimientoactivo.com), esto hace posible la modificación de la aplicación, y la continua mejora de la aplicación.

##### 4.2.1.1. Tiempo de respuesta de la aplicación

La apertura de la aplicación tarda unos 4.2 segundos



**Ilustración 16-4:** Tiempo necesario para la apertura de las aplicaciones

Realizado por: Jácome Darío, 2022

#### 4.3. Capacitación del manejo de la aplicación al personal de la empresa

La capacitación consistió en la utilización y modificación de las bases de datos, así como la utilización de las diferentes herramientas creadas para la gestión de la MTU. Además, se proporcionó un video para la utilización del software.



## CONCLUSIONES

Los modelos seleccionados para el análisis de confiabilidad fueron, Weibull, normal, logística, Gumbel, log-normal y log-logística y modelos de Markov. En cambio, para la gestión de tareas, se incorporó el método de la ruta crítica y para el análisis de costos en costo anual uniforme equivalente.

La información necesaria para la gestión de las M.T.U. fue realizada con la aplicación *Notion*, en cambio, el procesamiento de esta información fue desarrollada con el *software shiny-sever*.

La implementación se la hizo en una máquina virtual AWS EC2, utilizando shiny-server para desplegar las herramientas de gestión desarrolladas con un coste en dólares de 237,40 para tres años, es decir alrededor de 6.60 por mes. Este costo solo representa el pago por la infraestructura dado que solo se utilizó Rstudio para la creación de la aplicación y no se requirió de ningún software comercial.

En síntesis, la aplicación es una plantilla en *notion* que incorpora diferentes aplicaciones embebidas para realizar los diferentes cálculos referentes a fiabilidad, logística y costos desarrolladas en lenguaje R, disponibles en la página web [mantenimientoactivo.com](http://mantenimientoactivo.com)

Los informes desarrollados con las herramientas para la gestión desarrolladas cuentan con datos de componentes disponibles en el mercado o presenta un ejemplo práctico con componentes de las MTU.

## RECOMENDACIONES

Los modelos matemáticos para el análisis pueden ser utilizados para diferentes ámbitos, como por ejemplo equipos, personal por este motivo se recomienda incorporara estos modelos en aplicaciones de manera genérica, es decir que se pueda utilizar para varios ámbitos.

La implementación de las aplicaciones en lenguaje R pueden ser implementados mediante contenedores “*Docker*” en vez de máquinas virtuales

Se recomienda investigar acerca de árbol de eventos, Jack-nife, redundancia modular y selección de estrategias de mantenimiento.

Se sugiere a la empresa incluir en sus registros el modelo de confiabilidad que más se ajuste a las diferentes partes de las MTU.

Se recomienda realizar un diagrama en donde consten los estados y configuraciones de las partes del equipo, también realizar diagramas del flujo de datos para facilitar el desarrollo de las herramientas de gestión.

Se recomienda a la institución incluir en sus programas de estudio, el manejo de la gestión de mantenimiento mediante software

## GLOSARIO

**Amazon Elastic Compute Cloud (Amazon EC2):** Servicio web para iniciar y administrar de Linux/UNIX y Windows

**Dirección IP:** Dirección numérica (por ejemplo, 192.0.2.44) que los dispositivos de red utilizan para comunicarse entre sí mediante el protocolo de Internet (IP). A todas las Instancia EC2 se les asignan dos direcciones de IP en el momento del lanzamiento. Estas direcciones se asignan directamente una a otra mediante la traducción de direcciones de red (NAT): una dirección IP privada (conforme con RFC 1918) y una dirección IP pública. A las instancias que se lanzan en una VPC se les asigna solo una dirección IP privada. A las instancias que se lanzan en su VPC predeterminada se les asigna una dirección IP privada y una dirección IP pública.

**Instancia:** Una copia de una imagen de la máquina de Amazon (AMI) Se ejecuta como un servidor virtual en el directorio nube de AWS.

**Amazon Simple Storage Service (Amazon S3):** Almacenamiento para Internet. Se puede usar para almacenar y recuperar cualquier cantidad de datos en cualquier momento y desde cualquier parte de la Web.

**CNAME:** *Canonical Name Record*, registro de nombre canónico. Tipo de registro de recursos del Sistema de nombres de dominio (DNS) que especifica que el nombre de dominio es un alias de otro nombre de dominio canónico. En concreto, es una entrada de una tabla DNS que puede utilizar para asignar un nombre de dominio completamente calificado como alias a otro.

**Par de claves:** Conjunto de credenciales de seguridad que se usa para demostrar la identidad electrónicamente. Un par de claves consta de una clave pública y una clave privada.

**Docker:** Docker le proporciona una manera estándar de ejecutar su código. Docker es un sistema operativo para contenedores. De manera similar a cómo una máquina virtual virtualiza (elimina la necesidad de administrar directamente) el hardware del servidor, los contenedores virtualizan el sistema operativo de un servidor. Docker se instala en cada servidor y proporciona comandos sencillos que puede utilizar para crear, iniciar o detener contenedores.

## BIBLIOGRAFÍA

**AISHA, H.S.**, *Markov Decision Model For Maintenance Problem Of Deteriorating Equipment With Valueiteration* [en línea]. Bida-Nigeria: IOSR-JM, 2017. [Consulta: 20 diciembre 2022].

Disponible en: <https://iosrjournals.org/iosr-jm/papers/Vol13-issue5/Version-3/O1305038893.pdf>.

**ANTONOV, A.V., GALIVETS, E.Yu., CHEPURKO, V.A. y CHERNIAEV, A.N.**, *Fault tree analysis in the R programming environment* [en línea]. Dependability, ISSN 2500-3909, 1729-2646. DOI 10.21683/1729-2646-2018-18-1-4-13. [Consulta: 3 noviembre 2022]. Disponible en: [https://www.researchgate.net/publication/323699519\\_Fault\\_tree\\_analysis\\_in\\_the\\_R\\_programming\\_environment](https://www.researchgate.net/publication/323699519_Fault_tree_analysis_in_the_R_programming_environment)

**CRAWLEY, M.J.**, *The R book* [en línea]. Chichester-England: Wiley. 2007. ISBN 978-0-470-51024-7. [Consulta: 1 septiembre 2022]. Disponible en: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118448908>

**DHILLON, B.S.**, *Engineering Maintenance: A Modern Approach*. USA: CRC Press, 2002. ISBN 978-0-429-13220-9, pp. 102-206.

**GANO D.L.** *RealityCharting© Seven Steps To Effective Problem-Solving And Strategies For Personal Success*. Richland, Washington – USA: Apollonian Publications, 2011. ISBN 978-1-883677-14-5, pp. 43-50.

**IEEE-1413.1.** IEEE Guide for Selecting and Using Reliability Predictions Based on IEEE 1413

**JIN T.** *Reliability Engineering and Services*. Hoboken - USA : John Wiley & Son, 2019. ISBN: 978-1-119-16701-3, pp. 59-65.

**HBK**, *Life Data Analysis Reference Book - ReliaWiki*. [en línea], 2021. [consulta: 26 noviembre 2021]. Disponible en: [https://www.reliawiki.com/index.php/Life\\_Data\\_Analysis\\_Reference\\_Book](https://www.reliawiki.com/index.php/Life_Data_Analysis_Reference_Book).

**PHAM H.** *Handbook of reliability engineering* [en línea]. London -New York: Springer. ISBN 978-1-85233-453-6. Disponible en: <https://link.springer.com/book/10.1007/b97414.TA169.H358.2003>



**RINNE H.** *The Weibull distribution: a handbook*. New York. Boca Raton – Estados Unidos: CRC Press. 2008. ISBN 978-1-4200-8743-7, pp. 27-53.

**TRIPOINT.** Bombeo hidráulico [en línea], 2021. [consulta: 29 noviembre 2021]. Disponible en: <https://www.tripoint-ec.com/esp/servicios2.html>.

**UNE-EN 15341: 2008.** *Indicadores clave de rendimiento del mantenimiento*

**VIVEROS, P., STEGMAIER, R., KRISTJANPOLLER, F., BARBERA, L. y CRESPO, A.,** *Propuesta de un modelo de gestión de mantenimiento y sus principales herramientas de apoyo. Ingeniare. Revista chilena de ingeniería*, 2013. vol. 21, no. 1, ISSN 0718-3305. DOI 10.4067/S0718-33052013000100011

**YE, Y. GROSSMANN, I.E., PINTO, J.M. y RAMASWAMY, S.,** *Modeling for reliability optimization of system design and maintenance based on Markov chain theory. Computers & Chemical Engineering*, 2019. vol. 124, ISSN 00981354. DOI 10.1016/j.compchemeng.2019.02.016

**ZIO E.** *An introduction to the basics of reliability and risk analysis*. Singapore, Hackensack, N.J: World Scientific. Series on quality, reliability & engineering statistics, 2007. ISBN 978-981-270-639-3, pp. 77-126.

**ZIO E.** *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. London: Springer London. Springer Series in Reliability Engineering, 2013. ISBN 978-1-4471-4587-5, pp. 192-193.

## ANEXOS

### ANEXO A: DESPLIEGUE DE APLICACIONES EN R

#### Introducción a R

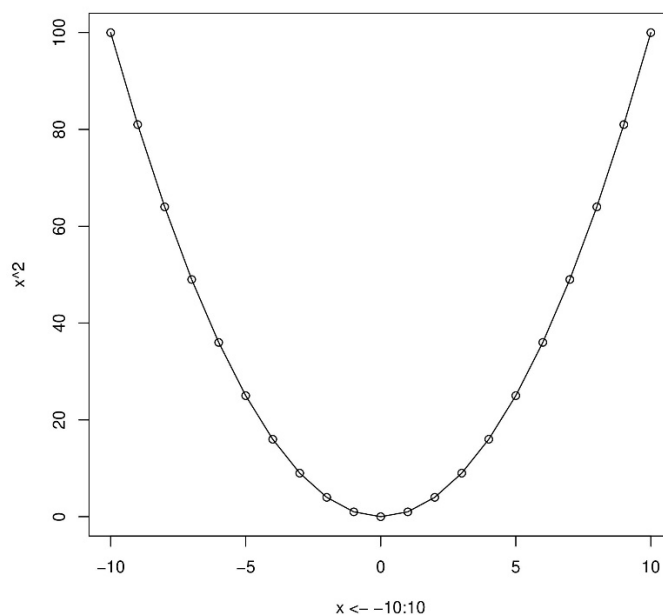
En la consola de Rstudio cada instrucción se ejecuta al presionar *enter*. Las instrucciones son ignoradas si se antepone el signo numeral (#) antes del código, esto es un comentario y si se desea parar la ejecución se presiona Ctrl + c.

Los valores se guardan con el operador "<-" por ejemplo `x <- 6`, del mismo modo para dos valores: `x1 <- x2 <- 4`, lo cual guardaría el valor, pero no lo mostraría; Para mostrarlo se coloca entre paréntesis (`x <- 6`). Además de que el operador puede ir dentro de las funciones, también se puede ocupar para guardar diferentes objetos así `x <- sin()`, en este caso estamos guardando la función seno en la variable `x`

Para colocar más de una instrucción en una sola línea, éstas se separan con punto y coma `a <- 2; b <- 3`.

Se destaca que al ejecutar las líneas de código, éstas se muestran con "##" y [1], lo que representa el primer elemento del resultado.

```
plot(x<- -10:10, x^2, type="o")
```



La instalación de Instalar librerías aumenta más el número de funciones a nuestro entorno de trabajo se lo hace mediante un comando, por ejemplo, `install.packages("ggplot2")`. Una vez instalado, es necesario cargarlo a nuestro entorno de trabajo, para el ejemplo resulta `library(ggplot2)`

Una de las librerías ampliamente utilizadas es `magrittr`, lo que hace es que el código sea más legible.

```
x %>% f()      # equivale a f(x)
x %>% f(y)     # equivale a f(x,y)
x %>% f(y,#)   # equivale a f(y,x)
x %>% f(y,z=#) # equivale a f(y,z=x)

%>% # pipe. Canaliza valores hacia el lado derecho.
%T>% # tee pipe. Canaliza y devuelve los valores originales en vez del resultado
%<>% # assignment pipe. Canaliza los valores y asigna los valores
```

```
library(magrittr)
iris[1:3] %T>% pairs %>% summary
```

Múltiples valores se pueden guardar de la siguiente forma `x <- c(3,4,1)` donde la letra `c` hace referencia a una columna de datos, que comúnmente se le conoce como vector (el cual no hace referencia al vector utilizado en la Física). Todos los datos del vector deben ser del mismo tipo (numérico en este caso) y que el vector no tiene dimensiones.

**Lista.** Cuando se requiera guardar varios tipos de valores; que puede ser número o texto, se procede de la siguiente manera `x <- list("ene",31)`. es decir, que hemos ocupado una lista para asignarle valores a la variable "x".

**Matriz.** Una matriz es otro tipo de objeto y se crea a partir de un vector del cual se distribuye primero en columnas y luego en filas por ejemplo el vector 1:6 distribuido en 3 columnas y 2 filas se escribe así `m <- matrix(1:6,3,2)`. Note que el vector va de 1 hasta 6 con paso 1 y en el caso que se hubiese querido distribuir primero por filas se añade a las opciones `byrow=TRUE`. También se puede escribir de la siguiente manera `m <- matrix(1:6,ncol=3,nrow=2)`. Otra forma de crear una matriz es utilizando la función `cbind(...)` en donde se pueden colocar varios vectores. La matriz resultante será de un solo tipo, es decir o solo numérico o solo texto, si existe uno varios vectores de tipo texto la función convertirá los demás valores en texto, y si se pone un solo valor este se replicará hasta completar la columna, por ejemplo, en `m <- cbind(1:4, 2.1)` el valor 2.1 se replicará 4 veces. En los resultados las filas tienen se identifican de la siguiente forma `[n, ]` y

las columnas [,n] por ejemplo si necesitamos los valores de la segunda columna se lo hace con `m[,2]`.

**Array.** Si se necesita crear estructuras de más de dos dimensiones, de igual que las matrices puede formar a partir de un vector, para luego distribuirlos como se requiera. La primera matriz de la tercera dimensión se representa en la consola con `[, ,1]` la segunda con `[, ,2]` y así sucesivamente, en cambio la primera matriz de la cuarta dimensión se muestra cómo `[, ,1,1]`, la segunda matriz como `[, ,2,1]` y así sucesivamente.

En el siguiente ejemplo se distribuye el vector `1:30` en 4 dimensiones. `a <- array(seq(30), dim = c(3,5,2,1))`. Una vez creado si se desea mover los datos de una dimensión a otra y ocupa `aperm` donde `aperm(a,c(1,2,3,4))` devolvería el mismo array, y si se cambia por `aperm(a,c(2,1,3,4))` esto intercambiaría las filas por columnas.

**Aproximar.** Algunas veces el resultado contiene muchos decimales. Si se desea aproximar  $x$  a un decimal se escribiría `round(x,1)`, a dos cifras significativas para redondear `signif(x,2)`. En cambio, `floor(x)` aproxima a un inmediato inferior y `ceiling(x)` aproxima al inmediato superior, ambos devolviendo un número entero.

**Concatenar caracteres.** Para concatenar dos valores se lo hace de la siguiente forma `cat("A",2)` cuyo resultado es `A2`. En cambio, `paste0` junta los vectores uno a uno, y si los vectores no son iguales, completa como si fuera una serie, ejemplo:

Para unir los vectores `c("A","B","C")` con `c(1,2)` devolvería `"A1" "B2" "C1"` en cambio si se necesita concatenar con un separador "-" se utiliza `paste()` con opciones `sep = "-"` y devolvería `"A-1" "B-2" "C-1"`. Esta función por defecto utiliza un espacio como separador. No obstante, si se requiere un solo elemento como resultado, se coloca en las opciones `collapse = ""`.

Los operadores más habituales son `[+ - * /]` suma, resta, multiplicación y división

Al momento de utilizar funciones trigonométricas, siempre se escribe sus valores en radianes y si por ejemplo se necesita conocer el seno de  $30^\circ$  se escribiría así: `sin(30*pi/180)`, y si se desea omitir la constante  $\pi$  quedaría así: `sinpi(30/180)` en ambos casos el resultado es 0.5. Lo mismo aplica para el coseno y tangente.

Para recordar en nombre de las funciones hiperbólicas se le agrega "h" después de la función, y a las funciones inversas se la antepone una "a" que equivale al arco; resultando: `sinh(x)` y `asin(x)`.

Cuando se ejecuta el código, y si se trabaja con divisiones para cero; se pueden obtener valores como `Inf`, `-Inf` o cuando se ingresa un valor que no es admitido por la función se obtienen resultados como `NaN` que significa "Not a number", es decir, "no es un número". Por otra parte, `NA` significa "Not Available" lo que quiere decir, "no hay valores" y aparece comúnmente cuando la función no admite dicho valor. Los cuales podemos comprobar con las siguientes funciones: `is.nan(x)` `is.na()` los cuales devolverán un `TRUE` o `FALSE` según el caso.

**Polinomios.** Los polinomios se escriben en forma de vector por ejemplo `c(10, 3, 3)`, siguiendo el siguiente orden para encontrar sus raíces se ocupa la función `polyroot`.

El número complejo  $3 + 3i$  se escribe como `complex(real = 3, imaginary = 3)` o de manera simplificada `3+3i`. En cambio  $2\angle 40^\circ$  resulta `complex(modulus = 2, argument = 40*pi/180)`. Note que el argumento debe estar en radianes.

Se puede realizar operaciones con *arrays* como en los vectores, por ejemplo, el producto, por entre una matriz `m` y vector `x` quedaría `outer(x,m,"*")`, además se puede utilizar otras funciones como `sum()`, `min()`, etc.

Para crear un vector con nombres para cada elemento se escribe por ejemplo `x <- c("ene"=31, "feb"=27, "mar"= 30)`. En donde puedo seleccionar el primer elemento de `x["ene"]`, cuando no tenga un nombre se selecciona `x[1]`, y se puede seleccionar varios elementos como `x[1:2]`, es decir los dos primeros elementos, adicionalmente se puede utilizar una condición, ejemplo:

`x[x>=30]`, devolvería los elementos mayores o iguales a 30.

La manera de escribir se lo hace como en el siguiente ejemplo `x <- list(meses = c("ene", "feb", "mar"), dias = c(31,29,31))`. Y se puede acceder a su información mediante `x$meses` y `x$dias`.

Se le asigna el nombre mediante una lista en la opción `dimnames` por ejemplo `m <- cbind(0:4, 2.1, dimnames = l)` siendo `l` una lista con un vector para la fila y otro para la columna. Y si solo se necesitaba el nombre de las columnas `cbind("C1"=0:4,"C2"=4:8)` sería una mejor opción. Además, se puede asignar nombres una vez creada la matriz mediante `dimnames()<-`, `colnames()<-`, `rownames()<-`.

Al igual que en la matriz se le asigna los nombres con una lista.

## Programación en R

**Condiciones.** las condiciones como  $x \geq 1$  retorna un valor verdadero TRUE o FALSE, si la cumplen o no. También se pueden agrupar de la siguiente manera  $x < 1 \ \&\& \ x > 2$   $x$  es menor a uno y  $x$  es mayor a dos, es decir que devolverá TRUE si el valor está entre uno y dos sin incluir estos dos valores. Adicionalmente, se tienen las funciones `all()` que devuelve TRUE si todas las condiciones son verdaderas, y `any()` devuelve TRUE si al menos una condición es verdadera. En cambio, si se desea comparar vectores quedaría  $x > 1 \ \& \ x > 2$  donde  $x$  puede ser `c(1,2,4)`.

A continuación, un ejemplo simple con `&`

Cuando se requiera elegir entre una instrucción u otra según la condición dada, se escribe como en el siguiente ejemplo `ifelse(x==1, print("ON"), print("OFF"))` que imprime o muestra la palabra ON si la condición  $x==1$  es verdadera, si no lo es devuelve OFF. Otra estructura sería `if(x==1) print("ON") else print("OFF")`. Por el contrario, cuando se desea elegir una instrucción u otra, es decir entre la primera, segunda, etc. se escribe por ejemplo `switch(2, PRINT("LOW"), PRINT("NORM"), PRINT("HIGH"))`. En caso tener más instrucciones para ejecutar se puede agrupar como en el ejemplo a continuación:

```
x <- 0

if(x==1) {
  print("ON")} else {
  print("OFF")
}

## [1] "OFF"
```

Cuando se tenga la necesidad de repetir la misma instrucción varias veces por ejemplo deseo dividir 1 para 1, 2, 3 hasta 10, necesitaría el vector 1:10 un elemento  $i$  a la vez, es decir  $i < 1$ , luego 2, luego 3 y así sucesivamente por lo que se escribe `for(i in 1:10) print(1/i)`.

Cuando no trabaja con el número de iteraciones sino con una condición se escribe `while` y la condición en el paréntesis, la cual itera hasta que la condición sea falsa lo cual puede ser un problema cuando esta siempre se cumpla. No obstante, se puede colocar la instrucción `break` para salir del bucle o `next` para saltar al principio de la instrucción.

```

i <- 1

while (i<3){
  print(i)
  i <- i + 1
}

## [1] 1
## [1] 2

```

Utilizamos repeat para iterar varias veces sobre un bloque de código y break para salir del ciclo

```

i <- 1

repeat {
  print(i) ## [1] 2
  i <- i + 1
  if(i == 3) break
}

## [1] 1
## [1] 2

```

## Crear funciones

Las funciones pueden ser guardadas en una variable, por ejemplo, z <- min en el cual se puede utilizar "z" como una función así z(c(0,1,2,3)), lo que daría como resultado 0. Y si se desea crear una función propia por ejemplo mifuncion(x), se utiliza la siguiente estructura mifuncion <- function() y luego de esta, se escribe la instrucción de lo que requiere que haga, si se requiere más instrucciones se coloca entre corchetes {}.

```

mifuncion <- function(a, b) sqrt(a^2 + b^2)

mifuncion(3,4)
## [1] 5

```

En otro caso cuando el número de argumentos se variable, se utiliza los signos ... y se puede conocer su longitud con ...length(). Por el contrario, si se desea seleccionar un elemento de los argumentos en la posición n queda ...elt(n).

```

prom <- function(...) sum(...)/...length()

prom (0, 0.25, 0.75, 1)
## [1] 0.5

```

## Funciones de ejemplo

```
myexp <- function(x) {  
  n <- 0:100  
  sum(x^n/factorial(n))  
}  
  
# Ahora usemos La función:  
  
x <- 3  
exp(x) - myexp(x)  
  
## [1] 0
```

## Cadenas de texto

```
library(stringr)
```

Para unir cadenas de texto dispuesto en vectores utilizar `str_c`.

Para detectar si existe o no una cadena, contar, y la localización inicial y final de todas las coincidencias `str_detect`, `str_count`, `str_locate`, `str_locate_all`. Si se desea cambiar a minúscula, mayúscula y mayúscula en cada palabra respectivamente utilice `str_upper`, `str_lower` `str_title` `str_replace` reemplaza el texto por otro, pero solo en la primera coincidencia, en cambio para reemplazar todo utilice `str_replace_all` y si se desea cambiar una parte del texto según la posición o índice `str_sub()`<->. Si se requiere ordenar o encontrar los índices un vector: `str_sort` `str_order`. Finalmente, para duplicar o repetir cadena de texto se utiliza `str_dup("A",times=4)`

La función `ymd_hms("2017-11-28T14:02:00")` convierte la cadena o números a data-times, en donde las letras `ymd` representan en año, mes y días respectivamente y `hms` las horas minutos y segundos, se puede obtener una gran cantidad de funciones, variando estas letras, por ejemplo: `ymd_hm()` y `mdy()` según lo que se requiera. De igual forma en la que se puede varias las funciones se puede ingresar una gran variedad de formatos de fechas como:

```
2018-01-31 11:59:59  
2017-11-28T14:02:00  
11/28/2017 1:02:03  
1 Jan 2017 23:59:59  
July 4th, 2000  
4th of July '99  
4th of July 99
```



Para conocer el año, mes, día, hora, minuto y segundo se utiliza: year, month, hour, minute, second.  
Además, se puede optar por el prefijo mili, micro, nano, pico así: milliseconds()

Para operar con fechas se dispone de tres clases.

Periodos: years, months, days, hours, minutes, seconds

Duraciones: dyears, dmonths, ddays, dhours, dminutes, dseconds

Intervalos: interval(ymd("2017-01-01"), ymd("2017-02-11"))

Periodo, en una fecha específica en el tiempo, duración es un rango e intervalo es la referencia a dos fechas.

```
require(readr)
## Loading required package: readr
write_csv(iris, file = "iris.csv")
```

## Aplicaciones shiny

```
library(shiny)
```

En el siguiente ejemplo se ocupa el diseño del esquema FluidRow y dentro de esta función están los widgets de entrada y salida de datos, que vendrían a ser las entradas de los textos, números y datos en general. A cada widget se le asigna un identificador para poder referenciarlo en el procesamiento de los datos los cuales se muestran con la función renderPlot.

**Entrada de datos.** Una entrada numérica simple o con selector se puede formar con numericInput y sliderInput respectivamente, y se escribe de la siguiente manera:

```
numericInput("num1", min = 1, max = 10, value = 7)
```

En cambio, selectInput, checkboxGroupInput, radioButtons que son cajas de selección simple, casillas de selección simple y múltiple se utiliza la siguiente forma: selectInput("sel", choices = c("Lunes"="L", "Martes"="M", "Miércoles"="X"))

Adicionalmente un conjunto de variables, entrada de texto, fecha y rango de fecha de la siguiente manera:

```

varSelectInput("variable", mtcars)

textInput("txt1")

dateInput("dt1", value="2019-01-15", language="es", weekstart=1)

dateRangeInput("dtr1", start="2017-01-01", end="2019-12-31",
  separator="hasta", language="es", weekstart=1)

```

En los ejemplos anteriores la primera casilla es el identificador de cada objeto.

```

actionButton, actionLink, checkboxGroupInput, checkboxInput, fileInput,
passwordInput, selectizeInput, submitButton

```

## Salida de datos

A cada salida le corresponde un tipo de renderizado, por ejemplo:

```

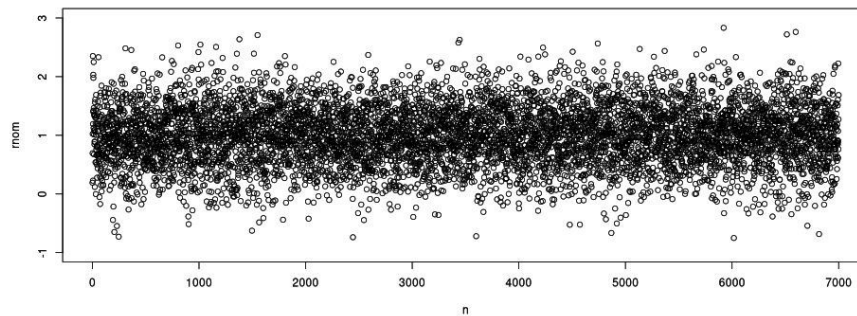
plotOutput - renderPlot
imageOutput - renderImage
tableOutput - renderTable
dataTableOutput - renderDataTable
textOutput - renderText
verbatimTextOutput - renderPrint
htmlOutput - renderUI
uiOutput - renderUI

```

```

fluidRow(
  sliderInput(inputId = "selector", # input$selector
    label = "valores para n de 1 a 10mil",
    value = 7000,
    min = 1,
    max = 10000),
  plotOutput(outputId = "grafico") # output$grafico
)
output$grafico <- renderPlot({
  plot(rnorm(input$selector, mean = 1, sd = 0.5),
    ylim = c(-1,3),
    xlab = "n",
    ylab = "rnom")
})

```



## Instalación de AWS en el servidor

### Paso 1. Crear el primer servidor EC2

Una vez verificados y con el usuario y contraseña que hemos dado accedemos a la plataforma y buscamos en la pestaña de servicios la opción EC2.

### Paso 2. Elegir sistema operativo AMI

En Amazon los llaman AMI (*Amazon Machine Image*) que básicamente son sistemas operativos preinstalados

Uno de las AMI más sencillas y robustas en la de Ubuntu Server, que está en varias versiones. Se selecciona en la más moderna Ubuntu Server 20.04 LTS.

Una vez elegido Ubuntu, hay que seleccionar en la siguiente pantalla las características del ordenador asignado, digamos el cuerpo del PC en la nube. Marca por defecto t2.micro que es la opción gratuita y tiene una CPU con un núcleo y 1GB de RAM, si elegimos una más potente puede generar gastos en la tarjeta, así que para empezar esto es suficiente.

El servicio permite seleccionar el almacenamiento, por defecto AWS otorga 8Gb de espacio, pero la cuenta gratuita parece que permite hasta 30Gb.

EC2 permite cambiar estas cosas según las necesidades y si con posterioridad queremos ampliar o crear otro volumen de almacenamiento es super fácil desde el panel de control:

Aquí en *Security Groups* tenemos que hacer algún cambio, aunque también lo podemos dejar para después en el panel de control. Lo principal es añadir además del puerto 22 que ya está por defecto, los puertos 80, 8787 para Rstudio y 3838 para Shiny y abrir el puerto 80.

### **Paso 3.** Crear pareja de claves de seguridad

Esta pareja de claves sirve para proteger el acceso al servidor, ya que, una vez en marcha está abierto a Internet, por lo que creamos una clave pública y otra privada que nos permitirá acceder. (no se debe perder estos archivos, porque no se podrá acceder)

Se escribe un nombre y descarga el fichero \*.pem que se crea. Tras esto se tendrá acceso al servidor disponible en AWS

### **Paso 4.** Acceder al servidor

Si se dispone de la clave con el nombre key.pem y la ip es 18.216.112.138 se puede acceder con la siguiente línea de código:

```
ssh -i "key.pem" ubuntu@ec2-18-216-112-138.us-east-2.compute.amazonaws.com
```

Instalar R y shiny:

Una vez tengamos acceso al servidor puede utilizarse los siguientes comandos:

```
sudo apt-get install r-base
sudo apt-get install gdebi-core
wget https://download3.rstudio.org/ubuntu-18.04/x86_64/shiny-server-1.5.19.995-
amd64.deb
sudo gdebi brillante-servidor-1.5.19.995-amd64.deb
```

La instalación de paquetes se lo realiza de la siguiente forma:

```
sudo su - \  
-c "R -e \"install.packages('shiny', repos='https://cran.rstudio.com/')\""
```

Cabe señalar que todas las librerías deberían de proceder desde el repositorio oficial de otro modo existen errores por las dependencias de las librerías.

### **Paso 5.** Enviar archivos al servidor *shiny*

Los archivos para el despliegue de las aplicaciones web se encuentran en la carpeta “/srv/shiny-server”, si disponemos de todos los documentos necesarios en un archivo comprimido de nombre archivos.tar.gz se envía este archivo de la siguiente forma:

```
scp -i "key.pem" archivos.tar.gz ubuntu@ec2-18-216-112-139.us-east-
```

## Configuración de grupos de seguridad

Reglas de entrada:

Tipo	Origen
HTTP	anywhere-ipv4
TCP Personalizado (3838)	anywhere-ipv4
HTTPS	anywhere-ipv4
SSH	anywhere-ipv4

Reglas de salida:

tipo	Origen
HTTP	anywhere-ipv4
HTTPS	anywhere-ipv4
Todo el tráfico	anywhere-ipv4

## Registros para Route S3

Registros tipo A, y CNAME redirigido al balanceador de carga

## Obtener certificados SSL con AWS Certificate Manager

Este servicio provee los registros SOA NS que serán registrados en Route S3

ID de certificado	Nombre de dominio	Tipo	Estado	¿Está en uso?	Requisitos que se deben cumplir para la renovación	Emitido a las
d77d5694-2396-49c6-830f-6d615dd9b9ec	mantenimientoactivo.com	Emitido por Amazon	Emitido	Sí	Se cumplen los requisitos	diciembre 25, 2021, 20:03:48 (UTC-05:00)

La configuración de los balanceadores de carga se lo realiza como muestra la imagen:

## Listener details

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

Protocol

HTTPS ▼

Port

443

1-65535

### Default actions [Info](#)

Specify the default actions for traffic on this listener. Default actions apply to traffic that does not meet the conditions of rules on your listener. Rules can be configured after the listener is created.

▼ 1. Forward to [Info](#)

Remove

Target group



Weight (0-999)

mytargetMA

HTTP ▼

1



Target type: Instance, IPv4

Traffic distribution: 100%

Select a target group ▼

0



[Create target group](#)

Enable group-level stickiness [Info](#)

If you enable stickiness for your target group, requests routed to it remain in the same group for the duration you specify.

Add action ▼

## Secure listener settings [Info](#)

### Security policy

Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration, known as a security policy, to negotiate SSL connections with clients.

ELBSecurityPolicy-2016-08 ▼

[Compare security policies](#)

### Default SSL/TLS certificate

The certificate used if a client connects without SNI protocol, or if there are no matching certificates. This certificate will automatically be added to your listener certificate list.

From ACM ▼

mantenimientoactivo.com ▼

d77d5694-2396-49c6-830f-6d615dd9b9ec ▼



[Request new ACM certificate](#)

**ANEXO B: RELACIÓN ENTRE SEIS FUNCIONES QUE DESCRIBEN LA VIDA ÚTIL ESTOCÁSTICA**

	$f(x)$	$F(x)$	$R(x)$	$h(x)$	$H(x)$	$\mu(x)$
$f(x)$	-	$\int_0^x f(z) dz$	$\int_x^\infty f(z) dz$	$\frac{f(x)}{\int_x^\infty f(z) dz}$	$-\ln \left\{ \int_x^\infty f(z) dz \right\}$	$\frac{\int_0^\infty z f(x+z) dz}{\int_x^\infty f(z) dz}$
$F(x)$	$F'(x)$	-	$1 - F(x)$	$\frac{F'(x)}{1 - F(x)}$	$-\ln \{1 - F(x)\}$	$\frac{\int_x^\infty (1 - F(z)) dz}{1 - F(x)}$
$R(x)$	$-R'(x)$	$1 - R(x)$	-	$\frac{-R'(x)}{R(x)}$	$-\ln[R(x)]$	$\frac{\int_x^\infty R(z) dz}{R(x)}$
$h(x)$	$h(x) \exp \left\{ -\int_0^x h(z) dz \right\}$	$1 - \exp \left( -\int_0^x h(z) dz \right)$	$\exp \left( -\int_0^x h(z) dz \right)$	-	$\int_0^x h(z) dz$	$\frac{\int_x^\infty \exp \left\{ -\int_0^z h(v) dv \right\} dz}{\exp \left\{ -\int_0^x h(z) dz \right\}}$
$H(x)$	$-\frac{d\{\exp[-H(x)]\}}{dx}$	$1 - \exp\{-H(x)\}$	$\exp\{-H(x)\}$	$H'(x)$	-	$\frac{\int_x^\infty \exp\{-H(z)\} dz}{\exp\{-H(x)\}}$
$\mu(x)$	$\frac{1 + \mu'(x)}{\mu^2(x)} \times \mu(0) \times \exp \left\{ -\int_0^x \frac{1}{\mu(z)} dz \right\}$	$1 - \frac{\mu(0)}{\mu(x)} \times \exp \left\{ -\int_0^x \frac{1}{\mu(z)} dz \right\}$	$\frac{\mu(0)}{\mu(x)} \times \exp \left\{ -\int_0^x \frac{1}{\mu(z)} dz \right\}$	$\frac{1}{\mu(x)} \{1 + \mu'(x)\}$	$\ln \left\{ \frac{\mu(x)}{\mu(0)} \right\} + \int_0^x \frac{1}{\mu(z)} dz$	-

Fuente: (Rinne 2009, p. 31)

**ANEXO C: LISTADO DE CÓDIGOS**

**Ecuaciones:** Markov

A)

$$\begin{cases} P_0(t) = e^{-\alpha t} \\ P_1(t) = 1 - e^{-\alpha t} \end{cases}$$

B)

$$a = \frac{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}}{\lambda_1 + \lambda_2}$$

$$\begin{cases} P_0(t) = e^{-(\lambda_1 + \lambda_2)t} \\ P_1(t) = (1 - e^{-(\lambda_1 + \lambda_2)t}) a_1 \\ P_2(t) = (1 - e^{-(\lambda_1 + \lambda_2)t}) a_2 \end{cases}$$

C)

$$a = \frac{\begin{bmatrix} \mu \\ \lambda \end{bmatrix}}{\mu + \lambda}$$

$$\begin{cases} P_0(t) = a_1 + a_2 e^{-(\mu + \lambda)t} \\ P_1(t) = a_2 - a_2 e^{-(\mu + \lambda)t} \end{cases}$$

D)

$$B = \mu_1 + \mu_2 + \lambda_1 + \lambda_2$$

$$C = \mu_2 \mu_1 + \lambda_1 \mu_2 + \lambda_2 \mu_1$$

$$m = (-B \pm \sqrt{B^2 - 4C}) / 2$$

$$a = \frac{\begin{bmatrix} \mu_2 \mu_1 \\ \lambda_1 \mu_2 \\ \lambda_2 \mu_1 \end{bmatrix}}{\mu_1 + \mu_2 + \lambda_1 + \lambda_2} \quad b = \frac{\begin{bmatrix} (\mu_2 + m_1)(\mu_1 + m_1), (\mu_2 + m_2)(\mu_1 + m_2) \\ (\lambda_1 + m_1)(\mu_2 + m_1), (\lambda_1 + m_2)(\mu_2 + m_2) \\ (\lambda_2 + m_1)(\mu_1 + m_1), (\lambda_2 + m_2)(\mu_1 + m_2) \end{bmatrix}}{m_2(m_1 - m_2)}$$

$$\begin{cases} P_0(t) = a_1 + b_{1,1} e^{m_1 t} - b_{2,1} e^{m_2 t} \\ P_1(t) = a_2 + b_{2,1} e^{m_1 t} - b_{2,2} e^{m_2 t} \\ P_2(t) = a_3 + b_{3,1} e^{m_1 t} - b_{3,2} e^{m_2 t} \end{cases}$$

E)



$$a = \frac{\begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1 \mu_2 \\ 1 & \lambda_1 & \mu_2 & \lambda_1 \mu_2 \\ 1 & \mu_2 & \lambda_2 & \lambda_2 \mu_1 \\ 1 & \lambda_2 & \lambda_2 & \lambda_1 \lambda_2 \end{bmatrix}}{(\mu_1 + \lambda_1)(\mu_2 + \lambda_2)}$$

$$\begin{cases} P_0(t) = a_{1,1} e^{-(\lambda_2 + \mu_2 + \lambda_1 + \mu_1)t} + a_{1,2} e^{-(\lambda_2 + \mu_2)t} + a_{1,3} e^{-(\lambda_1 + \mu_1)t} + a_{1,4} \\ P_1(t) = -a_{2,1} e^{-(\lambda_2 + \mu_2 + \lambda_1 + \mu_1)t} + a_{2,2} e^{-(\lambda_2 + \mu_2)t} - a_{2,3} e^{-(\lambda_1 + \mu_1)t} + a_{2,4} \\ P_2(t) = -a_{3,1} e^{-(\lambda_2 + \mu_2 + \lambda_1 + \mu_1)t} - a_{3,2} e^{-(\lambda_2 + \mu_2)t} + a_{3,3} e^{-(\lambda_1 + \mu_1)t} + a_{3,4} \\ P_3(t) = a_{4,1} e^{-(\lambda_2 + \mu_2 + \lambda_1 + \mu_1)t} - a_{4,2} e^{-(\lambda_2 + \mu_2)t} - a_{4,3} e^{-(\lambda_1 + \mu_1)t} + a_{4,4} \end{cases}$$

F)

$$B = \mu_1 + \mu_2 + \mu_3 + \lambda_1 + \lambda_2 + \lambda_3$$

$$C = \lambda_3 \mu_2 + \mu_1 (\mu_2 + \mu_3) + \lambda_2 (\mu_2 + \mu_3) + \lambda_1 \mu_3 + \lambda_3 (\lambda_1 + \lambda_2) + \lambda_2 \mu_1$$

$$m = \left( -B \pm \sqrt{B^2 - 4C} \right) / 2$$

$$a = \frac{\begin{bmatrix} \lambda_3 \mu_2 + \mu_1 (\mu_2 + \mu_3), \\ \lambda_2 (\mu_2 + \mu_3) + \lambda_1 \mu_3, \\ \lambda_1 \mu_2, \lambda_2 \mu_1 \end{bmatrix}}{m_1 m_2} \quad b = \frac{\begin{bmatrix} (\mu_2 + m_1)(\mu_1 + m_1), \\ (\lambda_1 + m_1)(\mu_2 + m_1), \\ (\lambda_2 + m_1)(\mu_1 + m_1) \end{bmatrix}}{m_2 (m_1 - m_2)} \quad c = \frac{\begin{bmatrix} (\mu_2 + m_2)(\mu_1 + m_2), \\ (\lambda_1 + m_2)(\mu_2 + m_2), \\ (\lambda_2 + m_2)(\mu_1 + m_2) \end{bmatrix}}{m_2 (m_1 - m_2)}$$

$$\begin{cases} P_0(t) = a_1 + b_1 e^{m_1 t} + (1 - a_1 - b_1) e^{m_2 t} \\ P_1(t) = a_2 + b_2 e^{m_1 t} + (a_2 + b_2) e^{m_2 t} \\ P_2(t) = a_3 + b_3 e^{m_1 t} + (a_3 + b_3) e^{m_2 t} \end{cases}$$

G)

$$B = 3\lambda + \mu$$

$$C = 2\lambda^2$$

$$m = \left( -B \pm \sqrt{B^2 - 4C} \right) / 2$$

$$b = \frac{\begin{bmatrix} \lambda + \mu + m_1 \\ 2\lambda \\ m_2 \end{bmatrix}}{m_1 - m_2}$$

$$\begin{cases} P_0(t) = b_1 (e^{m_1 t} - e^{m_2 t}) \\ P_1(t) = b_2 (e^{m_1 t} - e^{m_2 t}) \\ P_2(t) = b_3 (e^{m_1 t} - e^{m_2 t}) + 1 \end{cases}$$

$$\begin{cases} P_0 = \left[ 1 + D_1 + 2\lambda^2 D + \frac{1}{\mu_m} (\lambda_{m1} + D_1 \lambda_{m2}) \right]^{-1} \\ P_1 = P_0 D_1 \\ P_2 = P_0 (\lambda_{m1} + D_1 \lambda_{m2}) / \mu_m \\ P_3 = P_0 2\lambda^2 D \end{cases}$$

**Listado de código:**     Distribuciones global.R

```
library(shiny)
library(actuar)
library(fitdistrplus)
library(DT)
library(FAdist)
```

**Listado de código:**     Distribuciones server.R

```
shinyServer(function(input, output) {

  dataInput <- reactive({

    x <- as.numeric(unlist(strsplit(input$vec1, ",")))
    y <- as.numeric(unlist(strsplit(input$vec1, " ")))
    if (length(x)==1){x<-y}

    if (all(x>=0))
    {
      Aj.ex <- fitdist(x, "exp")
      Aj.wb <- fitdist(x, "weibull")
      Aj.wbp <- fitdist(x, "weibull3",start = list(shape =3, scale = 1 ))
      Aj.gm <- fitdist(x, "gamma")
      Aj.ln <- fitdist(x, "lnorm")
      Aj.ll <- fitdist(x, "llogis")
      Aj.nr <- fitdist(x, "norm")
      Aj.lg <- fitdist(x, "logis")
      Aj.gb <- fitdist(x, "gumbel",start = list(scale = 2,location = 5 )) # necesita
condición inicial

      Ajuste <- list(Aj.ex,Aj.wb,Aj.wbp,Aj.gm,Aj.ln,Aj.ll,Aj.nr,Aj.lg,Aj.gb)
      Ajute.name <- c("exp","Weibull","Weibull3","gamma","log-normal","log-
logística","normal","logística","Gumbel")
      dt <- gofstat(Ajuste,fitnames = Ajute.name)

    }else{

      Aj.nr <- fitdist(x, "norm")
      Aj.lg <- fitdist(x, "logis")
      Aj.gb <- fitdist(x, "gumbel",start = list(scale = 2,location = 5 )) # necesita
condición inicial

      Ajuste <- list(Aj.nr,Aj.lg,Aj.gb)
      Ajute.name <- c("normal","logística","Gumbel")
      dt <- gofstat(Ajuste,fitnames = Ajute.name)
```

```

    }

    data1 <- data.frame(
      Kolmogorov.S=dt$ks,
      Cramer.v.M=dt$cvm,
      Anderson.D=dt$ad,
      AIC=dt$aic,
      BIC=dt$bic)

    list(x,Ajuste,Ajute.name,data1)
  })

  output$oid1 <- renderPrint({
    cat("Datos:\n")
    print(dataInput()[[1]])
    cat("\nParámetros:\n")
    print(dataInput()[[2]])
  })

  output$distPlot1 <- renderPlot({
    x <- dataInput()[[1]]
    plotdist(x, histo=TRUE, demp=TRUE)
  })

  output$distPlot2 <- renderPlot({
    cdfcomp(dataInput()[[2]],
             xlogscale = FALSE,
             ylogscale = TRUE,
             legendtext = dataInput()[[3]],
             main="Distribución acumulada CDF empírica y teórica",
             xlab="Datos"
    )
  })

  output$contents1 <- renderDataTable({
    datatable(dataInput()[[4]],extensions = c('Responsive'),
              options = list(responsive = TRUE))
  })
})

```

**Listado de código:**     Distribuciones ui.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      textInput('vec1', 'Datos separados por comas o espacios, mínimo 8',
"0.5472, 0.0461, 2.3514, 2.713, 12.1391, 2.5687, 0.5952, 1.6512, 2.3374, 2.7594,
3.493, 1.1225, 10.1915, 4.3583, 3.5354, 0.6457, 0.8055, 0.5855, 1.9845, 0.6449,
0.2062, 1.4841, 3.2389, 2.5976, 4.6836, 0.0504, 0.0804, 1.409, 11.3227"),
      verbatimTextOutput("oid1")
    ),
    mainPanel(
      tabsetPanel(
        tabPanel("Gráficas",
          plotOutput("distPlot1",height="490px"),

```

```

        plotOutput("distPlot2",height="490px")
      ),
      tabPanel("Estadísticos",
        dataTableOutput("contents1")
      )
    )
  )
))

```

**Listado de código:**      Calculadora Ui.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      textAreaInput('vec1', 'Evaluar expresión',
"rnorm(10,2,0.5)+3*rweibull(10,2.2,0.1)",
      textAreaInput('vec2', 'Gr. expresión', "x<-
rnorm(10,2,0.5)+3*rweibull(10,2.2,0.1); hist(x)"
    ),
    mainPanel(
      verbatimTextOutput("oid1"),
      plotOutput("distPlot")
    )
  )
))

```

**Listado de código:**      Calculadora server.R

```

shinyServer(function(input, output) {

  output$oid1 <- renderPrint({
    x <- eval({expression(input$vec1)})
    eval(str2expression(x))
  })

  output$distPlot <- renderPlot({
    x <- eval({expression(input$vec2)})
    eval(str2expression(x))
  })
})

```

**Listado de código:**      Calculadora global.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      textAreaInput('vec1', 'Evaluar expresión',
"rnorm(10,2,0.5)+3*rweibull(10,2.2,0.1)",
      textAreaInput('vec2', 'Gr. expresión', "x<-
rnorm(10,2,0.5)+3*rweibull(10,2.2,0.1); hist(x)"
    ),
    mainPanel(
      verbatimTextOutput("oid1"),

```

```

        plotOutput("distPlot")
    )
  )
))

```

**Listado de código:** Correlación global.R

```

library(shiny)
library(DT)
library(readxl)
library(ggplot2)
library(ggcorrplot)
library(dplyr)
options(shiny.maxRequestSize = 25*1024^2)

```

**Listado de código:** Correlación Ui.R

```

shinyUI(fluidPage(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "custom.css"),
    tags$link(rel = "icon", href = "favicon.png")
  ),
  tabsetPanel(
    tabPanel("Datos",
      fileInput('file1', 'Subir archivo',
        accept = c('text/csv', 'text/comma-separated-values',
          'text/tab-separated-values', 'text/plain', '.csv', '.tsv', '.xlsx')
      ),
      dataTableOutput("contents")
    ),
    tabPanel("Gráficos",
      plotOutput('ima1')
    )
  )
))

```

**Listado de código:** Correlación server.R

```

shinyServer(function(input, output) {

  datos_falla <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)
    read_excel(inFile$datapath)
  })

  # GGLOT GENERAL
  output$ima1 <- renderPlot({
    if (is.null(datos_falla()))
      return(NULL)
    correlacion <- datos_falla()
  })

```

```

    corr <- round(cor(correlacion), 2)
    p.mat <- cor_pmat(correlacion)
    ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE)
  })

# MOSTRAR TABLA

output$contents <- renderDataTable({
  datatable(datos_falla(),extensions = c('Responsive'),
    options = list(responsive = TRUE))
})
})

```

### Listado de código: Costos global.R

```
library(shiny)
```

### Listado de código: Costos Ui.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      textInput('vec1', 'Flujo de caja primera opción (separados por comas)',
"70000, 1000, 1220, 2001"),
      textInput('vec2', 'Flujo de caja primera opción (separados por comas)',
"95000, 1011, 2000, 2010, 2000"),
      sliderInput("interes", "Interés %", min=0, max=100, value = 12)
    ),
    mainPanel(
      verbatimTextOutput("oid1"),
      verbatimTextOutput("oid2"),
      verbatimTextOutput("oid3"),
      verbatimTextOutput("oid4")
    )
  )
))

```

### Listado de código: Costos server.R

```

shinyServer(function(input, output) {

  output$oid1 <- renderPrint({
    x <- as.numeric(unlist(strsplit(input$vec1, ",")))
    cat("PRIMERA OPCIÓN:\n")
    cat("VP:\n")
    n <- length(x)
    i <- input$interes/100
    k <- 1:n
    PV = x/((1+i)^k)
    print(PV)
    cat("Vp de una anualidad:\n")
    print(sum(PV))
    cat("CAE:\n")
    print(sum(PV)*(i*(i+1)^n)/((1+i)^n-1))

    y <- as.numeric(unlist(strsplit(input$vec2, ",")))
    cat("SEGUNDA OPCIÓN:\n")
    cat("VP:\n")
    n <- length(y)
    k <- 1:n

```

```

    PV <- y/((1+i)^k)
    print(PV)
    cat("Vp de una anualidad:\n")
    print(sum(PV))
    cat("CAE:\n")
    print(sum(PV)*(i*(i+1)^n)/((1+i)^n-1))
  })
})

```

#### Listado de código: Cronograma global.R

```

library(shiny)
library(DT)
library(readxl)
library(ggplot2)
library(dplyr)
library(plotrix)
options(shiny.maxRequestSize = 25*1024^2)

```

```

indicereal <- function(i,f){
  min(which(((1:52-i)%f==1)*1==1))
}

```

#### Listado de código: Cronograma Ui.R

```

shinyUI(fluidPage(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "custom.css"),
    tags$link(rel = "icon", href = "favicon.png")
  ),
  tabsetPanel(
    tabPanel("Datos",
      fileInput('file1', 'Subir archivo',
        accept = c('text/csv','text/comma-separated-values',
          'text/tab-separated-values','text/plain','.csv','.tsv','.xlsx'
        ),
      ),
    dataTableOutput("contents")
  ),
  tabPanel("Gráficos",
    plotOutput('ima1',height = "3780px",width = "98%")
  ),
  tabPanel("Resumen",
    dataTableOutput("contents2")
  )
)
))

```

#### Listado de código: Cronograma server.R

```

shinyServer(function(input, output) {

  datos_falla <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)
    read_excel(inFile$datapath)
  })

  # GGLOT GENERAL
  output$ima1 <- renderPlot({

```

```

if (is.null(datos_falla()))
  return(NULL)

tareas <- datos_falla()
tareas %>%
  mutate(orden=1:n()) %>%
  arrange(desc(frec),duracion) %>%
  mutate(indice=1:n()) -> tareas

n <- length(tareas$tarea)
x <- numeric(n)

for(i in 1:n){
  x[i] <- min(which(((1:52-tareas$indice[i]+1)%tareas$frec[i]==1)*1==1))
}

tareas%>%
  mutate(ind = x)-> tareas

nom <- numeric()
dat <- numeric()
color <- numeric()
dur <- numeric()
f <- numeric()
indd <- numeric()

for( i in 1:n){
  xx <- which(((1:52-tareas$ind[i])%tareas$frec[i]==1)*1:52!=0)
  nn <- length(xx)
  nom <- c(nom,rep(tareas$tarea[i],nn))
  dat <- c(dat,xx)
  dur <- c(dur,rep(tareas$duracion[i],nn))
  f <- c(f,rep(tareas$frec[i],nn))
  indd <- c(indd,rep(tareas$ind[i],nn))
}

info2 <- list(labels=nom,starts=dat,ends=dat+1)

ganttt.chart(info2, half.height=0.5, label.cex=1, taskcolors="lightgray", hgrid=TRUE, border.col="lightgray", vgridlab=1:52, vgridpos=1:52)

})

# MOSTRAR TABLA

output$contents <- renderDataTable({
  datatable(datos_falla(), extensions = c('Responsive'),
            options = list(responsive = TRUE))
})

output$contents2 <- renderDataTable({

  if (is.null(datos_falla()))
    return(NULL)

  tareas <- datos_falla()
  tareas %>%

```



```

mutate(orden=1:n()) %>%
  arrange(desc(frec),duracion) %>%
  mutate(indice=1:n()) -> tareas

datatable(tareas,extensions = c('Responsive'),
  options = list(responsive = TRUE))
})
})

```

### Listado de código: Causa.efecto global.R

```

library(shiny)
library(DT)
library(readxl)
library(ggplot2)
library(dplyr)
library(qcc)
options(shiny.maxRequestSize = 25*1024^2)

```

### Listado de código: Causa.efecto Ui.R

```

shinyUI(fluidPage(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "custom.css"),
    tags$link(rel = "icon", href = "favicon.png")
  ),
  tabsetPanel(
    tabPanel("Datos",
      fileInput('file1', 'Subir archivo',
        accept = c('text/csv','text/comma-separated-values',
          'text/tab-separated-values','text/plain','.csv','.tsv','.xlsx'
        ),
      ),
    dataTableOutput("contents")
  ),
  tabPanel("Gráficos",
    numericInput("parte","Elemento",min = 1,max=10,value=1),
    plotOutput('ima1')
  )
)
))

```

### Listado de código: Causa.efecto server.R

```

shinyServer(function(input, output) {

  datos_falla <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)
    read_excel(inFile$datapath)
  })
})

```

```

# GGLOT GENERAL
output$ima1 <- renderPlot({
  if (is.null(datos_falla()))
    return(NULL)
  causaraiz <- datos_falla()

  if (is.na(input$parte)){return(NULL)}else{
    inn<-input$parte

    causaraiz %>% dplyr::select(efecto) %>% unique -> z

    causaraiz %>%
      filter(campo== "Medidas",efecto==z$efecto[inn]) %>%
      dplyr::select(causa)-> var_medidas

    causaraiz %>%
      filter(campo== "Materiales",efecto==z$efecto[inn]) %>%
      dplyr::select(causa) -> var_materiales

    causaraiz %>%
      filter(campo== "Personal",efecto==z$efecto[inn]) %>%
      dplyr::select(causa) -> var_personal

    causaraiz %>%
      filter(campo== "Ambiente",efecto==z$efecto[inn]) %>%
      dplyr::select(causa) -> var_ambiente

    causaraiz %>%
      filter(campo== "Métodos",efecto==z$efecto[inn]) %>%
      dplyr::select(causa) -> var_metodos

    causaraiz %>% dplyr::select(causa,campo,efecto) %>%
      filter(campo== "Máquinas",efecto==z$efecto[inn]) %>%
      dplyr::select(causa) -> var_maquinas

    cause.and.effect(cause=list(Medidas=var_medidas$causa,
                                Materiales=var_materiales$causa,
                                Personal=var_personal$causa,
                                Ambiente=var_ambiente$causa,
                                Métodos=var_metodos$causa,
                                Máquinas=var_maquinas$causa),
                      effect=z$efecto[inn], title="Diagrama causa-efecto")
  }
})

# MOSTRAR TABLA
output$contents <- renderDataTable({
  datatable(datos_falla(),extensions = c('Responsive'),
            options = list(responsive = TRUE))
})
})

```

**Listado de código:** Markov tiempo continuo global.R

```

library(shiny)
library(latex2exp)
library(ggplot2)

mk06 <- function(x,l,u,model="P0"){
  H <- c(u,l)/(1+u)
  switch(model,
    "P0" = H[1] + H[2]*exp(-(1+u)*x),
    "P1" = H[2] - H[2]*exp(-(1+u)*x),
    "A_s" = (1/l)/(1/u+1/l)
  )
}

mk01 <- function(x,l,u,model="P0"){
  a <- l[1]+u[1]+l[2]+u[2]
  b <- l[1]+u[1]
  c <- l[2]+u[2]
  H <- c(l[1]*l[2], l[2]*u[1], u[2]*l[1], u[2]*u[1]) / (b*c)
  I <- c(l[2], l[2], u[2], u[2])*l[1] / (b*c)
  J <- c(l[1], u[1], l[1], u[1])*l[2] / (b*c)
  K <- c(1, 1, 1, 1)*l[1]*l[2] / (b*c)
  switch(model,
    "P0" = H[1]*exp(-a*x) + H[2]*exp(-b*x) + H[3]*exp(-c*x) + H[4],
    "P1" = -I[1]*exp(-a*x) + I[2]*exp(-b*x) - I[3]*exp(-c*x) + I[4],
    "P2" = -J[1]*exp(-a*x) - J[2]*exp(-b*x) + J[3]*exp(-c*x) + J[4],
    "P3" = K[1]*exp(-a*x) - K[2]*exp(-b*x) - K[3]*exp(-c*x) + K[4],
    "H" = H[4]
  )
}

mk02 <- function(x,l,u,model="P0"){
  a <- l[1]+u[1]+l[2]+u[2]
  b <- l[1]+u[1]
  c <- l[2]+u[2]
  H <- c(1, -1, -1, 1)*l[1]*l[2]/(b*c)
  I <- c(u[1], l[1], -u[1], -l[1])*l[2]/(b*c)
  J <- c(u[2], -u[2], l[2], -l[2])*l[1]/(b*c)
  K <- c(u[2]*u[1], l[1]*u[2], u[1]*l[2], l[1]*l[2])/ (b*c)
  switch(model,
    "P0"= H[1]*exp(-a*x) + I[1]*exp(-c*x) + J[1]*exp(-b*x) + K[1],
    "P1"= H[2]*exp(-a*x) + I[2]*exp(-c*x) + J[2]*exp(-b*x) + K[2],
    "P2"= H[3]*exp(-a*x) + I[3]*exp(-c*x) + J[3]*exp(-b*x) + K[3],
    "P3"= H[4]*exp(-a*x) + I[4]*exp(-c*x) + J[4]*exp(-b*x) + K[4],
    "H"= K[1],
    "P0P"=1- (H[3]*exp(-a*x) + I[3]*exp(-c*x) + J[3]*exp(-b*x) + K[3]),
    "HP"= 1-K[3]
  )
}

mk03 <- function(x,l,u,model="P0"){
  A <- u[1] + u[2] + l[1] + l[2]
  B <- u[1]*u[2] + l[1]*u[2] + l[2]*u[1]
  m1 <- (-A+sqrt(A^2-4*B)) / 2
  m2 <- (-A-sqrt(A^2-4*B)) / 2
  H <- c(u[2]*u[1], l[1]*u[2], l[2]*u[1])/ (m1*m2)
  I <- c((u[2]+m1)*(u[1]+m1), l[1]*(u[2]+m1), l[2]*(u[1]+m1)) / (m1*(m1-m2))
  J <- c((u[2]+m2)*(u[1]+m2), l[1]*(u[2]+m2), l[2]*(u[1]+m2)) / (m2*(m1-m2))
}

```

```

switch(model,
  "P0" = H[1] + I[1]*exp(m1*x) - J[1]*exp(m2*x),
  "P1" = H[2] + I[2]*exp(m1*x) - J[2]*exp(m2*x),
  "P2" = H[3] + I[3]*exp(m1*x) - J[3]*exp(m2*x),
  "As" = u[1]*u[2]/(B)
)
}

mk04 <- function(x,l,u,model="P0"){
  B <- l[1]+l[2]+l[3]+u[1]+u[2]+u[3]
  k1 <- (-B+sqrt(B^2-
4*(l[3]*u[2]+u[1]*(u[2]+u[3])+l[2]*u[3]+l[1]*(u[2]+u[3])+l[1]*l[3]+l[2]*(u[1]+l[3])))
) / 2
  k2 <- (-B-sqrt(B^2-
4*(l[3]*u[2]+u[1]*(u[2]+u[3])+l[2]*u[3]+l[1]*(u[2]+u[3])+l[1]*l[3]+l[2]*(u[1]+l[3])))
) / 2
  H <- c(l[3]*u[2]+u[1]*(u[2]+u[3]),l[2]*u[3]+l[1]*(u[2]+u[3]),
l[1]*l[3]+l[2]*(u[1]+l[3])) / (k1*k2)
  I <- c(l[3]*(u[2]+k1)+(u[1]+k1)*(u[2]+u[3]+k1), l[2]*u[3]+l[1]*(u[2]+u[3]+k1),
l[1]*l[3]+l[2]*(u[1]+l[3]+k1)) / (k1*(k1-k2))
  switch(model,
    "P0" = H[1] + I[1]*exp(k1*x) + (1-H[1]-I[1])*exp(k2*x),
    "P1" = H[2] + I[2]*exp(k1*x) - (H[2] + I[2])*exp(k2*x),
    "P2" = H[3] + I[3]*exp(k1*x) - (H[3] + I[3])*exp(k2*x)
  )
}

mk05 <- function(x,l,u,model="P0"){
  B = u[1]+u[2]+l[1]+l[2]
  C = u[1]*u[2]+l[2]*u[1]+l[1]*u[2]
  m1 <- (-B + sqrt(B^2-4*C)) / 2
  m2 <- (-B - sqrt(B^2-4*C)) / 2
  H <- c(u[2]*u[1], u[2], u[1])/(m1*m2)
  I <- c((u[2]+m1)*(u[1]+m1), u[2]+m1, (u[1]+m1)) / (m1*(m1-m2))
  J <- c((u[2]+m2)*(u[1]+m2), u[2]+m2, (u[1]+m2)) / (m2*(m1-m2))
  switch(model,
    "P0" = H[1] + I[1]*exp(m1*x)-J[1]*exp(m2*x),
    "P1" = (H[2] + I[2]*exp(m1*x)-J[2]*exp(m2*x))*l[1],
    "P2" = (H[3] + I[3]*exp(m1*x)-J[3]*exp(m2*x))*l[2],
    "AV_ss" = u[1]*u[2]/C
  )
}

mk07 <- function(x,l,u,model="P0"){
  c1 <- (- (3*l[1]+u[1])+sqrt((3*l[1]+u[1])^2-8*l[1]^2)) / 2
  c2 <- (- (3*l[1]+u[1])-sqrt((3*l[1]+u[1])^2-8*l[1]^2)) / 2
  H <- c(l[1]+u[1]+c1, 2*l[1], c2) / (c1-c2)
  xx <- log(c1/c2)*(1/(c2-c1))
  switch(model,
    "P0" = H[1]*exp(c1*x)-H[1]*exp(c2*x),
    "P1" = H[2]*exp(c1*x)-H[2]*exp(c2*x),
    "P2" = H[3]*exp(c1*x)-H[3]*exp(c2*x)+1,
    "H"= H[1]*exp(c1*xx)-H[1]*exp(c2*xx)
  )
}

```

**Listado de código:** Markov tiempo continuo Ui.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      selectInput("tipo", label = "Modelo", choices =
c('A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7')),
      numericInput("ndatos", "Eje x", min=0, value = 20),
      conditionalPanel(condition = "input.tipo == 'A1'",
        textInput("lambda1", "tasa de falla:", "0.3"),
        textInput("mu1", "tasa de reparación:", "0.09")),
      conditionalPanel(condition = "input.tipo == 'A2'",
        textInput("lambda2", "tasa de falla:", "0.3, 0.2"),
        textInput("mu2", "tasa de reparación:", "0.3, 0.1")),
      conditionalPanel(condition = "input.tipo == 'A3'",
        textInput("lambda3", "tasa de falla:", "0.3, 0.2"),
        textInput("mu3", "tasa de reparación:", "0.3, 0.1")),
      conditionalPanel(condition = "input.tipo == 'A4'",
        textInput("lambda4", "tasa de falla:", "0.3, 0.2"),
        textInput("mu4", "tasa de reparación:", "0.3, 0.1")),
      conditionalPanel(condition = "input.tipo == 'A5'",
        textInput("lambda5", "tasa de falla:", "0.3, 0.2"),
        textInput("mu5", "tasa de reparación:", "0.3, 0.1")),
      conditionalPanel(condition = "input.tipo == 'A6'",
        textInput("lambda6", "tasa de falla:", "0.3, 0.2"),
        textInput("mu6", "tasa de reparación:", "0.3, 0.1")),
      conditionalPanel(condition = "input.tipo == 'A7'",
        textInput("lambda7", "tasa de falla:", "0.3, 0.2, 0.2"),
        textInput("mu7", "tasa de reparación:", "0.3, 0.1, 0.1")),

      numericInput("x", "tiempo a evaluar", min=0, value = 1.5),
      verbatimTextOutput("texto")
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
))

```

### Listado de código: Markov tiempo continuo server.R

```

shinyServer(function(input, output) {
  output$distPlot <- renderPlot({

    if (input$tipo=='A1'){
      if (is.na(input$lambda1)||is.na(input$mu1)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
        u <- as.numeric(unlist(strsplit(input$mu1,",")))
        l <- as.numeric(unlist(strsplit(input$lambda1,",")))

        ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
          geom_function(fun = mk06, colour = "black", args = list(l = l,u=u, model =
"P0"))+
          geom_function(fun = mk06, colour = "grey", args = list(l = l,u=u, model =
"P1"))+
          geom_hline(yintercept=mk06(l=l,u=u,model = "A_s"),linetype="dashed")+
          labs(subtitle = "mk06",
              caption = paste("l=",l),

```

```

        x = "tiempo", y = "probabilidad")+theme_minimal()
    }
}
else if (input$tipo=='A2'){
    if (is.na(input$lambda2)||is.na(input$mu2)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
    u <- as.numeric(unlist(strsplit(input$mu2,",")))
    l <- as.numeric(unlist(strsplit(input$lambda2,",")))

    ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
      geom_function(fun = mk07, colour = "black",args = list(l = l,u=u, model =
"P0"))+
      geom_function(fun = mk07, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
      geom_function(fun = mk07, colour = "grey",args = list(l = l,u=u, model =
"P2"))+
      geom_hline(yintercept=mk07(l=l,u=u,model = "H"),linetype="dashed")+
      labs(subtitle = "mk07",
           caption = paste("l=",l),
           x = "tiempo", y = "probabilidad")+theme_minimal()
    }
}
else if (input$tipo=='A3'){
    if (is.na(input$lambda3)||is.na(input$mu3)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
    u <- as.numeric(unlist(strsplit(input$mu3,",")))
    l <- as.numeric(unlist(strsplit(input$lambda3,",")))

    ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
      geom_function(fun = mk01, colour = "black",args = list(l = l,u=u, model =
"P0"))+
      geom_function(fun = mk01, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
      geom_function(fun = mk01, colour = "grey",args = list(l = l,u=u, model =
"P2"))+
      geom_function(fun = mk01, colour = "grey",args = list(l = l,u=u, model =
"P3"))+
      geom_hline(yintercept=mk01(l=l,u=u,model = "H"),linetype="dashed")+
      labs(subtitle = "mk01",
           caption = paste("l=",l),
           x = "tiempo", y = "probabilidad")+theme_minimal()
    }
}
else if (input$tipo=='A4'){
    if (is.na(input$lambda4)||is.na(input$mu4)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
    u <- as.numeric(unlist(strsplit(input$mu4,",")))
    l <- as.numeric(unlist(strsplit(input$lambda4,",")))

    ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
      geom_function(fun = mk02, colour = "black",args = list(l = l,u=u, model =
"P0"))+
      geom_function(fun = mk02, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
      geom_function(fun = mk02, colour = "grey",args = list(l = l,u=u, model =
"P2"))+

```

```

geom_function(fun = mk02, colour = "grey",args = list(l = l,u=u, model =
"P3"))+
geom_hline(yintercept=mk02(l=l,u=u,model = "H"),linetype="dashed")+
labs(subtitle = "mk02",
caption = paste("l=",l),
x = "tiempo", y = "probabilidad")+theme_minimal()
}
}
else if (input$tipo=='A5'){
if (is.na(input$lambda5)||is.na(input$mu5)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
u <- as.numeric(unlist(strsplit(input$mu5,",")))
l <- as.numeric(unlist(strsplit(input$lambda5,",")))

ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
geom_function(fun = mk03, colour = "black",args = list(l = l,u=u, model =
"P0"))+
geom_function(fun = mk03, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
geom_function(fun = mk03, colour = "grey",args = list(l = l,u=u, model =
"P2"))+
geom_hline(yintercept=mk03(l=l,u=u,model = "As"),linetype="dashed")+
labs(subtitle = "mk03",
caption = paste("l=",l),
x = "tiempo", y = "probabilidad")+theme_minimal()
}
}
else if (input$tipo=='A6'){
if (is.na(input$lambda6)||is.na(input$mu6)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
u <- as.numeric(unlist(strsplit(input$mu6,",")))
l <- as.numeric(unlist(strsplit(input$lambda6,",")))

ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
geom_function(fun = mk05, colour = "black",args = list(l = l,u=u, model =
"P0"))+
geom_function(fun = mk05, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
geom_function(fun = mk05, colour = "grey",args = list(l = l,u=u, model =
"P2"))+
geom_hline(yintercept=mk05(l=l,u=u,model = "AV_ss"),linetype="dashed")+
labs(subtitle = "mk05",
caption = paste("l=",l),
x = "tiempo", y = "probabilidad")+theme_minimal()
}
}
else if (input$tipo=='A7'){
if (is.na(input$lambda7)||is.na(input$mu7)||is.na(input$ndatos)) {ggplot()+
theme_void()}else {
u <- as.numeric(unlist(strsplit(input$mu7,",")))
l <- as.numeric(unlist(strsplit(input$lambda7,",")))

ggplot(data.frame(x = 0:input$ndatos), aes(x)) +
geom_function(fun = mk04, colour = "black",args = list(l = l,u=u, model =
"P0"))+

```

```

    geom_function(fun = mk04, colour = "grey",args = list(l = l,u=u, model =
"P1"))+
    geom_function(fun = mk04, colour = "grey",args = list(l = l,u=u, model =
"P2"))+
    labs(subtitle = "mk04",
         caption = paste("l=",l),
         x = "tiempo", y = "probabilidad")+theme_minimal()
  }
}
})

```

```

output$texto <- renderPrint({
  if (input$tipo=='A1'){
    u <- as.numeric(unlist(strsplit(input$mu1,"")))
    l <- as.numeric(unlist(strsplit(input$lambda1,"")))
    print(
      c(mk06(input$x, l, u ,model="P0"),
        mk06(input$x, l, u,model="P1")))
  }
  else if (input$tipo=='A2'){
    u <- as.numeric(unlist(strsplit(input$mu2,"")))
    l <- as.numeric(unlist(strsplit(input$lambda2,"")))
    print(
      c(mk07(input$x, l, u ,model="P0"),
        mk07(input$x, l, u,model="P1"),
        mk07(input$x, l, u,model="P2")))
  }
  else if (input$tipo=='A3'){
    u <- as.numeric(unlist(strsplit(input$mu3,"")))
    l <- as.numeric(unlist(strsplit(input$lambda3,"")))
    print(
      c(mk01(input$x, l, u ,model="P0"),
        mk01(input$x, l, u,model="P1"),
        mk01(input$x, l, u,model="P2"),
        mk01(input$x, l, u,model="P3")))
  }
  else if (input$tipo=='A4'){
    u <- as.numeric(unlist(strsplit(input$mu4,"")))
    l <- as.numeric(unlist(strsplit(input$lambda4,"")))
    print(
      c(mk02(input$x, l, u ,model="P0"),
        mk02(input$x, l, u,model="P1"),
        mk02(input$x, l, u,model="P2"),
        mk02(input$x, l, u,model="P3"),
        mk02(input$x, l, u,model="P4")))
  }
  else if (input$tipo=='A5'){
    u <- as.numeric(unlist(strsplit(input$mu5,"")))
    l <- as.numeric(unlist(strsplit(input$lambda5,"")))
    print(
      c(mk03(input$x, l, u ,model="P0"),
        mk03(input$x, l, u,model="P1"),
        mk03(input$x, l, u,model="P2"),
        mk03(input$x, l, u,model="P3")))
  }
  else if (input$tipo=='A6'){
    u <- as.numeric(unlist(strsplit(input$mu6,"")))

```



```

    l <- as.numeric(unlist(strsplit(input$lambda6,",")))
    print(
      c(mk05(input$x, l, u ,model="P0"),
        mk05(input$x, l, u,model="P1"),
        mk05(input$x, l, u,model="P2")))
  }
  else if (input$tipo=='A7'){
    u <- as.numeric(unlist(strsplit(input$mu7,",")))
    l <- as.numeric(unlist(strsplit(input$lambda7,",")))
    print(
      c(mk04(input$x, l, u ,model="P0"),
        mk04(input$x, l, u,model="P1"),
        mk04(input$x, l, u,model="P2")))
  }
})
})

```

### Listado de código: Confiabilidad global.R

```

library(shiny)
library(ggplot2)
library(actuar)
library(latex2exp)
library(FAdist)

# Distribución exponencial
Rexp <- function(x,rate) 1-pexp(x,rate=rate)
texp <- function(x,rate) rate
mtbfexp <- function(rate) 1/rate

# Distribución de Weibull
Rweibull <- function(x, shape, scale) 1 - pweibull(x, shape = shape, scale = scale)
tweibull <- function(x,shape,scale) pweibull3(x, shape = shape, scale = scale,)/(1-
pweibull3(x, shape = shape, scale = scale))
mtbfweibull <- function(shape,scale) scale*gamma(1+1/shape)

# Distribución de Weibull3
Rweibull3 <- function(x, shape, scale, thres) 1 - pweibull3(x, shape = shape, scale =
scale, thres = thres)
tweibull3 <- function(x,shape,scale, thres) pweibull3(x, shape = shape, scale =
scale, thres = thres)/(1-pweibull3(x, shape = shape, scale = scale, thres = thres))
mtbfweibull3 <- function(shape,scale, thres) thres+scale*gamma(1+1/shape)

# Distribución gamma
Rgamma <- function(x, rate, shape) 1 -pgamma(x, rate=rate, shape=shape)
tgamma <- function(x, rate, shape) dgamma(x, rate=rate, shape=shape)/(1-pgamma(x,
rate=rate, shape=shape))
mtbfgamma <- function(rate,shape) shape/rate

# Distribución normal
Rnorm <- function(x, mean, sd) 1 - pnorm(x, mean = mean, sd = sd)
tnorm <- function(x, mean, sd) dnorm(x, mean=mean, sd=sd)/(1-pnorm(x, mean=mean,
sd=sd))
mtbfnorm <- function(mean) mean

# Distribución log-normal

```

```

Rlnorm <-function(x,meanlog,sdlog) 1 - plnorm(x, meanlog = meanlog, sdlog = sdlog)
tlnorm <- function(x, meanlog, sdlog) dlnorm(x, meanlog = meanlog, sdlog = sdlog)/(1-
plnorm(x, meanlog = meanlog, sdlog = sdlog))
mtbflnorm <- function(meanlog, sdlog) exp(meanlog+(1/2)*sdlog^2)

# Distribución inversa gaussiana
Rinvgauss <- function(x, mean, shape) 1 - pinvgauss(x, mean = mean, shape = shape)
tinvgauss <- function(x, mean, shape) dinvgauss(x, mean=mean, shape=shape)/(1-
pinvgauss(x, mean=mean, shape=shape))
mtbfinvgauss <- function(mean) mean

# Distribución logística
Rlogis <- function(x, location, scale) 1 -plogis(x, location = location, scale =
scale)
tlogis <- function(x, location, scale) dlogis(x, location = location, scale =
scale)/(1 -plogis(x, location = location, scale = scale))
mtbfllogis <- function(location) location

# Distribución log-logística
Rllogis <- function(x, shape, scale) 1 -pllogis(x, shape = shape, scale = scale)
tllogis <- function(x, shape, scale) dllogis(x, shape = shape, scale = scale)/(1 -
pllogis(x, shape = shape, scale = scale))
mtbfllogis <- function(shape, scale) mllogis(1, shape=shape, scale=scale)

# Distribución de Gumbel
Rgumbel <- function(x, scale,location) 1 -pgumbel(x, scale = scale,location =
location)
tgumbel <- function(x, scale,location) dgumbel(x, scale = scale,location =
location)/(1 -pgumbel(x, scale = scale, location = location))
mtbfgumbel <- function(scale,location) location+scale*0.57721566

```

## Listado de código: Confiabilidad Ui.R

```

shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      selectInput("dis", label = "Distribución", choices =
c('exponencial','weibull','weibull3','gamma','normal', 'log-normal','logística','log-
logística','gumbel')),
      conditionalPanel(condition = "input.dis == 'exponencial'",
        numericInput("lambda","lambda:",min=0,value = 0.9)
      ),
      conditionalPanel(condition = "input.dis == 'logística'",
        numericInput("mdata","m:",min=0,value = 7),
        numericInput("sdata","s:",min=0,value = 2)
      ),
      conditionalPanel(condition = "input.dis == 'log-logística'",
        numericInput("allogis","a:",min=0,value = 1.1),
        numericInput("sllogis","s:",min=0,value = 0.5)
      ),
      conditionalPanel(condition = "input.dis == 'gumbel'",
        numericInput("adata","scale:",min=0,value = 2),
        numericInput("bdata","location:",min=0,value = 4)
      ),
      conditionalPanel(condition = "input.dis == 'normal'",
        numericInput("mean","mean:",min=0,value = 5),
        numericInput("sd","sd:",min=0,value = 1)
      )
    )
  )
)

```

```

),
  conditionalPanel(condition = "input.dis == 'weibull'",
    numericInput("aweibull", "alpha:", min=0, value = 4),
    numericInput("bweibull", "beta:", min=0, value = 7)
  ),
),
  conditionalPanel(condition = "input.dis == 'weibull3'",
    numericInput("aweibull3", "alpha:", min=0, value = 4),
    numericInput("bweibull3", "beta:", min=0, value = 7),
    numericInput("gweibull3", "gamma:", min=0, value = 1)
  ),
),
  conditionalPanel(condition = "input.dis == 'gamma'",
    numericInput("agamma", "rate:", min=0, value = 2),
    numericInput("sgamma", "shape:", min=0, value = 7)
  ),
),
  conditionalPanel(condition = "input.dis == 'log-normal'",
    numericInput("meanlog", "meanlog:", min=0, value = 0.7),
    numericInput("sdlog", "sdlog:", min=0, value = 1)
  ),
),
  fluidRow(column(6, numericInput("adatos", "Eje min x:", min=0, value = 0)),
    column(6, numericInput("ndatos", "Eje max x:", min=0, value = 10))
  ),
),
  numericInput("ylim", "Eje y:", min=0, value = 1),
  numericInput("xdato", "tiempo a evaluar", min=0, value = 0),
  verbatimTextOutput("texto")
),
mainPanel(
  plotOutput("distPlot", height="490px")
)
)
))

```

### Listado de código: Confiabilidad server.R

```

shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    if (is.na(input$ylim) || is.na(input$xdato)
||is.na(input$ndatos)||is.na(input$adatos)){
      ggplot+geom_blank()+theme_classic()
    } else{
      adatos <- input$adatos
      ndatos <- input$ndatos
      ylim <- input$ylim
      xdato <- input$xdato

      if (input$dis=='exponencial'){
        if (is.na(input$lambda)){ggplot+geom_blank()+theme_classic()} else{
          ggplot(data.frame(x = adatos:ndatos), aes(x))+
            geom_vline(xintercept=mtbfexp(input$lambda), linetype="dotted")+
            geom_function(fun = dexp, args = list(rate =
input$lambda), linetype="longdash")+
            geom_function(fun = Rexp, args = list(rate = input$lambda))+
            geom_function(fun = texp, args = list(rate =
input$lambda), linetype="dotted", color="red")+
            labs(caption = TeX(paste("$\\lambda = ", input$lambda, "$")),

```

```

        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
    }
}

else if (input$dis=='logística'){
  if (is.na(input$mdata)||is.na(input$sdata)){ggplot+geom_blank()+theme_classic()}
else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfllogis(location = input$mdata),linetype="dotted")+
    geom_function(fun = dlogis, args = list(location = input$mdata, scale =
input$sdata),linetype="longdash")+
    geom_function(fun = Rlogis, args = list(location = input$mdata, scale =
input$sdata))+
    geom_function(fun = tlogis, args = list(location = input$mdata, scale =
input$sdata),linetype="dotted",color="red")+
    labs(caption = TeX(paste("$\mu = ",input$mdata,"$, $s = ",input$sdata,"$")),
        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}

else if (input$dis=='log-logística'){
  if
(is.na(input$allogis)||is.na(input$sllogis)){ggplot+geom_blank()+theme_classic()}
else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfllogis(shape = input$allogis, scale =
input$sllogis),linetype="dotted")+
    geom_function(fun = dllogis, args = list(shape = input$allogis, scale =
input$sllogis),linetype="longdash")+
    geom_function(fun = Rllogis, args = list(shape = input$allogis, scale =
input$sllogis))+
    geom_function(fun = tllogis, args = list(shape = input$allogis, scale =
input$sllogis),linetype="dotted",color="red")+
    labs(caption = TeX(paste("$\lambda = ",input$allogis,"$, $s =
",input$sllogis,"$")),
        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}

else if (input$dis=='gumbel'){
  if (is.na(input$adata)||is.na(input$bdata)){ggplot+geom_blank()+theme_classic()}
else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfgumbel(scale = input$adata, location =
input$bdata),linetype="dotted")+
    geom_function(fun = dgumbel, args = list(scale = input$adata, location =
input$bdata),linetype="longdash")+
    geom_function(fun = Rgumbel, args = list(scale = input$adata, location =
input$bdata))+
    geom_function(fun = tgumbel, args = list(scale = input$adata, location =
input$bdata),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$a=",input$adata,"\\, b=",input$bdata,"$")),
        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}

else if (input$dis=='normal'){

```

```

if (is.na(input$mean)||is.na(input$sd)){ggplot+geom_blank()+theme_classic()} else {
  ggplot(data.frame(x = adatos:ndatos), aes(x)) +
    geom_vline(xintercept=mtbfnorm(mean = input$mean),linetype="dotted")+
    geom_function(fun = dnorm, args = list(mean = input$mean, sd =
input$sd),linetype="longdash")+
    geom_function(fun = Rnorm, args = list(mean = input$mean, sd = input$sd))+
    geom_function(fun = tnorm, args = list(mean = input$mean, sd =
input$sd),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$\\mu = ",input$mean,"$, $\\sigma =
",input$sd,"$")),
      x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}

else if (input$dis=='log-normal'){
  if (is.na(input$meanlog)||is.na(input$sdlog)){ggplot+geom_blank()+theme_classic()}
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbflnorm(meanlog = input$meanlog, sdlog =
input$sdlog),linetype="dotted")+
    geom_function(fun = dlnorm, args = list(meanlog = input$meanlog, sdlog =
input$sdlog),linetype="longdash")+
    geom_function(fun = Rlnorm, args = list(meanlog = input$meanlog, sdlog =
input$sdlog))+
    geom_function(fun = tlnorm, args = list(meanlog = input$meanlog, sdlog =
input$sdlog),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$\\mu = ",input$meanlog,"$, $\\sigma =
",input$sdlog,"$")),
      x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }

else if (input$dis=='weibull'){
  if
(is.na(input$aweibull)||is.na(input$bweibull)){ggplot+geom_blank()+theme_classic()}
else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfweibull(shape = input$aweibull, scale =
input$bweibull),linetype="dotted")+
    geom_function(fun = dweibull, args = list(shape = input$aweibull, scale =
input$bweibull),linetype="longdash")+
    geom_function(fun = Rweibull, args = list(shape = input$aweibull, scale =
input$bweibull))+
    geom_function(fun = tweibull, args = list(shape = input$aweibull, scale =
input$bweibull),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$a=",input$aweibull,",\\,
", "b=",input$bweibull,"$")),
      x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}

else if (input$dis=='weibull3'){
  if
(is.na(input$aweibull3)||is.na(input$bweibull3)||is.na(input$gweibull3)){ggplot+geom_
blank()+theme_classic()} else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfweibull3(shape = input$aweibull3, scale =
input$bweibull3, thres = input$gweibull3),linetype="dotted")+

```

```

    geom_function(fun = dweibull3, args = list(shape = input$aweibull3, scale =
input$bweibull3, thres = input$gweibull3),linetype="longdash")+
    geom_function(fun = Rweibull3, args = list(shape = input$aweibull3, scale =
input$bweibull3, thres = input$gweibull3))+
    geom_function(fun = tweibull3, args = list(shape = input$aweibull3, scale =
input$bweibull3, thres = input$gweibull3),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$a=",input$aweibull3,"\\,
", "b=",input$bweibull3,"\\, ", "gamma=",input$gweibull3,"$")),
        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}
else if (input$dis=='gamma'){
  if
(is.na(input$agamma)||is.na(input$sgamma)){ggplot+geom_blank()+theme_classic()} else{
  ggplot(data.frame(x = adatos:ndatos), aes(x))+
    geom_vline(xintercept=mtbfgamma(rate = input$agamma, shape =
input$sgamma),linetype="dotted")+
    geom_function(fun = dgamma, args = list(rate = input$agamma, shape =
input$sgamma),linetype="longdash")+
    geom_function(fun = Rgamma, args = list(rate = input$agamma, shape =
input$sgamma))+
    geom_function(fun = tgamma, args = list(rate = input$agamma, shape =
input$sgamma),linetype="dotted",color="red")+
    labs( caption = TeX(paste("$\\alpha = ",input$agamma,"$, $\\beta =
",input$sgamma,"$")),
        x = "tiempo", y = "probabilidad")+ylim(0, ylim)+theme_classic()
  }
}
}
})

output$texto <- renderPrint({

  if (input$dis=='exponencial'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rexp(input$xdato,input$lambda)))
    print(paste("tasa de falla: ",texp(input$xdato,input$lambda)))
    print(paste("tiempo medio entre fallas:",mtbfexp(input$lambda)))
  }

  else if (input$dis=='logística'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rlogis(input$xdato,input$mdata,input$sdata)))
    print(paste("tasa de falla: ",tlogis(input$xdato,input$mdata,input$sdata)))
    print(paste("tiempo medio entre fallas:",mtbflogis(input$mdata)))
  }

  else if (input$dis=='log-logística'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rllogis(input$xdato,input$allogis,input$sllogis)))
    print(paste("tasa de falla:
",tllogis(input$xdato,input$allogis,input$sllogis)))
    print(paste("tiempo medio entre
fallas:",mtbfllogis(input$allogis,input$sllogis)))
  }

  else if (input$dis=='gumbel'){

```

```

    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rgumbel(input$xdato,input$adata,input$bdata)))
    print(paste("tasa de falla: ",tgumbel(input$xdato,input$adata,input$bdata)))
    print(paste("tiempo medio entre fallas:",mtbfgumbel(input$adata,input$bdata)))
  }

  else if (input$dis=='normal'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rnorm(input$xdato,input$mean,input$sd)))
    print(paste("tasa de falla: ",tnorm(input$xdato,input$mean,input$sd)))
    print(paste("tiempo medio entre fallas:",mtbfnorm(input$mean)))
  }

  else if (input$dis=='log-normal'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rlnorm(input$xdato,input$meanlog,input$sdlog)))
    print(paste("tasa de falla: ",tlnorm(input$xdato,input$meanlog,input$sdlog)))
    print(paste("tiempo medio entre fallas:",mtbflnorm(input$meanlog,input$sdlog)))
  }

  else if (input$dis=='weibull'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:"
,Rweibull(input$xdato,input$aweibull,input$bweibull)))
    print(paste("tasa de falla:
",tweibull(input$xdato,input$aweibull,input$bweibull)))
    print(paste("tiempo medio entre
fallas:",mtbfweibull(input$aweibull,input$bweibull)))
  }

  else if (input$dis=='weibull3'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:"
,Rweibull3(input$xdato,input$aweibull3,input$bweibull3,input$gweibull3)))
    print(paste("tasa de falla:
",tweibull3(input$xdato,input$aweibull3,input$bweibull3,input$gweibull3)))
    print(paste("tiempo medio entre
fallas:",mtbfweibull3(input$aweibull3,input$bweibull3,input$gweibull3)))
  }

  else if (input$dis=='gamma'){
    print(paste("t: " ,input$xdato))
    print(paste("fiabilidad:" ,Rgamma(input$xdato,input$agamma,input$sgamma)))
    print(paste("tasa de falla: ",tgamma(input$xdato,input$agamma,input$sgamma)))
    print(paste("tiempo medio entre fallas:",mtbfgamma(input$agamma,input$sgamma)))
  }
})
})

```

### Listado de código: Repuestos global.R

```

shinyUI(fluidPage(

  sidebarLayout(
    sidebarPanel(numericInput("qdata","Número de piezas utilizadas:",min=0,value =
4),

```

```

        numericInput("ldata", "tasa de falla de la pieza:", min=0, value =
0.077),
        fluidRow(column(6, numericInput("n1data", "número de repuestos
mínimo:", min=0, value = 0)),
                column(6, numericInput("n2data", "número de repuestos
máximo:", min=0, value = 3))
                ),
        fluidRow(column(6, numericInput("t1data", "tiempo a
evaluar:", min=0, value = 10)),
                column(6, numericInput("t2data", "eje (t):", min=0, value =
20))
                ),
        tags$h5("Factor de seguridad"),
        verbatimTextOutput("texto")
    ),
    mainPanel(
        plotOutput("distPlot", height="490px")
    )
)
))

```

#### Listado de código: Repuestos Ui.R

```

library(shiny)

Ps <- function(x,q,lambda,n){
  res <- numeric()
  for (i in x){
    E <- exp(-q*lambda*i)
    res <- c(res, E*sum((-log(E))^(0:n))/factorial(0:n)))
  }
  res
}

```

#### Listado de código: Repuestos server.R

```

shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    if
(is.na(input$t1data) || is.na(input$t2data) || is.na(input$n1data) || is.na(input$n2data) ||
is.na(input$qdata))){} else{
      x <- seq(0, input$t2data, by=1)
      n <- seq(input$n1data, input$n2data, by=1)
      q <- input$qdata
      lambda <- input$ldata
      nn <- length(n)

      tt <- numeric()
      for (i in n){
        tt <- cbind(tt, Ps(x, q, lambda, i))
      }
      matplot(tt, type = "l", xlab = "tiempo", ylab = "factor de seguridad")
      legend("bottomleft", legend = paste("n. rep. en bodega:
", n), lty=1:nn, col=1:nn)
    }
  })
  output$texto <- renderPrint({

```



```

        if
(is.na(input$t1data)||is.na(input$t2data)||is.na(input$n1data)||is.na(input$n2data)||
is.na(input$qdata))){} else{
  n <- seq(input$n1data,input$n2data,by=1)
  for (i in 1:length(n)){
    valor <- Ps(input$t1data,input$qdata,input$ldata ,n[i])
    print(paste(n[i],"repuestos : ",round(valor,4)))
  }
}
})

```

})  
Listado de código 2: Ruta crítica global.R

```

library(shiny)
library(DT)
library(readxl)
library(dplyr)
library(DiagrammeR)
library(criticalpath)
options(shiny.maxRequestSize = 25*1024^2)

```

### Listado de código: Ruta crítica Ui.R

```

shinyUI(fluidPage(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "custom.css"),
    tags$link(rel = "icon", href = "favicon.png")
  ),
  tabsetPanel(
    tabPanel("datos",
      fileInput('file1', 'Subir archivo',
        accept = c('text/csv','text/comma-separated-values',
          'text/tab-separated-values','text/plain','.csv','.tsv','.xlsx'
        )
      ),
      dataTableOutput("contents")),
    tabPanel("gráficos",grVizOutput('ima1,height = "640px"'))
  )
)
)

```

### Listado de código: Ruta crítica server.R

```

shinyServer(function(input, output) {

  datos_falla <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)
    read_excel(inFile$datapath)
  })

  # GGLOT GENERAL
  output$ima1 <- renderGrViz({
    if (is.null(datos_falla()))
      return(NULL)
    datacmp <- datos_falla()
  })

```

```

datacmp$orden -> orden
orden <- orden[!is.na(orden)]
datacmp$tarea -> name
name <- name[!is.na(name)]
datacmp$tiempo-> tiempo
tiempo <- tiempo[!is.na(tiempo)]

datacmp$inicia -> inicia
datacmp$finaliza -> finaliza

n <-length(name)

sch <- sch_new() %>%
  sch_add_activities(
    id = as.integer(orden),
    name = name,
    duration = as.integer(tiempo)
  ) %>%
  sch_add_relations(
    from = as.integer(inicia),
    to = as.integer(finaliza)
  ) %>%
  sch_plan()

ES <- round(sch$activities$early_start,2)
EF <- round(sch$activities$early_finish,2)
LS <- round(sch$activities$late_start,2)
LF <- round(sch$activities$late_finish,2)

CR <- sch$activities$critical
CR <- factor(CR)
levels(CR) <- list(' '=FALSE', 'crítico'=TRUE')

HITO <- sch$activities$milestone
HITO <- factor(HITO)
levels(HITO) <- list(' '=FALSE', '*'=TRUE')

grViz(paste(
  "digraph dot {
graph [layout = dot, compound = true]
node [style = filled, fillcolor = '#f9f9f9', shape = box, fixedsize = false]",
  paste0(sch$activities$id," [label='@1-",1:n,"']",collapse = "\n"),
  "edge [arrowhead = vee]",
  paste0(sch$relations$i_from,"->",sch$relations$i_to, sep = "",collapse = " "),
  "}"
  [1]: paste(sch$activities$name,'\\n','inicia :',ES,'->',EF,'
finaliza:',LS,'->',LF,'\\n',CR,HITO)
"
  ))
})
# MOSTRAR TABLA

output$contents <- renderDataTable({
  datatable(datos_falla(),extensions = c('Responsive'),
    options = list(responsive = TRUE))
})

```

```
)
```

### Listado de código: Confiabilidad boole global.R

```
library(shiny)
```

### Listado de código: Confiabilidad boole Ui.R

```
shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(
      textInput('vec1', 'Ingrese los valoresde R(t) separados por comas', "0.3
0.9 0.98"),
      textInput('vec2', 'Sistema k-out-of-n (R,k,n) ', "0.8 3 5"),
      textInput('vec3', 'Redundancia en espera (lambda,t,n) ', "0.3 12.5 3")
    ),
    mainPanel(
      verbatimTextOutput("oid1"),
      tags$img(src="bloque1.svg",width="200px"),
      verbatimTextOutput("oid2"),
      tags$img(src="bloque2.svg",width="200px"),
      verbatimTextOutput("oid3"),
      tags$img(src="bloque3.svg",width="200px"),
      verbatimTextOutput("oid4"),
      tags$img(src="bloque4.svg",width="200px"),
      verbatimTextOutput("oid5")
    )
  )
))
```

### Listado de código: Confiabilidad boole server.R

```
shinyServer(function(input, output) {

  output$oid1 <- renderPrint({
    cat("Valores ingresados:\n")
    cat(input$vec1)
  })

  output$oid2<-renderPrint({
    x <- as.numeric(unlist(strsplit(input$vec1," ")))
    x <- prod(x)
    cat("Serie:\n")
    print(x)
  })

  output$oid3<-renderPrint({
    x <- as.numeric(unlist(strsplit(input$vec1," ")))
    x <- 1-prod(1-x)
    cat("Paralelo:\n")
    print(x)
  })

  output$oid4<-renderPrint({
    x <- as.numeric(unlist(strsplit(input$vec2," ")))
    Rt <- x[1]
    k <- x[2]
    n <- x[3]
    i <- k:n
  })
})
```

```

Rs <- sum(choose(n,i)*((Rt)^i)*(1-Rt)^(n-i))
cat("k-out-of-n:\n")
print(Rs)
})

output$oid5<-renderPrint({
  x <- as.numeric(unlist(strsplit(input$vec3," ")))
  lambda <- x[1]
  t <- x[2]
  n <- x[3]

  nn <- 0:n
  y <- numeric()
  for (i in t){
    y <- c(y,exp(-lambda*i)*sum((lambda*i)^nn/factorial(nn)))
  }
  cat("redundancia activa:\n")
  print(y)
})
})

```

#### Listado de código: Stock seguridad global.R

```

library(shiny)
library(ggplot2)
library(latex2exp)

```

#### Listado de código: Stock seguridad Ui.R

```

shinyUI(fluidPage(

  sidebarLayout(
    sidebarPanel(sliderInput("zdatos","Nivel de servicio deseado:",min=0,max=1,value
= 0.95),
      fluidRow(column(6,numericInput("udatos","u:",min=0,value = 200)),
        column(6,numericInput("sigmadatos","sigma:",min=0,value = 2))
      ),
    verbatimTextOutput("texto")
  ),
  mainPanel(
    plotOutput("distPlot",height="490px")
  )
)

))

```

#### Listado de código: Stock seguridad server.R

```

shinyServer(function(input, output) {

  output$distPlot <- renderPlot({
    if
(is.na(input$udatos)||is.na(input$sigmadatos)||is.na(input$zdatos)){ggplot+geom_blank
()+theme_classic()} else{
      u <- input$udatos
      sigma <- input$sigmadatos
      z <- qnorm(input$zdatos)

      ST = ceiling(z*sigma)

```

```

ODP = ceiling(u + z*sigma)

ggplot(data.frame(x = c(u-3*sigma, u+3*sigma)), aes(x = x)) +
  stat_function(fun = dnorm, args = list(mean = u, sd = sigma)) +
  geom_vline(xintercept=ODP,linetype="dashed", color = "#00a06e") +
  geom_vline(xintercept=u,linetype="dashed")+
  labs(caption = paste("u =",u,",sigma =",sigma),
       x = "número de repuestos", y = "probabilidad")+theme_classic()
}
})
output$texto <- renderPrint({
  print(paste("punto de pedido: ",ceiling( input$udatos +
qnorm(input$zdatos)*input$sigmadatos)))
  print(paste("stock de seg.:
",ceiling(qnorm(input$zdatos)*input$sigmadatos)))
})
})

```

### Listado de código: Inspecciones global.R

```

library(shiny)
library(ggplot2)

TDT <- function(x,Ti,Tb,c) {x*Ti+c*Tb/x}
TDTpart1 <- function(x,Ti) {x*Ti}
TDTpart2 <- function(x,Tb,c) {c*Tb/x}

```

### Listado de código: Inspecciones Ui.R

```

shinyUI(fluidPage(

  sidebarLayout(
    sidebarPanel(numericInput("cdata","constante c:",min=0,value = 3),
                 numericInput("ndata","n. inspecciones :",min=0,value = 3),
                 fluidRow(column(6,numericInput("tbdata","tb:",min=0,value = 0.6)),
                          column(6,numericInput("tidata","ti:",min=0,value = 0.9))
                ),
    verbatimTextOutput("texto")
  ),
  mainPanel(
    plotOutput("distPlot",height="490px")
  )
))

```

### Listado de código: Inspecciones server.R

```

shinyUI(fluidPage(

  sidebarLayout(
    sidebarPanel(numericInput("cdata","constante c:",min=0,value = 3),
                 numericInput("ndata","n. inspecciones :",min=0,value = 3),
                 fluidRow(column(6,numericInput("tbdata","tb:",min=0,value = 0.6)),
                          column(6,numericInput("tidata","ti:",min=0,value = 0.9))
                ),
    verbatimTextOutput("texto")
  ),
  mainPanel(
    plotOutput("distPlot",height="490px")
  )
))

```

```
)
))
Listado de código: Árbol de fallas global.R
```

```
library(shiny)
library(DT)
library(readxl)
library(dplyr)
library(DiagrammeR)
library(FaultTree)
options(shiny.maxRequestSize = 25*1024^2)
```

**Listado de código:** Árbol de fallas Ui.R

```
shinyUI(fluidPage(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "custom.css"),
    tags$link(rel = "icon", href = "favicon.png")
  ),
  tabsetPanel(
    tabPanel("datos",
      fileInput('file1', 'Subir archivo',
        accept = c('text/csv','text/comma-separated-values',
          'text/tab-separated-values','text/plain','.csv','.tsv','.xlsx'
        )
      ),
    dataTableOutput("contents")),
    tabPanel("gráficos",grVizOutput('ima1',height = "640px"))
  )
)
```

**Listado de código 2:** Árbol de fallas server.R

```
shinyServer(function(input, output) {

  datos_falla <- reactive({
    inFile <- input$file1
    if (is.null(inFile))
      return(NULL)
    read_excel(inFile$datapath)
  })

  # GGLOT GENERAL
  output$ima1 <- renderGrViz({
    if (is.null(datos_falla()))
      return(NULL)
    a <- datos_falla()

    n <- dim(a)[1]
    tree1 <- ftree.make(type="priority",reversible_cond=TRUE,name=a$name[1])

    for(i in 1:n){
      if (!is.na(a$priority[i]) && a$priority[i]=="and"){
```

```

    tree1 <- addLogic(tree1, at=a$at[i], type=a$priority[i], name=a$name[i])
  } else if (!is.na(a$priority[i]) && a$priority[i]=="or"){
    tree1 <- addLogic(tree1, at=a$at[i], type=a$priority[i], name=a$name[i])
  } else if (!is.na(a$priority[i]) && a$priority[i]=="inhibit"){
    tree1 <- addLogic(tree1, at=a$at[i], type=a$priority[i], name=a$name[i])
  }

  if(!is.na(a$mttf[i]) && !is.na(a$mttr[i])&& !is.na(a$inspect[i])){
    tree1 <- addLatent(tree1,
at=a$at[i],name=a$name[i],inspect=a$inspect[i],mttr=a$mttr[i],mttf=a$mttf[i])
  } else if(!is.na(a$mttf[i]) && is.na(a$mttr[i])&& is.na(a$inspect[i])){
    tree1 <- addDemand(tree1, at=a$at[i], name=a$name[i], mttf=a$mttf[i])
  }else if (!is.na(a$prob[i])){
    tree1 <- addProbability(tree1, at=a$at[i],name=a$name[i],prob=a$prob[i])
  }
}
tree2 <- ftree.calc(tree1)

id <- tree2$ID
name <- tree2$Name
CFR <- round(tree2$CFR,7)
CFR[CFR==1]<-NA
PBF <- round(tree2$PBF,7)
PBF[PBF==1]<-NA
CRT <- round(tree2$CRT,7)
CRT[CRT==1]<-NA
inicia <-tree2$GParent[2:9]
termina <-tree2$ID[2:9]

n <-length(a)[1]

a$priority
HITO <- a$priority
HITO <- as.factor(HITO)
HITO[is.na(HITO)]<-"-"

grViz(paste(
  "digraph dot {
graph [layout = dot, compound = true]
node [style = filled, fillcolor = '#f9f9f9', shape = box, fixedsize = false]",
  paste0(id," [label='@1-",1:9,"']",collapse = "\n"),
  "edge [arrowhead = vee]",
  paste0(inicia,"->",termina, sep = "",collapse = " "),
  "}")
[1]: paste(name, '\n', 'tasa de falla :', CFR, '\n', 'probabilidad
:', PBF, '\n', 'tiempo para reparar :', CRT, '\n', '_____', '\n', HITO)
"
  ))
})

# MOSTRAR TABLA
output$contents <- renderDataTable({
  datatable(datos_falla(),extensions = c('Responsive'),
    options = list(responsive = TRUE))
})
})

```