



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA AUTOMOTRIZ

**OPTIMIZACIÓN DE LA CONFIABILIDAD DE FLOTAS DE TRANSPORTE CON
EL MÉTODO CROW AMSAA A TRAVÉS DE HERRAMIENTAS DE SOFTWARE
LIBRE**

Trabajo de titulación previo a la obtención del
título de Ingeniero Automotriz

AUTORES: JONNATHAN ISRAEL VALLADARES CASTILLO
RICHARD VINICIO SAQUIPAY LOJANO
TUTOR: ING. CRISTIAN LEONARDO GARCÍA GARCÍA, PhD.

Cuenca - Ecuador
2024

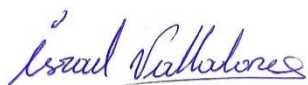
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jonathan Israel Valladares Castillo con documento de identificación N° 0107523730 y Richard Vinicio Saquipay Lojano con documento de identificación N° 0106691074; manifestamos que:

Somos autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 24 de julio del 2024

Atentamente,



Jonathan Israel Valladares Castillo

0107523730



Richard Vinicio Saquipay Lojano

0106691074

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Jonathan Israel Valladares Castillo con documento de identificación N° 0107523730 y Richard Vinicio Saquipay Lojano con documento de identificación N° 0106691074, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Optimización de la confiabilidad de flotas de transporte con el método Crow AMSAA a través de herramientas de software libre”, el cual ha sido desarrollado para optar por el título de: Ingeniero Automotriz en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 24 de julio del 2024

Atentamente,

Jonathan Israel Valladares Castillo

0107523730

Richard Vinicio Saquipay Lojano

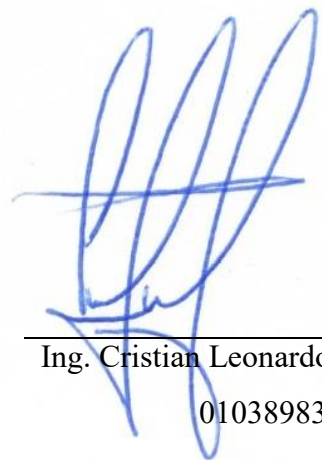
0106691074

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Cristian Leonardo García García con documento de identificación N° 0103898318, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: OPTIMIZACIÓN DE LA CONFIABILIDAD DE FLOTAS DE TRANSPORTE CON EL MÉTODO CROW AMSAA A TRAVÉS DE HERRAMIENTAS DE SOFTWARE LIBRE, realizado por Jonathan Israel Valladares Castillo con documento de identificación N° 0107523730 y por Richard Vinicio Saquipay Lojano con documento de identificación N° 0106691074, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 24 de julio del 2024

Atentamente,



Ing. Cristian Leonardo García García

0103898318

Dedicatoria

Dedico esta tesis a mi Dios por haberme dado esas fuerzas necesarias para cumplir este logro y a mi familia. A mis Abuelos que siempre me brindaron su cariño, su amor y sus valores. A mis padres, por sus sacrificios para que todos mis sueños se hagan realidad y su amor condicional de igual manera de mi hermana. A mis tíos y tías que con sus consejos supieron guiarme. A mi primos y primas, por su constante ánimos y mis amigos y compañeros que siempre me brindaron sus buenos deseos. Finalmente, mi a tutor y profesores que contribuyeron a este logro y aquellas personas que creyeron en mí.

Jonnathan Israel Valladares Castillo

Dedicatoria

Dedico mi tesis a Dios por haberme bendecido con una familia tan maravillosa y por poner en mi vida a personas tan especiales que han sido mi guía y mi apoyo incondicional.

A mi madre, Lucía, cuyo amor incondicional y paciencia infinita han sido el pilar sobre el cual he construido mis sueños. Gracias por enseñarme el valor del esfuerzo y la dedicación, y por creer en mí incluso en los momentos más difíciles.

A mi padre, Ángel, por ser mi ejemplo de perseverancia y fortaleza. Tus palabras de aliento y tu apoyo inquebrantable me han guiado en cada paso de este camino. Gracias por enseñarme que, con trabajo duro y determinación, todo es posible.

A mis hermanos, Pedro y Tamara, ³⁷ por ser mi fuente constante de inspiración y por compartir conmigo no solo risas y alegrías, sino también desafíos y aprendizajes. Su apoyo y amor fraternal han sido fundamentales en mi vida.

A mi novia, Jacqueline, por tu amor, comprensión y por estar siempre a mi lado. Gracias por tu paciencia y por darme fuerzas cuando más lo necesitaba. Tu presencia ha sido un faro de luz y esperanza en este viaje.

Y a mis abuelos, Inés, Luz, y abuelo Alfonso por su amor y sabiduría. Gracias por sus abrazos cálidos y sus consejos que siempre me han guiado.

Richard Vinicio Saquipay Lojano

Agradecimiento

Deseo expresar mi profundo agradecimiento, a Dios por haberme cuidado, darme esa valentía, esa fuerza y ganas de nunca rendirme con el fin de lograr mi objetivo a pesar de todas las adversidades que se presentaron en mi camino. A mi abuela Gloria, con su cariño, apoyo y sus valores permitieron que sea un buen chico logrando así ser lo que soy. A mis padres, Israel y Magaly, y a mi hermana, quienes, a pesar de la distancia, siempre me brindaron su cariño y apoyo emocional, haciendo todo lo posible para que nunca me faltara nada y permitiéndome culminar con mis estudios. A mis tíos, tías y el resto de la familia que nunca desconfiaron de mí, me brindaron sus consejos y sobre todo siempre me apoyaron. Amigos y compañeros que me ofrecieron su confianza en todo este trayecto.

Expresar mi gratitud a mi tutor de tesis, Ing. Cristian Garcia, por su paciencia, su apoyo constante y por haberme guiado a lo largo de este proceso ya que su experiencia y su sabiduría permitieron acabar con este trabajo. De igual manera un sincero agradecimiento a todos los docentes de la Universidad Politécnica Salesiana en especial de mi facultad de Ingeniería Automotriz.

Gracias a todos

Jonnathan Israel Valladares Castillo

Agradecimiento

Agradezco a Dios por darme a las personas que han hecho realidad de mis sueños como lo son mis padres Ángel Saquipay y Lucia Lojano que con gran sacrificio han cuidado y apoyado en todas instancias de mi vida, A mis hermanos Pedro y Tamara Saquipay que a pesar de la distancia siempre los he sentido presente, A mi novia Jacqueline Tenesaca quien a estado conmigo en buenas y malos momentos de mi vida, todos se han convertido en una fuente de inspiración para continuar sin rendirme y los estoy muy agradecidos.

También quiero agradecer a todos los docentes que han aportado para mi desarrollo estudiantil en especial a mi tutor de tesis, Ing. Cristian García por compartir de sus conocimientos dentro de las aulas como fuera de ellas, y por último a mi compañero de tesis Israel Valladares quien a través del tiempo se ha ido ganando mi confianza y convirtiéndose en un buen amigo.

Richard Vinicio Saquipay Lojano

Resumen

La confiabilidad es esencial para evaluar la capacidad de un sistema de operar sin fallos durante un periodo determinado, especialmente en la gestión de flotas de transporte, donde la eficiencia y la seguridad dependen de la confiabilidad de los vehículos. Los métodos más destacados para su cálculo incluyen el análisis de la distribución Weibull y el modelo Crow-AMSAA. Weibull es ideal para productos no reparables, proporcionando información valiosa sobre la tasa de fallas a lo largo del tiempo y la vida útil del producto. Por otro lado, Crow-AMSAA es más adecuado para sistemas reparables, como los vehículos de una flota, ya que permite identificar tendencias de fallas y estimar parámetros clave como la tasa de fallas (Λ) y el tiempo medio entre fallas (MTBF), ofreciendo una visión dinámica y precisa de la confiabilidad del sistema a lo largo del tiempo. En el contexto de las flotas de transporte, la aplicación del modelo Crow-AMSAA facilita la planificación del mantenimiento preventivo y la reducción de tiempos de inactividad, permitiendo una gestión más eficaz y eficiente de los recursos. Para implementar este modelo, se estudió y seleccionó Python como el lenguaje de programación debido a su robustez, versatilidad y la disponibilidad de bibliotecas especializadas en análisis estadístico y confiabilidad. Se desarrolló una herramienta informática en Python que utiliza el método Crow-AMSAA para calcular la confiabilidad, adaptada a las necesidades específicas de las flotas de transporte. Esta herramienta fue sometida a pruebas con datos reales de fallas, demostrando su precisión en las estimaciones y su utilidad en la gestión de la confiabilidad operativa. Se recomienda el uso del modelo Crow-AMSAA sobre Weibull en este contexto, debido a su capacidad para manejar sistemas reparables y ofrecer una evaluación dinámica de la confiabilidad. Esta herramienta representa un avance significativo en la gestión proactiva de la confiabilidad y el mantenimiento de flotas de transporte, proporcionando una base sólida para la toma de decisiones informadas y la mejora continua de los procesos operativos.

Abstract

Reliability is essential for evaluating a system's ability to operate without failures over a given period, especially in fleet management, where vehicle efficiency and safety depend on reliability. The most notable methods for calculating reliability include Weibull distribution analysis and the Crow-AMSAA model. Weibull is ideal for non-repairable products, providing valuable information on the failure rate over time and the product's lifespan. Conversely, Crow-AMSAA is more suitable for repairable systems, like fleet vehicles, as it allows for identifying failure trends and estimating key parameters such as failure rate (Λ) and mean time between failures (MTBF), offering a dynamic and accurate view of the system's reliability over time. In the context of transportation fleets, applying the Crow-AMSAA model facilitates preventive maintenance planning and reduces downtime, enabling more effective and efficient resource management. To implement this model, Python was studied and selected as the programming language due to its robustness, versatility, and availability of specialized libraries for statistical analysis and reliability. A software tool was developed in Python using the Crow-AMSAA method to calculate reliability, tailored to the specific needs of transportation fleets. This tool was tested with real failure data, demonstrating its accuracy in estimates and usefulness in operational reliability management. The Crow-AMSAA model is recommended over Weibull in this context due to its ability to handle repairable systems and provide a dynamic evaluation of reliability. This tool represents a significant advance in proactive reliability management and fleet maintenance, providing a solid foundation for informed decision-making and continuous process improvement.

TABLA DE CONTENIDO

Contenido

Dedicatoria.....	iv
Agradecimiento.....	vi
Resumen	viii
Abstract.....	ix
INTRODUCCIÓN	2
2. PROBLEMA.....	2
19 2.1. Antecedentes	2
2.2. Importancia y Alcances.....	2
2.3. Delimitación.....	4
3. OBJETIVOS	5
3.2. Objetivo General.....	5
3.3. Objetivos Específicos.....	5
CAPÍTULO I	6
4. MARCO TEÓRICO REFERENCIAL	6
4.1. Flota de transportes	6
4.2. Confiabilidad.....	8
4.3. Modelos estadísticos de predicción	9
Modelo Log-normal:.....	10
Procesos de Poisson no homogéneos (NHPP):.....	10
Modelos de Confiabilidad de Cox (Modelos de Riesgos Proporcionales):.....	10
Modelos de Fracaso Competitivo:	10
Modelo Binomial Negativo:	11

4.3. Método Crow AMSAA.....	11
4.3.1. Desarrollo del Método Crow AMSAA.....	11
4.3.2. Modelo matemático Crow AMSAA solución grafica del modelo.....	13
4.4. Modelo Weibull	15
4.4.1. Modelo matemático Weibull.....	16
4.5. Comparación entre los modelos Weibull – Crow AMSAA.....	19
CAPÍTULO II	21
5. MARCO METODOLÓGICO.....	21
5.1. Lenguajes De Programación.....	22
5.2. Software libre.....	23
5.3. Softwares analizados en este proyecto.....	24
5.3.1. JavaScript.....	24
5.3.2. El lenguaje C++	27
5.3.3. Python	30
5.4. Librerías de interés para el cálculo de confiabilidad	32
5.4.1. Reliability.....	32
5.4.2. NumPy	33
5.4.3. Pandas	34
5.4.4. Matplotlib.....	35
5.5. Comparativa Entre Los Lenguajes De Programación.....	35
CAPÍTULO III.....	37
6. Desarrollo de una Herramienta Informática para el Análisis de Confiabilidad.....	37
6.1. Diagrama de flujos.....	37
6.1.1. Diagrama de flujo para modelo Crow AMSAA	38
6.1.2. Diagrama de flujo para modelo Weibull.....	40

6.2. Instalación de Recursos Necesarios	43
6.2.1. Pasos para instalar anaconda.....	Error! Bookmark not defined.
6.2.2. Pasos para lanzar Jupyter Notebook	44
6.2.3. Pasos para codificación en Python 3.....	45
6.3. Codificación para el Modelo Crow AMSAA	47
6.4. Calculo por el modelo Crow AMSAA para comprobar su funcionamiento.....	49
6.5. Codificación por el Modelo Weibull	54
6.6. Calculo por el modelo Weibull para comprobar su funcionamiento.	56
6.7. Comparación entre modelo Weibull y Crow AMSAA.....	60
6.8. Ventajas del modelo Crow-AMSAA frente al modelo Weibull.....	61
6.9. Interfaz de la herramienta	62
6.9.1. Diagrama del flujo de la interfaz	63
6.10. Validación de la herramienta con datos reales por Crow AMSAA	64
6.11. Validación de la herramienta con datos reales por WEIBULL	71
6.12. Conclusión del estudio.....	75
7. CONCLUSIONES	76
8. RECOMENDACIONES.....	78
9. BIBLIOGRAFIA.....	Error! Bookmark not defined.
10. ANEXOS.....	81

ÍNDICE DE TABLAS

Tabla 1 Clasificación de flotas de transporte.....	7
Tabla 2 Diferencias de método Weibull y Crow AMSAA	19
Tabla 3 Cuadro comparativo en base a sus características, modelo de Weibull-Crow AMSAA.	20
Tabla 4 Ranking Software libre	24
Tabla 5 Comparación de lenguajes de programación	36
Tabla 6 Diagrama de flujo	37
Tabla 7 Datos ejemplo 1	49
Tabla 8 Parámetros calculados ejemplo 1 Crow AMSAA	50
Tabla 9 Datos Weibull ejemplo1	56
Tabla 10 parámetros de cálculo ejemplo 1 Weibull.....	57
Tabla 11 ejemeplo2 Crow AMSAA	64
Tabla 12 parámetros de cálculo ejemplo 2 Crow AMSAA	65
Tabla 13 Datos ejemplo 2 Weeibull.....	71
Tabla 14 parámetros de cálculo ejemplo 2 Weibull.....	72

ÍNDICE DE FIGURAS

Figura 1: Modelo de una flota de transporte	6
Figura 2: Evolución del mantenimiento.....	9
Figura 3: Modelo Crow AMSAA	13
Figura 4: Modelo de Weibull.....	16
Figura 5: Reliability	33
Figura 6: NumPy.....	34
Figura 7: pandas.....	34
Figura 8: matplotlib.....	35
Figura 9 diagrama de flujo Crow AMSAA.....	39
Figura 10 Diagrama de flujo Weibull	41
Figura 11 : Anaconda.....	43
Figura 12 Anaconda navigator.....	44
Figura 13 jupyter Notebook	45
Figura 14 select kernel.....	45
Figura 15 interprete Python	46
Figura 16 SMD	46
. Figura 17 ajuste Crow AMSAA ejemplo 1.....	51
Figura 18 confiabilidad y des confiabilidad ejemplo1	52
Figura 19 tasa de fallos	53
Figura 20 MTBF acumulados	54
Figura 21 ajuste Weibull.....	58
Figura 22 tasa de fallos	59
Figura 23 comparación de modelos	60
Figura 24 interfaz.....	62
Figura 25 Flujo del interfaz.....	63
Figura 26 ajuste crow amsaa ejemplo2	67
Figura 27 Confiabilidad ejemplo2	68
Figura 28 Tasa de fallos ejemplo2	69
Figura 29 MTBF ejemplo2	70
Figura 30 ajuste Weibull ejemplo2	73

Figura 31 tasa de fallos	74
--------------------------------	----

INTRODUCCIÓN

El proyecto propuesto sobre “Optimización de la confiabilidad de flotas de transporte con el método Crow AMSAA a través de herramientas de software libre”, Incentiva a la investigación y desarrollo de una herramienta en software libre en la cual se busca calcular la confiabilidad de las flotas de transporte dentro de una empresa, con la finalidad de brindar mantenimiento y de esta manera minimizar costos y tiempos de parada en las flotas.

Se desarrollará una herramienta en un software libre generando un lenguaje de programación específico para el análisis de datos los cuales se obtienen de los históricos de las empresas de flotas de transporte.

INTRODUCCIÓN

2. PROBLEMA

2.1. Antecedentes

Los costos asociados con los softwares de gestión del mantenimiento representan una barrera para muchas empresas, ya sea por limitaciones económicas o por la falta de evidencia sobre los beneficios técnicos y económicos que estos pueden proporcionar. Evaluar la confiabilidad de las flotas de transporte no solo conlleva una mejor organización empresarial, sino que también puede reflejar un crecimiento económico al reducir los tiempos de parada mediante mantenimientos más programados, mejorando así la disponibilidad de los vehículos. Sin embargo, la inversión en software de gestión de la confiabilidad es desalentada por el alto costo de las licencias, lo que limita los logros empresariales.

2.2. Importancia y Alcances

El cálculo de la confiabilidad de los vehículos en función del tiempo o de las veces que un elemento falla y requiere mantenimiento refleja un punto crítico de interés para las empresas. Hasta la fecha, no existen herramientas informáticas desarrolladas en software libre que se enfoquen en el cálculo de la confiabilidad basadas en datos históricos de empresas.

Por lo tanto, trabajar en la propuesta de una herramienta de este tipo, adaptable al contexto operativo de las flotas de transporte, puede brindar una ventaja competitiva a las empresas. Este tema de tesis se relaciona con el objetivo número uno y cuatro del plan de creaciones de oportunidades que hace referencia a garantizar la gestión de las finanzas públicas de manera sostenible y transparente también a la generación de oportunidades en condiciones laborables.

En cuanto a los objetivos de desarrollo sostenible hace referencia al objetivo número cuatro que permite ²³ una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje permanente para todos.

2.3. Delimitación

14 El presente proyecto, se llevará a cabo en la Provincia del Azuay, Ciudad de Cuenca, ubicada al sur del Ecuador, la cual tiene una altitud de 2500 m.s.n.m, una extensión de 70.59 km² y una población aproximada de 580000 habitantes.

El resultado del proyecto se puede aplicar en cualquier región del país o del mundo, tomando en cuenta que se trata de un software libre.

3. OBJETIVOS

3.2. Objetivo General

Desarrollar una herramienta informática utilizando software libre para el cálculo de la confiabilidad mediante el Método Crow-AMSAA.

3.3. Objetivos Específicos

- Realizar un estudio comparativo de las técnicas estadísticas utilizadas para el cálculo de la confiabilidad en flotas de transporte, analizando diferentes métodos.
- Evaluar las distintas herramientas de software libre disponibles, incluyendo varios lenguajes de programación, para determinar la más adecuada para el cálculo de la confiabilidad en flotas de transporte.
- Desarrollar una herramienta informática utilizando un lenguaje de programación en software libre para el cálculo de la confiabilidad con el método Crow-AMSAA, adaptándola a las necesidades específicas de las flotas de transporte.

CAPÍTULO I

4. MARCO TEÓRICO REFERENCIAL

4.1. Flota de transportes

Las flotas de transporte se denominan a un conjunto de vehículos destinados a transportar mercancías o persona que van a depender de la misma empresa. El transporte debe ser profesional por parte del personal que labora, por el mismo hecho de transportar ya sea personas o mercancías, porque es de esencial garantizar el cuidado con la finalidad de generar réditos económicos a la empresa. (Fernández, 2016)

Figura 1: Modelo de una flota de transporte



Tomado de (Latam, 2022)

En estas flotas de transporte el combustible es de vital relevancia en la estructura de los costos, por lo tanto, para el adecuado desarrollo de la empresa es necesario la realización de una gestión eficiente en el consumo de combustible. Para una gestión eficiente en este tipo de empresas la idea fundamental será siempre usar la menor cantidad de recursos posibles. Entonces, dependiendo del tamaño de la flota de transportes tienen por lo general un departamento que se encarga del tráfico, que está encargado de realizar varias tareas como la elección de vehículos, selección de rutas, gestión de cargas, gestión movilidad, costos relacionados con la reparación y cambios de repuestos, y la gestión del consumo de combustible. (Casilla,2018)

Por lo general las flotas de transporte según su tamaño se puede clasificar en:

- **Flotas pequeñas:** se las nombra así cuando es administrado por la familia o son de carácter autónomo, se los considera hasta los 6 vehículos, y todo el trabajo que realizan es para una persona o para una empresa de gran tamaño.

- **Flotas medianas:** estas cuentan con un número que van de 6 a 30 unidades. También se pueden tratar de empresas familiares, pero que han crecido aprovechando en base a la buena gestión y se han especializado en el mercado. Por lo general estas empresas ya tienen varios clientes tanto a nivel nacional como internacional, que, si bien cuentan con un departamento de tráfico de relieve, está gestionado solo por una persona. Sin embargo, a medida que aumente el número de vehículos la estructura de la empresa puede llegar hasta tener talleres de reparación o depósitos de almacenamiento de combustible.

- **Flotas grandes:** son empresas bien estructuradas en todos los ámbitos y cuentan con un número elevado de vehículos. Estos vehículos pueden ser propios, trabajar bajo contrato y autónomos. También pueden llegar a tener varias delegaciones o sucursales en diferentes ciudades o países que van a actuar como operadores logísticos y de distribución de varias marcas posibles. En este sentido el departamento de tráfico suele tener una estructura propia donde cuenta con un gerente al mando, y varios delegados que se encargarán de supervisar los locales y todas las flotas de vehículos que prestan servicios a la empresa. (Fernandez Gomez, 2016)

Tabla 1 Clasificación de flotas de transporte

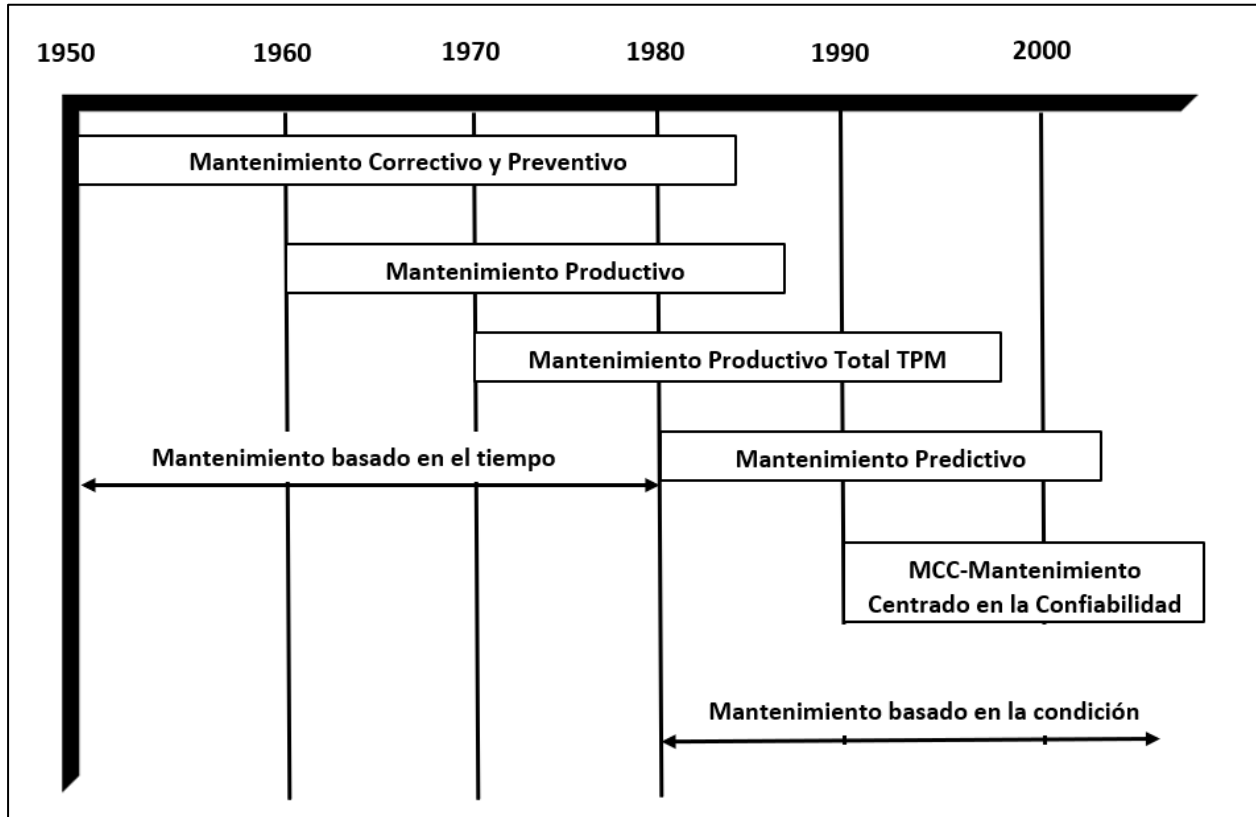
Por el tamaño	Pequeña (=, < 10 vehículos)
	Mediana (=, < 30 vehículos)
	Grande (> 30 vehículos)
Por la aplicación	Transporte de personas
	Transporte de mercadería

	Transporte de maquinaria pesada
	Gestión de obras publicas
	Gestión de actividad agrícola
Tipo de vehículos	Liviano o turismo
	Maquinaria semi pesada
	Maquinaria pesada
Por la distribución geográfica	Mantenimiento taller único
	Mantenimiento en varios talles
Por ruta y vialidad	Fuera de ruta
	Vehículo de carretera
Tomado de: (Fernández, 2016)	

4.2. Confiabilidad

La confiabilidad puede ser definida como la “confianza” que se tiene de que un componente, equipo o sistema desempeñe su función básica, durante un período de tiempo preestablecido, bajo condiciones estándares de operación. Otra definición importante de confiabilidad es; probabilidad de que un ítem pueda desempeñar su función requerida durante un intervalo de tiempo establecido y bajo condiciones de uso definidas. (Scientia, 2006)

Figura 2: Evolución del mantenimiento



Obtenido de (Autores)

4.3. Modelos estadísticos de predicción

En el ámbito del mantenimiento de flotas de vehículos, la implementación de modelos estadísticos de predicción representa un avance significativo hacia la optimización de recursos y la mejora de la confiabilidad. Estos modelos permiten la anticipación de los fallos, la programación de mantenimientos y la asignación de recursos de manera más eficiente, lo cual es crucial en sectores como el transporte y la logística.

Los modelos estadísticos de predicción se han convertido en herramientas indispensables para los profesionales del mantenimiento. Estos modelos permiten el análisis del comportamiento de los equipos y anticipar fallos, lo cual es esencial para la planificación de los mantenimientos que no interrumpan la operatividad normal de las instalaciones. El análisis estadístico proporciona

una base sólida para la toma de decisiones informadas útil para realizar las intervenciones de mantenimiento, basándose en la probabilidad de fallo y no en rutinas preestablecidas. Entre los modelos más importantes se evidencian los siguientes:

Modelo Exponencial: este modelo asume una tasa constante de fallo, lo cual es adecuado para componentes que no envejecen o se deterioran con el tiempo. Es uno de los modelos más simples y se utiliza frecuentemente para datos de vida hasta el primer fallo.

Modelo Log-normal: adecuado para la descripción de los tiempos de fallo donde los datos tienen una distribución sesgada, común en sistemas mecánicos y electrónicos donde los fallos ocurren debido a un desgaste acumulativo.

Procesos de Poisson no homogéneos (NHPP): estos modelos son útiles para datos de fallo de sistemas reparables, donde la tasa de fallo puede cambiar con el tiempo. El Método Crow AMSAA es un ejemplo de este tipo de modelos.

Modelos de Confiabilidad de Cox (Modelos de Riesgos Proporcionales): este modelo puede incorporar múltiples variables explicativas para evaluar el efecto sobre el tiempo hasta llegar al fallo, siendo útil para análisis de supervivencia y estudios de confiabilidad donde los factores de riesgo varían con el tiempo. En el caso de un hospital, este modelo podría aplicarse para analizar el tiempo hasta el fallo de un dispositivo médico, considerando factores como la intensidad de uso y el mantenimiento recibido. El modelo de Cox permite evaluar los factores que impactan en la confiabilidad del dispositivo (Casilla, 2022).

Modelos de Fracaso Competitivo: estos modelos consideran la posibilidad de que existan múltiples modos de fallo, cada uno con sus propias características de riesgo. Permiten un análisis más detallado de las causas potenciales de fallo en sistemas complejos.

Modelo Binomial Negativo: es utilizado para modelar la variabilidad en tasas de fallo de componentes donde los datos de conteo muestran una varianza mayor que la media, lo que es típico en datos de fallo agrupados o sobre dispersos.

4.3. Método Crow AMSAA

El Método Crow AMSAA, un modelo de proceso Poisson no homogéneo con intensidad de ley de potencia, es fundamental en el análisis de la fiabilidad de sistemas reparables y ha evolucionado significativamente desde su introducción por Larry H. Crow en 1975. La literatura reciente sobre el modelo Crow AMSAA revela avances en la incorporación de covariables que varían en el tiempo, lo que permite analizar cómo las condiciones operativas y ambientales influyen en la fiabilidad del producto. (Talkit, 2023)

El modelo Crow AMSAA se basa en la estimación de parámetros a través de métodos como la regresión por rangos y la estimación de máxima verosimilitud (MLE), que se ajusta particularmente bien para pruebas de fiabilidad en sistemas reparables. Estos parámetros, incluyendo la intensidad de fallos (λ) y los parámetros de forma (β) y escala (α), se calculan en función del número de fallos y el tiempo total de prueba. El modelo de Crow-AMSAA es un modelo estadístico desarrollado a partir del historial de fallas y los tiempos acumulados hasta dichas fallas (Casilla, 2022).

4.3.1. Desarrollo del Método Crow AMSAA

El modelo de Crow AMSAA conocida como una herramienta fundamental para el análisis y la predicción de la confiabilidad en una amplia gama de sistemas y aplicaciones industriales. La elección de este modelo se basa en su reconocida eficacia para comprender el comportamiento de

la confiabilidad a lo largo del tiempo, lo cual es fundamental en entornos donde la disponibilidad, la seguridad y el rendimiento son críticos.

Para el desarrollo del método, se aplica un proceso estructurado que puede ser desglosado en los siguientes pasos:

Paso 1. Recolección de datos: este paso es el punto de partida, corresponde a la recopilación de datos históricos de fallos. Estos datos deben incluir los tiempos en los que se han producido fallos en el sistema a lo largo de su periodo operativo. Es crucial que estos datos sean precisos y estén bien documentados, ya que formarán la base del análisis de confiabilidad.

Paso 2. Análisis preliminar: antes de aplicar el modelo, se realiza un análisis preliminar de los datos para verificar su adecuación y limpieza. Esto puede incluir la verificación de la integridad de los datos y la eliminación de anomalías o datos atípicos que podrían distorsionar el análisis.

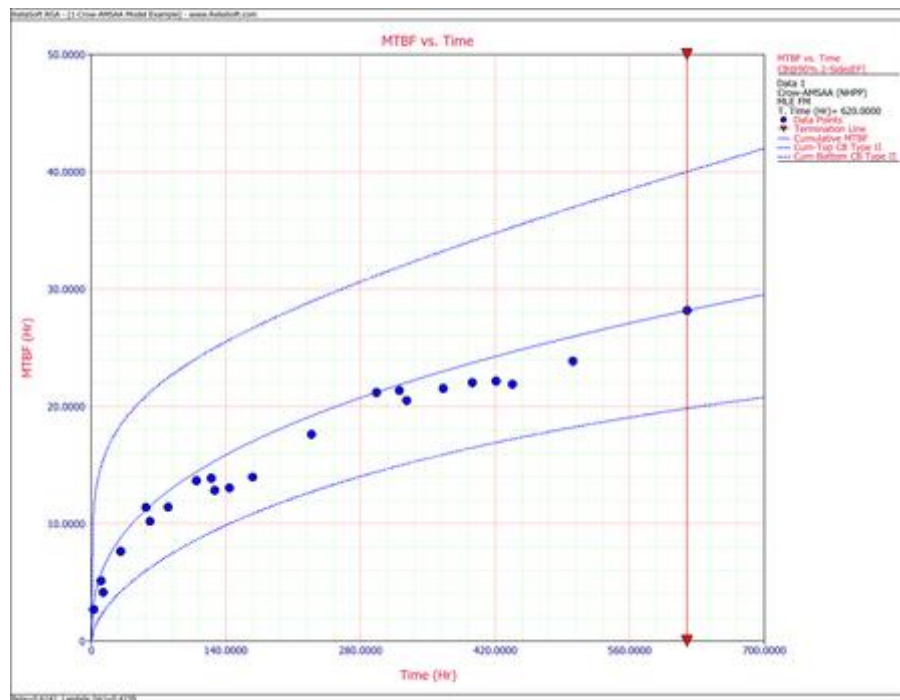
Paso 3. Estimación de parámetros: utilizando los datos de fallos, se estiman los parámetros del modelo Crow AMSAA, principalmente el parámetro de escala (α) y el parámetro de forma (β). Estos parámetros se calculan comúnmente usando métodos de estimación de máxima verosimilitud (MLE) o regresión por rangos. El parámetro de forma (β) indica el tipo de tasa de fallo (decreciente, constante o creciente), y el parámetro de escala (α) relaciona la intensidad de fallo con el tiempo (Villanueva, 2019).

Paso 4. Validación del Modelo: una vez estimados los parámetros, se procede a validar el modelo para asegurar que describe adecuadamente los datos de fallos. Esto incluye la comparación de las tasas de fallos observadas con las predicciones del modelo. Herramientas como pruebas de bondad de ajuste y análisis de residuos pueden emplearse para evaluar si el modelo es un buen ajuste (Wang, 2020).

Paso 5. Interpretación y Aplicación: con los parámetros del modelo validados, los resultados se interpretan para tomar decisiones de mantenimiento. Si β es mayor que 1, indica que la tasa de fallos está aumentando, lo que puede justificar intervenciones de mantenimiento más frecuentes o una revisión del diseño del sistema. Si β es menor que 1, sugiere una mejora en la fiabilidad con el tiempo (Casilla, 2022).

Finalmente, el proceso no termina con una sola aplicación del modelo. La confiabilidad de los sistemas es dinámica, por lo que el modelo Crow AMSAA debe aplicarse periódicamente con nuevos datos de fallos para ajustar las estrategias de mantenimiento y respuesta a las condiciones

Figura 3: Modelo Crow AMSAA



Tomando de (Luzuriaga, 2022)

4.3.2. Modelo matemático Crow AMSAA ¹³ solución grafica del modelo

El modelo de la función para medir una rata de falla instantánea de tiempo acumulativo está determinado por:

$$p(t) = \lambda \beta t^{\beta-1}$$

P(t): solución grafica del modelo.

(λ): la intensidad de fallos.

(β): parámetros de forma.

t: tiempo.

Sistemas reparables

$$n(t) = \lambda \times t^\beta$$

n(t): número de fallos acumulados con respecto al tiempo.

λ: ordenada en el origen, parámetro escala.

β: pendiente de la recta, parámetro de forma.

Regresión de rango

$$\beta = \frac{n}{(n-1) \ln t_n - \sum_{i=1}^{n-1} \ln t_i}$$

$$\lambda = \frac{n}{T^\beta}$$

Máximo verisimilitud

$$\beta = \ln\left(\frac{n \sum n(t)t - \sum n(t) \sum t}{n \sum t^2 - (\sum t)^2}\right)$$

$$\lambda = \ln(n(t) - t^\beta)$$

Tomando logaritmos en la ecuación anterior:

$$\ln n(t) = \lambda + \beta \times \ln t$$

Si $\beta > 1$, tasa de creciente, Deterioro del sistema.

Si $\beta < 1$, la tasa de fallo decreciente. Fiabilidad creciente.

Un $\beta = 1$ tasa de fallo constante.

Para $t = 1$; $\lambda = 0.49$

Predicción de fallos

$$n(t) = \lambda \times t^{\beta}$$

N= Numero de Fallas o costos.

t = Número de meses, días, horas.

λ = Valor de primera Falla o costo.

β = Pendiente constante y está relacionado con la rata de mejoramiento.

Tiempo futuro del próximo fallo

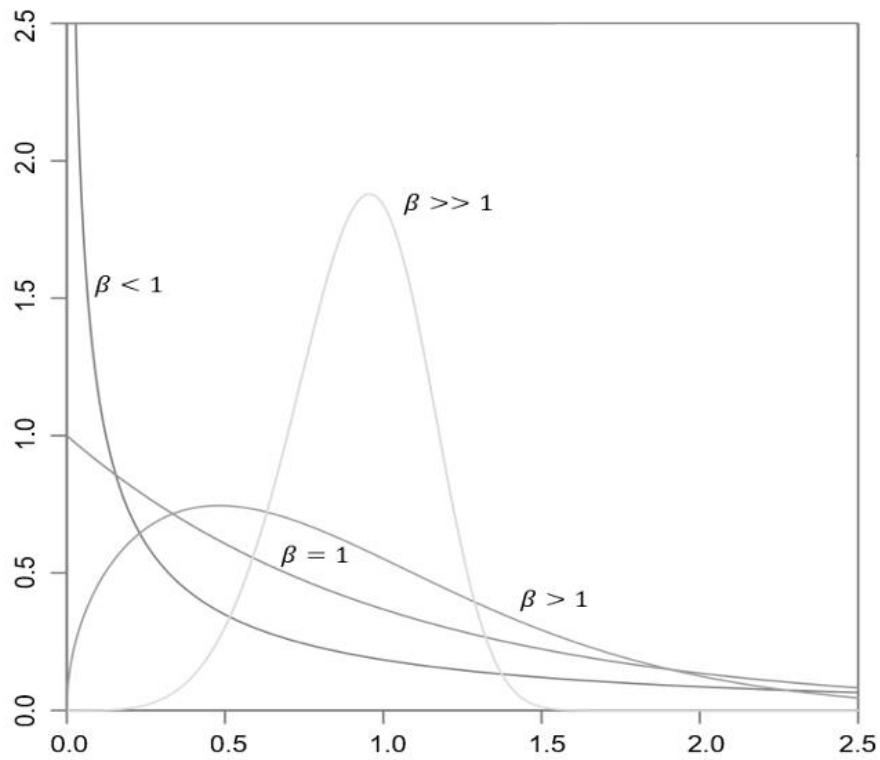
$$t + k = \left(\frac{n(t) + 1}{\lambda} \right)^{1/\beta}$$

k el tiempo que transcurre desde el instante de tiempo hasta el siguiente fallo.

4.4. Modelo Weibull

muy utilizado por su flexibilidad para modelar los diversos tipos de datos de fallo, ya que puede adaptarse a diferentes patrones de tasa de fallo aumentando, disminuyendo o siendo constante, a través de sus parámetros de forma y escala. Por ejemplo, un modelo Weibull es ideal para analizar la vida útil de un componente mecánico en un automóvil, como la transmisión, donde el riesgo de fallo puede aumentar con el tiempo debido al desgaste. Si un componente tiene un parámetro de forma mayor que 1, indica un aumento en la tasa de fallo a medida que el componente envejece. (Prasety & Kindhi,2023)

Figura 4: Modelo de Weibull



Tomado de (Ochoa & Villa, 2022)

4.4.1. Modelo matemático Weibull

Modelo de 3 parámetros

Considerando los parámetros para calcular la distribución de densidad de Weibull:

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

f(t): función de densidad de probabilidad.

β : es el parámetro de forma.

α : es el parámetro de escala.

γ : es el parámetro de localización t : tiempo de análisis de datos.

t: tiempo de análisis de datos.

Modelo 2 parámetros

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

Donde:

f(t): función de densidad de probabilidad.

β: es el parámetro de forma.

α: es el parámetro de escala.

t: tiempo de análisis de datos.

Los parámetros de Weibull se obtendrá los valores que influirán en al cálculo de la confiabilidad de esta forma se puede encontrar la confiabilidad según la distribución de Weibull.(Sequeda & Diaz, 2008)

$$R(t) = e^{-\left(\frac{t-\gamma}{\alpha}\right)^{\beta}}$$

Donde:

R(t): Confiabilidad.

β: es el parámetro de forma.

α: es el parámetro de escala.

γ: es el parámetro de localización.

t: tiempo de análisis de datos.

Distribución de probabilidad acumulada F(t):

$$F(t) = 1 - e^{\left(-\frac{t}{\alpha}\right)^{\beta}}$$

Donde:

F(t): distribución de probabilidad acumulada.

β : es el parámetro de forma.

α : es el parámetro de escala.

t: tiempo de análisis de datos.

La función de la tasa de fallos se representa por:

$$\lambda(t) = \frac{\beta}{\alpha} \left(\frac{t - y}{\alpha}\right)^{\beta-1} = \frac{f(t)}{R(t)}$$

$\lambda(t)$: tasa de fallos en el tiempo.

β : es el parámetro de forma.

α : es el parámetro de escala.

γ : es el parámetro de localización.

t: tiempo de análisis de datos.

la tasa de falla constante ($\lambda(t) = \text{constante}$), entonces se puede calcular de la siguiente manera:

$$\lambda(t) = \frac{\#fallas}{t_{operación}}$$

Donde:

$\lambda(t)$: tasa de fallos en el tiempo.

#fallas: número de fallas.

t operación: es el tiempo en operación del análisis.

4.5. Comparación entre los modelos Weibull – Crow AMSAA.

Dentro de la tabla 2 se puede apreciar las diferentes características que poseen cada modelo ya que ambos modelos se usan para el análisis y estimación de fallas y posterior a su elección adecuada para realizar los cálculos.

Tabla 2 Diferencias de método Weibull y Crow AMSAA

MODELO DE WEIBULL	MODELO DE CROW AMSAA
Su desempeño es superior en las fallas más tempranas al analizar los tiempos de operación hasta el fallo.	Su labor se enfoca en el estudio de los modos de falla en sistemas combinados.
No se consideran las causas y los modos de falla.	Utilizado para evaluar y supervisar el aumento de la confiabilidad.
Su estudio se enfoca en modos de falla individuales o en identificar el eslabón más débil.	Se utiliza para representar el comportamiento de sistemas reparables, es decir, que permiten modelar un componente que ha sido reemplazado varias veces.
Permite la flexibilidad para ajustarse a las distintas etapas de la curva de la bañera.	Permite gestionar combinaciones de modos de falla.

Tomado de: (Ochoa & Villa, 2022)

Una vez identificado y analizado las diversas diferencias entre Weibull y Crow AMSAA, se elabora un cuadro comparativo detallado de las características entre estos dos modelos que están presentes en esta tabla 3. Con base a los objetivos de nuestro plan de trabajo, se tomará la decisión de emplear una metodología específica.

Tabla 3 Cuadro comparativo en base a sus características, modelo de Weibull-Crow AMSAA

CARACTERÍSTICAS	MODELO	
	Weibull	Crow AMSAA
<i>Tiempos de falla</i>	Tiempos entre fallas	Tiempos acumulados
<i>Campo de operación</i>	Elementos	Sistemas combinados
<i>Uso en elementos reemplazados</i>	No	Si
<i>Análisis de la confiabilidad</i>	Confiabilidad de un tiempo concreto	Crecimiento acumulado de confiabilidad
<i>Toma en cuenta causas y modos de falla</i>	No	Si
<i>Adaptación a la curva de la bañera</i>	Si	Si

Tomado de: (Ochoa & Villa, 2022)

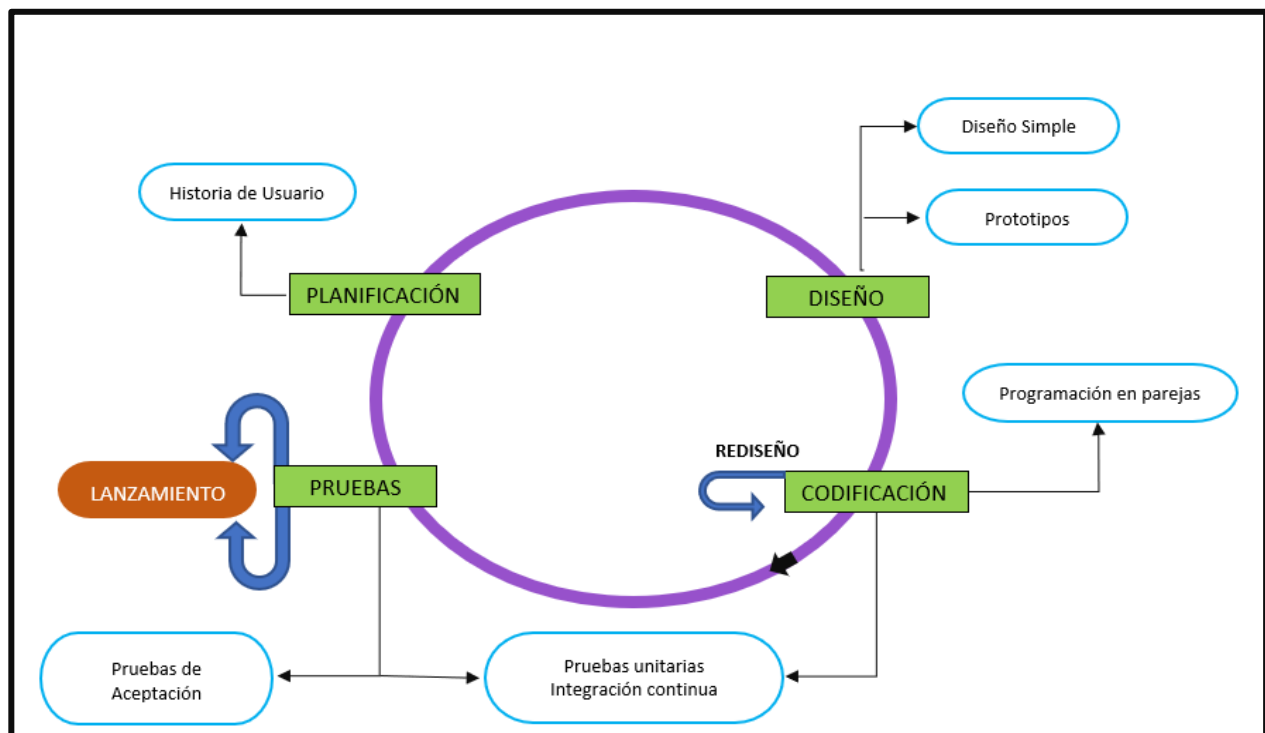
De la misma manera estos dos modelos estadísticos pueden ser puestos a comparación con distintas distribuciones que serán aplicados de acuerdo a su campo de aplicación y a sus distintas características.

CAPÍTULO II

5. MARCO METODOLÓGICO

En esta sección, se describen los procedimientos y técnicas utilizados para aplicar el modelo Crow - AMSAA en el análisis de la confiabilidad. Este método es fundamental para evaluar el crecimiento de la confiabilidad a través del tiempo y permite identificar tendencias en los datos de fallas acumuladas. En base a nuestro proyecto se estructura en varias etapas, comenzando con la preparación de los datos y finalizando con el análisis estadístico y la interpretación de los resultados. En la siguiente grafica se muestra detalladamente los pasos que se seguirá para la creación de la herramienta para análisis de confiabilidad en el software libre, esta metodología se basa en un conjunto de reglas y principios que ha permitido a lo largo de su historia manejar un buen funcionamiento de desarrollo de software.

Figura 5: Metodología para creación de una herramienta informática en software libre



Obtenido de (Autores)

Planificación: En esta fase se incluyen varios principios, tales como la ⁵ realización de pruebas, el proceso de planificación, la presencia del cliente en el sitio y la programación.

Diseño: ¹¹ Facilita la integración continua, la refactorización (evaluando el diseño del sistema a lo largo de todo el proyecto y realizando ajustes en el código cuando sea necesario) y la entrega de pequeñas actualizaciones.

Codificación: En esta categoría se establecen ⁵ criterios como la creación de un diseño sencillo, el uso de tarjetas CRC (Clase, Responsabilidad y Colaboración), y la elaboración de una metáfora del sistema o una historia completa.

Pruebas: Todos los módulos deben pasar las pruebas unitarias, al encontrar un error debe ser corregido inmediatamente, de la misma manera se debe ¹⁷ corregir para verificar que el error haya sido resuelto.

5.1. Lenguajes De Programación

El lenguaje ⁴¹ de programación es una herramienta fundamental en el campo de la informática que permite a los desarrolladores crear instrucciones que una computadora puede interpretar y ejecutar. Estos lenguajes están diseñados con una sintaxis y reglas específicas que ⁵² facilitan la comunicación entre el humano y la máquina. Existen diversos tipos de lenguajes de programación, cada uno con características y usos particulares, desde la creación de aplicaciones web y móviles hasta la automatización de tareas ⁴⁷ complejas y el análisis de datos. En esencia, el lenguaje de programación es el medio por el cual los ingenieros de software transforman ideas abstractas en soluciones prácticas y funcionales.

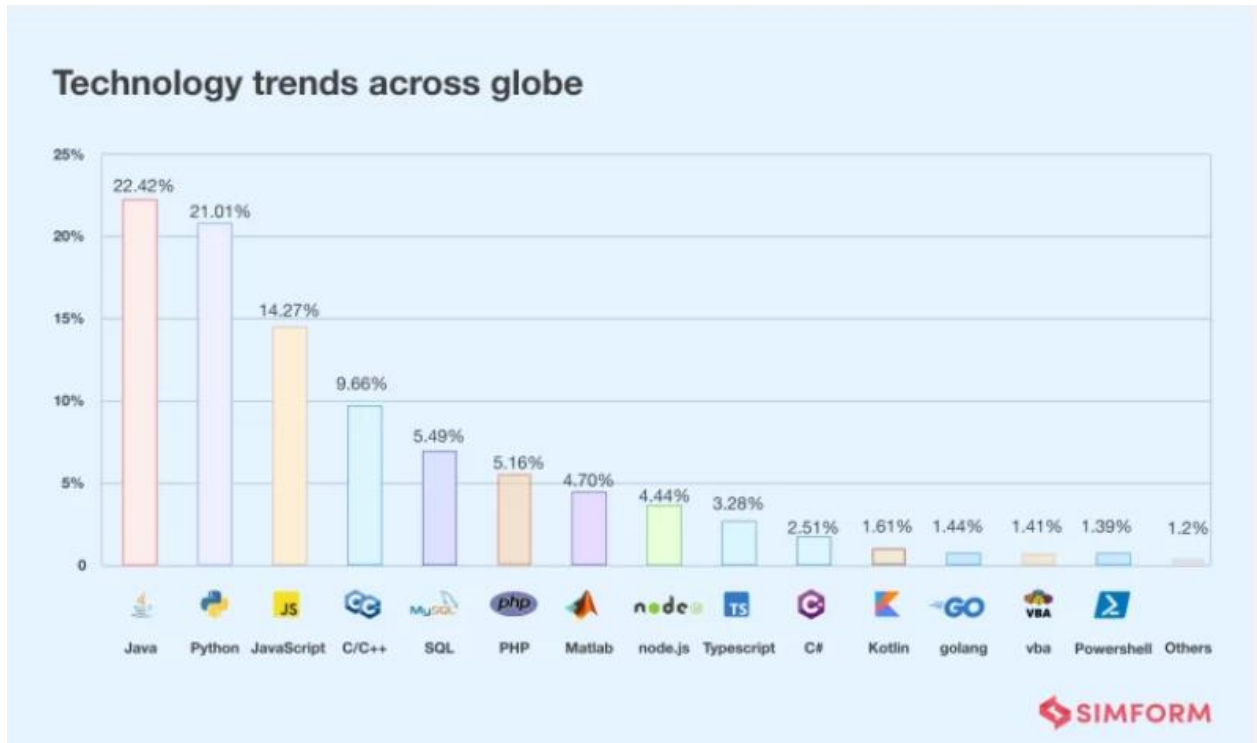
5.2. Software libre

El software libre ofrece una serie de características que lo hacen particularmente atractivo para su uso en el desarrollo y aplicación de modelos estadísticos. La libertad para usar el software para cualquier propósito, el acceso al código fuente y la capacidad de modificarlo permiten a los usuarios y desarrolladores adaptar herramientas estadísticas a sus necesidades específicas sin restricciones de licencias costosas o limitativas. Además, la capacidad de distribuir copias y mejoras facilita la colaboración y el intercambio de conocimientos dentro de la comunidad científica y académica, lo cual es esencial para el progreso en campos dinámicos como la estadística y el análisis de datos. (Mera, 2023)

Este enfoque abierto y colaborativo no solo mejora la transparencia y la seguridad del software, al permitir que una comunidad más amplia revise y mejore el código, sino que también promueve la sostenibilidad tecnológica al reducir la dependencia de proveedores específicos. Para los investigadores y profesionales que aplican modelos estadísticos, el software libre ofrece una plataforma flexible y coste-efectiva para desarrollar y testar nuevos métodos analíticos, asegurando que puedan adaptarse rápidamente a nuevos desafíos y cambios en el entorno de investigación sin barreras legales o técnicas.

Entre las entidades que ofrecen este tipo de licencias libres, es la Open Source Initiative (OSI) creada en el año de 1998 que también ofrece sus versiones, aunque con el término de códigos abiertos. La Free Software Foundation (FSF), también ofrece al usuario una lista de licencias libres y una lista adicional de licencias que pueden ser consideradas como libres. La FSF definió dos de las licencias de software libre más utilizadas en todo el mundo: la GNU GPL (General Public License) y la GNU LGPL (Lesser General Public License, inicialmente conocida como Library General Public License).

Tabla 4 Ranking Software libre



Tomado de (Akiwatkar, 2022)

5.3. Softwares analizados en este proyecto

5.3.1. JavaScript

JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo, programación funcional).

Figura 6: de java script



Tomado de: (*The Evolution of JavaScript*, s. f.)

Breve historia del lenguaje

En 1995, Brendan Eich trabajaba en Netscape Communications Corporation, la compañía que publicó el famoso Netscape Navigator, entonces principal competidor de Internet Explorer. 7 Brendan desarrolló Live Script un lenguaje de script que se basa en el lenguaje Java, y que estaba destinado a ser instalado en los servidores desarrollados por Netscape. Netscape inició el desarrollo de una versión del cliente LiveScript, que pasó a llamarse JavaScript, en homenaje al lenguaje Java creado por Sun Microsystems.(Lebbos, 2023)

Ventajas de JavaScript

Velocidad - JavaScript tiende a ser muy rápido porque a menudo se ejecuta inmediatamente en el navegador. Entonces mientras no requiera recursos externos, JavaScript no tiene permitido retrasarse por llamados del servidor backend.

Simplicidad - La sintaxis de JavaScript está inspirada por Java y es relativamente sencillo de aprender comparado a otros lenguajes de programación populares como C++.

Popularidad - JavaScript esta por todas partes de la web, y con la llegada de Node.js, se ha incrementado su uso en backend. Hay incontables recursos para aprender JavaScript.

Tanto StackOverflow como GitHub muestran un creciente número de proyectos que usan JavaScript, y la popularidad que ha alcanzado en los recientes años se espera que siga creciendo.

Compatibilidad- A diferencia de PHP u otros lenguajes scripting, JavaScript puede ser usado en cualquier página web. JavaScript puede ser usado en diferentes tipos de aplicaciones gracias al soporte en otros lenguajes como Pearl y PHP.

Server Load- JavaScript es client-side, entonces esto reduce la demanda de servidores en general, y las aplicaciones sencillas puede que no necesiten el servidor del todo.

Interfaces sencillas- JavaScript puede ser usado para crear características como arrastrar y soltar, y componentes tales como las diapositivas, lo cual mejora enormemente la interfaz de usuario y la experiencia del sitio.

Funcionalidad extendida- Los desarrolladores pueden extender la funcionalidad de las páginas web mediante fragmentos de JavaScript para un tercer grupo de extensiones como Greasemonkey.

Versatilidad- Hay muchos métodos para usar JavaScript mediante servidores Node.js. Si tú estás en bootstrap Node.js con Express, puedes usar un documento database como MongoDB, y usar JavaScript en el frontend para clientes. Es posible desarrollar una aplicación completa de JavaScript desde principio a fin solo usando JavaScript.

Actualizaciones- Desde que la llegada de ECMAScript 5 (la especificación escrita en que se basa JavaScript), ECMA International se ha dedicado a actualizar JavaScript anualmente.

Desventajas de JavaScript

Seguridad Client-Side- Desde que el código en JavaScript es ejecutado en el client-side, bugs y descuidos pueden ser explotados algunas veces para malos propósitos. Por esto, algunas personas deciden desactivar JavaScript por completo.

Soporte del navegador- Mientras server-side script siempre produce el mismo resultado, algunas veces diferentes navegadores interpretan el código JavaScript de manera distinta. Estos días las diferencias son mínimas, y no deberías tener que preocuparte mientras compruebes tu código en la mayoría de los navegadores. (Zubukarai,2019)

5.3.2. ²El lenguaje C++

C++ es un lenguaje de programación potente y no está diseñado para crear aplicaciones web y no es tan fácil de usar como otros lenguajes de desarrollo web como JavaScript, etc.

Figura 7: software C++



Tomado de (dev_creative, 2023)

Breve historia del lenguaje

fue creado a mediados de los años 80 y es uno de los lenguajes de programación más utilizados. ²De hecho, muchos de los programas que se utiliza diariamente están hechos en lenguaje C++, por ejemplo: los sistemas operativos Windows, Linux y macOS, los programas de Office, el sistema Android, entre muchos otros.

Ventajas de c++

A pesar de su complejidad, C++ tiene muchos beneficios que vale la pena conocer (versatilidad, rendimiento y popularidad).

En primer lugar, ⁴⁰C++ es un lenguaje orientado a objetos, lo que significa que ayuda a los desarrolladores a crear código modular que es fácil de entender y mantener. Esto lo convierte en una opción ideal para proyectos grandes con múltiples desarrolladores. En segundo lugar, C++ es muy rápido y eficiente, lo que lo hace adecuado para aplicaciones que consumen muchos recursos, como videojuegos o simulaciones científicas. En tercer lugar, C++ tiene una amplia variedad de bibliotecas disponibles que brindan soluciones listas para usar para tareas de programación comunes.

En general, aprender C++ puede resultar difícil al principio, pero es una experiencia muy gratificante. Aquellos que estén dispuestos a dedicar tiempo y esfuerzo encontrarán que es una habilidad valiosa que pueden utilizar para crear cosas asombrosas.

Desventajas de c++

C++ es un lenguaje muy difícil de aprender, e incluso los desarrolladores experimentados pueden tener dificultades al intentar utilizarlo. Aquí están 5 de las mayores desventajas de aprender C++:

- La sintaxis puede resultar increíblemente confusa, especialmente para los principiantes.
- Hay muchas formas diferentes de hacer las cosas en C++, lo que puede dificultar saber cuál es la mejor manera de hacer algo.
- Los compiladores de C++ pueden ser muy exigentes e incluso un pequeño error puede hacer que su código no se compile correctamente.
- El código C++ puede ser difícil de leer, especialmente si no está bien escrito.

- La biblioteca estándar de C++ es grande y complicada, y comprender todas las diferentes funciones y clases puede llevar mucho tiempo.

Aplicaciones con lenguaje de c++

C++³³ es un poderoso lenguaje de programación que permite a los desarrolladores crear sistemas sofisticados. Con C++ es posible desarrollar software para una amplia variedad de plataformas, desde pequeños sistemas integrados hasta grandes sistemas corporativos, además C++ se utiliza ampliamente en aplicaciones científicas y de ingeniería debido a su capacidad para manejar cálculos complejos. En resumen, C++ proporciona a los desarrolladores una herramienta versátil para crear software de alta calidad.

Como puedes imaginar, C++ es un lenguaje sumamente importante y, por lo tanto, todavía se utiliza en muchos proyectos.

C++² es un lenguaje versátil que se puede utilizar para una variedad de propósitos. El lenguaje de programación es ampliamente utilizado en:

- Desarrollo de software para diversos sistemas.
- Creación de aplicaciones.
- Programación de sistemas embebidos.
- Desarrollo de motores de juegos.
- Creación de aplicaciones de escritorio.

5.3.3. Python

Es un lenguaje de programación orientado a objetos, creado por el Holandés Guido van Rossum en los inicios de los años 90, se encuentra bajo una licencia de software libre, de una implementación canónica conocida como CPython.(Ochoa &Villa, 2022)

Figura 8: Python



Tomado de (Python, 2024)

Breve historia del lenguaje

Python fue creado por Guido van Rossum, un programador holandés a finales de los 80 y principio de los 90 cuando se encontraba trabajando en el sistema operativo Amoeba. ¹Primariamente se concibe para manejar excepciones y tener interfaces con Amoeba como sucesor del lenguaje ABC. El 16 de octubre del 2000 se lanza Python 2.0 que contenía nuevas características como completa recolección de basura y completo soporte a Unicode. Pero el mayor avance lo constituye que este comenzó a ser verdaderamente desarrollado por la comunidad, bajo la dirección de Guido. El Python 3.0 es una versión mayor e incompatible con las anteriores en muchos aspectos, que llega después de un largo período de pruebas el 3 de diciembre del 2008. Muchas de las características introducidas en la versión 3 han sido compatibilizadas en la versión 2.6 para hacer de forma más sencilla la transición entre estas. A Guido van Rossum le fue otorgado el Free Software Award (Premio del Software Libre) en el 2001, por sus trabajos en la creación y

desarrollo del lenguaje Python. En el 2005 fue contratado por Google, donde trabaja en la actualidad, aunque sigue liderando los esfuerzos en el desarrollo del Python.

Python es un lenguaje que tiene como característica su enfoque multiparadigma y una alta legibilidad de su código, lo que lo ha convertido en uno de los lenguajes más populares en todo el planeta.(Challenger, 2014)

Características de Python

Posee un nivel de aprendizaje alto y rápido.

- Pose gran cantidad de estructura de datos.
- Tiene sintaxis muy legible y expresiva.
- Numerosas bibliotecas de acuerdo con el campo de aplicación.
- Disponible para diferentes sistemas operativos (Windows, Mac, Linux, otros).
- Está orientado a objetos con estilo de programación imperativo procedimental
- Se puede programar en alto nivel con aplicaciones de gran tamaño.
- Se tiene el soporte de una comunidad muy grande en el ámbito de programar.
- Es gratuito y libre • Se utiliza de forma comercial a gran escala.

1 Sintaxis

La sintaxis de Python es muy sencilla, tanto que en algunas ocasiones parece pseudocódigo. Es muy interesante observar las diferencias que existen entre el programa Hola Mundo de Python y el de otro lenguaje de alto nivel como C++.

1 Librería Estándar

Una de las fortalezas de Python, y quizás la mayor, es la librería estándar con que cuenta. Con decenas de módulos cubre la mayoría de las necesidades básicas de un programador y mucho más. En esta se le da cobertura de forma muy intuitiva a tópicos como:

- Cadenas
- Estructura de datos
- Funciones numéricas y matemáticas
- Compresión de datos
- Formatos de archivo
- Criptografía
- Servicios de los Sistemas Operativos
- Comunicación entre Procesos
- Manejo de datos de Internet
- Servicios multimedia
- Manejo de excepciones

Existen más de 40 tópicos en la referencia de la librería de Python4 por lo que se puede afirmar que es una de las más completas con que se cuenta en la actualidad, comparable con la de Java y .NET.

5.4. Librerías de interés para el cálculo de confiabilidad

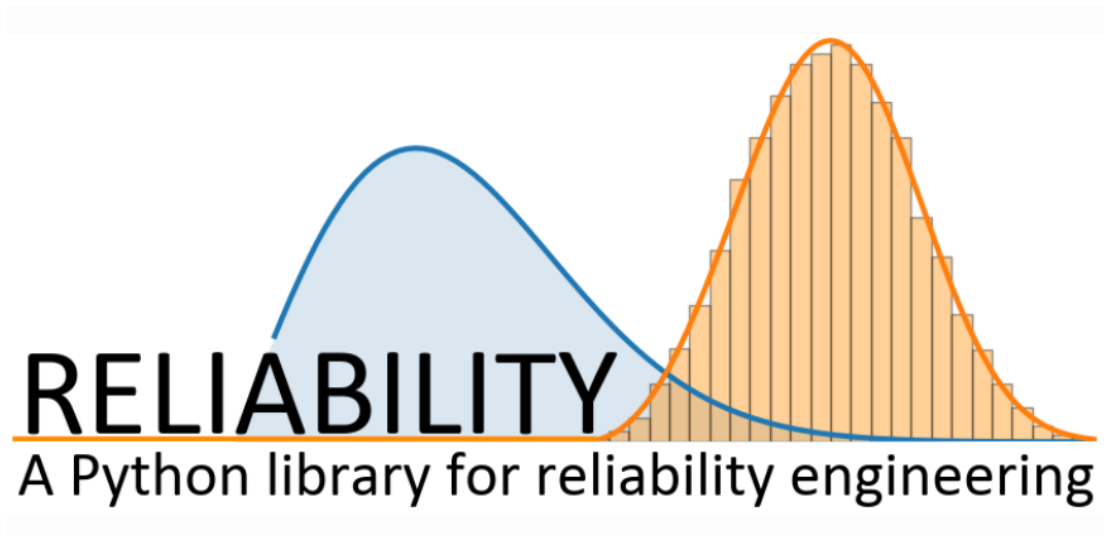
5.4.1. Reliability

Una de las principales librerías de interés para el cálculo de confiabilidad es “Reliability” y se puede instalar directamente o desde el intérprete de Python con la línea de código: “pip install reliability”. Es esta librería proporciona una guía de distintos procesos estadísticos enfocados a la ingeniería de confiabilidad.

Los procesos y modelos estadísticos se encuentran de forma gratuita en esta página web que se los puede usar sin restricción, a diferencia de otras herramientas de pago como Reliasoft.

La biblioteca puede ser expuesta mediante un navegador o un archivo PDF como un libro de estudio, sin embargo, se recomienda utilizar la información en la web debido a la facilidad de búsqueda de elementos, ejemplos y códigos que brinda reliability, así también el beneficio de tener acceso a la traducción automática del navegador Google Chrome. (Matthew, 2023)

Figura 5: Reliability



Tomado de: (Matthew, 2023)

5.4.2. NumPy

NumPy⁹ ofrece funciones matemáticas integrales, generadores de números aleatorios, rutinas de álgebra lineal, transformadas de Fourier y más. admite una amplia gama de plataformas informáticas y de hardware, y funciona bien con bibliotecas distribuidas, de GPU y de matrices dispersas.

El núcleo de NumPy es el código C bien optimizado. Disfrute de la flexibilidad de Python con la velocidad del código compilado.

La sintaxis de alto nivel de NumPy lo hace accesible y productivo para programadores de cualquier nivel o experiencia.¹⁶ Distribuido bajo una licencia BSD liberal, es desarrollado y mantenido públicamente en GitHub por una comunidad vibrante, receptiva y diversa.

Casi todos los científicos que trabajan en Python aprovechan el poder de NumPy.

NumPy aporta el poder computacional de lenguajes como C y Fortran a Python, un lenguaje mucho más fácil de aprender y usar. (NumPy -, s. f.)

Figura 6: NumPy



Tomado de: (NumPy, 2024)

5.4.3. ⁶ Pandas

Es una biblioteca de software escrita como extensión de Numpy para manipulación y análisis de datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.

Figura 7: pandas



Tomado de (pandas, 2024)

5.4.4. Matplotlib

3 Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Matplotlib hace que las cosas fáciles sean fáciles y las difíciles posibles.

- Crea gráficos con calidad de publicación.
- Crea figuras interactivas que puedan hacer zoom, desplazarse y actualizarse.
- Personaliza el estilo visual y el diseño.
- Exporte a muchos formatos de archivo.
- Incrustar en JupyterLab y en interfaces gráficas de usuario.
- Utilice una amplia gama de paquetes de terceros creados en Matplotlib.

Figura 8: matplotlib



Tomado de (matplotlib, 2024)

5.5. Comparativa Entre Los Lenguajes De Programación

Actualmente Java Script, C++ y Python representan ser una poderosa herramienta en el análisis de datos estadísticos, trabajan directamente con expresiones matemáticas, así como de herramientas de visualización muy intuitivas con los usuarios. Sin embargo, a continuación, se expone una diferenciación entre lenguajes en función de la capacidad e información que se tiene en el mundo de la ingeniería.

En la siguiente tabla se presenta las comparaciones entre lenguajes de programación para lo cual se tiene en consideración los siguientes iconos:

- Aplica
- χ No aplica

Tabla 5 Comparación de lenguajes de programación

Lenguaje de Programación	Python	C++	Java Script
Características			
Lenguaje de programación fácil e intuitivo	•	•	•
Disponible en Anaconda Distribution	•	χ	χ
Ejecutable en Jupyter Lab o Jupyter Notebook	•	•	•
Procesos de Weibull integrados	•	χ	•
Procesos de Crow-AMSSA integrados	•	χ	χ
Compatible con librería Reliability	•	•	•
Compatible con librerías: NumPy, pandas, matplotlib	•	χ	•
Compatible con paquetes de interfaz gráfica como producto	•	χ	χ

Obtenido de (Autores)

De acuerdo a estas cualidades, la herramienta informática se desarrollará en el ambiente de Python ya que tiene gran dominio y enfoque al análisis de confiabilidad, con gran información y recursos disponibles en la web.

CAPÍTULO III

6. Desarrollo de una Herramienta Informática para el Análisis de Confiabilidad

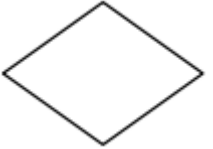
Una vez establecida la teoría, se realiza un ejemplo con datos recopilados, a continuación, se procede a utilizar la herramienta estadística Python con la cual se realizará la interfaz para la herramienta informática explicando su funcionamiento a continuación mediante diagramas de flujo.

6.1. Diagrama de flujos

Un diagrama de flujo es una representación gráfica que ilustra los pasos secuenciales de un proceso o sistema. Utiliza símbolos estandarizados, como rectángulos, diamantes y flechas, para mostrar acciones, decisiones y el flujo de información o materiales. Esta herramienta facilita la comprensión y comunicación de cómo funciona un proceso, identificando puntos clave y posibles áreas de mejora. Los diagramas de flujo son ampliamente utilizados en diversas disciplinas, como ingeniería, programación y gestión de proyectos, para planificar, documentar y optimizar procesos.

Tabla 6 Diagrama de flujo

Nombre	Descripción	Componente
Inicio/Final	Indica el inicio o el fin del proceso.	
Entrada/Salida	Representa una actividad o tarea.	
Entrada por teclado	Se entiende que el programa espera una entrada manual de datos por parte del usuario.	

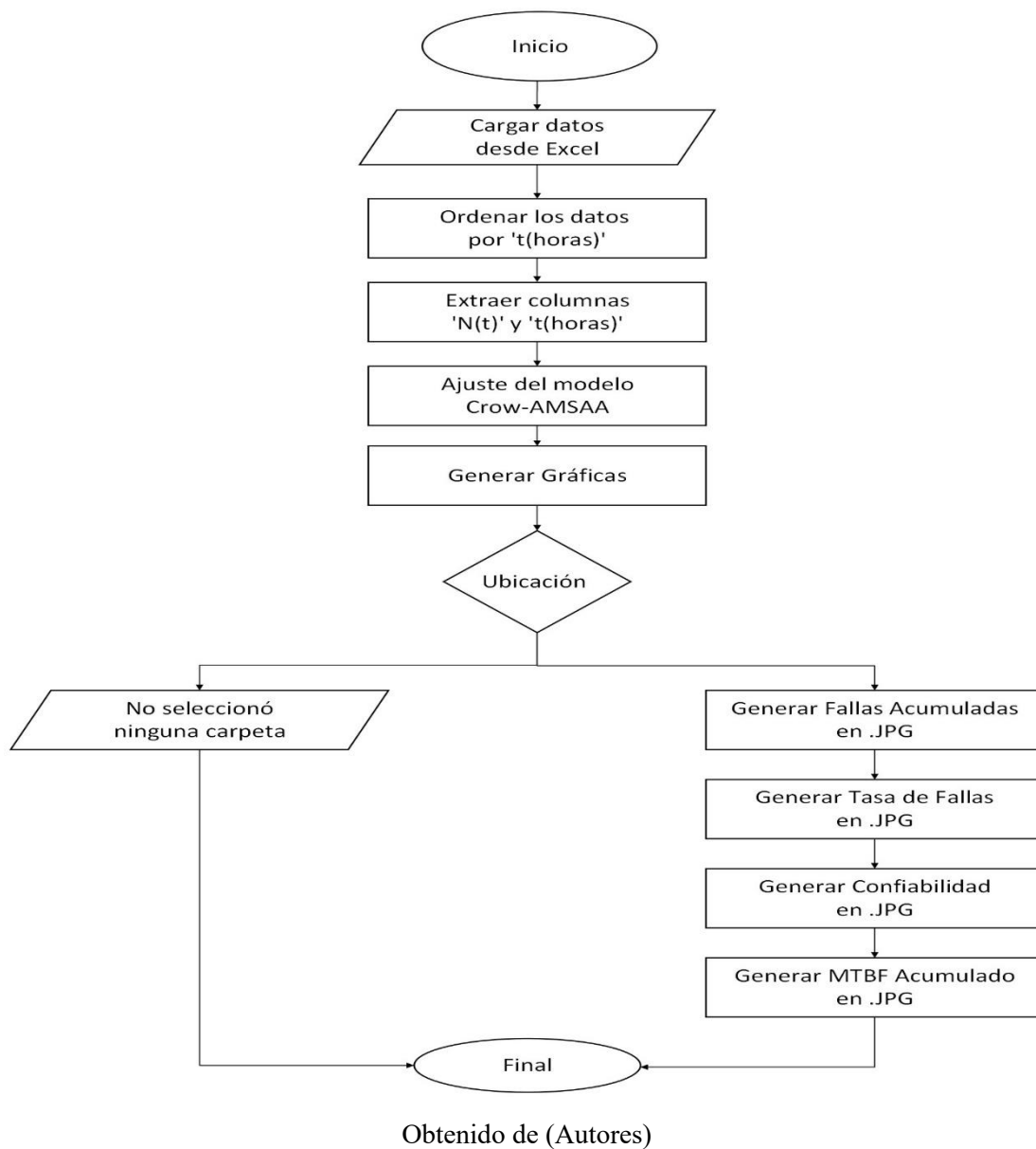
Flechas de dirección	Indica el sentido de la ejecución de las operaciones en el diagrama.	
Acción	Es la acción general que se debe de realizar.	
Decisión	Realiza la comparación de los datos, depende si es falso o verdadero toma la decisión.	
Impresora	Muestra la impresión de un resultado del diagrama.	
Pantalla	Muestra un mensaje o el resultado en la pantalla del usuario.	

Obtenido de (Autores)

6.1.1. Diagrama de flujo para modelo Crow AMSAA

Este diagrama de flujo sigue las convenciones estándar de diagramación para representar claramente cada paso y la toma de decisiones en el proceso de generación de gráficas utilizando el modelo Crow-AMSAA.

Figura 9 Diagrama de flujo Crow AMSAA



Inicio: Representado por un óvalo, es el punto de partida del proceso.

Cargar datos desde Excel: Representado por un paralelogramo, indica la entrada de datos desde un archivo Excel.

Ordenar los datos por 't(horas)': Representado por un rectángulo, se ordenan los datos según la columna de tiempos 't(horas)'.

Extraer columnas 'N(t)'(representa el número acumulado de fallas hasta un tiempo t) y **'t(horas)'**(representa los tiempos de buen funcionamiento): Representado por un rectángulo, se extraen las columnas necesarias para el análisis.

Ajuste del modelo Crow-AMSAA: Representado por un rectángulo, se realiza el ajuste del modelo Crow-AMSAA a los datos.

Generar Gráficas: Representado por un rectángulo, se preparan las gráficas necesarias.

Ubicación: Representado por un rombo, verifica si se selecciona una ubicación para guardar las gráficas.

No seleccionó ninguna carpeta: Si no se selecciona una carpeta, el proceso termina (representado por un paralelogramo y un óvalo).

Sí seleccionó carpeta: Si se selecciona una carpeta, se generan las siguientes gráficas en formato JPG (todos representados por rectángulos):

Generar Fallas Acumuladas en .JPG: Generar la gráfica de fallas acumuladas.

Generar Tasa de Fallas en .JPG: Generar la gráfica de tasa de fallas.

Generar Confiabilidad en .JPG: Generar la gráfica de confiabilidad.

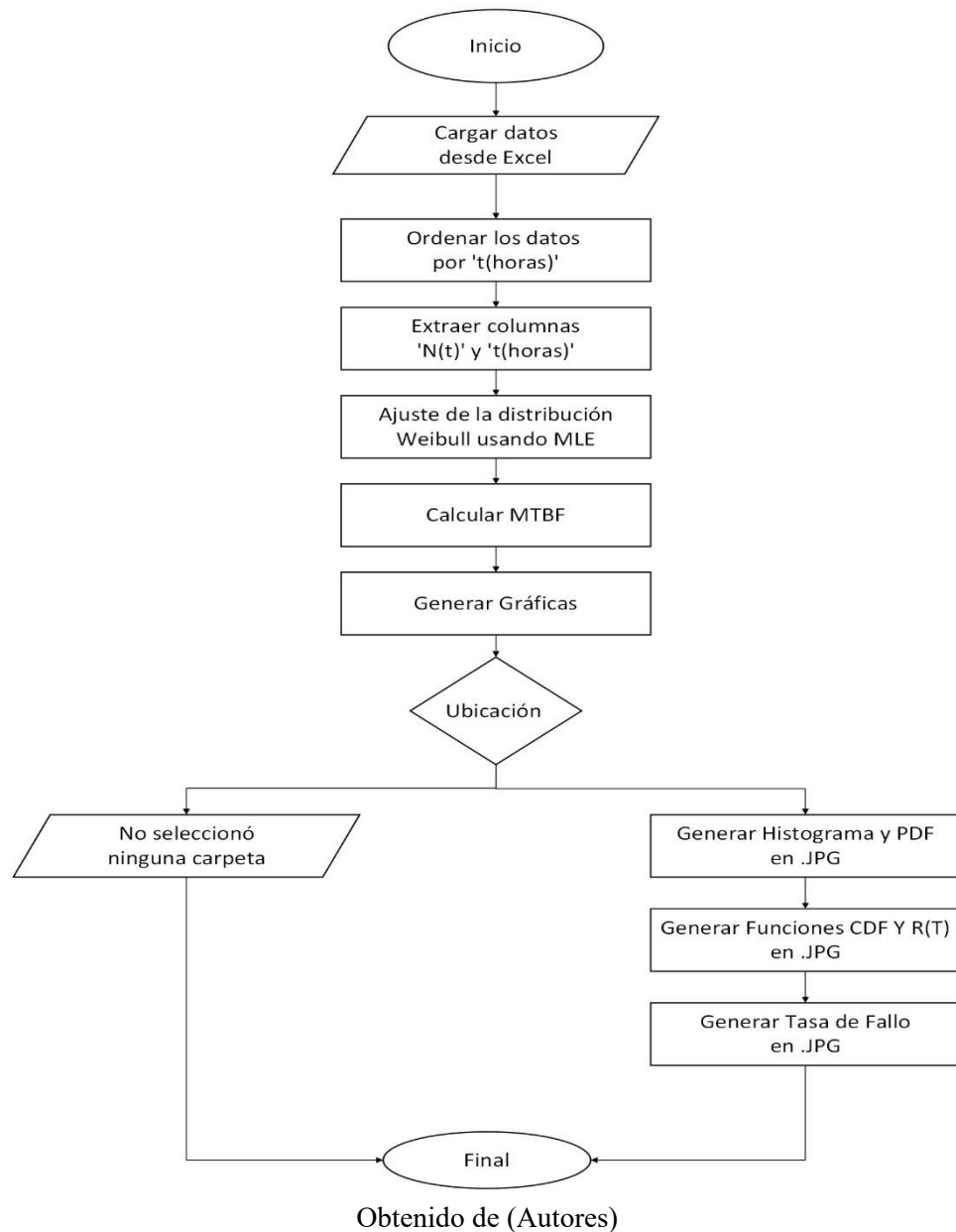
Generar MTBF Acumulado en .JPG: Generar la gráfica de MTBF acumulado.

Final: Representado por un óvalo, es el punto de terminación del proceso.

6.1.2. Diagrama de flujo para modelo Weibull

Este diagrama de flujo sigue las convenciones estándar de diagramación para representar claramente cada paso y la toma de decisiones en el proceso de generación de gráficas utilizando el modelo Weibull.

Figura 10 Diagrama de flujo Weibull



Inicio: Representado por un óvalo, es el punto de partida del proceso.

Cargar datos desde Excel: Representado por un paralelogramo, indica la entrada de datos desde un archivo Excel.

Ordenar los datos por 't(horas)': Representado por un rectángulo, se ordenan los datos según la columna de tiempos 't(horas)'.

Extraer columnas 'N(t)' y 't(horas)': Representado por un rectángulo, se extraen las columnas necesarias para el análisis.

Ajuste de la distribución Weibull usando MLE: Representado por un rectángulo, se realiza el ajuste de la distribución Weibull a los datos utilizando el método de máxima verosimilitud (MLE).

Calcular MTBF: Representado por un rectángulo, se calcula el tiempo medio entre fallos (MTBF) utilizando los parámetros ajustados de la distribución Weibull.

Generar Gráficas: Representado por un rectángulo, se preparan las gráficas necesarias.

Ubicación: Representado por un rombo, verifica si se selecciona una ubicación para guardar las gráficas.

No seleccionó ninguna carpeta: Si no se selecciona una carpeta, el proceso termina (representado por un paralelogramo y un óvalo).

Sí seleccionó carpeta: Si se selecciona una carpeta, se generan las siguientes gráficas en formato JPG (todos representados por rectángulos):

Generar Histograma y PDF en .JPG: Generar el histograma de los datos y la función de densidad de probabilidad ajustada.

Generar Funciones CDF y R(T) en .JPG: Generar las funciones de distribución acumulada (CDF) y confiabilidad (R(T)).

Generar Tasa de Fallo en .JPG: Generar la gráfica de tasa de fallo en función del tiempo.

Final: Representado por un óvalo, es el punto de terminación del proceso.

6.2. Instalación de Recursos Necesarios.

Python³⁹ es un lenguaje de programación orientado a objetos que funciona directamente sobre el sistema operativo donde está instalado. Para este proyecto de análisis e investigación, se utilizará un intérprete de Python que proporciona un entorno de aprendizaje mejorado, conocido como Jupyter. Este intérprete está disponible de forma gratuita en la web, gracias al soporte de Anaconda Distribution, una plataforma de código abierto.

Figura 11 : Anaconda



Obtenido de (Anaconda, 2024)

Anaconda es una **suite de código abierto** que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la ciencia de datos con Python.⁸ Se trata de una distribución de Python que básicamente funciona como un gestor de entorno, de paquetes y que posee una colección de más de 720 cuya característica primordial es que son de código abierto.

Anaconda Distribution

se agrupa en cuatro sectores o soluciones tecnológicas:

Anaconda Navigator: Interfaz gráfica de Anaconda Python

Anaconda Project

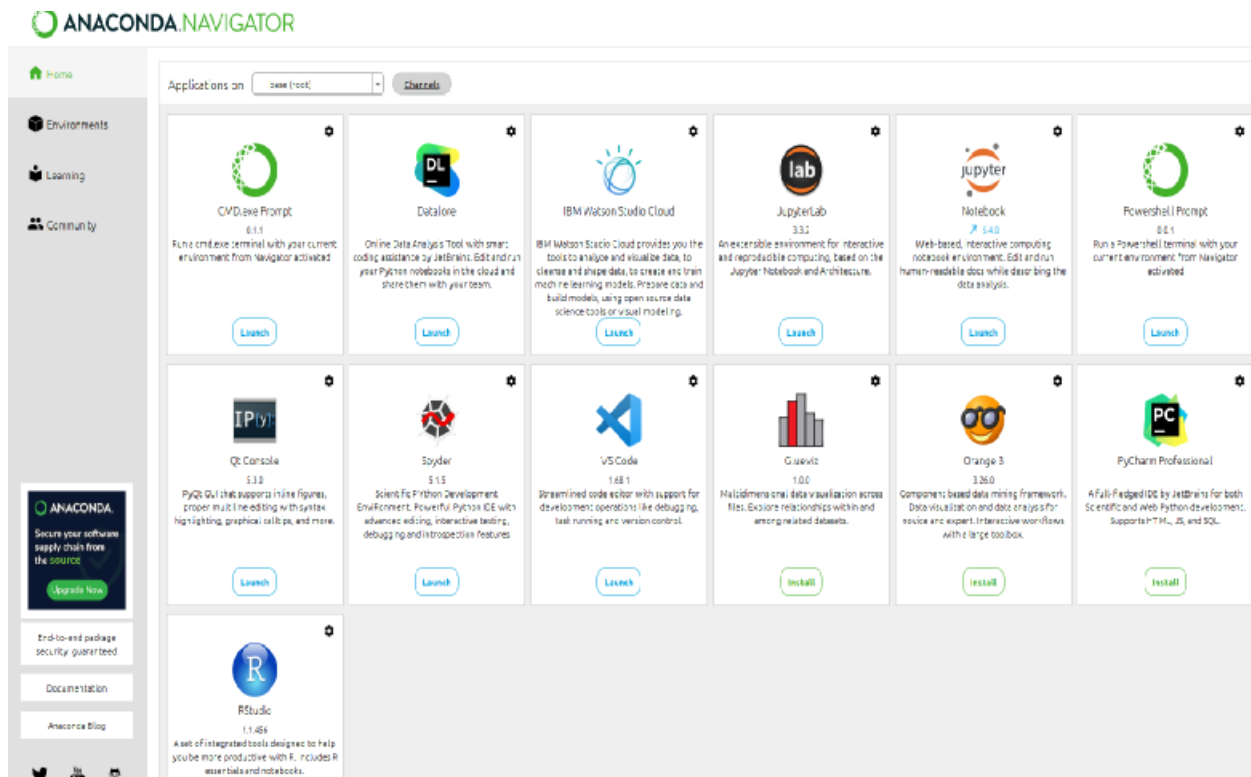
Librerías de Ciencia de Datos

Conda: Gestor de código del Anaconda Python.(Mira, 2021)

6.2.1 Iniciar Anaconda Navegador

Al abrir anaconda navegadora se puede encontrar diversas aplicaciones orientadas a la programación, incluyendo R y otros intérpretes de lenguajes de programación, como se ilustra a continuación. JupyterLab y Jupyter Notebook son plataformas diseñadas para desarrollar códigos en Python3 de manera abierta.

Figura 12 Anaconda navigator

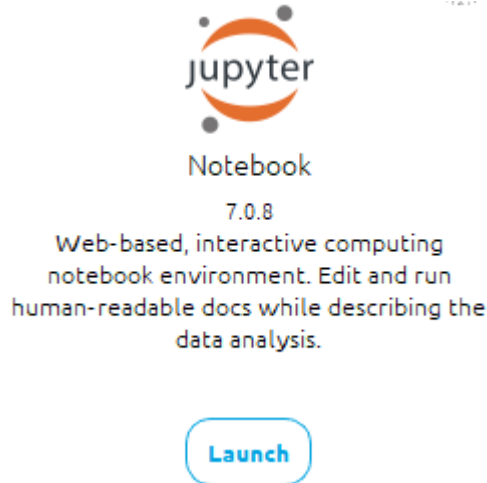


Obtenido de (Captura local)

6.2.2. Jupyter Notebook

¹² Jupyter Notebook es una aplicación web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto explicativo. Es ampliamente utilizado para ²² la limpieza y transformación de datos, simulación numérica, modelado estadístico, aprendizaje automático y mucho más.

Figura 13 jupyter Notebook

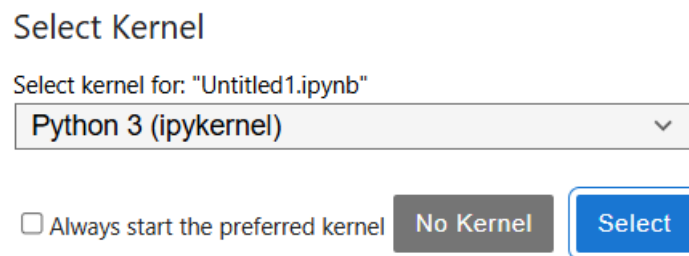


Obtenido de (Captura local)

6.2.3. Pasos para codificación en Python 3

En la interfaz de Jupyter Notebook, haz clic en "New" y selecciona "Python 3" para crear un nuevo notebook.

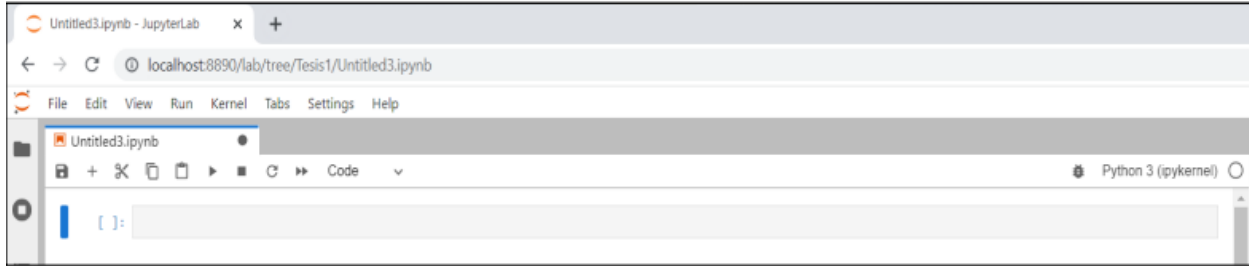
Figura 14 Select kernel



Obtenido de (Captura local)

A continuación, se abrirá la intérprete de Python donde se ejecutará el desarrollo o la codificación.

Figura 15 interprete Python



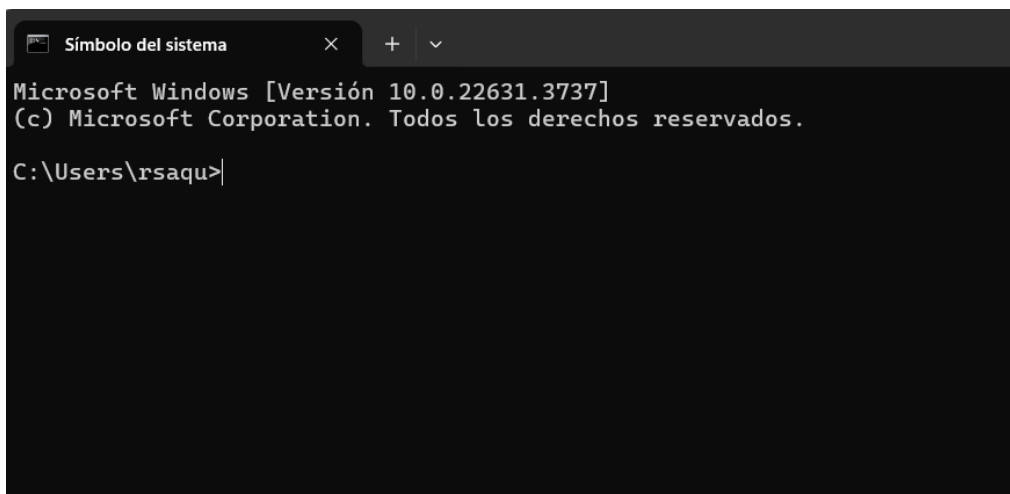
Obtenido de (Python)

Dentro del intérprete de Python, es importante instalar las librerías que serán utilizadas para desarrollar el código fuente. Las librerías a instalar son: Reliability, NumPy, pandas y matplotlib. Excepto Reliability, las demás librerías están incluidas en Anaconda Distribution. Otra manera para instalar estas librerías lo puedes hacer usando el comando:

```
!pip install
```

Otra manera de realizar la instalación de librerías es ingresando en el buscador de Windows el código CMD y se abrirá el símbolo de sistema en donde se puede descargar y verificar cuales son las librerías que están descargadas en nuestro sistema.

Figura 16 SMD



Obtenido de (Captura local)

6.3. Codificación para el Modelo Crow AMSAA

- **Importar librerías necesarias:**

Se importan las librerías `pandas`, `numpy`, `matplotlib.pyplot` y `curve_fit` de `scipy.optimize`, que son esenciales para la manipulación de datos, cálculos numéricos, visualización y ajuste de parámetros, respectivamente.

```
21 import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

- **Leer datos desde un archivo Excel:**

Se carga un archivo Excel y se extraen los valores de las columnas 't(horas)' y 'N(t)'.

```
df = pd.read_excel(r"C:\Users\rsaqu\Desktop\PROGRAMA\tbf.xlsx")
tiempos = df['t(horas)'].values
fallas_acumuladas = df['N(t)'].values
```

- **Verificación de los datos:**

Se verifica que los datos no contengan valores negativos, NaN (Not a Number) o infinitos

```
if any(tiempos <= 0):
    raise ValueError("Todos los valores de tiempo deben ser positivos y mayores que cero.")
if any(np.isnan(tiempos)) or any(np.isnan(fallas_acumuladas)):
    raise ValueError("Los datos no deben contener valores NaN.")
if any(np.isinf(tiempos)) or any(np.isinf(fallas_acumuladas)):
    raise ValueError("Los datos no deben contener valores infinitos.")
```

- **Definir la función Crow-AMSAA:**

Se define la función Crow-AMSAA que modela el comportamiento de las fallas acumuladas.

```
def crow_amsaa(t, beta, eta):
    t_safe = np.maximum(t, np.finfo(float).eps)
    eta_safe = np.maximum(eta, np.finfo(float).eps)
```

```
return (t_safe / eta_safe) ** beta
```

- **Ajustar los parámetros usando `curve_fit`:**

Se utilizan los datos para ajustar los parámetros `beta` y `eta` del modelo mediante el método de mínimos cuadrados no lineales (`curve_fit`).

```
params, _ = curve_fit(crow_amsaa, tiempos, fallas_acumuladas, p0=[0.5, 1000])
beta, eta = params
```

- **Calcular tasa de crecimiento y MTBF:**

Se calcula la tasa de crecimiento y el tiempo medio entre fallas (MTBF).

```
growth_rate = 1 - beta
mtbf = np.sum(tiempos) / len(fallas_acumuladas)
```

- **Calcular la tasa de fallas (Lambda) y la confiabilidad R(t):**

Se calcula la tasa de fallas y la confiabilidad en función del tiempo.

```
lambda_t = (beta / eta) * (tiempos / eta) ** (beta - 1)
R_t = np.exp(-(tiempos / eta) ** beta)
```

- **Calcular el MTBF acumulado:**

Se calcula el MTBF acumulado a lo largo del tiempo.

```
mtbf_acumulado = np.cumsum(tiempos) / np.arange(1, len(tiempos) + 1)
```

- **Imprimir los resultados:**

Se muestran los parámetros ajustados y los resultados calculados.

```
print(f"Beta: {beta:.2f}")
print(f"Eta: {eta:.2f}")
print(f"Growth Rate: {growth_rate:.2f}")
print(f"MTBF: {mtbf:.2f} horas")
```


6.4. Calculo por el modelo Crow AMSAA para comprobar su funcionamiento

- **Carga y Preparación de Datos**

En la tabla 7 se muestran valores utilizados para la comprobación del funcionamiento del sistema, Se recomienda que los valores a ingresar deben estar ordenados de manera ascendente, para garantizar esto el programa hace una revisión previa al cálculo, los datos deben ser ingresados en dos columnas llamadas N(t) y t(horas).

Tabla 7 Datos ejemplo 1

N(t)	t(horas)
1	98
2	115
3	126
4	126
5	135
6	139
7	148
8	153
9	164
10	166
11	168
12	175
13	193
14	198

- **Cálculo de Parámetros**

Tabla 8 Parámetros calculados ejemplo 1 Crow AMSAA

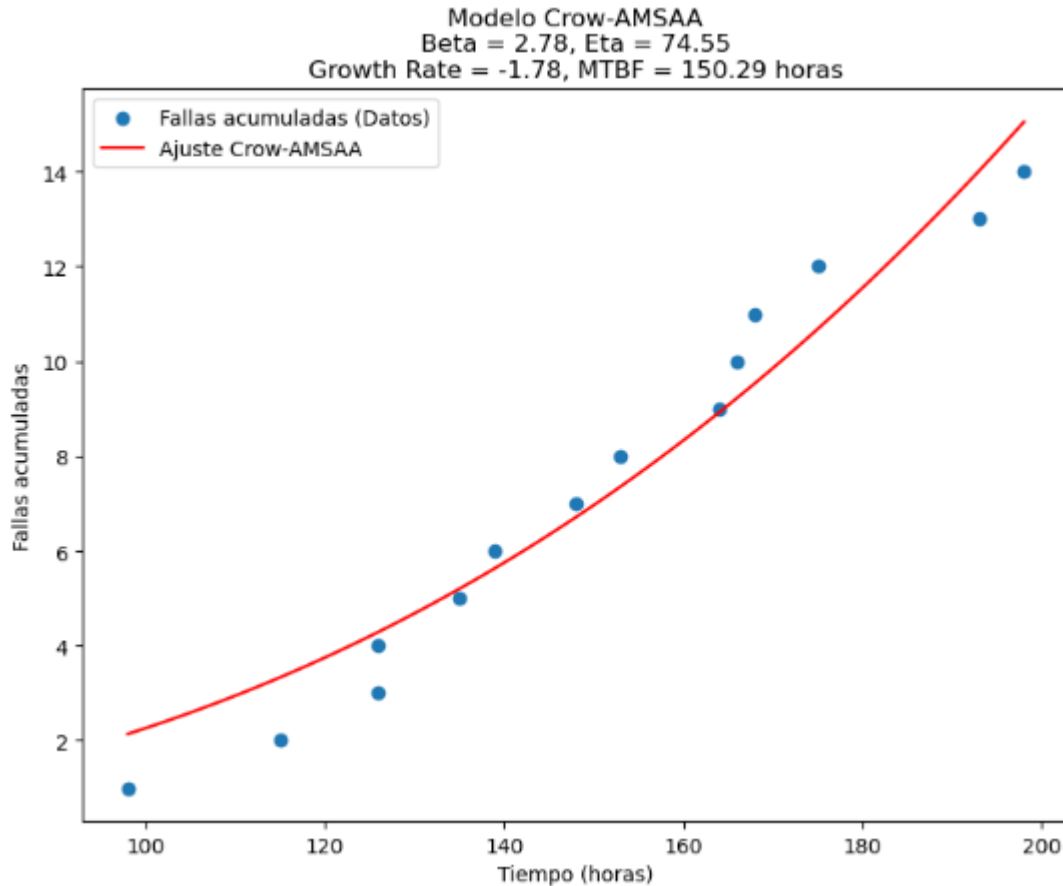
PARAMETROS DEL MODELO Crow AMSAA	
BETA	2,78
ETA	74,55
Growth Rate	-1,78
MTBF	150.29 horas

Obtenido de (Autores)

Los parámetros β y η para describen el comportamiento de las fallas en el sistema. En este análisis, se encontró que $\beta=2.78$, lo que indica una tasa de fallas en aumento con el tiempo. Esto es típico en sistemas sujetos a envejecimiento y desgaste, donde los componentes se deterioran con el uso continuo, resultando en una frecuencia de fallas cada vez mayor. El parámetro $\eta=74.55$ horas representa la vida característica del sistema, sugiriendo que, en promedio, se espera que las fallas se vuelvan frecuentes después de este tiempo. La tasa de crecimiento negativa de -1.78 confirma que la tasa de fallas está incrementándose de manera acelerada. A pesar de este comportamiento adverso, el tiempo medio entre fallas (MTBF) estimado es de 150.29 horas. Este valor de MTBF es útil para planificar mantenimientos y reparaciones, indicando que, aunque el sistema está fallando más frecuentemente, las intervenciones de mantenimiento están ayudando a extender el tiempo medio entre fallas. Sin embargo, la necesidad de un mantenimiento proactivo y posiblemente de mejoras en el diseño del sistema se vuelve evidente para garantizar una mayor confiabilidad a largo plazo.

- **Ajuste Crow AMSAA**

. Figura 17 Ajuste Crow AMSAA ejemplo 1

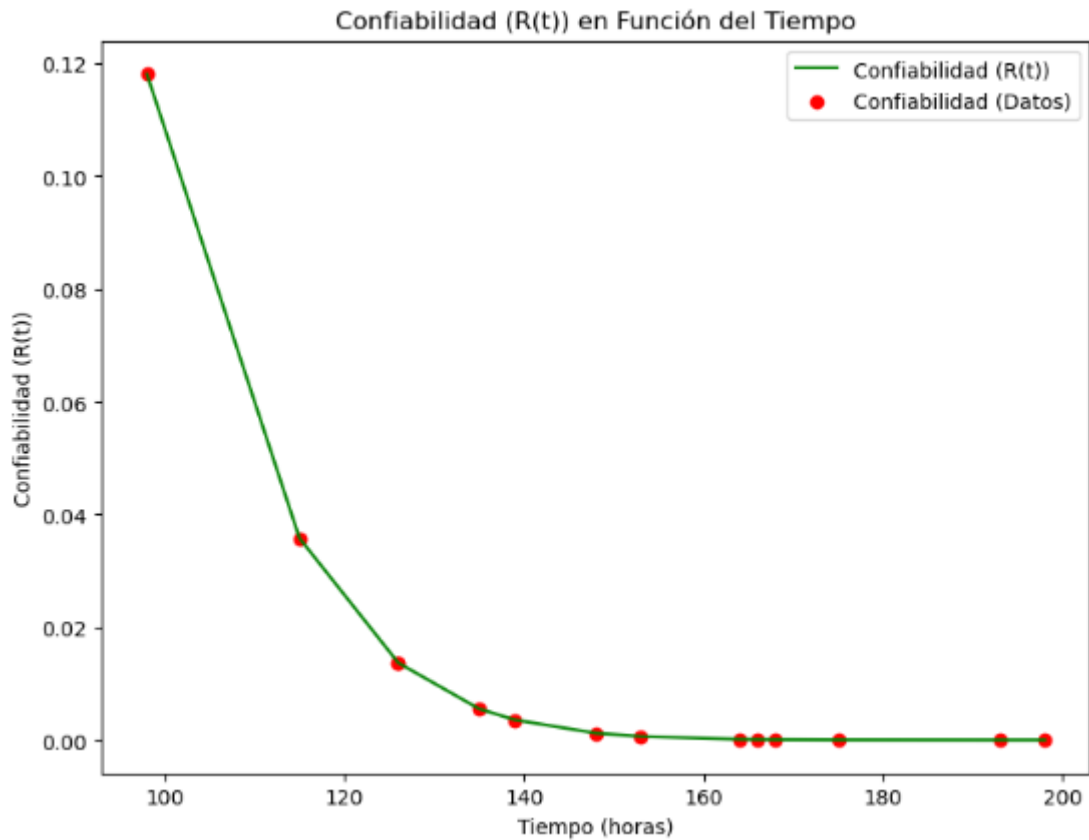


Obtenido de (Autores)

La Figura 15 de fallas acumuladas ajustadas con el modelo Crow-AMSAA indica que el sistema está en una fase de deterioro, con un aumento en la frecuencia de fallas a medida que pasa el tiempo. El buen ajuste del modelo a los datos observados valida su uso para predecir futuras fallas y planificar estrategias de mantenimiento. La tendencia ascendente y el valor de β mayor que 1 sugieren la necesidad de intervenciones de mantenimiento más frecuentes para prolongar la vida útil del sistema y mejorar su confiabilidad a largo plazo

- **Confiabilidad**

Figura 18 Confiabilidad y des confiabilidad ejemplo1



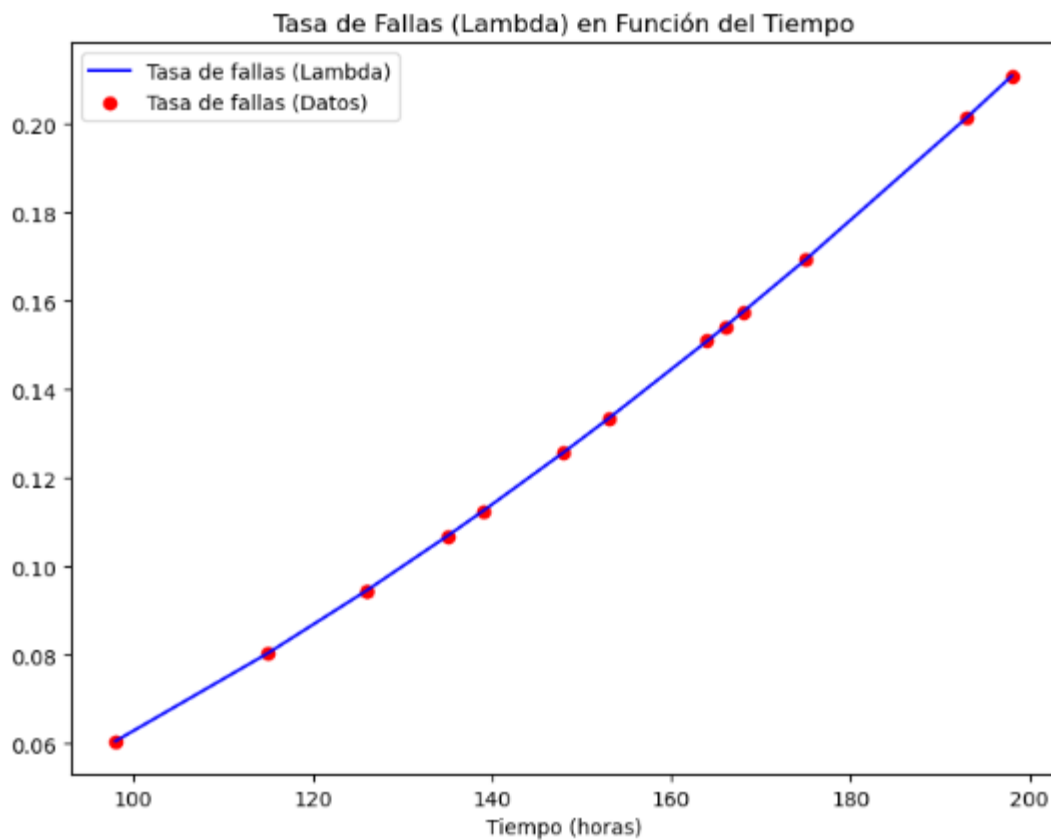
Obtenido de (Autores)

La figura 16 de confiabilidad $R(t)$ en función del tiempo revela una disminución rápida en la probabilidad de que el sistema opere sin fallas a medida que pasa el tiempo. Esta tendencia descendente indica que la confiabilidad del sistema se deteriora significativamente con el tiempo, lo cual es característico de sistemas que están en una fase de envejecimiento y desgaste. La curva verde, que representa el ajuste del modelo Crow-AMSAA, sigue de cerca los puntos rojos observados, validando la precisión del modelo para describir la dinámica de confiabilidad del sistema. Esta concordancia sugiere que las predicciones del modelo son confiables y pueden utilizarse para planificar mantenimientos preventivos.

- **MTBF acumulados y Tasa de fallos**

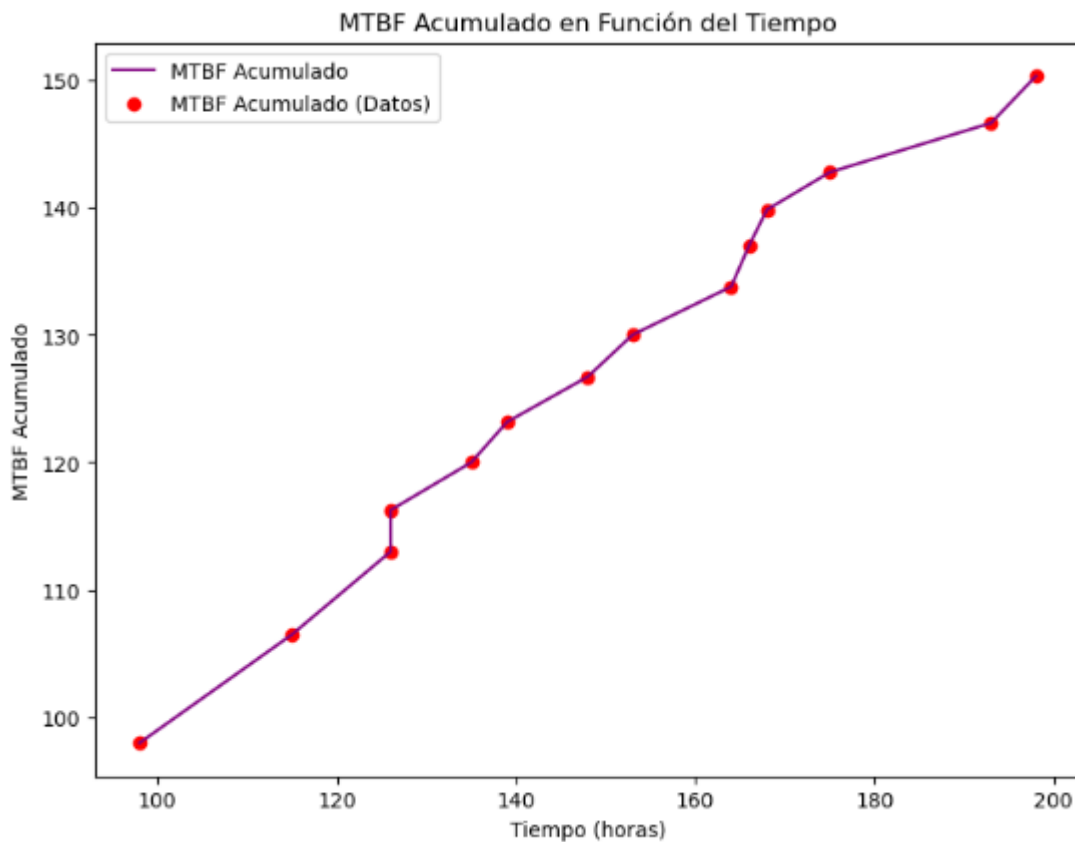
Adicionalmente el programa facilita las gráficas del comportamiento de los MTBF y Tasa de fallos los cuales indican o facilitan el momento idóneo en el que se puede realizar la corrección o si el mantenimiento está siendo efectivo.

Figura 19 Tasa de fallos



Obtenido de (Autores)

Figura 20 MTBF acumulados



Obtenido de (Autores)

6.5. Codificación por el Modelo Weibull

- **Importación de Bibliotecas:**

Importar las bibliotecas necesarias para manejar datos, realizar cálculos estadísticos, y crear gráficos.

```
15 import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import weibull_min
from scipy.special import gamma, gammaln, gammalncc
```

- **Carga de Datos:**

Leer los datos del archivo Excel y extraer las columnas de $N(t)$ y $t(\text{horas})$.

```
file_path = r"C:\Users\rsaqu\Desktop\PROGRAMA\VINI.xlsx"
df = pd.read_excel(file_path)
N_t = df['N(t)'].
t_horas = df['t(horas)'].values
```

- **Ajuste de la Distribución Weibull:**

Ajustar la distribución Weibull a los datos de tiempo ¹⁸ utilizando el método de máxima verosimilitud para obtener los parámetros de forma, escala y ubicación.

```
shape, loc, scale = weibull_min.fit(t_horas, loc=0, floc=0) # Ajustar Weibull
y obtener parámetros

# Parámetros de la distribución Weibull
beta = shape # 38 Parámetro de forma ( $\beta$ )
eta = scale # Parámetro de escala ( $\eta$ )
gamma_val = loc # Parámetro de ubicación ( $\gamma$ )
```

- **Cálculo del MTBF:**

Calcular el Tiempo Medio Entre Fallos (MTBF) usando los parámetros ajustados.

```
mtbf = eta * gamma(1 + 1/beta)
print(f"MTBF (Mean Time Between Failures): {mtbf:.2f} horas")
```

- **Cálculo de las Funciones ³⁰ $F(T)$ y $R(T)$:**

Calcular la función de distribución acumulada (CDF) y la función de confiabilidad para un rango de valores de tiempo.

```
t_values = np.linspace(np.min(t_horas), np.max(t_horas), 100) # Valores de
tiempo para la evaluación
F_t = gammainc(shape, (t_values / eta)**beta)
R_t = gammaincc(shape, (t_values / eta)**beta)
```

- **Generación de la Curva de Densidad de Weibull Ajustada:**

Generar datos para la curva de densidad de probabilidad ajustada de Weibull usando los parámetros ajustados.

```
x = np.linspace(np.min(t_horas), np.max(t_horas), 10000)
```

```
pdf = weibull_min.pdf(x, shape, loc, scale
```

- 27 **Cálculo de la Tasa de Fallo:**

Calcular la tasa de fallo como la densidad de probabilidad dividida por la confiabilidad en cada punto en el tiempo. Con el fin de asegurar que los arreglos tengan las mismas dimensiones para evitar errores de broadcasting.

```
t_values_for_hazard = np.linspace(np.min(t_horas), np.max(t_horas),
len(t_values))
pdf_for_hazard = weibull_min.pdf(t_values_for_hazard, shape, loc, scale
R_t_for_hazard = gammaincc(shape, (t_values_for_hazard / eta)**beta)
hazard_rate = pdf_for_hazard / R_t_for_hazard
```

- Visualización:**

Crear tres gráficos:

Un histograma de los datos junto con la curva de densidad ajustada.

Las funciones CDF y confiabilidad.

La tasa de fallo ajustada y los datos originales para comparar visualmente.

6.6. Calculo por el modelo Weibull para comprobar su funcionamiento.

- Carga de Datos:**

Tabla 9 Datos Weibull ejemplo1

N(t)	t(horas)
1	98
2	115
3	126
4	126
5	135
6	139

7	148
8	153
9	164
10	166
11	168
12	175
13	193
14	198

- **Ajuste de la Distribución Weibull y cálculo del MTBF:**

Tabla 10 parámetros de cálculo ejemplo 1 Weibull

PARAMETROS DEL MODELO WEIBULL	
BETA	6,06
ETA	161,93
GAMMA	0
MTBF	150,31 horas
Confiabilidad del sistema	0,53

Obtenido de (Autores)

Los parámetros estimados de la distribución Weibull para los datos proporcionados son $\beta = 6.06$, $\eta = 161.93$, y $\gamma = 0.00$. El parámetro de forma (β) mayor a 1 indica una tasa de fallo que aumenta, lo que sugiere una fase de desgaste conocido como un período inicial de "infancia" o ajustes iniciales de los componentes.

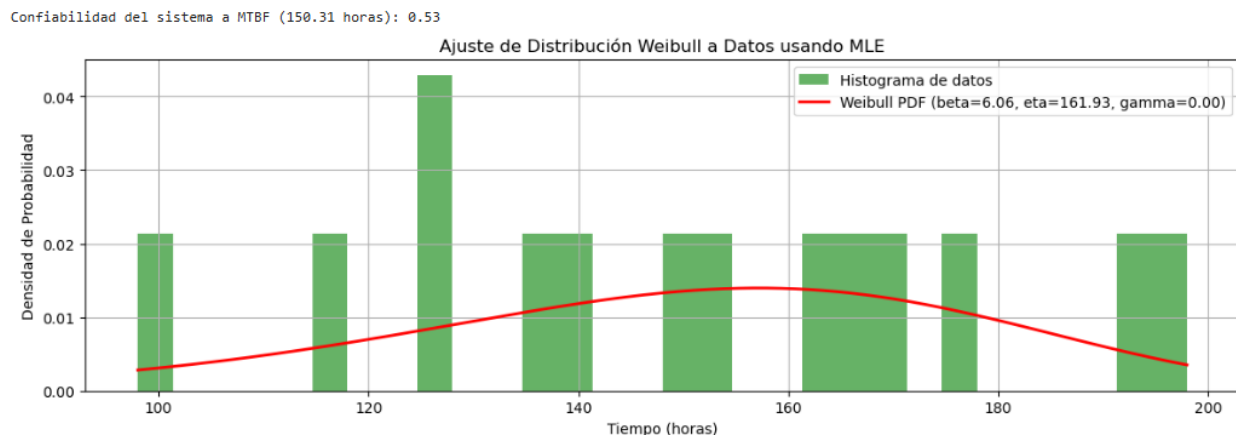
El parámetro de escala (η) refleja el tiempo característico en el cual ocurre el 63.2% de los fallos, situándose en 161.93 horas para este caso. Esto sugiere que la mayoría de los fallos ocurren antes de las 161.93 horas, lo cual se alinea con el gráfico de la función de densidad de probabilidad

(PDF) que muestra un pico alrededor de las 140 horas. La ubicación (γ) igual a 0 implica que no hay un tiempo de retraso antes de que comiencen a ocurrir las fallas.

El MTBF (Mean Time Between Failures) calculado es de 158.31 horas. El MTBF es una métrica crucial en la ingeniería de confiabilidad, proporcionando una estimación del tiempo promedio que se espera entre fallos sucesivos de un sistema. En este caso, la confiabilidad del sistema a las 158.31 horas es del 53%, lo que indica que hay una probabilidad significativa de que las unidades fallen cerca de este tiempo. Esto se puede usar para programar mantenimientos preventivos y planificar reemplazos de componentes antes de que se alcance este tiempo, mejorando así la disponibilidad y confiabilidad del sistema.

- **Ajuste Weibull**

Figura 21 ajuste Weibull

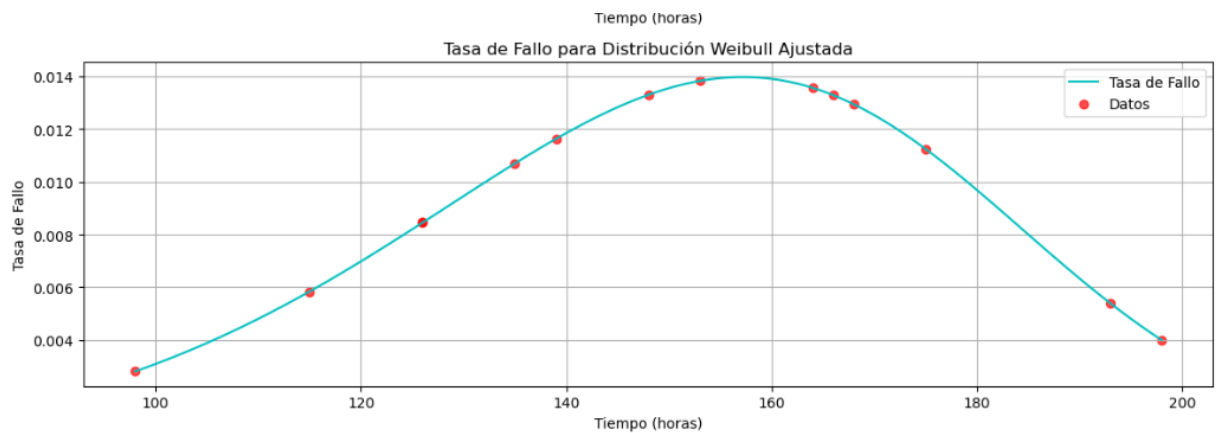


Obtenido de (Autores)

La figura 19 de ajuste Weibull compara la distribución teórica ajustada con los datos empíricos de fallas, mostrando un histograma de densidad de fallas y una función de densidad de probabilidad (PDF) Weibull ajustada. Las barras verdes indican la frecuencia de fallas observadas, con un pico notable alrededor de las 120 horas. La curva roja de la PDF ajustada refleja un comportamiento de desgaste, con parámetros estimados de Beta (6.06) y Eta (161.93), lo que

sugiere una tasa de fallo creciente seguida de una disminución. La correspondencia entre la PDF ajustada y el histograma valida la adecuación del modelo Weibull para describir los patrones de fallas del sistema, proporcionando una base sólida para la planificación de mantenimientos y predicciones de confiabilidad.

Figura 22 tasa de fallos

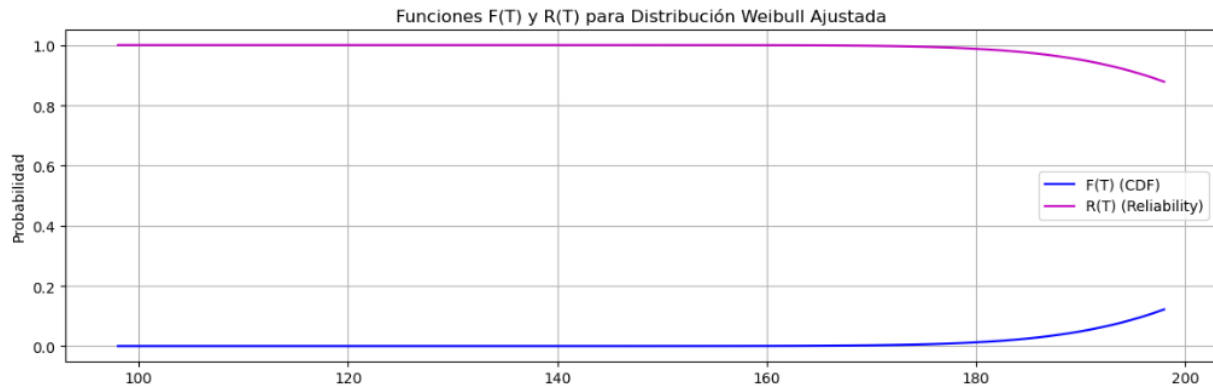


Obtenido de (Autores)

La figura 20 de tasa de fallos revela un patrón claro de desgaste, con un incremento en la probabilidad de fallos hasta aproximadamente las 140 horas, seguido de una disminución. Este comportamiento es característico de muchos sistemas mecánicos y electrónicos que experimentan desgaste con el tiempo. La correspondencia entre la tasa de fallos observada y la curva Weibull ajustada valida la precisión del modelo, proporcionando una base sólida para la planificación de mantenimientos y mejoras en la confiabilidad del sistema.

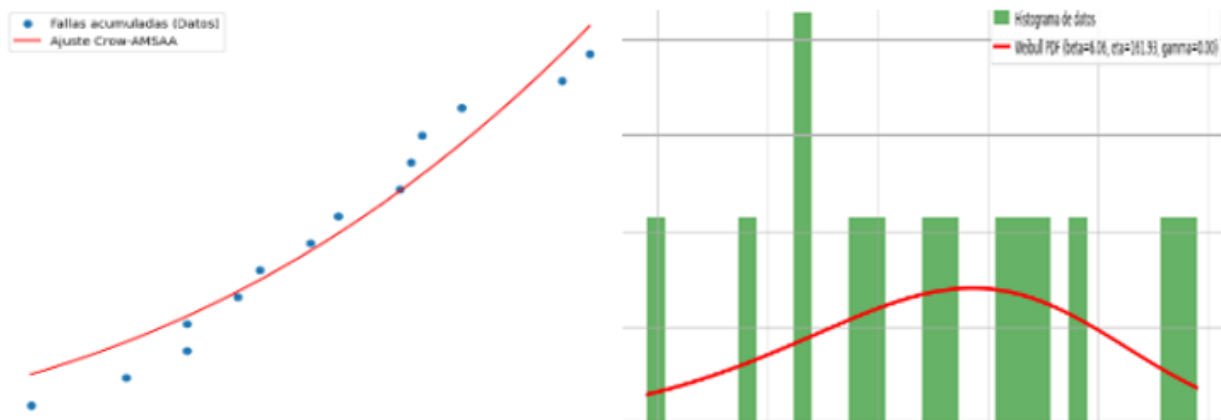
- **Función $R(t)$ y $F(t)$**

Adicional el modelo Weibull muestra las gráficas de confiabilidad y probabilidad de fallas a medida que avanza el tiempo.



6.7. Comparación entre modelo Weibull y Crow AMSAA

Figura 23 comparación de modelos



La gráfica Crow-AMSAA muestra las fallas acumuladas en el tiempo con un ajuste del modelo Crow-AMSAA (línea roja), reflejando cómo las fallas se acumulan en sistemas reparables, donde la tasa de fallas puede cambiar con el tiempo. Este modelo proporciona una representación

directa del crecimiento de fallas, siendo adecuado para sistemas con mantenimiento continuo. Por otro lado, la gráfica Weibull presenta un histograma de los tiempos entre fallas (TBF) con una función de densidad de probabilidad (PDF) ajustada (línea roja). Aunque la distribución Weibull es flexible y útil para sistemas no reparables, en este caso, la PDF no encaja perfectamente con todos los picos del histograma, sugiriendo que puede no capturar completamente la variabilidad de los datos. Mientras el modelo Crow-AMSAA es más adecuado para sistemas reparables y captura la dinámica de fallas acumuladas, la distribución Weibull es ideal para analizar la vida útil de componentes individuales, pero puede tener limitaciones en la representación de la complejidad de los datos de fallas acumuladas.

6.8. Ventajas del modelo Crow-AMSAA frente al modelo Weibull

Crow-AMSAA es especialmente adecuado para sistemas reparables donde se pueden realizar acciones correctivas. Permite evaluar si estas acciones están mejorando la confiabilidad del sistema.

El parámetro $\beta < 1$ sugiere una mejora en la confiabilidad, mientras que $\beta > 1$ indicaría un deterioro.

- **Especialización en la fase de crecimiento:** Crow-AMSAA está diseñado para capturar y modelar de manera efectiva la tasa de fallos durante la etapa inicial del ciclo de vida del producto. Esto es crucial para empresas que necesitan evaluar y mejorar la fiabilidad antes de la comercialización masiva.
- **Mejor ajuste inicial de los datos:** Debido a su enfoque en la fase inicial, Crow-AMSAA puede proporcionar ajustes más precisos y relevantes para los datos recopilados durante las pruebas de introducción al mercado. Esto ayuda a las empresas a identificar y abordar problemas de fiabilidad tempranos.

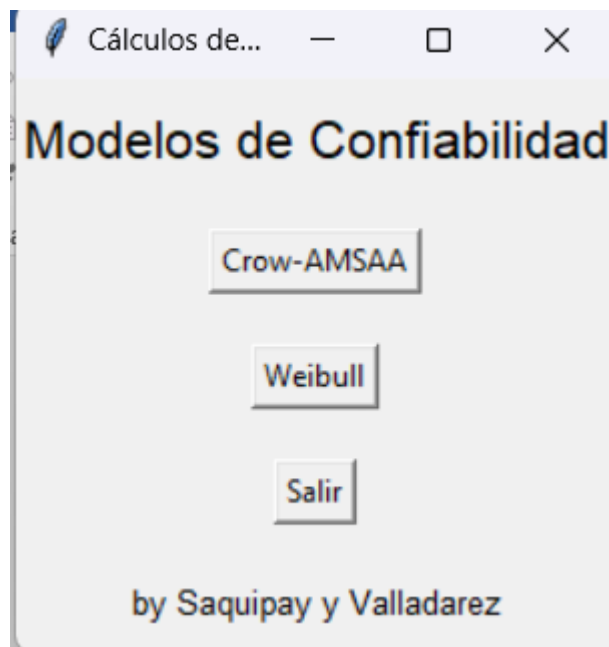
- **Modelo de crecimiento de fallos:** Incorpora un modelo explícito para la tasa de crecimiento de fallos, lo cual es esencial para comprender cómo evolucionará la fiabilidad a medida que el producto se someta a condiciones operativas reales.

En resumen, si estás trabajando en la fase inicial de desarrollo de un producto o sistema y necesitas evaluar y mejorar la fiabilidad durante esta etapa crítica, Crow-AMSAA podría ser más adecuado debido a su enfoque específico en el crecimiento de fallos y la fase inicial del ciclo de vida del producto.

6.9. Interfaz de la herramienta

La interfaz de usuario presentada en la imagen es un programa diseñado para realizar cálculos de confiabilidad utilizando los modelos Crow-AMSAA y Weibull. La interfaz, que lleva por título "Modelos de Confiabilidad", proporciona una manera simple y directa para que los usuarios accedan a las herramientas necesarias para llevar a cabo sus análisis de confiabilidad.

Figura 24 interfaz

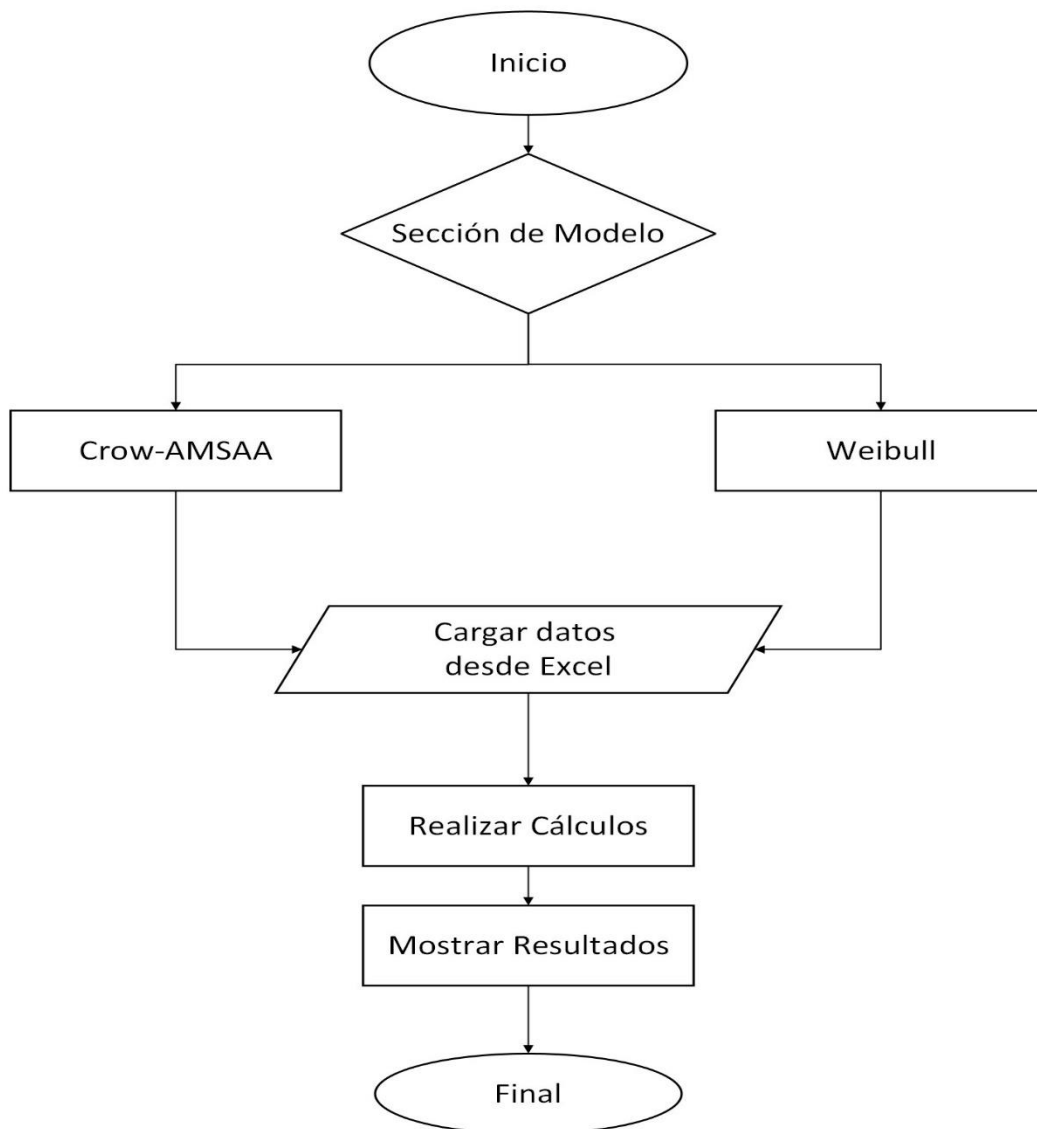


Obtenido de (Autores)

6.9.1. Diagrama del flujo de la interfaz

Este diagrama de flujo sigue las convenciones estándar de diagramación para representar claramente cada paso y la toma de decisiones en el proceso de utilización de la interfaz para análisis de confiabilidad con los modelos Crow-AMSAA y Weibull.

Figura 25 Flujo del interfaz



Obtenido de (Autores)

Inicio: Representado por un óvalo, es el punto de partida del proceso.

Selección de Modelo: Representado por un rombo, es el punto donde se decide si se utilizará el modelo Crow-AMSAA o Weibull.

Crow-AMSAA: Representado por un rectángulo, se selecciona el modelo Crow-AMSAA.

Weibull: Representado por un rectángulo, se selecciona el modelo Weibull.

Cargar datos desde Excel: Representado por un paralelogramo, indica la entrada de datos desde un archivo Excel.

Realizar Cálculos: Representado por un rectángulo, se realizan los cálculos necesarios según el modelo seleccionado.

Mostrar Resultados: Representado por un rectángulo, se muestran los resultados obtenidos de los cálculos.

Final: Representado por un óvalo, es el punto de terminación del proceso.

6.10. Validación de la herramienta con datos reales por Crow AMSAA

- **Carga de datos.**

Tabla 11 ejemplo2 Crow AMSAA

N(t)	t(horas)
1	68
2	235
3	250
4	500
5	525
6	654
7	654
8	750
9	750
10	966
11	1000
12	1000

13	1250
14	1250
15	1310
16	1320
17	1421
18	1430
19	1500
20	1500
21	1750
22	1750
23	2000
24	2250
25	2500
26	2750
27	3000
28	3250
29	3500
30	3750
31	3750
32	4000
33	23158
34	23158
35	23170

- **Calculo de los parámetros Crow AMSAA**

Tabla 12 parámetros de cálculo ejemplo 2 Crow AMSAA

PARAMETROS DEL MODELO Crow AMSAA

BETA	0,31
ETA	0,19
Growth Rate	0,69
MTBF	3487.69 horas

Obtenido de (Autores)

El parámetro β de 0.31 es particularmente revelador, ya que ⁴⁵ indica que la tasa de fallas disminuye con el tiempo. Este valor sugiere que el sistema experimenta una fase inicial de mayor

tasa de fallas, posiblemente debido a un período de asentamiento o "burn-in", donde los defectos iniciales se manifiestan y son eliminados. Con el tiempo, a medida que el sistema se estabiliza, la tasa de fallas disminuye, mejorando la confiabilidad.

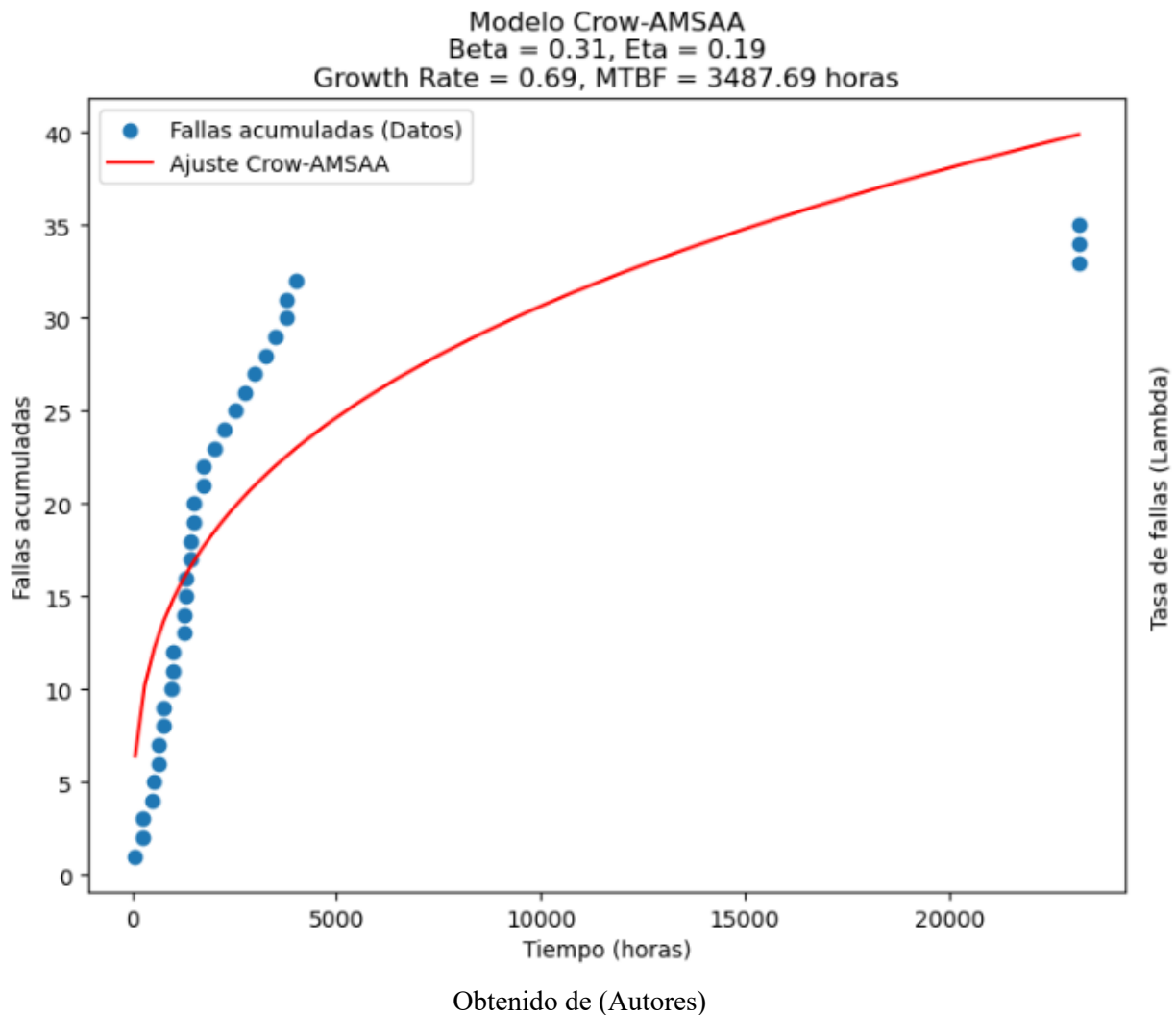
El parámetro η (eta) de 0.19 refuerza esta interpretación. Eta es un parámetro de escala que indica la rapidez con la que ocurren las fallas. Un valor bajo de η sugiere una alta frecuencia de fallas al inicio del ciclo de vida del sistema, lo cual es consistente con el valor de β menor que 1. Esta alta tasa de fallas inicial que disminuye rápidamente indica que el sistema pasa por un proceso de estabilización efectivo, reduciendo significativamente la frecuencia de fallas con el tiempo.

La tasa de crecimiento del modelo Crow AMSAA de 0.69 indica que el número de fallas acumuladas crece a una tasa más lenta que el tiempo transcurrido. Esto es coherente con un valor de β menor que 1, donde el incremento en las fallas es menos que proporcional al incremento en el tiempo, reflejando una mejora en la confiabilidad del sistema conforme pasa el tiempo.

Finalmente, el MTBF calculado de 3487.69 sugiere que, en promedio, el sistema opera durante aproximadamente 3487.69 horas antes de experimentar una falla. Este MTBF elevado indica un nivel de confiabilidad significativo, especialmente después de superar la fase inicial de alta tasa de fallas. En conjunto, estos parámetros y el MTBF muestran que el sistema, tras un período inicial de ajustes y correcciones, logra una estabilidad notable, con una frecuencia de fallas que disminuye y un tiempo operativo entre fallas que se prolonga considerablemente.

- **Ajuste Crow AMSAA**

Figura 26 Ajuste Crow-AMSAA ejemplo2

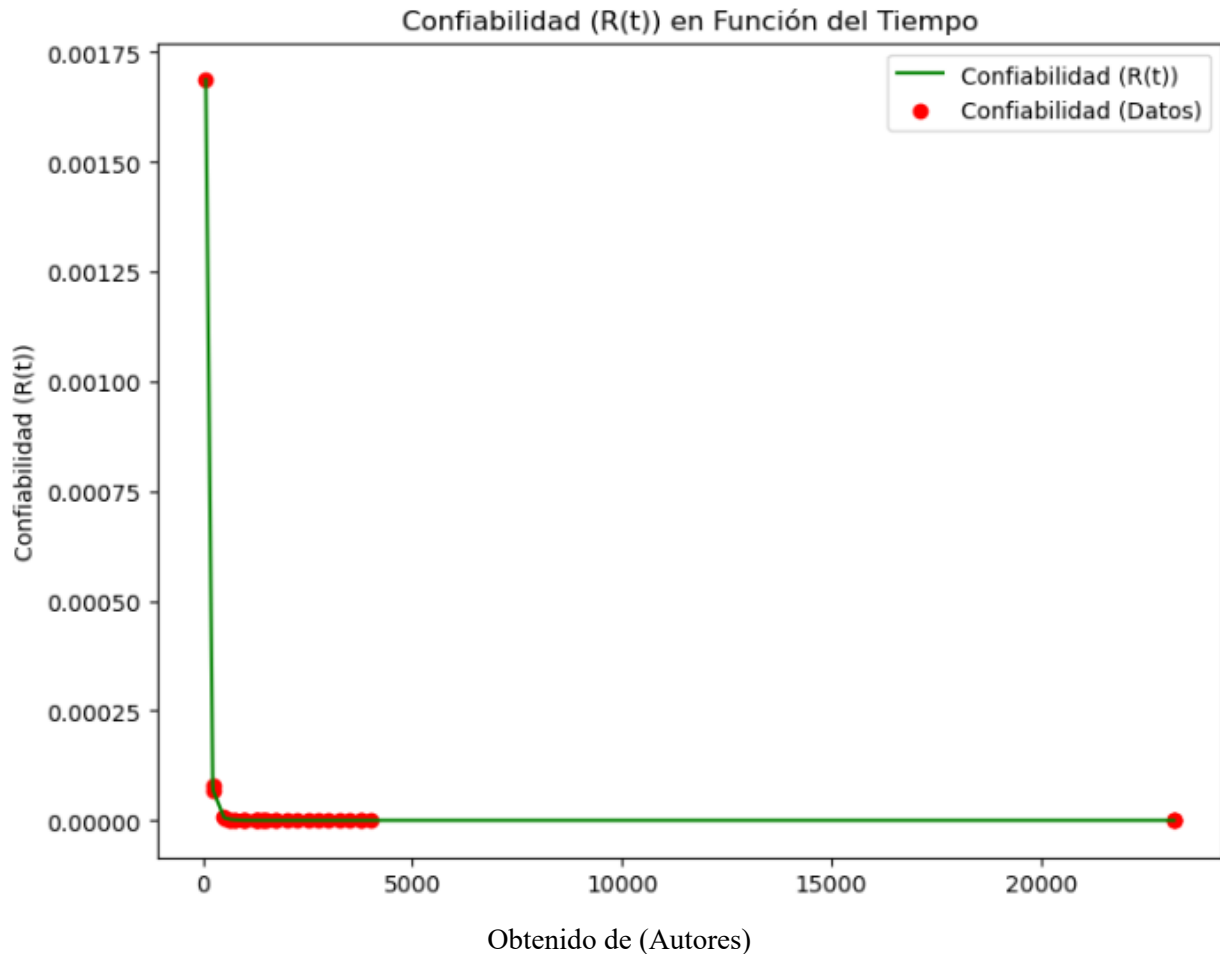


La figura 26 de ajuste del modelo Crow-AMSAA muestra la cantidad de fallas acumuladas en función del tiempo. Los puntos azules representan los datos observados de fallas acumuladas, mientras que la línea roja representa el ajuste del modelo Crow-AMSAA. Se observa que el modelo se ajusta bien a los datos, siguiendo de cerca la tendencia de las fallas acumuladas. El parámetro β (0.31) indica una tasa de fallas decreciente, lo cual es coherente con la forma de la curva que se aplana con el tiempo. Esto sugiere una alta tasa de fallas inicial que disminuye conforme el sistema

se estabiliza. La concordancia entre el modelo y los datos valida el uso del modelo Crow-AMSAA para predecir el comportamiento de fallas del sistema.

- **Confiabilidad**

Figura 27 Confiabilidad ejemplo2



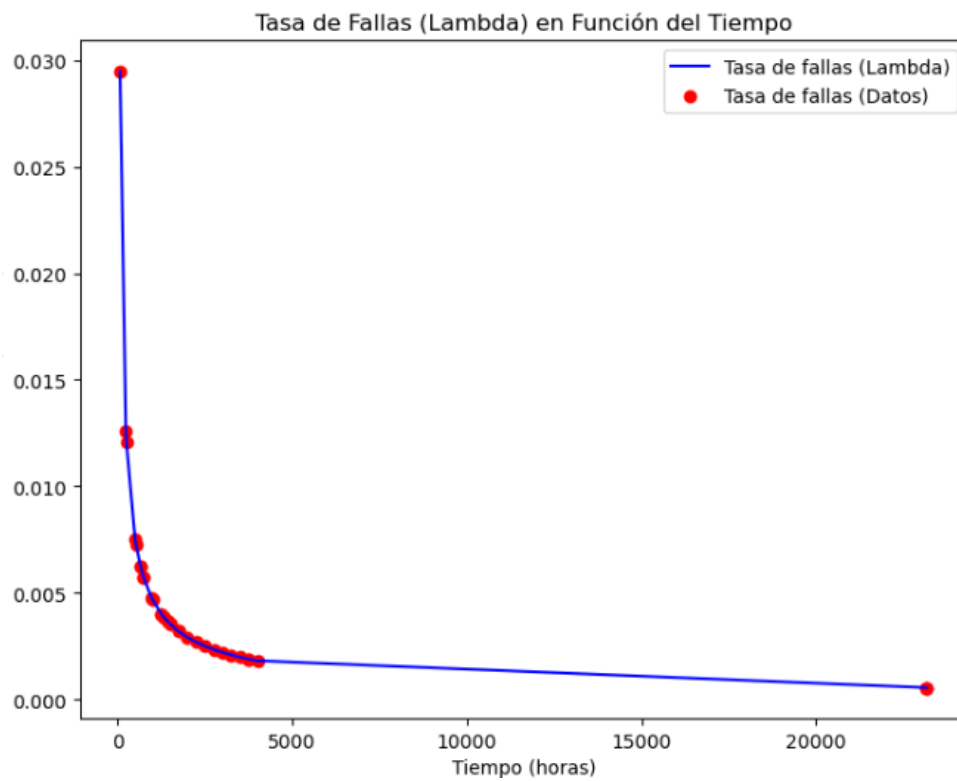
La figura 27 de confiabilidad $R(t)$ muestra una marcada disminución al inicio, lo que indica una alta tasa de fallas tempranas, típica de un período de "burn-in". Posteriormente, la confiabilidad se estabiliza en niveles bajos, reflejando una mejora en la fiabilidad del sistema tras superar los defectos iniciales. La correspondencia entre los datos observados y el modelo Crow-AMSAA sugiere que el modelo es adecuado para predecir la confiabilidad del sistema. Esta

información es crucial para planificar mantenimientos preventivos, optimizando la gestión de la confiabilidad del sistema a lo largo del tiempo.

- **MTBF acumulados y Tasa de fallos**

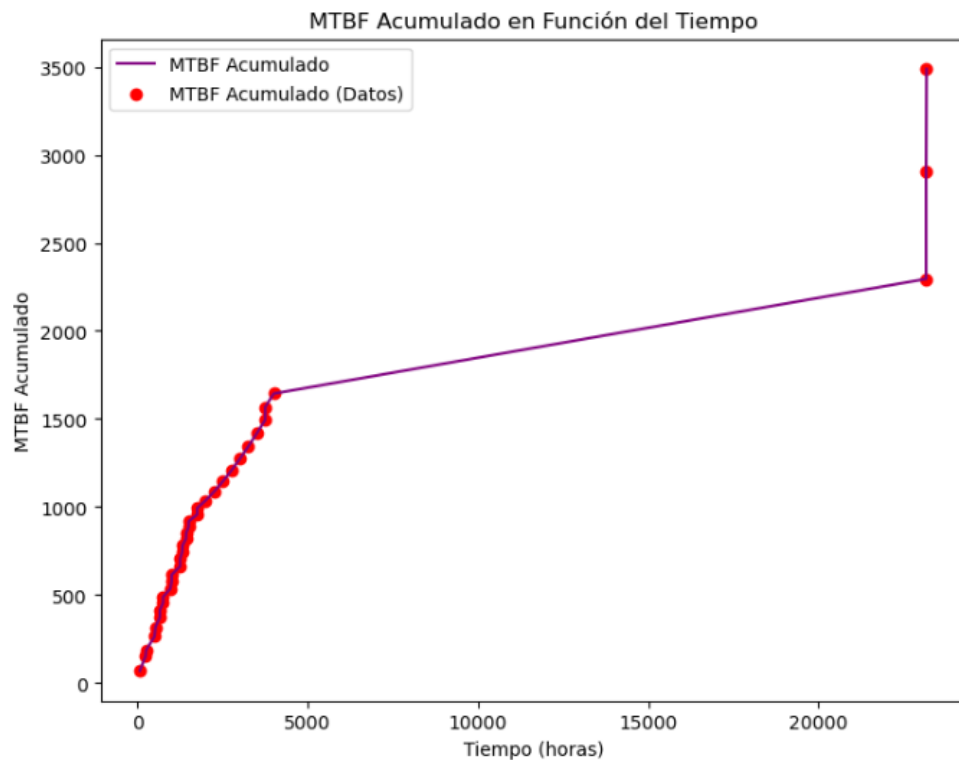
Se agregan graficas adicionales para mejorar el análisis:

Figura 28 Tasa de fallos ejemplo2



Obtenido de (Autores)

Figura 29 MTBF ejemplo2



Obtenido de (Autores)

La tasa de fallas es alta al inicio y decrece rápidamente, estabilizándose a niveles cercanos a cero con el tiempo. Este error está relacionado en la instalación del sistema, o problemas de diseño que se manifiestan poco después de que el sistema entra en operación. Las fallas durante esta etapa son corregidas a medida que se identifican, lo que lleva a una rápida disminución de la tasa de fallas.

Después de la eliminación de los problemas iniciales, el sistema entra en una fase de operación estable como indica el MTBF donde las fallas ocurren de manera aleatoria y la tasa de fallas se estabiliza a un nivel bajo constante. Durante esta fase, el sistema opera de manera confiable y la tasa de fallas es predecible.

6.11. Validación de la herramienta con datos reales por WEIBULL

- Cargar gatos.

Tabla 13 Datos ejemplo 2 Weibull

N(t)	t(horas)
1	68
2	235
3	250
4	500
5	525
6	654
7	654
8	750
9	750
10	966
11	1000
12	1000
13	1250
14	1250
15	1310
16	1320
17	1421
18	1430
19	1500
20	1500
21	1750
22	1750
23	2000
24	2250
25	2500
26	2750
27	3000
28	3250
29	3500
30	3750
31	3750
32	4000
33	23158
34	23158
35	23170

- **Ajuste de la Distribución Weibull y cálculo del MTBF:**

PARAMETROS DEL MODELO WEIBULL

Tabla 14 parámetros de cálculo ejemplo 2 Weibull

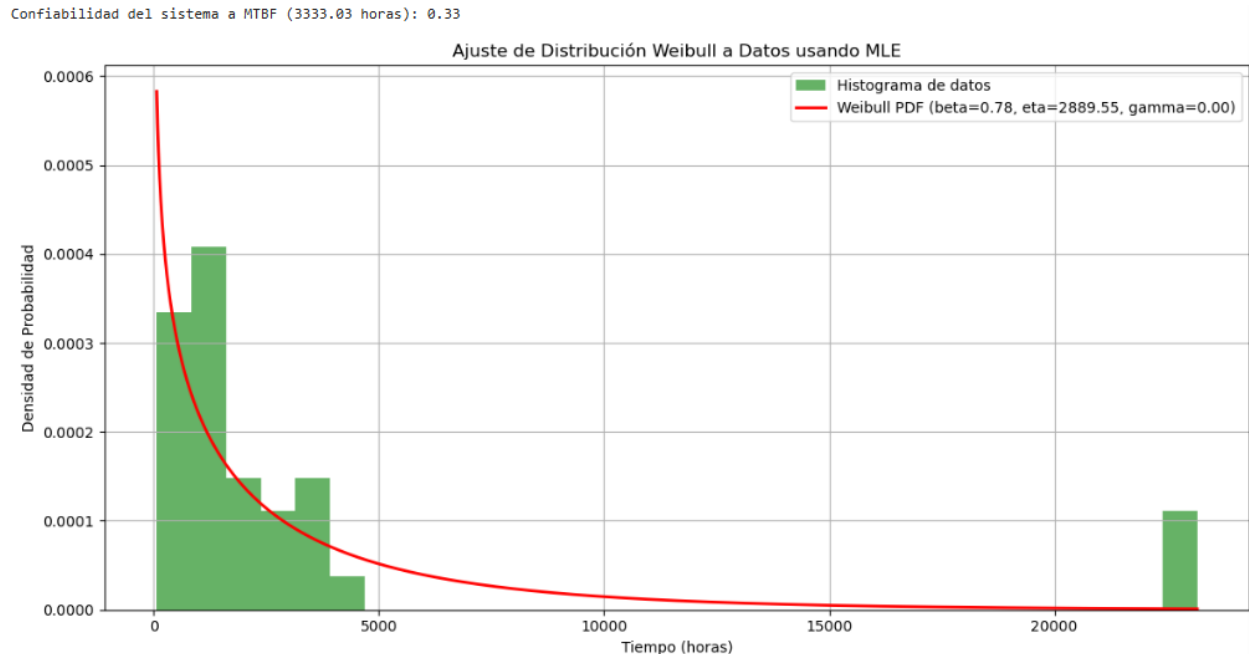
BETA	0,78
ETA	2889,55
GAMMA	0
MTBF	3333.03 horas
Confiabilidad del sistema	0,33

Obtenido de (Autores)

Los resultados del análisis de Weibull indican que el parámetro de forma (Beta) es 0.78, lo que sugiere una tasa de fallas decreciente, se puede concluir que el sistema se encuentra en la fase de fallas iniciales (infantiles) de la curva de la bañera. Esta fase se caracteriza por una alta tasa de fallas al inicio, que disminuye con el tiempo a medida que se corrigen los defectos iniciales. El parámetro de escala (Eta) es 2889.55, lo que representa el tiempo característico en el cual el 63.2% de las unidades habrán fallado. El parámetro de ubicación (Gamma) es 0.00, indicando que no hay un retraso en el inicio de las fallas. El MTBF calculado es de 3333.03 horas, lo que proporciona una medida del tiempo promedio entre fallas, y la confiabilidad a MTBF es 0.33, indicando que hay un 33% de probabilidad de que el sistema funcione sin fallas hasta alcanzar el MTBF. Estos parámetros sugieren que el sistema experimenta una mejora en la confiabilidad después de un período inicial, con una significativa proporción de fallas que ocurren en las primeras 2889.55 horas de operación.

- **Confiabilidad**

Figura 30 ajuste Weibull ejemplo2

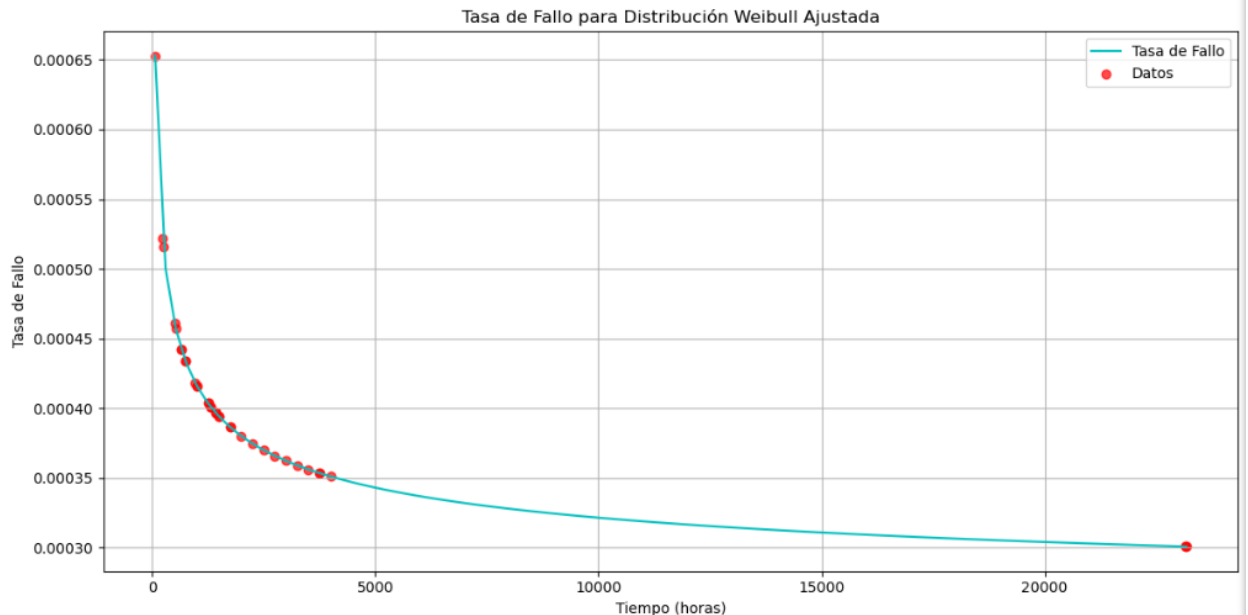


Obtenido de (Autores)

La gráfica muestra el ajuste de la distribución Weibull a los datos de fallas utilizando el método de máxima verosimilitud (MLE). El histograma verde representa la densidad de probabilidad de los datos observados, mientras que la línea roja corresponde a la función de densidad de probabilidad (PDF) de la distribución Weibull ajustada con parámetros Beta (0.78), Eta (2889.55) y Gamma (0.00). La alta densidad de probabilidad inicial, que decrece rápidamente, confirma que el sistema se encuentra en la fase de fallas infantiles de la curva de la bañera, caracterizada por una alta frecuencia de fallas tempranas que disminuye con el tiempo.

- **Tasa de fallos**

Figura 31 tasa de fallos

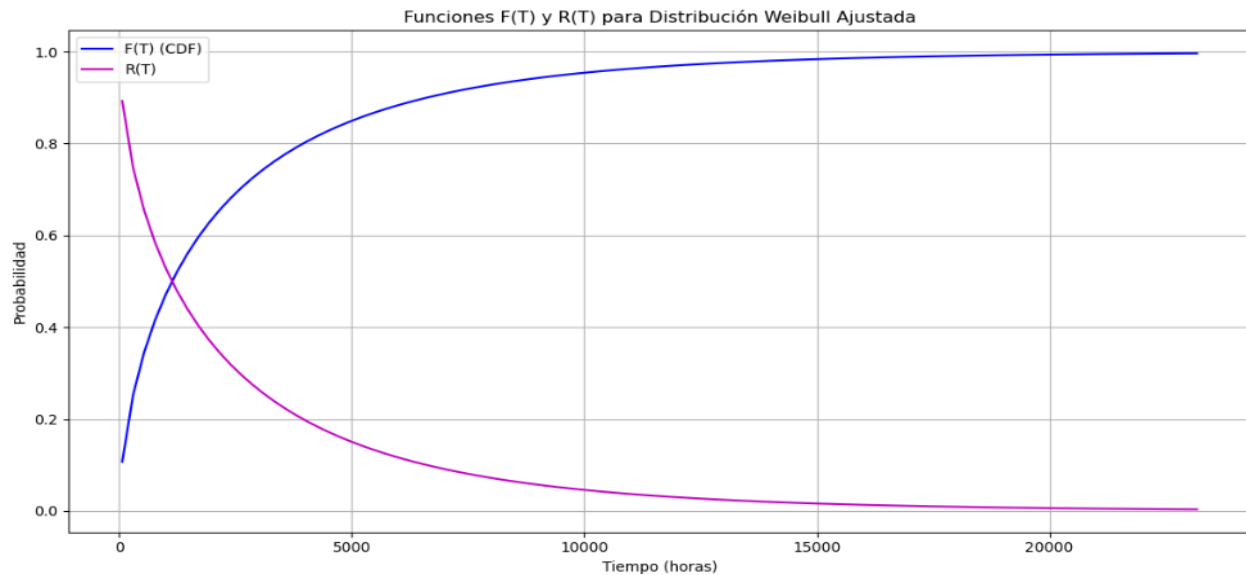


Obtenido de (Autores)

La correspondencia estrecha entre los puntos de datos y la curva ajustada sugiere que el modelo Weibull proporciona un ajuste adecuado a los datos reales, permitiendo una representación precisa de la confiabilidad y la tasa de fallos del sistema a lo largo del tiempo. Esta información es crucial para la toma de decisiones en cuanto a la planificación del mantenimiento y la mejora de la confiabilidad operativa.

- **Función $R(t)$ y $F(t)$**

Adicional el modelo Weibull muestra las gráficas de confiabilidad y probabilidad de fallas a medida que avanza el tiempo.



Obtenido de (Autores)

6.12. Conclusión del estudio

Se recomienda utilizar el modelo Crow AMSAA ya que facilita la predicción de la ocurrencia de fallas futuras como se muestra en la figura 28, lo que permite una planificación proactiva del mantenimiento. Conocer cuándo es probable que ocurran fallas ayuda a programar mantenimientos preventivos, reduciendo el tiempo de inactividad y los costos asociados, a diferencia del Weibull que indica si el sistema es confiable o no confiable Crow AMSAA me ayuda a identificar como se está comportando los sistemas en caso de que se haya realizado un mantenimiento si este está dando resultados o no.

7. CONCLUSIONES

17 El desarrollo de una herramienta informática utilizando software libre para el cálculo de la confiabilidad mediante el Método Crow AMSAA representa un avance significativo en la gestión y mantenimiento de flotas de transporte. Los objetivos específicos alcanzados en este proyecto han permitido no solo la creación de una herramienta eficiente y adaptada, sino también un entendimiento más profundo y comparativo de las técnicas estadísticas y las soluciones de software disponibles en el ámbito de la confiabilidad.

Al comparar técnicas estadísticas para calcular la confiabilidad en flotas de transporte, el modelo Crow-AMSAA demuestra ser superior en varios aspectos. Por ejemplo, en el ejemplo 2, Crow-AMSAA con $\beta = 0.31$, $\eta = 0.19$, Growth rate = 0.69 y un MTBF de 3487.69 horas reveló una mejora significativa en la confiabilidad con una reducción de la tasa de fallas, este detalle se demuestra en su valor de β menor con un MTBF mayor frente al modelo Weibull $\beta = 0.78$, $\eta = 2889.55$, $\gamma = 0$ y un MTBF de 3333.03 horas que mostró un 33% por ciento de confiabilidad reduciendo en MTBF por un ajuste menos exacto, Crow-AMSAA permite un seguimiento dinámico de la confiabilidad en sistemas reparables, algo que el análisis de supervivencia no maneja tan eficientemente. Así, Crow-AMSAA ofrece una visión más detallada y adaptable, optimizando mejor las estrategias de mantenimiento y gestión de riesgos en flotas de transporte.

El desarrollo de la herramienta informática utilizando Python como lenguaje de programación ha resultado en una aplicación robusta y específica para las necesidades de las flotas de transporte. La herramienta incorpora capacidades avanzadas de análisis, permitiendo a los usuarios ingresar datos de fallas, realizar cálculos detallados de confiabilidad, y visualizar resultados a través de gráficos intuitivos y detallados. La adaptabilidad de la herramienta permite

su ajuste a distintos escenarios y tipos de datos, ofreciendo una solución versátil y eficiente para los gestores de flotas.

La implementación de esta herramienta informática tiene un impacto significativo en la gestión de flotas de transporte. Permite una planificación proactiva del mantenimiento, basada en datos precisos y análisis confiables, lo que ³⁵ reduce el tiempo de inactividad y los costos asociados con las fallas inesperadas. Además, la capacidad de predecir fallas futuras y entender las tendencias de fallas a lo largo del tiempo permite a los gestores optimizar los recursos, mejorar la confiabilidad de la flota ² y, en última instancia, aumentar la satisfacción del cliente.

8. RECOMENDACIONES

Para maximizar el impacto y la efectividad de la herramienta informática desarrollada para el cálculo de la confiabilidad mediante el Método Crow AMSAA, se recomienda adoptar un enfoque holístico y proactivo que abarque la integración continua, capacitación de usuarios, ampliación de funcionalidades, automatización de procesos de datos, y colaboración con expertos. Es crucial establecer un sistema de monitoreo y retroalimentación constante para evaluar el rendimiento de la herramienta y realizar mejoras basadas en datos y experiencias de los usuarios. Además, se debe asegurar la seguridad y protección de los datos, fomentar el desarrollo de una comunidad activa alrededor de la herramienta y mantenerla adaptable a nuevas tecnologías emergentes. Este enfoque garantizará que la herramienta no solo cumpla con los requisitos actuales de las flotas de transporte, sino que también pueda evolucionar y mejorar continuamente para enfrentar los desafíos futuros en la gestión de la confiabilidad operativa.

9. REFERENCIAS

- Akiwatkar, R. (2022). Top Programming Languages of 2022: A Compilation of Key Statistics. *Simform - Product Engineering Company*.
<https://www.simform.com/blog/top-programming-languages/>
- Casilla, A. (2018). *Confiabilidad de una flota de cargadores de bajo perfil aplicando el modelo estocástico de Crow Amsaa—Unidad Minera Arcata, Arequipa*.
<http://repositorio.unac.edu.pe/handle/20.500.12952/6051>
- Challenger, I., Díaz, Y., & Becerra, A. (2014). El lenguaje de programación Python. *Ciencias Holguín, XX* (2), 1-13.
<https://www.redalyc.org/pdf/1815/181531232001>.
- Fernandez, J. (2016). *Introducción a la gestión de flotas de vehículos*.
https://www.researchgate.net/profile/Jose-Fernandez-Gomez/publication/303864416_Introduccion_a_la_gestion_de_flotas_de_vehiculos
- Latam, S. (2022). Tipos de flotas: Una visión general. *Skydata Latam*.
<https://skydatalatam.com/blog/flotas-de-transporte-cuales-son-los-principales-tipos>.
- Luzuriaga, O., & Paul, A. (s. f.). *Desarrollo de una herramienta informática mediante el uso de software libre para el cálculo de la confiabilidad en flotas de transporte*.
<http://dspace.ups.edu.ec/handle/123456789/23144>
- Mira, A. (2021). *Ciencia de datos: Distribución con Anaconda Python*. Tokio School.
<https://www.tokioschool.com/noticias/ciencia-datos-anaconda-python/>
- NumPy -. (s. f.). Recuperado 29 de mayo de 2024, de <https://numpy.org/>
- Pandas documentation - Pandas 2.2.2 documentation. (s. f.). Recuperado 29 de mayo de 2024, de <https://pandas.pydata.org/docs/>

Sequeda, B. & Diaz, L. (2008). Estudio de confiabilidad de los equipos críticos de la línea de producción planta 1 de Propilco s.a. y la línea de producción compuestos 3 de Ajover s.a.

<https://hdl.handle.net/20.500.12585/1417>

The Advantages and Disadvantages of JavaScript. (5 de diciembre de 2019).

Freecodecamp.Org.

<https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/>

The Evolution of JavaScript: A Journey from ES1 to the Latest Version (Part 1). (s. f.).

Recuperado (29 de mayo de 2024, de <https://www.linkedin.com/pulse/evolution-javascript-journey-from-es1-latest-version-part-lebbos-za9fe>

10. ANEXOS

Figura 32 codificación Weibull Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import weibull_min
from scipy.special import gamma, gammainc, gammaincc

# Cargar datos desde Excel
file_path = r"C:\Users\rsagu\Desktop\PROGRAMA\tbf.xlsx"
df = pd.read_excel(file_path)

# Extraer y ordenar los datos de las columnas
df = df.sort_values(by='t(horas)')
N_t = df['N(t)'].values
t_horas = df['t(horas)'].values

# Ajuste de la distribución Weibull usando MLE
shape, loc, scale = weibull_min.fit(t_horas, loc=0, floc=0)

# Parámetros de la distribución Weibull
beta = shape # Parámetro de forma
eta = scale # Parámetro de escala
gamma_val = loc # Parámetro de ubicación

# Calcular MTBF
mtbf = eta * gamma(1 + 1/beta)

# Imprimir MTBF
print(f"MTBF (Mean Time Between Failures): {mtbf:.2f} horas")

# Calcular F(T) y R(T) para los datos
t_values = np.linspace(np.min(t_horas), np.max(t_horas), 100)
F_t = gammainc(shape, (t_values / eta)**beta)
R_t = gammaincc(shape, (t_values / eta)**beta)

# Generar datos para la curva de densidad de Weibull ajustada
x = np.linspace(np.min(t_horas), np.max(t_horas), 10000)
pdf = weibull_min.pdf(x, shape, loc, scale)

# Calcular la confiabilidad del sistema para el tiempo MTBF
reliability_at_mtbf = weibull_min.sf(mtbf, shape, loc, scale)

# Calcular la tasa de fallo
t_values_for_hazard = np.linspace(np.min(t_horas), np.max(t_horas), len(t_values))
pdf_for_hazard = weibull_min.pdf(t_values_for_hazard, shape, loc, scale)
R_t_for_hazard = gammaincc(shape, (t_values_for_hazard / eta)**beta)
hazard_rate = pdf_for_hazard / R_t_for_hazard

# Mostrar los resultados en una tabla
results_df = pd.DataFrame({
    'N(t)': N_t,
    't(horas)': t_horas,
    'F(T) (CDF)': gammainc(shape, (t_horas / eta)**beta),
    'R(T) (Reliability)': gammaincc(shape, (t_horas / eta)**beta)
})
```

Obtenido de (Autores)

Figura 33 codificación Crow AMSAA Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Cargar el archivo Excel sin especificar el nombre de la hoja
archivo_excel = r"C:\Users\rsaqu\Desktop\PROGRAMA\tbf.xlsx"
df = pd.read_excel(archivo_excel)

# Asumiendo que la columna de interés se llama 't(horas)'
columna_tiempo_fallas = 't(horas)'

# Ordenar los tiempos de fallas en orden ascendente
df_sorted = df.sort_values(by=columna_tiempo_fallas)

# Calcular el ln de la columna 't(horas)' y crear una nueva columna 'ln_t(horas)'
df_sorted['ln_' + columna_tiempo_fallas] = np.log(df_sorted[columna_tiempo_fallas])

# Calcular el MTBF acumulativo
df_sorted['MTBF_cumulative'] = df_sorted[columna_tiempo_fallas].cumsum() / (df_sorted.index + 1)

# Calcular  $\theta$ 
n = len(df_sorted)
ultimo_tiempo_falla = df_sorted[columna_tiempo_fallas].iloc[-1]
suma_ln_fallas = np.log(df_sorted[columna_tiempo_fallas]).sum()
beta = n / (n * np.log(ultimo_tiempo_falla) - suma_ln_fallas)

# Calcular Lambda ( $\lambda$ )
lambda_value = n / (ultimo_tiempo_falla ** beta)

# Calcular la curva de ajuste Crow-AMSAA
df_sorted['Ajuste_Crow_AMSAA'] = lambda_value * df_sorted[columna_tiempo_fallas]**beta

# Calcular la confiabilidad
df_sorted['Confiabilidad'] = np.exp(-lambda_value * df_sorted[columna_tiempo_fallas]**beta)

# Calcular la tasa de fallos
df_sorted['Tasa_de_fallos'] = lambda_value * df_sorted[columna_tiempo_fallas] ** (beta - 1)

# Calcular la tasa de cambio de la confiabilidad
df_sorted['Cambio_Confiabilidad'] = -lambda_value * beta * df_sorted[columna_tiempo_fallas] ** (beta - 1) * np.exp(-lambda_value * df_sorted[columna_tiempo_fallas]**beta)

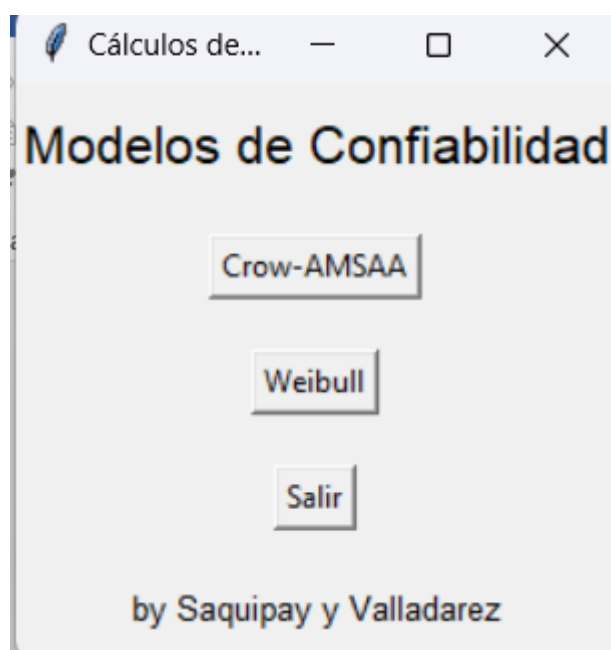
# Calcular el MTBF total
mtbf_total = df_sorted[columna_tiempo_fallas].sum() / len(df_sorted)

# Mostrar los tiempos de fallas ordenados, sus respectivos ln y MTBF acumulativo
print("Tiempos de fallas ordenados, ln(tiempos) y MTBF acumulativo:")
print(df_sorted[[columna_tiempo_fallas, 'ln_' + columna_tiempo_fallas, 'MTBF_cumulative']].to_string(index=False))

# Mostrar  $\theta$ ,  $\lambda$  y MTBF total
print(f"beta calculado: {beta}")
print(f"Lambda calculado: {lambda_value}")
print(f"MTBF total calculado: {mtbf_total}")
```

Obtenido de (Autores)

Figura 34 interfaz



Obtenido de (Autores)

● 9% de similitud general

Principales fuentes encontradas en las siguientes bases de datos:

- 7% Base de datos de Internet
- Base de datos de Crossref
- 7% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

FUENTES PRINCIPALES

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	es.slideshare.net Internet	2%
2	coursehero.com Internet	<1%
3	Infile on 2022-09-15 Submitted works	<1%
4	iti.es Internet	<1%
5	Universidad Nacional Tecnologica De Lima Sur on 2024-07-22 Submitted works	<1%
6	ribuni.uni.edu.ni Internet	<1%
7	slideshare.net Internet	<1%
8	repositorio.unajma.edu.pe Internet	<1%

9	Universidad Nacional de San Cristóbal de Huamanga on 2021-12-09	<1%
	Submitted works	
10	researchgate.net	<1%
	Internet	
11	R. E Lopez. "Agile software development applied to the management o...	<1%
	Crossref	
12	Corporación Universitaria Minuto de Dios, UNIMINUTO on 2024-03-30	<1%
	Submitted works	
13	docplayer.es	<1%
	Internet	
14	clubensayos.com	<1%
	Internet	
15	Imperial College of Science, Technology and Medicine on 2023-06-27	<1%
	Submitted works	
16	repositorio.unap.edu.pe	<1%
	Internet	
17	Universidad Ricardo Palma on 2018-05-16	<1%
	Submitted works	
18	Universidad de Santiago de Chile on 2021-01-14	<1%
	Submitted works	
19	Universidad Politecnica Salesiana del Ecuador on 2022-07-28	<1%
	Submitted works	
20	Universidad TecMilenio on 2024-07-25	<1%
	Submitted works	

21	isolution.pro Internet	<1%
22	palomarecuero.wordpress.com Internet	<1%
23	sdg.humanrights.dk Internet	<1%
24	Universidad TecMilenio on 2024-02-05 Submitted works	<1%
25	Universidad Politecnica Salesiana del Ecuador on 2022-01-06 Submitted works	<1%
26	monografias.com Internet	<1%
27	Infile on 2019-10-07 Submitted works	<1%
28	Universiti Malaysia Perlis on 2018-12-28 Submitted works	<1%
29	es.scribd.com Internet	<1%
30	idoc.pub Internet	<1%
31	mit.ocw.universia.net Internet	<1%
32	Universidad Nacional Abierta y a Distancia, UNAD,UNAD on 2024-05-08 Submitted works	<1%

33	fr.slideshare.net Internet	<1%
34	pt.scribd.com Internet	<1%
35	rigaku.com Internet	<1%
36	Clara Riba, Xavier Ballart. "La motivación para el servicio público de los..." Crossref	<1%
37	Lent, Sally Peilynn. "Local and Landscape Effects on Wild Bees Along a..." Publication	<1%
38	Tecsup on 2021-08-23 Submitted works	<1%
39	es.unionpedia.org Internet	<1%
40	hackaboss.com Internet	<1%
41	sebd.ehu.es Internet	<1%
42	yayiinacloseyoureyes.blogspot.com Internet	<1%
43	"Productos de precipitación satelital de alta resolución espacial y temp..." Crossref posted content	<1%
44	Universidad Internacional de la Rioja on 2022-02-09 Submitted works	<1%

45	Universidad Tecnologica del Peru on 2016-12-12 Submitted works	<1%
46	atvise.vesterbusiness.com Internet	<1%
47	repositorio.uide.edu.ec Internet	<1%
48	repositorio.unican.es Internet	<1%
49	solocannabis.com Internet	<1%
50	economiaindustrial.com Internet	<1%
51	rediris.es Internet	<1%
52	rte.espol.edu.ec Internet	<1%

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Fuentes excluidas manualmente
- Material citado
- Coincidencia baja (menos de 9 palabras)
- Bloques de texto excluidos manualmente

FUENTES EXCLUIDAS

dspace.ups.edu.ec	7%
Internet	
hdl.handle.net	1%
Internet	
1library.co	<1%
Internet	
Submitted works	<1%
Universidad Católica de Santa María on 2021-04-05	<1%
Submitted works	
upcommons.upc.edu	<1%
Internet	
catalonica.bnc.cat	<1%
Internet	
Universidad Distrital FJDC on 2021-05-06	<1%
Submitted works	
documentop.com	<1%
Internet	

d.documentop.com	<1%
Internet	
Universidad TecMilenio on 2024-01-17	<1%
Submitted works	
pdfslide.tips	<1%
Internet	
Universidad Politecnica Salesiana del Ecuador on 2020-09-20	<1%
Submitted works	
Universidad Politecnica Salesiana del Ecuador on 2020-09-20	<1%
Submitted works	
ups.edu.ec	<1%
Internet	
Universitat Politècnica de València on 2024-07-16	<1%
Submitted works	
agualatinoamerica.com	<1%
Internet	
recercat.cat	<1%
Internet	
Universidad Carlos III de Madrid on 2019-09-23	<1%
Submitted works	
Dimy Nanclares Fernandes Sanches. "Nanotubos de hidróxidos lamelares: co...	<1%
Crossref posted content	

BLOQUES DE TEXTO EXCLUIDOS

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA CARRERA DE INGENIERÍA ...

dspace.ups.edu.ec

TUTOR: ING. CRISTIAN LEONARDO GARCIA GARCIA, PhD. Cuenca - Ecuador

dspace.ups.edu.ec

Trabajo de titulación previo a la obtención del título de Ingeniero Automotriz AUT...

dspace.ups.edu.ec

con documento de identificación N

dspace.ups.edu.ec

¡CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN...

dspace.ups.edu.ec

con documento de identificación N

dspace.ups.edu.ec

confiabilidad

dspace.ups.edu.ec

el cual ha sido desarrollado para optar por el título de: Ingeniero Automotriz en la U...

dspace.ups.edu.ec

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN Yo, con documento d...

Universidad Politecnica Salesiana del Ecuador on 2022-02-26

CONFIABILIDAD

dspace.ups.edu.ec

con documento de identificación N

dspace.ups.edu.ec

iiiDedicatoriaDedico

dspace.ups.edu.ec

que todos mis sueños se hagan realidad y

naruto.fandom.com

a Dios por haberme

repositorio.espam.edu.ec

han sido el pilar sobre elcual he construido mis sueños. Gracias por

dspace.ups.edu.ec

cuyo amor incondicional y

Berlin School of Business and Innovation on 2024-02-04

en cada paso de este camino. Gracias por

Carmen María Sánchez Arévalo. "Implementación de tecnología de membranas para la valorización de los c...

expresar mi

dspace.ups.edu.ec

a pesar de

dspace.ups.edu.ec

a mi tutor de tesis, Ing

dspace.ups.edu.ec

agradecimiento a todos losdocentes de la Universidad Politécnica Salesiana

www.coursehero.com

viAgradecimientoAgradezco a Dios por darme

dspace.ups.edu.ec

se han convertido en una fuente de inspiración para

www.iucn.org

especial a mi tutor de tesis

repositorio.ug.edu.ec

la gestión de flotas de transporte

docplayer.es

la vida útil del producto

www.slideshare.net

el lenguaje de programación debido a su

Universidad Nacional Pedro Ruiz Gallo on 2020-12-20

una base sólida para la toma de decisiones informadas y la mejora continua de los ...

mx.xamai.com

Dedicatoria

repositorio.uss.edu.pe

9.BIBLIOGRAFIA Error! Bookmark not defined.10.AN...

Universidad de Chile on 2017-07-02

ÍNDICE DE FIGURASFigura 1

Universidad Tecnológica del Perú on 2017-03-12

lenguaje de programación

revistas.ulasalle.edu.pe

ya sea por limitaciones económicas o por la falta de

www.acnur.org

garantizar la gestión de las finanzas públicas de manera sostenible y transparente

view.genial.ly

a un conjunto de vehículos destinados a transportar mercancías o

docplayer.es

a medida que

Universidad Internacional del Ecuador on 2020-01-20

cuentan con un

es.scribd.com

ConfiabilidadLa confiabilidad puede ser definida como la

tangara.uis.edu.co

proporcionauna base sólida para la toma de decisiones informadas

www.coursehero.com

en el análisis de la fiabilidad de sistemas reparables

www.ujaen.es

El modelo de Crow AMSAA

Universidad Cesar Vallejo on 2022-12-06

antes de aplicar el modelo, se

Universidad de Santiago de Chile on 2021-09-15

la verificación de la integridadde los datos y la

cloud.google.com

el parámetro de escala (α) y el parámetro de

www.ing.unlp.edu.ar

Numero de Fallas o costos

docplayer.es

es el parámetro de forma

www.slideshare.net

β : es el parámetro de forma

Instituto Superior de Artes, Ciencias y Comunicación IACC on 2021-12-28

es el parámetro de forma

www.slideshare.net

es el parámetro de forma

www.coursehero.com

con el análisis estadístico y la interpretación de los resultados

gacetasanitaria.org

JavaScriptJavaScript es un lenguaje de programación basada en prototipos, multi...

repositorio.uesiglo21.edu.ar

Ventajas de JavaScriptVelocidad - JavaScript tiende a ser muy rápido porque a me...

www.freecodecamp.org

Desventajas de JavaScriptSeguridad Client-Side- Desde que el código en JavaScri...

www.freecodecamp.org