

Diseño e implementación de software para optimización del volumen de material y análisis de cargas en el proceso de distribución de tubos.

Alonso Beltrán Gutiérrez y Junior Fabián Pacheco Moyano

Trabajo de Grado para Optar al Título de ingeniero mecánico

Director

Jabid Eduardo Quiroga Méndez

Doctor en ingeniería civil

Universidad Industrial de Santander

Facultad de fisicomecánicas

Escuela de Ingeniería mecánica

Ingeniería mecánica

Bucaramanga

2023

Dedicatoria

A mis padres Orlando Beltrán y Mercedes Gutiérrez, y a mis hermanos Sneider, Maritza, Fredy, Luz Neida y Mayer les dedico este gran trabajo con mucho cariño para ellos quienes en todo momento me apoyaron con su amor incondicional en toda mi carrera, gracias por sus enseñanzas valores transmitidos durante toda mi vida y en esta etapa académica más que nunca, gracias por infundir en mi pasión por lo que se quiere, gracias por su amor y apoyo en todo momento.

Alonso Beltrán Gutiérrez

A mi madre Luz Nelfy, mi padre que en el cielo esta Arnulfo León, a mis tíos Yibson, Sandra, Yolanda y finalmente a mi mujer Carolina Espitia por ese apoyo incondicional que me dieron a lo largo de toda mi carrera, gracias por siempre creer en mí y haber sido parte de este proceso, y siempre motivarme a no desistir, gracias de todo corazón.

Junior Fabián Pacheco Moyano

Agradecimientos

A mi familia por su apoyo incondicional, un agradecimiento incondicional a residencias universitarias UIS quien fue más que un hogar una familia, a mis compañeros colegas que hicieron parte de este tan anhelado sueño, un agradecimiento muy especial, a Junior Pacheco por su apoyo, responsabilidad y gran dedicación para cumplir este gran logro juntos, un agradecimiento a Hader Ardila un amigo incondicional en toda mi formación profesional. Gracias a Néstor D'cros profesor de la escuela de ingeniería mecánica por infundir en mi pasión por la profesión en todo momento gracias por sus enseñanzas sus consejos de profesional, su motivación y toda su ética profesional como también un agradecimiento incondicional a todos mis docentes de formación por brindarnos sus conocimientos en todo este proceso, Gracias a Jabid Quiroga por su confianza y permitirnos hacer parte de este proyecto, muy agradecido con todo su apoyo en todo momento que hicieron posible este tan gratificante logro.

Alonso Beltrán Gutiérrez

Quiero expresar mi sincero agradecimiento a mi familia por su apoyo incondicional. También deseo agradecer de manera incondicional a residencias universitarias UIS, quienes no solo fueron un hogar, sino una verdadera familia. Mis compañeros colegas, quienes formaron parte de este anhelado sueño, merecen un agradecimiento muy especial. A Alonso Beltrán, quiero agradecerle por su apoyo, responsabilidad y dedicación, que nos permitieron alcanzar juntos este gran logro. No puedo dejar de mencionar a Héctor Hernando y Carlos Jose, dos amigos que han sido pilares importantes a lo largo de mi carrera. Además, quiero expresar un agradecimiento incondicional a todos mis profesores, quienes compartieron sus

conocimientos durante todo este proceso de formación. A Jabid Quiroga, quiero agradecerle por su confianza y por permitirnos formar parte de este proyecto. Estoy enormemente agradecido por todo el apoyo brindado en cada momento, ya que hizo posible este logro tan gratificante.

Junior Fabián Pacheco Moyano

Tabla de Contenido

Introducción	12
1. Descripción del problema	13
1.1 Antecedentes.....	14
2. Justificación	15
3. Objetivos	16
3.1 Objetivo general.....	17
3.2 Objetivos específicos.....	17
Marco teórico.....	19
4. Referencias.....	22
4.1 ¿Qué es Python?	22
4.1.2 P.....	24
5. Diseño de la investigación.....	27
5.1 Población.....	27
5.2 Muestra	28
5.3 Etapas	30
5.4 Presupuesto	31
6. Diseño conceptual.....	32
6.1 Descripción diseño conceptual	34
Análisis	37
7. Análisis de resultados	37

7.1 Análisis de carga.....	44
8. Conclusiones.....	53
9. Recomendaciones.....	54
Bibliografía.....	55
Apéndices.....	56

Lista de Tablas

Tabla 1. Presupuesto del proyecto31

Lista de Figuras

	Pág.
Figura 1. Interfaz de usuario	33
Figura 2. Registro de datos	34
Figura 3. Listado de tubos	35
Figura 4. Listado de tubos	35
Figura 5. Volumen de container.....	36
Figura 6. Diseño análisis de tubos (expectativa)	37
Figura 7. Medidas container 20 pies.....	38
Figura 8. Primer ensayo.....	39
Figura 9. Muestra de tubos dentro del container.....	40
Figura 10. Propiedades del polietileno	44
Figura 11. Ficha técnica de la norma ASTM F714.....	45
Figura 12. Grafica esfuerzo deformación.....	46
Figura 13. Esfuerzo deformación del polietileno.....	46
Figura 14. Vista frontal del tubo (DCL).....	47
Figura 15. Distribución de fuerzas vista lateral	48
Figura 16. Diagrama de fuerza unificada	48
Figura 17. Grafica de análisis de cargas del primer ensayo	51

Lista de Apéndices

	pág.
Apéndice A. Codificación.....	56

Resumen

Título: Diseño e implementación de software para optimización del volumen de material y análisis de cargas en el proceso de distribución de tubos.

Autor: Alonso Beltrán Gutiérrez y Junior Fabián Pacheco Moyano

Palabras Clave: optimización, volumen, programación, Python, tubos.

Descripción: El presente proyecto corresponde a una propuesta de mejora empresarial, en la parte de logística y transporte dirigido a aquellas empresas dedicadas a la fabricación y distribución de tubos, basado en diseñar una solución que permita la mejora en transporte de tubos utilizando una herramienta tecnológica como aplicación. Como resultado obtenemos un servicio ágil y seguro que permite asegurar y optimizar el volumen de cada carga al momento de su transporte y distribución basado en análisis de cargas, esto permite un eficiente uso de cada volumen transportado según los requerimientos de cada cliente. La planeación del presente proyecto se realizó con base a información estándar obtenida por diferentes compañías dedicadas al transporte de este material.

* Trabajo de grado

** Facultad de ingeniería fisicomecánicas. Escuela de ingeniería mecánica. Director: Jabid

Abstract

Title: Diseño e implementación de software para optimización del volumen de material y análisis de cargas en el proceso de distribución de tubos

Author(s): Alonso Beltran Gutiérrez y Junior Fabián Pacheco Moyano

Key Words: Optimization, volume, programming, Python, tubes.

Description: This project corresponds to a proposal for business improvement in the logistics and transportation aimed at those companies engaged in the manufacture and distribution of pipes, based on designing a solution that allows the improvement in transportation of pipes using a technological tool as an application. As a result we obtain an agile and safe service that allows to ensure and optimize the volume of each load at the time of transport and distribution based on load analysis, this allows an efficient use of each transported volume according to the requirements of each client. The planning of this project was based on standard information obtained from different companies dedicated to the transportation of this material.

Eduardo Quiroga Méndez. Doctor en ingeniería civil.

* Degree Work

** Facultad de ingeniería fisicomecánicas. Escuela de ingeniería mecánica. Director: Jabid

Eduardo Quiroga Méndez. Doctor en ingeniería civil.

Introducción

Los grandes avances tecnológicos de hoy en día hacen que todo tienda automatizarse en busca de la mejora de procesos, generando productos más asequibles, siendo competitivos en precios de mercado. En la actualidad se busca optimizar todos los procesos es por ello que se implementan diferentes formas o métodos que faciliten las tareas que se realizan en la vida cotidiana de forma manual sin tener en cuenta las variables que influyen en estos procesos.

En los últimos años la tecnología empezó a jugar un papel importante en las compañías, es por esto que estas destinan recursos a desarrollar y expandir su negocio mediante aplicaciones web o programas offline desarrollados por empresas que proveen estos servicios, para que puedan cumplir una o varias tareas, la mayoría de estos software ayuda en la toma de decisiones. En muchas ocasiones las empresas optan por destinar personal encargado para estas responsabilidades sin tener en cuenta factores alteran el óptimo rendimiento aumentando costos y tiempo.

El principal propósito del presente trabajo es el diseño y creación de una aplicación que sea ágil, amigable y de fácil manejo que permita optimizar procesos dentro de una compañía que se dedique a la fabricación y distribución de tubos, esta ayudaría especialmente en el área de logística y transporte.

Se sabe de antemano que una compañía dedicada a la fabricación de tubos tiene un departamento encargado de logística y transporte, por eso el presente documento está dirigido a todas esas empresas que de una u otra manera deseen tener una mejora en esos procesos relacionando a ello también un eficiente y un seguro transporte.

El presente proyecto lleva, como ya se ha mencionado antes del diseño de una aplicación de escritorio realizada en lenguaje de programación Python, totalmente gratis para aquellas compañías grandes, medianas y pequeñas que quieran utilizarlo, cada una de las líneas de código que realizo en esta aplicación se verán reflejadas en el presente documento

1. Descripción del problema

Muchas empresas en la actualidad buscan una mejora en sus procesos especialmente cuando se trata de logística y transporte de materiales, un proceso que se hace manual es la carga de tubos de polietileno, este proceso se hace en un cálculo manual donde un operario a partir de su destreza y experiencia determina la mejor opción de ordenamiento de los tubos, a su vez este operario determina si hay tubos que se puedan telescopiar (Ingresar un tubo de menor diámetro en uno de mayor diámetro), para aprovechar la mayor cantidad de espacio y que la carga no esté constituida en su mayor parte por espacios vacíos que no tienen un aporte en términos de dinero, siempre teniendo en cuenta el estado de la carga (tubos). El apilar muchos tubos podría afectar a los que se encuentren en la parte inferior porque estos se ven afectados por la carga de los demás y la de sí mismos.

El transporte en una empresa implica varios factores, Según Mora (2023) afirma,

El transporte de carga en los países latinoamericanos se ha convertido en toda una problemática para el sector industrial las empresas generadoras, empresas transportistas y muchos conductores en general dicen que se ha venido creando un panorama de conflictos que solo contribuyen al retraso en el desarrollo de este tan importante sector. Este es uno de los elementos vitales para mantener la economía de un país y toda una globalización, el

transporte no solo se ve afectado por el sistema que manejan las empresas por su logística sino también en la forma en que se está ejecutando siendo métodos poco eficientes. (p.10)

Según la Universidad politécnica de Catalunya en el año (2005) se logró evidenciar una muestra de 500 empresas de cinco países del mundo, Canadá y EEUU permitió determinar en 1986 que el coste logístico transporte de una empresa era el 45%, representando esto un factor importante y a tener en cuenta dentro de una organización.

1.1 Antecedentes

En este apartado el lector encontrara los enfoques que se han tomado durante el diseño y construcción de invernaderos por parte de otros investigadores.

1.1.1 Antecedentes a nivel nacional

Diseño e implemento de una herramienta digital para la generación de informes de balance de material line pipe ofertado a Ecopetrol S.A bajo el contrato 3011938 por la empresa Tenaris Tubo Caribe LTDA.

EL objetivo principal de este proyecto fue diseñar una aplicación digital con el fin de generar informes mensuales de balance de material Line Pipe que la empresa Tenaris oferta a Ecopetrol S.A. todo a través del lenguaje de programa con visual Basic del aplicativo informativo Excel y power BI.

Para lograrlo, se diseñó una interfaz en Excel que permite consolidar toda la información de distintas bases de datos de entrada que corresponden a la cantidad en toneladas de:

Los requerimientos de órdenes de compra, el inventario y la carga de planta. Sobre la cual, se planteó en el programa dos macros para generar reportes que diferencien a la tubería por diámetro y tipo de proceso de fabricación (con o sin costura), se creó en la herramienta power BI un dashboard para monitorear los objetivos de ventas que se planearon al iniciar cada mes con una proyección de demanda por tipo de tubería, destino y gerencia de Ecopetrol, llamado Real vs Plan.

Como conclusión se puede observar la efectividad de implementar herramientas tecnológicas como el software a los procesos manuales, para este caso se logró reducir un 26% el tiempo empleado en la ejecución y mejora en la categorización a simple vista para reconocer cuáles materiales están abastecidos y cuáles no.

2. Justificación

Debido al cambio en la actualidad y la necesidad de innovar, crecer y fortalecer nuestros conocimientos nos valemos de herramientas que nos faciliten estos procesos mejorando cada vez más la calidad en lo que hacemos, el uso de tecnologías implementadas por el hombre en beneficio de toda una sociedad conlleva a una constante innovación en este tipo de industrias como lo son, la implementación de software en empresas que requieran de una u otra manera mejorar su rendimiento y por ende su productividad.

La ingeniería de software de la mano con la ingeniería mecánica convierte procesos que se realizan de forma manual a un forma automatizada, es por ello que la combinación de estas dos importantes ciencias crean un cambio considerable en innovación y desarrollo en donde sean aplicadas, por ende en el presente trabajo de investigación se quiere implementar un software que facilite el proceso de transporte reduciendo el tiempo de carga y el volumen desperdiciado por la misma por la forma particular que presentan las cargas de diferentes empresas.

Estas plataformas como servicios pueden verse como una especialización de utilidad para la empresa en donde se agregan servicios se en la cadena de valor sobre infraestructura base. Es decir, son servicios ubicados en una capa de abstracción más alta que permite no solamente utilizar recursos de tecnología sino también desarrollar y desplegar aplicaciones que contribuyan sobre esos recursos y entrega de recursos de plataforma. Otra característica que puede ofrecer este de carga, que se puede definir como la capacidad de distribución de la carga entre el volumen total disponible. Relacionando esto con el aprovisionamiento automático de instancias y la escalabilidad a demanda, las características de balancear la carga automáticamente entre muchos otros aspectos hace de ello un mejor uso del volumen total para la carga. Las compañías con cierto volumen de material necesitan optimizar al máximo este espacio y este tipo de aplicaciones pueden simular este tipo de proceso haciendo que este sea más rentable para la compañía.

3. Objetivos

A continuación se mencionan los objetivos de alcance de este proyecto tanto general

como específicos

3.1 Objetivo general

- Diseñar e implementar software para optimizar el volumen en el proceso de distribución de tubos en un contenedor para transporte, y analizar las cargas resultado de esta distribución, asegurando un proceso eficiente en espacio y tiempo.

3.2 Objetivos específicos

- Desarrollar software con los siguientes detalles:
 - Lenguaje de programación Python
 - Recibe los siguientes datos: Tipos de tubos (Material y medidas estandarizadas), cantidades de cada tipo de tubo y volumen de almacenamiento (Medio de transporte).
 - Entrega un documento PDF con la distribución óptima de los tubos para el volumen de almacenamiento de carga.
- Diseñar e implementar un software que realice las siguientes tareas:
 - Toma de datos según ingreso de usuario de los tipos de tubos con sus respectivas cantidades y el volumen disponible de almacenamiento.
 - Analizar los datos y determinar la mejor distribución de la carga en el volumen de almacenamiento, basados en el diámetro y longitud de los tubos que permita ingresar otros dentro de este, hasta que se ocupetodo el espacio disponible sin

- afectar la condición y calidad de la carga.
- Determinar los esfuerzos que interactúan en la distribución de la carga para prevenir daños en el material.
 - Exportar un documento donde se observe la configuración óptima de la carga con sus respectivas descripciones que permitan al usuario entender la distribución de una manera confiable, amigable con el usuario y fácil de comprender.
 - Analizar los resultados del software teniendo en cuenta los siguiente:
 - Comparar la duración en los cálculos realizados por el software y de forma manual.
 - Comparar los costos de funcionamiento del software y del operario teniendo en cuenta lo siguiente:
 - Promedio del costo de hora trabajada por operario y software, realizando la misma acción.
 - Cantidad de material que se aprovecha en el mismo volumen de carga.

Marco teórico

Para tener un buen desempeño y desarrollo en todo el proceso, se debe tener un conocimiento previo o haber realizado una investigación conceptual, que permita avanzar de manera concreta y profunda en las actividades investigativas. Por esto, se hace necesario realizar una indagación sobre el contexto a tratar.

Historia de Python: El creador de Python fue en la década de los 80 en sus últimos años, Guido van Rossum, el nombre del lenguaje de programación se debe a que este hombre es fan de la serie Monty Python's y en un diciembre de 1990 tomó la decisión de llamarlo de este modo a su proyecto considerado uno de los más importantes en esa época. A grandes rasgos la historia de Python nos dice que nació como una búsqueda de un sucesor para el lenguaje de programación ABC, que era capaz de llevar y manejar excepciones para interactuar con Amoeba, un sistema operativo. En el año 1991 cuando por primera vez se publicó el código de Python en su versión 0.9.0 y a partir de allí se vinieron sus actualizaciones cada vez más lucrativas y de mayor utilidad para aquellas personas que lo utilizaban, con el paso de los años la evolución se fue dando y se han venido presentando nuevas mejoras a las características ya conocidas y a su vez se plantean con base a su filosofía de Python. En la actualidad es un lenguaje que más se utiliza para muchas clases de desarrollo y hay grandes compañías detrás de su uso y mejora.

¿Qué es Python? Python es un lenguaje de programación que se define como interprete en donde la filosofía de sus códigos de trabajo hacen énfasis siempre a ser un código legible, del mismo modo este lenguaje de programación multiparadigma, puede soportar la orientación de objetos, siendo llamado también un lenguaje orientado a objetos, este lenguaje es una medida de programación fundamental útil para desarrollar mutile herramientas que lo hacen muy funcional.

Por otro lado Python es un lenguaje que se usa en todo el mundo el cual se identifica como un lenguaje de alto nivel razón por la que sus desarrolladores han trabajado para convertirlo en un lenguaje fácil de aprender y con una ventaja de ser código abierto.

Python también es un lenguaje de Scripting independiente de plataforma y orientad a objetos, preparado para realizar cualquier tipo de aplicación en servidores de red o incluso páginas web. Es un lenguaje interpretado, lo que significa que no es necesario compilar el código fuente para poder ejecutarlo, ofrece ventajas como rapidez en sus compilaciones tardando muy poco en comparación con otros lenguajes de programación, ofrece también una velocidad en escritura fácil de interpretar utilizando lectores de código.

¿Para qué sirve Python? Python cuenta con una variedad de usos tanto así que las compañías más grandes del mundo emplean este lenguaje para el desarrollo de sus aplicaciones. Uno de sus objetivos con este lenguaje de programación es que se automaticen los procesos para que así se consiga ahorrar tiempo. Este por esta razón las múltiples soluciones que se logran con pocas líneas de código. Por otra parte Python es muy pertinente para el trabajo de análisis de datos en grandes volúmenes que favorecen su extracción y el procesamiento en poco tiempo. Cuando requiere analizar grandes cantidades de datos se suele utilizar este lenguaje así que para trabajos de ciencia es muy útil.

Sus librerías, sus comandos, bibliotecas módulos y de más hacen a este lenguaje que se destaque por encima de otros. Estos son los usos más comunes:

- Cálculos científicos o de la rama de alguna ingeniería
- Desarrollo de aplicaciones tanto móviles como web
- Videojuegos o similitudes

- Programas de desarrollo gráfico
- entre otras aplicaciones

Características de lenguaje

- Multiplataforma: este lenguaje cuenta con versiones en distintos sistemas informáticos.

Originalmente se desarrolló para Unix aunque se puede ejecutar en cualquier sistema compatible con el lenguaje siempre y cuando exista un intérprete para él en el sistema instalado.

- Interpretado: en este lenguaje no se debe compilar el código antes de la ejecución como pasaría en otros lenguajes de programación. Lo que hace es que realiza una compilación, pero esta se realiza de manera transparente para el programador. En algunos casos cuando se ejecuta un código por primera vez se producen unos bits que se guardan automáticamente en el sistema y que de cierto no sirve para acelerar la compilación implícita que se realiza cada vez que ejecutamos el mismo código.

- Interactivo: Python dispone de un intérprete por líneas de comando en el que se puede introducir sentencias. Cada sentencia se compila y se transforma en un marco visible para ayudarnos a entender de manera más simple el lenguaje y probar los resultados de ejecución por líneas de código.

- Orientado a objetos: este es un lenguaje que permite y está diseñado para programación orientada a objetos y ofrece múltiples casos de manera sencilla de crear programas con componentes reutilizables.

- Funciones o Librerías: cuenta con muchas funciones incorporadas en el mismo lenguaje para el tratamiento numérico y alfabético de archivos entre otros, además de ello existen otras librerías que podemos exportar y utilizar en nuestro código para tratar desarrollar temas específicos.

•Sintaxis clara: este lenguaje se destaca entre otros por contar con una sintaxis muy visual y fácil de interpretar gracias a que cuenta con una notación dentada (con márgenes en sus líneas de código) de obligado cumplimiento. En otros lenguajes para separar códigos, se utilizan otros elementos como llaves, palabras clave entre otros. En cambio para separar el código por porciones en Python se debe tabular hacia dentro, esto permite llevar un orden y además para aquellos programadores que están iniciando los mantiene en su margen descartando cualquier probabilidad de que se pierdan en su propio código. Esto también ayuda a que todos los programadores adopten una misma notación, y que los programas de cualquier persona tengan un aspecto muy similar.

4. Referencias

Este apartado tiene como finalidad reunir información acerca de los temas a tratar durante el desarrollo del proyecto.

4.1 ¿Qué es Python?

Según la Universidad de Alcalá, Python es un lenguaje de programación interpretado, orientado a objetos de alto nivel y con semántica dinámica. Su sintaxis hace énfasis en la legibilidad del código, lo que facilita su depuración y, por tanto, favorece la productividad. Ofrece la potencia y la flexibilidad de los lenguajes compilados con una curva de aprendizaje suave. Esta es la razón por la que se incluye como lenguaje de programación básico en un Máster en Data Science.

Aunque Python fue creado como lenguaje de programación de uso general, cuenta con

una serie de librerías y entornos de desarrollo para cada una de las fases del proceso de Data Science. Esto, sumado a su potencia, su carácter open source y su facilidad de aprendizaje le ha llevado a tomar la delantera a otros lenguajes propios de la analítica de datos por medio de Data Science como pueden ser SAS (software comercial líder hasta el momento) y R (también open source, pero más propio de entornos académicos o de investigación).

Figura 1. Logo de Python



Nota: la figura representa el logo de Python. Tomado de <https://www.python.org/>

4.1.1 Software de código abierto (Open source software)

El término Open Source, hace referencia a un tipo de código o de programa escrito de manera que se pueda modificar por cualquier usuario que así lo desee. Un programa hecho de esta manera se puede modificar según los deseos particulares de aquella persona que lo va a usar. Open Source al principio salió con el nombre de software gratis, sin embargo los desarrolladores de la iniciativa: software que se puede compartir con todo el mundo, decidieron que la palabra “gratis” (Del inglés, free software) no se adecuaba exactamente a lo que se quería expresar. La palabra gratis, implicaba por lo general términos como, fácil, descuidado etc (Francisco Cano, 2014).

4.1.2 Principio investigación de operaciones según Churchman, Ackoff y Arnoff

La investigación de operaciones es la aplicación, por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (hombre-máquina), a fin de que se produzcan soluciones que mejor sirvan los objetivos de la organización.

Este principio busca coordinar las operaciones independientemente de la naturaleza de estos y se basa en “hacer investigación sobre las operaciones”. Este empieza por la observación luego se pasa a construir un modelo que por lo general es matemático que intenta extraer la idea principal del problema, luego se plantean hipótesis sobre de que el modelo sea suficiente para que las soluciones que este plantea sean validas para el problema real y por último se llevan a cabo los experimentos adecuados para probar esta hipótesis, modificarla si es necesario y eventualmente verificarla. (Roberto Carro, 2009)

4.1.3 Programación Python usando un framework (Tkinter)

En Python existen diferentes librerías que facilitan al programador al momento de crear un proyecto, entre ellas están los frameworks, que nos permiten crear aplicaciones con las que trabajamos en nuestro ordenador, en modo de texto o graficas, en modo de texto o gráficas. Tkinter nos permite crear ventanas diseñadas para interactuar entre el ordenador y el usuario por medio de una interfaz gráfica (conexión), donde dispondremos de múltiples elementos gráficos, como botones, cuadros de texto, ventanas, menús y de más para tratar información.

Utilizando tkinter Python hay muchas formas de crear una interfaz gráfica todas ellas muy sencillas de aprender y de entender, es multiplataforma y, además, viene incluido con Python en su versión para Windows, para Mac y para la mayoría de las distribuciones GNU/Linux. A pesar de su larga historia, su uso no es demasiado extendido con otros usuarios de equipos personales porque su integridad visual con los sistemas operativos no era buena y proporcionaba pocos widgets (controles), para construir programas gráficos. Se le considera el estándar de facto en la programación GUI con Python. Sin embargo las nuevas actualizaciones han venido mejorando cada día más este aspecto tanto así hasta llegar a tener una interfaz con un aspecto notablemente en lo que se refiere a integridad visual, también las nuevas actualizaciones han desarrollado más controles permitiendo al programador una amplia gama en variedad en cuanto a sus diseños gráficos visuales.

4.1.4 Matplotlib

Matplotlib es una librería de Python más utilizada para dibujar gráficos y visualizar análisis de datos, para reproducción fácilmente figuras para su publicación y ser utilizados en diferentes plataformas. Matplotlib es una biblioteca para hacer múltiples gráficos en conjunto con otras librerías sirve para mostrar análisis estadístico de varias formas visualizando todos y cada uno de sus puntos con las concavidades y elongaciones que su gráfica replantaría utilizando otro software. Aunque Matplotlib está escrito principalmente en Python puro, hace un uso intensivo de NumPy y otros códigos de extensión para proporcionar un buen rendimiento incluso para matrices grandes.

Matplotlib fue desarrollado en el 2002 por John Hunter neurobiólogo de profesión creó esta librería con la finalidad de observar en el cerebro las señales eléctricas de personas que

padecieran de epilepsia. Lo que buscaba era imitar las funcionalidades de creación gráfica que tenía Matlab con Python.

Debido a su carácter *open source* o de código abierto, esta librería, pese a la muerte de su creador en 2012, ha seguido teniendo significativos avances, hasta el punto de que actualmente por medio de ella se pueden crear trazados, histogramas, diagramas de barra y otros tantos tipos de gráficas con poco código. Matplotlib se utiliza en servidores de aplicaciones web, *shells* y *scripts* de Python y es representativamente más útil para aquellas personas que trabajan con NumPy

4.1.5 Visual studio

Visual Studio es un entorno de desarrollo integrado (IDE) para los equipos que construyen aplicaciones que funcionan en Microsoft Windows, Microsoft Office, o en la web de Microsoft y plataformas de dispositivos móviles-. Visual Studio Ultimate Edition incluye todas las de depuración, el diagnóstico y la funcionalidad de las pruebas de la edición Premium, además de herramientas para la gestión del laboratorio, la arquitectura y el modelado. Los desarrolladores que utilizan Visual Studio Ultimate Edition puede optar por el programa a largo Lenguajes de programación de Microsoft, como Visual Basic.NET, Visual C + +. NET, Visual C #. NET, JScript, y docenas más o en varios idiomas que se acaban de apoyo con Visual Studio 2010 como Visual C #, IronRuby y IronPython. Visual Studio 2010 también incluye una mayor integración con SharePoint Server que las versiones anteriores, ayudando a los desarrolladores que necesitan para personalizar una instalación de SharePoint Server 2010.

Balestrini (2000) señala que el marco metodológico “es el conjunto de procedimientos a seguir con la finalidad de lograr objetivos de la información de forma válida y con una alta precisión” (p.44). En otras palabras, es la estructura sistemática para la recolección, ordenamiento y análisis de la información, que permite la interpretación de los resultados en función de los problemas que se investigan.

En este capítulo se dará a conocer los pasos que se llevarán a cabo para el desarrollo del proyecto, los cuales están subdivididos de la siguiente manera.

5. Diseño de la investigación

Dado que el objetivo de estudio será el desarrollo del software y su incidencia en la mejora de la atención al cliente en la de las empresas en general que requieran de la implementación de este software, se recurrió a un diseño experimental

Este proyecto es de carácter descriptivo transversal, para conocer a detalle la forma en que este software mejoraría la calidad del servicio en el transporte del material (tubos) que se ofrece a los clientes el cual tiene como objetivo indagar la incidencia y los valores en los que se manifiestan una o más variables dentro del enfoque cuantitativo

5.1 Población

La población se define como “un conjunto de todos los elementos que estamos estudiando, acerca de los cuales intentamos sacar conclusiones” (Levin y Rubin 1996,p.20)

Como en este proyecto se analizan variables en un entorno cuantitativo para mejorar

el transporte de carga optimizando su volumen, la población a estudiar estará conformada por los elementos de carga en este caso tubos, la cantidad de tubos va a depender del tamaño disponible en volumen de carga, de la misma manera que del pedido del cliente.

5.2 Muestra

La muestra es definida por Fortín (1999) como “un subconjunto de una población o grupo de sujetos que conforman parte de una misma población” (p.160) en este mismo sentido, Pineda Alvarado (1994) señala que es “un subconjunto de la población en que se llevará a cabo la investigación con el fin posterior de generalizar los hallazgos del todo” (p.108)

En este trabajo se utiliza el muestreo no probabilístico, ya que se fijan cuotas que consisten en un número determinado de tubos que va a depender del pedido del cliente este número de tubos de diferentes medidas será un dato conocido para posteriormente hacer el análisis mediante el software que será el que determine su cantidad de tubos que serán transportados para su entrega.

5.2.1 Técnicas de recolección de datos

Falcon y Herrera (2005) se refieren a la técnica de recolección de datos como “el procedimiento o forma particular de obtener datos o información (...) la aplicación de una técnica conduce a la aplicación de información la cual debe ser resguardada mediante un instrumento de recolección de datos” (p.12)

La técnica de recolección de datos que se utilizará en la presente investigación será la información suministrada por búsquedas en portales de navegación de internet (imágenes y

datos numéricos), datos relevantes como tiempo y costos se toman de personas que por su experiencia transportando carga de este tipo nos brindaron esta información

Instrumento de recolección de datos

Un instrumento de recolección de datos es un principio:

Cualquier recurso de que pueda valerse el investigador para acercarse a los fenómenos y extraer de ellos la información. De este modo el instrumento sintetiza en toda la labor de la investigación, resume los aportes del marco teórico al seleccionar datos que correspondan a los indicadores y por lo tanto a las variables o conceptos utilizados (Sabino, 1992, p.83).

El instrumento de recolección de datos es la personal de la empresa que antes de implementar el software realizaba esta labor a mano lo que lo hace improductivo, esta persona nos suministrará toda la información necesaria como las diferentes medidas de los tubos fabricados, los volúmenes disponibles para el transporte entre otras cosas para llevar a cabo el desarrollo de este proyecto.

Para la recolección de datos se tendrán en cuenta las diferentes variables del proceso como lo son:

1. diferentes medidas de los tubos:
 - Ancho
 - Largo
 - Espesor
2. material de tubos, para analizar las cargas que actúan entre ellos al momento de su transporte.

- | 3. volumen de transporte de almacenamiento, donde se transportan los tubos.

5.3 Etapas

En este apartado se llevará el paso a paso con el cual se realizarán las diferentes actividades planteadas en los objetivos específicos.

5.3.1 Etapa 1

- Revisión de papers, catálogos, revistas, artículos, libros y cualquier tipo de material bibliográfico a fin de recopilar información que nos ayude a tener claridad de ideas para aplicar mejoras a nuestro proyecto.
- Selección de la información más reciente y relevante para el desarrollo del proyecto.

5.3.2 Etapa 2

- Especificar las variables, se analiza los datos proporcionados por los empleados de la empresa, variables que actúan en el diseño de la aplicación tales como, longitudes de los tubos y diámetros de los mismos.
- Se definen parámetros a implementar en la estructura del software. Teniendo en cuenta que la aplicación se desarrollará en Python, un lenguaje de programación que permite desarrollar aplicaciones con interfaz y análisis de datos.

5.3.3 Etapa 3

- Estipulado los valores de medidas de los diferentes tubos, se procede a realizar la programación del software teniendo en cuenta que estos valores son variables y van a depender del pedido de los clientes.

- se procederá a simular la estructura del código escrito.

5.3.4 Etapa 4

- Ajuste en la programación
- Verificación del programa
- Puesta en marcha con ajuste en el programa

5.3.5 Etapa 5

- Registro de datos.
- Análisis, conclusiones y recomendaciones.

5.4 Presupuesto

Tabla 1. Presupuesto del proyecto

Descripción	Unidad	Cantidad	Valor unitario	Valor parcial
Recursos Humanos				
Director de proyecto	Hora	40	\$25.000	\$1.000.000
Estudiantes	Hora	160	\$12.000	\$1.920.000
Subtotal				\$2.920.000
Software				
Office 365	Año	1	\$449.999	\$449.999
Pycharm (profesional)	Año	1	\$480.000	\$480.000
Adobe	Año	1	\$189.000	\$189.000

Subtotal				\$1.118.999
Equipos				
Computador (asus)	Uno	2	\$42.000.000	\$12.600.000
Internet	Mes	2	\$80.000	\$240.000
Subtotal				\$12.840.000
Otros				
Transporte	Mes	2	\$100.000	\$200.000
Libros	Uno	3	\$180.000	\$360.000
Gel desinfectante	Uno	1	\$20.000	\$20.000
Tapabocas N95	Uno	10	\$20.000	\$200.000
Subtotal				\$780.000
Total				\$17.658.999

6. Diseño conceptual

El diseño conceptual comienza con la teoría que va a ser ilustrada por la aplicación y por el dispositivo móvil o de escritorio. El diseño conceptual de la aplicación consistirá de una interfaz de usuario donde se pueda acceder a esta de manera amigable y confiable.

Esta de la mano con el principio de investigación de operaciones, donde posterior a la toma de datos e investigación del problema, se propone el modelo de diseño conceptual que se llevará a experimentos donde se probará la hipótesis de solución del problema y en caso de ser necesario, modificarla y eventualmente verificarla.

La hipótesis plantea llevar a la automatización por medio de software la lógica que el operario utiliza para el almacenamiento de tubos de diferentes diámetros, teniendo en cuenta la calidad de los tubos evitando fallas o que pierdan sus propiedades mientras ocurre su transporte, todo a través de filtros de lógica de programación donde siempre se tendrá en cuenta el estado de los tubos para el caso de que su distribución atente con las propiedades del material tomarlo como una distribución no óptima.

La comprobación de este modelo será que cumpla con una correcta distribución igual o mejor que la hecha por el operario y que las propiedades del material no se vean afectadas por la misma distribución de estos y su propio peso.

Figura 2. Interfaz de usuario

volume optimizer software

DATOS DE ENTRADA

Ingrese aqui las características de los tubos

Material Referencia

Diametro nominal RDE

Presentación

Longitud Cantidad

Listado de tubos

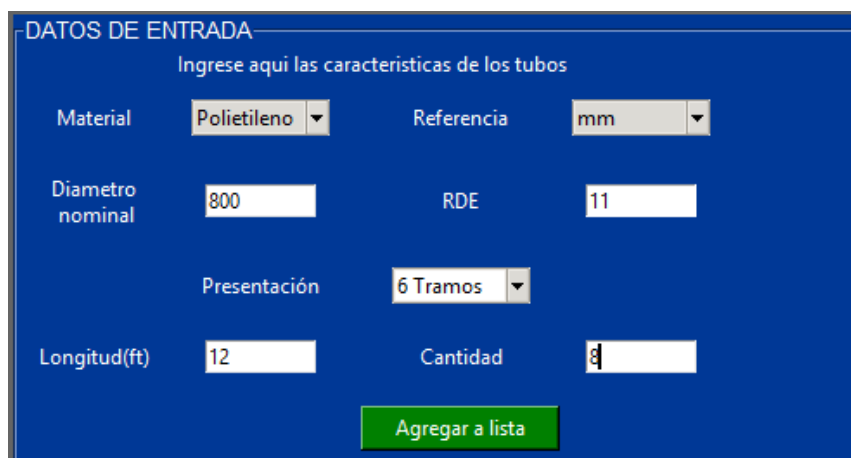
Cantidad	Longitud	Diametro externo	Diametro interno	Espesor	Delete
----------	----------	------------------	------------------	---------	--------

Tipo Containers

6.1 Descripción diseño conceptual

El diseño conceptual presenta dos grandes secciones, una de ingreso de datos y otra para el análisis de los resultados en base a los datos tomados por parte del usuario.

Figura 3. Registro de datos



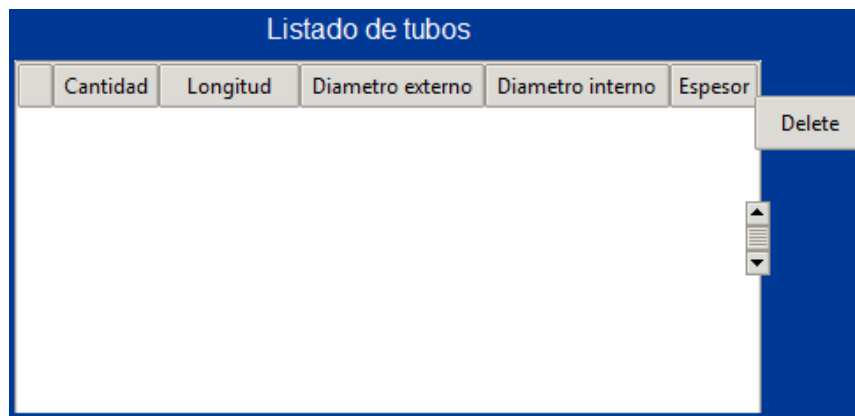
DATOS DE ENTRADA

Ingrese aquí las características de los tubos

Material	Polietileno	Referencia	mm
Diametro nominal	800	RDE	11
Presentación	6 Tramos		
Longitud(ft)	12	Cantidad	8

Agregar a lista

Esta sección está dividida a su vez en dos partes, la parte superior donde se ingresan los datos de cada tubo con sus características y donde luego de agregarse se ingresa a la parte inferior que es una tabla donde se ven todos los tubos a los que se desea hacer el análisis.

Figura 4. Listado de tubos

Cantidad	Longitud	Diametro externo	Diametro interno	Espesor	Delete
----------	----------	------------------	------------------	---------	--------

En el listado de tubos se tendrán botones para acciones como editar el tubo ingresado, en caso de haber digitado mal algún dato y la opción de eliminar un elemento de la lista, todo para cada elemento de la lista, dando control dinámico de la lista.

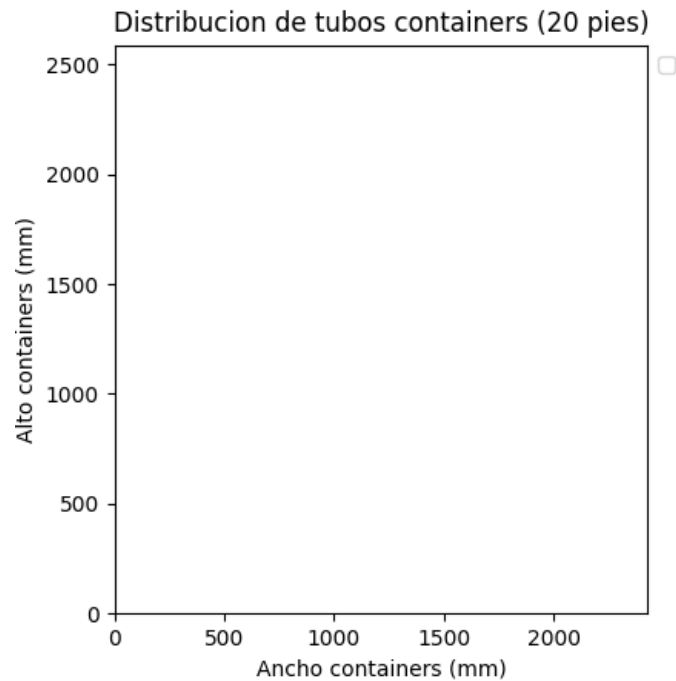
Figura 5. Listado de tubos

Cantidad	Longitud	Diametro externo	Diametro interno	Espesor	Delete
20	11	315.95	255.75	30.1	
23	3	114.3	87.38	13.46	
12	2	200.6	162.2	19.2	
6	4	168.3	135.89	16.205	

Luego de la toma de datos al hacer el análisis se tendrá la vista de los resultados, donde se podrá ver la distribución óptima de los tubos en la parte superior, junto con su leyenda

respectiva para identificar cada tipo de tubo.

Figura 6. Volumen de container



Por último en la parte inferior se tendrá el análisis de los tubos inferiores de mayor diámetro que serán los que soportan las mayores cargas, en esta parte se mostrarán gráficas que permitan visualizar de forma óptima que tan propenso es el material a fallar o a tener deformaciones que puedan afectar el propósito de los tubos para el cual fueron diseñados.

Figura 7. Diseño análisis de tubos (expectativa)

Análisis

7. Análisis de resultados

Los ensayos del uso de la aplicación se realizaron en una empresa dedicada al transporte de materiales, donde se obtuvo los datos necesarios para el cálculo de volumen y la distribución de la carga en el mismo, los tiempos se obtuvieron de forma manual y con la aplicación, fue necesario el punto de vista de varios transportadores quienes se dedican a diario a dicha industria, ellos con su experiencia en el transporte y la logística dicen que el tiempo y los resultados de esta aplicación son verdaderamente significantes.

Para este ensayo se tomó como referencia contenedor de 20 pies como media de transporte, cabe aclarar que todas sus medidas ya están establecidas dentro de la aplicación.

A continuación se observan sus medidas

Figura 8. Medidas container 20 pies

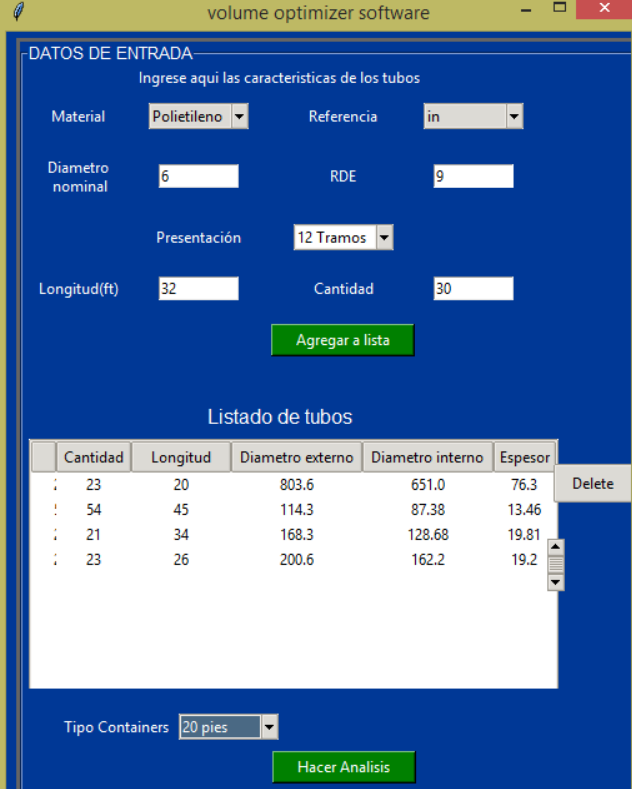
Capacidad	Dimensiones	Techo
Volumen 31.1 m ³ Máxima carga 22,000 kg Tara 2,250 kg Peso Bruto máximo 30,480 kg	Exterior Largo 6,06 m. Ancho 2,43 m. Alto 2,59 m. Interior Largo 5,89 m. Ancho 2,35 m. Alto 2,38 m. Puerta Ancho 2,434 m. Alto 2,27 mm,	Anchura 2,340 mm Altura 2,280 mm

Nota. Adaptado de <https://roscontainer.es/tipos-de-contenedores-maritimos/>

Debido a que la aplicación cuanta con materiales ya definidos tomados de diferentes catálogos era complejo tener una aproximación exacta con los diferentes materiales que ellos transportaban, sin embargo la estimación tiene un margen de error comparativo muy bajo.

A continuación se presenta la descripción de la interfaz gráfica de la aplicación donde se irán viendo reflejados cada uno de los datos ingresados por el usuario, así como también el análisis de optimización de volumen y el análisis de cargas para cada tipo de tubo respectivamente, según sea ingresado por el usuario.

Figura 9. Primer ensayo

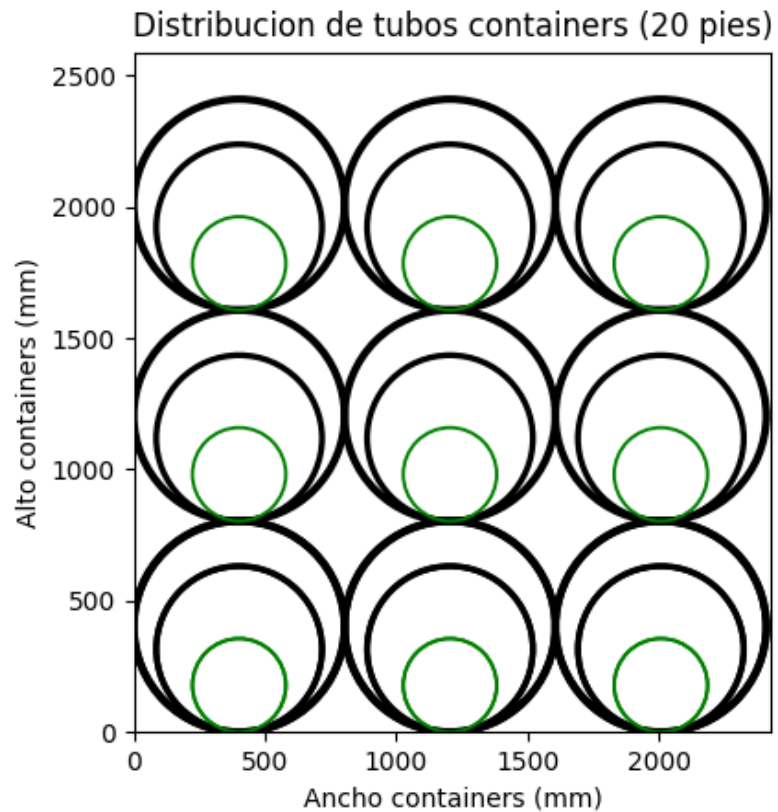


The screenshot shows the 'volume optimizer software' interface. The main window has a blue background and contains the following elements:

- DATOS DE ENTRADA:** A section for entering pipe characteristics with the following fields:
 - Material: Polietileno (dropdown)
 - Referencia: in (dropdown)
 - Diametro nominal: 6 (text input)
 - RDE: 9 (text input)
 - Presentación: 12 Tramos (dropdown)
 - Longitud(ft): 32 (text input)
 - Cantidad: 30 (text input)
- Agregar a lista:** A green button to add the current configuration to the list.
- Listado de tubos:** A table displaying the results of the optimization process.

	Cantidad	Longitud	Diametro externo	Diametro interno	Espesor	
:	23	20	803.6	651.0	76.3	Delete
:	54	45	114.3	87.38	13.46	
:	21	34	168.3	128.68	19.81	
:	23	26	200.6	162.2	19.2	
- Tipo Containers:** 20 pies (dropdown)
- Hacer Analisis:** A green button to perform the analysis.

En este primer ensayo se hizo el cálculo por medio de la aplicación donde se tardó 2 minutos en digitar la información y menos de 30 segundos en arrojar el análisis de la distribución de tubos como se observa en la siguiente imagen.

Figura 10. Muestra de tubos dentro del container

El acoplamiento de los tubos con el container es tan exacto que se aprovecha cada parte del mismo, cabe aclarar que dentro de la programación estos datos están ya establecidos, como también el volumen de los container, esto con referencia una investigación previa que se realizó para determinar las medidas de dichos elementos de carga.

En este ensayo utilizando la manera normal que se usa en las compañías, se tardó aproximadamente en realizar el cálculo 30 minutos, además el cálculo se hace a prueba y error dando como resultado una pérdida de tiempo innecesaria, Se comparó el tiempo realizado con la aplicación y el tiempo realizado manual quedando de la siguiente manera.

Usando la aplicación con un tiempo aproximado de 6 minutos

Forma manual un tiempo aproximado de 35 minutos

Costo de operación

Para el análisis de costo de operación se calculó de la siguiente manera

Partiendo del salario básico de un obrero el día de 8 horas de trabajo tiene un coste de \$40.000 pesos colombianos

$$C_t = t_g * C_h \quad (1)$$

Siendo

C_t El costo total gastado

t_g El tiempo gastado

C_h El costo por hora

Para el cálculo de costo por hora utilizamos la siguiente ecuación

$$C_h = \frac{V_d}{H_d} \quad (2)$$

$$C_h = \frac{40000}{8 \text{ horas}} = 5000 \text{ pesos por hora trabajada}$$

Calculando el costo de hora queda de la siguiente manera el costo total

Cuando se usa la aplicación

$$t_g = 0.1 \text{ horas} = 6 \text{ minutos}$$

$$C_t = 0.1 \text{ horas} * 5000 = 500 \text{ pesos la hora}$$

Ahora se hace el cálculo del costo total, realizando la labor de forma manual

$$t_g = 0.583 \text{ horas} = 35 \text{ minutos}$$

$$C_t = 0.583 \text{ horas} * 5000 = 2.917 \text{ pesos la hora}$$

Comparando estos resultados se observa una gran diferencia en las dos formas ahorrando \$2.400 pesos aproximadamente en cada cargue para despacho, en los datos recogidos por los operarios se calculó que en promedio en el día se realizan 40 cargues para transporte por todo el país, calculando el costo realizado de estos 40 cargues de ambas formas tanto con la aplicación como manual se obtuvo lo siguiente, utilizando la siguiente ecuación para realizar el cálculo.

Usando la aplicación

$$C_{t(cargues)} = (\# \text{ Cargadas realizadas por día}) * (\text{costo de carga}) \quad (3)$$

$$C_{t(cargues)} = 40 * 500 = 20.000 \text{ pesos por día}$$

Si se usa la aplicación en el día obtenemos un costo de \$20.000 pesos en todas las cargas realizadas.

Realizando el mismo calculo pero uso manual de carga, se obtiene lo siguiente.

$$C_{t(cargues)} = 40 * 2.917 = 116.680 \text{ pesos por día}$$

Si lo hacemos de forma manual todas las cargas nos da un costo de \$116.680 pesos por día.

Si comparamos los resultados obtenidos usando la aplicación y los resultados sin usar la misma,

Tendremos una diferencia de \$96.680 pesos que se estarían ahorrando con el uso de esta herramienta.

7.1 Análisis de carga

Para el análisis de esfuerzos para este ensayo se escogieron 2 materiales el polietileno y el polipropileno, en la siguiente tabla se verán las propiedades de cada uno de ellos así como también la curva de esfuerzo deformación.

Figura 11. Propiedades del polietileno

Propiedad	Polietileno de baja densidad (LDPE)	Polietileno de baja densidad lineal (LLDPE)	Polietileno de alta densidad (HDPE)
Densidad (g/cm ³)	0,92-0,93	0,822-0,926	0,95-0,96
Resistencia a la tensión, MPa	6,2-17,2	12,4-20,0	20,0-37,2
Alargamiento, %	550-600	600-800	20-120
Cristalinidad, %	65		95

Nota. Esta figura muestra las propiedades del polietileno para respectivos cálculos tanto de forma manual como con el software, adaptado de

https://www.upv.es/materiales/Fcm/Fcm15/Imagenes/Tabla15_2.jpg

Teniendo en cuenta la ficha técnica del polietileno, para ensayo obtenemos el esfuerzo de ruptura para analizar la gráfica. A continuación se observa la tabla de la ficha técnica de la norma ASTM F714

Figura 12. Ficha técnica de la norma ASTM F714

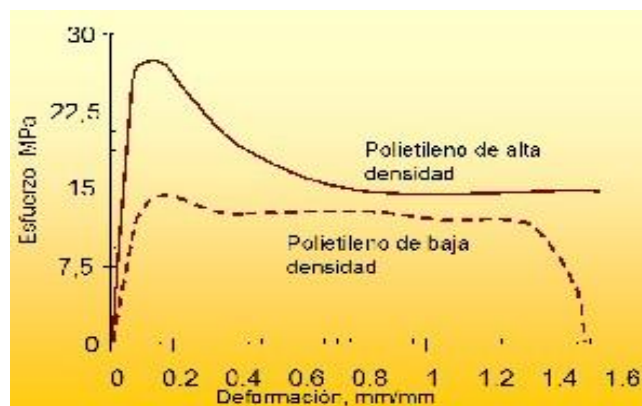
PROPIEDADES	TUBO PE100	UNIDAD
Densidad	$\geq 0,955$	g/cm ³
Negro de Humo	2,0 – 3,0	%
Índice de fluidez (190°C/2,16kg)	$\leq 0,15$	g/10 min %
Tiempo de inducción a la oxidación OIT	>20	min
Resistencia a la tensión aparente al anillo	>20 PE100(PE4710)	MPa
Presión de rotura RDE 7	$\geq 6,67$	MPa
Presión de rotura RDE 9	$\geq 5,00$	MPa
Presión de rotura RDE 11	$\geq 4,00$	MPa
Presión de rotura RDE 13,5	$\geq 3,20$	MPa
Presión de rotura RDE 17	$\geq 2,50$	MPa

Nota. Esta figura muestra las propiedades del polietileno

Teniendo en cuenta las características del material de ensayo se debe conocer también el comportamiento de la gráfica esfuerzo de formación para el polietileno, a continuación se observa la gráfica generalizada para los termoplásticos.

Figura 13. Grafica esfuerzo deformación

Nota. En esta figura se observa la curva esfuerzo deformación de los termoplásticos adaptado de Curvas típica tensión-deformación de materiales termoplásticos.

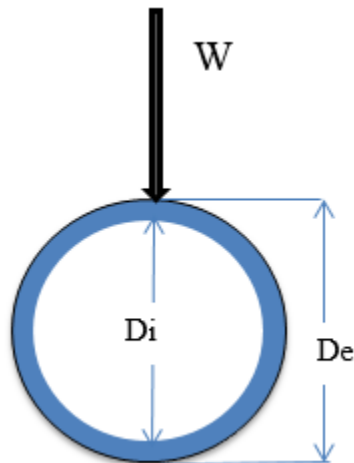
Figura 14. Esfuerzo deformación del polietileno

Nota. Grafica esfuerzo deformación del polietileno, adaptado de Curvas típica tensión-deformación de materiales termoplásticos.

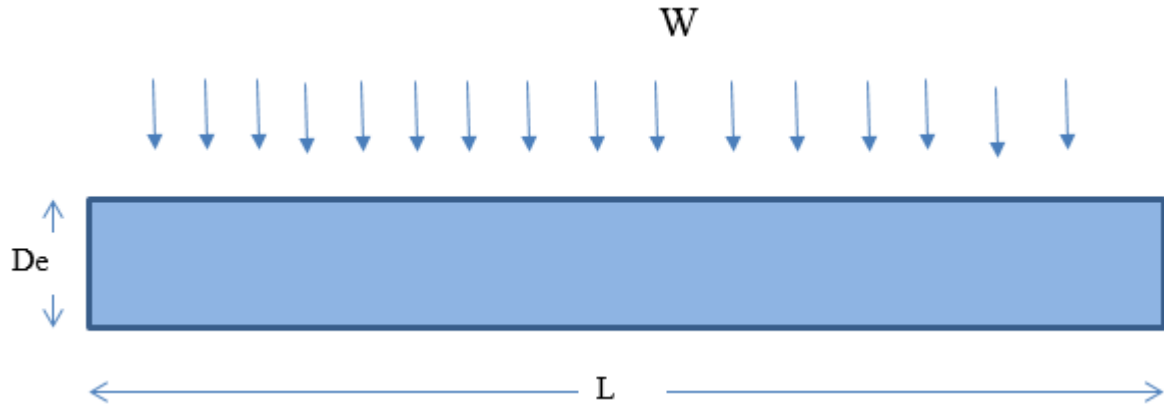
Para este ensayo los 2 materiales, el polietileno y el polipropileno en la aplicación, estos valores como su fórmula están programadas para que se genere automáticamente el análisis, también cabe resaltar que los tubos con mayor criticidad de fractura son los inferiores debido a que soportan toda la carga de los demás tubos.

A continuación se hace un diagrama donde se observan las cargas que interactúan en los tubos más críticos en este caso los inferiores.

Figura 15. Vista frontal del tubo (DCL)



La carga ejercida sobre la superficie del tubo está distribuida uniformemente a lo largo del mismo, como se observa a continuación.

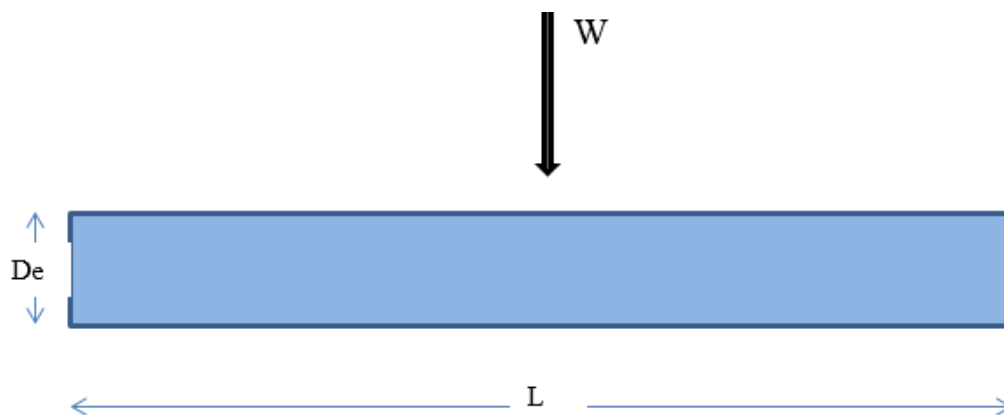
Figura 16. Distribución de fuerzas vista lateral

Siendo,

De = diámetro exterior

L = longitud

Como la carga ejercida es uniforme se puede unificar por una sola en el centro de gravedad de él tuvo, quedando de la siguiente manera.

Figura 17. Diagrama de fuerza unificada

Para calcular la resistencia de los tubos inferiores analizando su carga estática debemos primero calcular la masa de cada tubo, cabe aclarar que para calcular la masa debemos saber el material y sus propiedades como también su longitud, para ello utilizaremos la siguiente formula.

$$D = \frac{m}{v} \quad (4)$$

Siendo:

D = densidad del material escogido

m = masa del material escogido

v = volumen del material escogido

Una vez calculada la masa de cada tubo que esté por encima se procede a sumar la cantidad de tubos que tiene para calcular la carga total, quedando de la siguiente manera para este ensayo.

Primero calculamos el volumen del tubo con la siguiente formula,

$$v = \frac{\pi * d^2 * L}{4} \quad (5)$$

Siendo,

v = volumen del cilindro

d = diámetro del cilindro

L = longitud del cilindro

El tubo con mayor criticidad en este ensayo tiene las siguientes medidas

Diámetro externo (D_e)= 0,803 m

Diámetro interno (D_i) = 0,651 m

Longitud (L) = 20 ft= 6,096 m

Para calcular el volumen total es necesario calcular el volumen interno y externo para luego restar el interno quedando así el volumen del cilindro.

Volumen exterior

$$v = \frac{\pi * D_e^2 * L}{4} = \frac{\pi(0,803)^2 * 6,096}{4} = 3,07 \text{ m}^3$$

Volumen interior

$$v = \frac{\pi * D_i^2 * L}{4} = \frac{\pi(0,651)^2 * 6,096}{4} = 2,02 \text{ m}^3$$

Calculamos el volumen total quedando de la siguiente forma

$$V_{total} = \text{volumen exterior} - \text{volumen interior} \quad (6)$$

$$V_{total} = 3,07 - 2,02 = 1,03 \text{ m}^3$$

Con la densidad, calculamos la masa de cada uno de los usando la formula (4) quedando así de la siguiente forma.

$$d = 0,93 \frac{g}{cm^3} = 930 \frac{kg}{m^3}$$

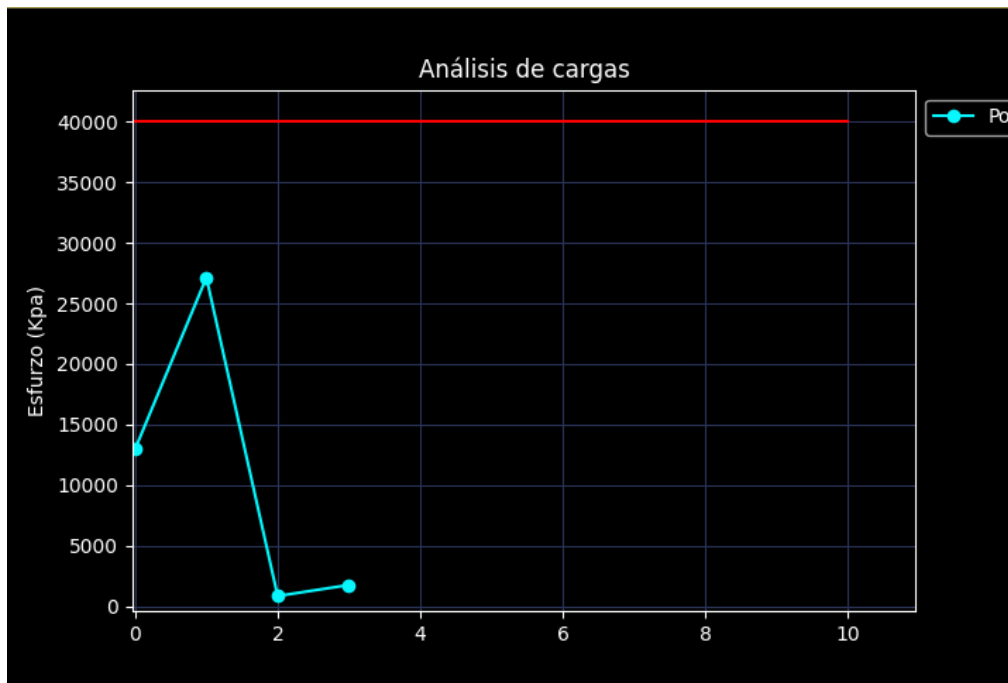
$$m = d * V_{total}$$

$$m = 930 * 1,03 = 967,2 \text{ Kg}$$

La masa de cada uno de los tubos es de 967 kg aproximadamente, pero se debe sumar también los tubos interiores, dando como resultado una masa total de 1.200 kg aproximadamente, se además de ello se debe sumar la cantidad de tubos que se tienen encima del inferior, para calcular así su carga total, para este primer ensayo son 2 tubos se debe multiplicar por 2 dando como carga total una masa aproximada de 2.400 Kg.

A continuación se observa la gráfica usando la aplicación.

Figura 18. Gráfica de análisis de cargas del primer ensayo



Como se puede observar para esa configuración de tubos, tenemos que el tubo más crítico es el de 800 mm de diámetro nominal con un RDE de 11 material polietileno, sabiendo que la ruptura para este material se encuentra por encima de los 40000 Kpa y que su carga máxima llega hasta alrededor de los 27000 Kpa muy por debajo del punto crítico.

8. Conclusiones

Para el desarrollo del presente trabajo se utilizó el lenguaje de programación Python, un lenguaje relativamente sencillo de comprender muy utilizado para el desarrollo de estas aplicaciones, la aplicación desarrollada en este proyecto será de mucha ayuda para que las empresas dedicadas a la industria del transporte de tubería se vean beneficiadas económicamente, usando esta herramienta informática hace que ahorre mucho tiempo al momento de cargar, además también optimiza todo el volumen del container disponible de tal manera que se aproveche al máximo, las empresas dedicadas a esta industria se ven afectadas por el tiempo que se demora cargando ya que en un tiempo considerado muerto, con la presente aplicación también reducimos considerablemente este tiempo, cuando desarrollamos esta aplicación también pensamos en los clientes, dado que es el tema de valor para estas industrias, es por eso que se incluyó un análisis de cargas garantizando que la carga no sufra daños al momento de su cargue o de transporte. El rendimiento y la eficiencia de esta aplicación permiten que el cliente deposite su confianza optando por el servicio. La herramienta utilizada es muy sencilla de implementar a pesar de que no se programó con seguridad de ingreso ni que las interfaces de este lenguaje tenga diseños visiblemente impresionantes, se tiene la certeza de que la aplicación creada fluirá con rapidez, además no tiene restricciones para ninguna ni su sistema operativo computadoras.

En un futuro se pretende hacer mejoras de acuerdo con la necesidades de cada compañía, dado que esta aplicación fue creada con valores aleatorios, se puede acoplar a datos establecidos según requerimientos de cada compañía que desee utilizarla, también se quiere implementar nuevas graficas como por ejemplo que permita visualizar un historial de órdenes que sirva para la

implementación de machine learnig que brinde más seguridad y confianza tanto en los clientes como en la compañía.

9. Recomendaciones

- Hacer ejercicios utilizando la aplicación y llevar un historial de las órdenes impartidas.
- Ya que el tema de visualización del análisis no es familiar para muchos, se recomienda documentarse para conocer mejor el tema
- Tomar referencias de antes y después del uso de la aplicación, como también tomar nota de las posibles mejoras o implantes a futuro para mejora de la misma
- Consultar varios puntos de vista tomados de diferentes compañías al respecto del uso del software y sus beneficios

Bibliografía

Cano, F. (septiembre, 2014).

<https://www.definicionabc.com/tecnologia/open-source.php> Definición de Open Source. Definición ABC. Desde

Carro, R. (2009). INVESTIGACION DE OPERACIONES EN ADMINISTRACION

(2009.aed.). Universidad Nacional de Mar del Plata.

<http://nulan.mdp.edu.ar/id/eprint/2180>

Herrera Velandia, D. (2019). Diseño e Implementación de una Herramienta Digital para La Generación de Informes de Balance de Material Line Pipe Ofertado a Ecopetrol SA Bajo el Contrato 3011938 por la Empresa TenarisTubo Caribe LTDA.

Scikit-Learn, herramienta básica para el Data Science en Python. (2019, 8 abril). Máster En Data Science. <https://www.master-data-scientist.com/scikit-learn-data-science/>.

2000 Como-se-elabora-un-proyecto-de-Investigacion-MIRIAM-BALESTRINI.pdf

https://www.academia.edu/35200583/2000_Como_se_Elabora_un_Proyecto_de_Investigacion_MIRIAN_BALESTRINI_pdf

Apéndices

Apéndice A. Codificación

```
from tkinter import*

from tkinter import ttk

from tkinter.ttk import Combobox

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.path as mpath

import matplotlib.patches as patches

import matplotlib.transforms as transforms

class ventanaprincipal(Frame):

    def __init__(self, master=None):

        super().__init__(master, width=1000, height=1000, bg="#003896")

        self.pack(expand=True, fill="both")

        self.master = master
```



```
self.frm_datos()

self.Configuracion_tubos()

def Configuracion_tubos(self):

    frm3=LabelFrame(self,text="Configuracion de tubos",fg="white",bg="#003896")

    frm3.place(x=800,y=5,width=500,height=300)

def frm_datos(self):

    style = ttk.Style()

    style.theme_use('clam')

    fframe1=LabelFrame(self,text="DATOS      DE      ENTRADA",font="arial
10",highlightbackground="blue",highlightthickness=3,fg="white",bg="#003896")

    fframe1.place(x=8,y=5,height=700,width=520)

    textContainer=Label(fframe1,text="Tipo Containers",fg="white",bg="#003896")

    textContainer.place(x=30,y=520)

    listado=["20 pies","40 pies"]
```

```
self.elegirContainer=Combobox(fframe1,values=listado,width=10,state="readonly",postcommand=self.sugerir)
```

```
self.elegirContainer.place(x=125,y=520)
```

```
def entradaDatos():
```

```
def Borrar():
```

```
DiametroExterno.delete(0,END)
```

```
RDEentrada.delete(0,END)
```

```
LongitudEntrada.delete(0,END)
```

```
CantidadEntrada.delete(0,END)
```

```
fx1=Frame(fframe1,bg="#003896")
```

```
fx1.place(x=0,y=0,width=500,height=500)
```

```
text=Label(fx1,bg="#003896",text="Ingrese aqui las características de los tubos",fg="white" )
```

```
text.grid(row=0,column=1,columnspan=2)
```

```
tex1=Label(fx1, text="Material",fg="white",bg="#003896")
```

```
tex1.grid(row=1,column=0,padx=10,pady=10)
```

```
text=Label(fx1,text="RDE",fg="white",bg="#003896")
```

```
text.grid(row=2,column=2)
```

```
text=Label(fx1,text="Referencia",fg="white",bg="#003896")
```

```
text.grid(row=1,column=2,padx=10,pady=10)
```

```
tex2=Label(fx1,bg="#003896", text="Diametro\n nominal",fg="white")
```

```
tex2.grid(row=2,column=0, padx=10,pady=10)
```

```
tex3_1=Label(fx1, text="Presentación",fg="white",bg="#003896",)
```

```
tex3_1.grid(row=3,column=1,columnspan=1,padx=10,pady=10)
```

```
presenOpcion=Combobox(fx1,width=10,values=["Tramos","Rollos"])
```

```
presenOpcion.grid(row=3,column=2)
```

```
tex3=Label(fx1, text="Longitud(ft)",fg="white",bg="#003896",)
```

```
tex3.grid(row=4,column=0, padx=10,pady=10)
```

```
tex3=Label(fx1, text="Cantidad",fg="white",bg="#003896",)
```

```
tex3.grid(row=4,column=2, padx=10,pady=10)
```

```
opc1 conbobox=
```

```
Combobox(fx1,width=10,values=["Polietileno", "Polipropileno", "Poliamida"], state="readonly")
```

```
opc1 conbobox.grid(row=1, column=1)
```

```
Refcombobox= Combobox(fx1,width=10,values=["mm","in"], state="readonly")
```

```
Refcombobox.grid(row=1, column=3)
```

```
DiametroExterno= Entry(fx1,width=10,bg="white")
```

```
DiametroExterno.grid(row=2, column=1)
```

```
RDEentrada=Entry(fx1,width=10)
```

```
RDEentrada.grid(row=2,column=3)
```

```
LongitudEntrada=Entry(fx1,width=10,bg="white")
```

```
LongitudEntrada.grid(row=4,column=1)
```

```
CantidadEntrada=Entry(fx1,width=10,bg="white")
```

```
CantidadEntrada.grid(row=4,column=3)
```

```
text=Label(fx1,font="Arial",bg="#003896",text="Listado de tubos",fg="white")
```

```
text.place(x=145,y=270)
```

```
tabla=ttk.Treeview(fx1,columns=("Cantidad","Longitud","Diametro  
externo","Diametro interno","espesor"))
```

```
tabla.place(x=5,y=300)
```

```
verscrbar = ttk.Scrollbar(fx1,
```

```
orient="vertical",
```

```
command = tabla.yview)
```

```
verscrbar.place(x=420,y=380)
```

```
tabla.configure(xscrollcommand = verscrbar.set)
```

```
tabla.column("#0", width=0)
```

```
tabla.column("Cantidad",width=60,anchor=CENTER)
```

```
tabla.column Longitud",width=80, anchor=CENTER)
```

```
tabla.column ("Diametro externo",width=105, anchor=CENTER)
```

```
tabla.column ("Diametro interno",width=105, anchor=CENTER)
```

```
tabla.column ("espesor",width=70, anchor=CENTER)
```

```
table.heading("#0", text="", anchor="w")
```

```
table.heading ("Cantidad",text="Cantidad")
```

```
table.heading ("Longitud",text="Longitud", anchor=CENTER)
```

```
table.heading ("Diametro externo",text="Diametro externo", anchor=CENTER)
```

```
table.heading ("Diametro interno",text="Diametro interno", anchor=CENTER)
```

```
table.heading ("espesor",text="Espesor", anchor=CENTER)
```

```
def get_diametros(diametros):
```

```
    groups = { }
```

```
    num = 1
```

```
    for diametro in diametros:
```

```
        if not f"g{num}" in groups:
```

```
            groups[f"g{ num}"] = [diametro]
```

```
            continue
```

```
list_groups = list(groups.values())

for id, group in enumerate(list_groups):

    currents = sorted(group, key=lambda x: x["e"], reverse=True)

    valid=True

    for idx, current in enumerate(currents):

        if diametro["e"] == current["i"] or diametro["e"] == current["e"] :

            valid=False

            break

    if valid:

        groups[f"g{ num}"].append(diametro)

        num = 1

        break

    else:

        num+=1

        if id == len(list_groups) -1:

            groups[f"g{ num}"] = [diametro]

            num = 1

return list(groups.values())

lista=[]
```

```
list_tubos={ }
```

```
def Agregar():
```

```
    if Refcombobox.get()=="mm":
```

```
        Diametro=float(DiametroExterno.get())
```

```
        rde=float(RDEentrada.get())
```

```
        Longitud=(LongitudEntrada.get())
```

```
        Cantidad=(CantidadEntrada.get())
```

```
        data = {
```

```
            'Diametro_Nominal': [
```

```
                16,20,25,32,40,50,63,75,90,110,160,200,250,315,400,
```

```
                16,20,25,32,40,50,63,75,90,110,160,200,250,315,355,400,450,500,630,
```

```
                20,25,32,40,50,63,75,90,110,125,160,180,200,225,250,280,315,355,400,450,500,630,800,
```

```
                25,32,40,50,63,75,90,110,125,160,180,200,225,250,280,315,355,400,450,500,560,630,710,800,
```

```
                32,40,50,63,75,90,110,125,160,180,200,225,250,280,315,355,400,450,500,560,630,710,800,
```

```
                40,50,63,75,90,110,125,160,180,200,225,250,280,315,355,400,450,500,560,630,710,800,
```


50,63,75,90,110,160,180,200,250,315,355,400,450,500,560,630,710,800,],

'Diametro_externo':[

16.15,20.15,25.15,32.15,40.2,50.2,63.2,75.25,90.3,110.35,160.5,200.6,250.75,315.95,401.2,

16.15,20.15,25.15,32.15,40.2,50.2,63.2,75.25,90.3,110.35,160.5,200.6,250.75,315.95,356.1,401.

2,451.35,501.5,631.9,

20.15,25.15,32.15,40.20,50.20,63.20,75.25,90.30,110.35,125.40,160.50,180.55,200.60,225.70,25

0.75,280.85,315.95,356.10,401.20,451.35,501.50,631.90,803.60,

25.15,32.15,40.2,50.2,63.2,75.25,90.3,110.35,125.4,160.5,180.55,200.6,225.7,250.75,280.85,315

.95,356.1,401.2,451.35,501.5,561.7,631.9,713.2,803.6,

32.15,40.2,50.2,63.2,75.25,90.3,110.35,125.4,160.5,180.55,200.6,225.7,250.75,280.85,315.95,35

6.1,401.2,451.35,501.5,561.7,631.9,713.2,803.6,

40.2,50.2,63.2,75.25,90.3,110.35,125.4,160.5,180.55,200.6,225.7,250.75,280.85,315.95,356.1,40

1.2,451.35,501.5,561.7,631.9,713.2,803.6,

50.2,63.2,75.25,90.3,110.35,160.5,180.55,200.6,250.75,315.95,356.1,401.2,451.35,501.5,561.7,6
31.9,713.2,803.6],

'RDE': [

7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,7.4,

9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,

11,

13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13.6,13
.6,13.6,13.6,13.6,13.6,

17,

21,21,21,21,21,21,21,21,21,21,21,21,21,21,21,21,21,21,21,

26,26,26,26,26,26,26,26,26,26,26,26,26,26,26,26,26,26,26],

'Diametro_interno': [

11.15,13.75,17.65,22.75,28.5,35.6,45,53.45,64.3,78.45,114.4,142.9,178.75,225.25,286.2,

11.85,15.15,18.75,24.45,30.6,38.3,48.1,57.45,68.9,84.35,122.8,153.4,192.05,241.85,272.6,307.2,
345.55,384.2,484.1,

15.85,20.15,25.75,32.30,40.40,50.90,60.85,72.90,89.25,101.30,129.70,145.95,162.20,182.50,202.95,227.35,255.75,288.30,324.80,365.35,406.00,511.60,651.00,

20.85,26.95,33.7,42.3,53.2,63.35,76.1,93.15,105.9,135.6,152.45,169.6,190.7,211.95,237.45,267.05,301.1,339.3,381.65,424.1,475,534.5,603.4,680,

27.85,35,43.8,55.1,65.65,78.8,96.35,109.7,140.4,157.95,175.5,197.4,219.55,245.85,276.55,311.6,351.3,395.15,439,491.8,553.2,624.6,703.9,

35.9,45,56.8,67.55,81.1,99.05,112.7,144.2,162.35,180.3,202.9,225.65,252.55,284.35,320.5,360.9,406.05,451.2,505.5,568.8,641.9,723.4,

45.9,57.8,69.05,82.8,101.35,147.3,165.95,184.3,230.45,290.35,327.4,368.9,415.05,461.2,516.6,581.1,655.9,739.2],

'Espesor': [

2.5,3.2,3.75,4.7,5.85,7.3,9.1,10.9,13,15.95,23.05,28.85,36,45.35,57.5,

2.15,2.5,3.2,3.85,4.8,5.95,7.55,8.9,10.7,13,18.85,23.6,29.35,37.05,41.75,47,52.9,58.65,73.9,

2.15,2.50,3.20,3.95,4.90,6.15,7.20,8.70,10.55,12.05,15.40,17.30,19.20,21.60,23.90,26.75,30.10,3
3.90,38.20,43.00,47.75,60.15,76.30,

2.15,2.6,3.25,3.95,5,5.95,7.1,8.6,9.75,12.45,14.05,15.5,17.5,19.4,21.7,24.45,27.5,30.95,34.85,38.
7,43.35,48.7,54.9,61.8,

2.15,2.6,3.2,4.05,4.8,5.75,7,7.85,10.05,11.3,12.55,14.15,15.6,17.5,19.7,22.25,24.95,28.1,31.25,3
4.95,39.35,44.3,49.85,

2.15,2.6,3.2,3.85,4.6,5.65,6.35,8.15,9.1,10.15,11.4,12.55,14.15,15.8,17.8,20.15,22.65,25.15,28.1,
31.55,35.65,40.1,

2.15,2.7,3.1,3.75,4.5,6.6,7.3,8.15,10.15,12.8,14.35,16.15,18.15,20.15,22.55,25.4,28.65,32.2]

}

```
df=pd.DataFrame(data)
```

```
for i in range(len(df)):
```

```
    if df.iloc[i]["Diametro_Nominal"]==Diametro and df.iloc[i]["RDE"]==rde:
```

```
        resultdiametro=float(df.iloc[i]["Diametro_interno"])
```

```
        resultespesor=float(df.iloc[i]["Espesor"])
```

```
p=tabla.insert("",END,text=Cantidad,values=(Cantidad,Longitud,df.iloc[i]["Diametro_externo"],
resultdiametro,resultespesor))
```

```
    for column in tabla["columns"]:
```

```
        values = [tabla.set(item, column) for item in tabla.get_children()]
```

```
        list_tubos[column] = values
```

```
    Borrar()
```

```
print(list_tubos["Cantidad"])
```

```
r=tabla.set(p)
```

```
lista.append(r["Diametro externo"])
```

```
print(list_tubos)
```

```
if Refcombobox.get()=="in":
```

```
    Diametro=float(DiametroExterno.get())
```

```
    rde=float(RDEentrada.get())
```

```
    Longitud=(LongitudEntrada.get())
```

```
    Cantidad=(CantidadEntrada.get())
```

```
data = {  
  
    'Diametro_Nominal': [  
  
        4,6,8,10,12,14,  
  
        4,6,8,10,12,14,  
  
        4,6,8,10,12,14,  
  
        4,6,8,10,12,14,  
  
        4,6,8,10,12,14],  
  
    'Diametro_externo': [  
  
        114.3,168.3,219.1,273.1,323.8,355.6,  
  
        114.3,168.3,219.1,273.1,323.8,355.6,  
  
        114.3,168.3,219.1,273.1,323.8,355.6,  
  
        114.3,168.3,219.1,273.1,323.8,355.6,  
  
        114.3,168.3,219.1,273.1,323.8,355.6,],  
  
    'RDE': [  
  
        7.3,7.3,7.3,7.3,7.3,7.3,  
  
        9,9,9,9,9,  
  
        11,11,11,11,11,11,  
  
        13.5,13.5,13.5,13.5,13.5,13.5,  
  
        17,17,17,17,17,17,],  
  
}
```

```
'Diametro_interno': [  
81.14,119.41,155.46,193.79,229.74,252.31,  
87.38,128.68,167.52,208.83,247.5,271.81,  
92.28,135.89,176.89,220.51,261.41,287.05,  
96.36,141.86,184.68,230.22,273.01,299.77,  
100.03,147.29,191.79,239.07,283.41,311.23,],  
'Espesor': [  
16.58,24.445,31.82,39.655,47.03,51.645,  
13.46,19.81,25.79,32.135,38.15,41.895,  
11.01,16.205,21.105,26.295,31.195,34.275,  
8.97,13.22,17.21,21.44,25.395,27.915,  
7.135,10.505,13.655,17.015,20.195,22.185,  
]  
}  
  
df=pd.DataFrame(data)  
  
for i in range(len(df)):  
  
    if df.iloc[i]["Diametro_Nominal"]==Diametro and df.iloc[i]["RDE"]==rde:  
  
        resultdiametro=float(df.iloc[i]["Diametro_interno"])  
  
        resultespesor=float(df.iloc[i]["Espesor"])
```

```
p=tabla.insert("",END,text=Cantidad,values=(Cantidad,Longitud,df.iloc[i]["Diametro_externo"],
resultdiametro,resultespesor))
```

```
    for column in tabla["columns"]:
```

```
        values = [tabla.set(item, column) for item in tabla.get_children()]
```

```
    list_tubos[column] = values
```

```
    Borrar()
```

```
    print(list_tubos["Cantidad"][0])
```

```
    r=tabla.set(p)
```

```
    lista.append(r["Diametro externo"])
```

```
def grafic_carga():
```

```
    limit_x=10
```

```
    valor_critico=40000
```

```
    plt.style.use("dark_background")
```

```
    colors = [
```

```
        '#08F7FE', # teal/cyan
```



```
'#FE53BB', # pink

'#F5D300', # yellow

'#00ff41', # matrix green

]

suma=[]

for m in range(len(list_tubos["Cantidad"])):

    list_tubos["Cantidad"][m]=float(list_tubos["Cantidad"][m])

for m in range(len(list_tubos["Longitud"])):

    list_tubos["Longitud"][m]=float(list_tubos["Longitud"][m])

for m in range(len(list_tubos["Diametro externo"])):

    list_tubos["Diametro externo"][m]=float(list_tubos["Diametro externo"][m])

for m in range(len(list_tubos["Diametro interno"])):

    list_tubos["Diametro interno"][m]=float(list_tubos["Diametro interno"][m])

for i in list_tubos["Diametro externo"]:

    ti=float(i/1000)

    for j in list_tubos["Longitud"]:

        pr=float(j*0.3048)

        for k in list_tubos["Diametro interno"]:
```

```
lk=float(k/1000)

vi=(np.pi*(lk**2)*pr)/4

vi+=vi

ve=(np.pi*(ti**2)*pr)/4

ve+=ve

vt=vi+ve

vt+=vt

d=930

w=float(d*vt**2)

suma.append(w)

df = pd.DataFrame({'Polietileno': suma,

})

fig, ax = plt.subplots()

df.plot(marker='o', color=colors, ax=ax)

x=[0,limit_x]

y=[valor_critico,valor_critico]
```

```
ax.plot(x,y, color = "red",)

ax.grid(color='#2A3459')

ax.legend(loc='upper left', bbox_to_anchor=(1, 1))

plt.title("Análisis de cargas")

plt.ylabel("Esfuerzo (Kpa)")

plt.show()
```

```
def Analisis():
```

```
    if opc1 conbobox.get()=="polietileno":
```

```
        fig, ax = plt.subplots()
```

```
    if self.elegirContainer.get()=="40 pies":
```

```
        ancho=2352
```

```
        alto=2393
```

```
        prepared =[]
```

```
        for x in range(len(list_tubos["Diametro externo"])):
```

```
            for i in range(int(list_tubos["Cantidad"][x])):
```

```
prepared.append({  
    "e":float(list_tubos["Diametro externo"][x]),  
    "i":float(list_tubos["Diametro interno"][x]),  
    "spe":float(list_tubos["espesor"][x])  
})  
  
colors = {  
    "1": "red",  
    "2": "blue"  
}  
  
def get_color(spe):  
    if spe < 10:  
        return "blue"  
  
    if spe >= 10 and spe < 50:  
        return "red"  
  
    if spe >= 50 and spe < 100:  
        return "green"  
  
    return "black"
```

```
def get_linewidth(spe):  
  
    if spe < 10:  
  
        return "blue"  
  
    if spe >= 10 and spe < 15:  
  
        return "red"  
  
    if spe >= 15 and spe < 25:  
  
        return "green"  
  
    return "black"  
  
  
max_x = ancho  
  
max_y = alto  
  
diam_array = get_diametros(prepared)  
  
  
fig, ax = plt.subplots()  
  
off_xmax = 0  
  
off_ymax = 0  
  
off_x = 0  
  
off_y = 0
```

```
restantes = []

fila_max_r=0

for idx,diametros in enumerate(diam_array):

    d_max = max(diametro["e"] for diametro in diametros)

    r_max = d_max/2

    if idx == 0:

        y=r_max

        off_ymax = d_max

    if (off_x+d_max) > max_x:

        if (off_y+d_max) > max_y:

            print("Ha alcanzado el limite maximo de altura")

            restantes.append(diametros)

            continue

        print("nueva fila")

        fila_max_r=r_max

        y=off_y+r_max

        x=r_max

        off_x = d_max

        off_y+=d_max
```

```

        off_ymax+=d_max

    else:

        x=off_x+r_max

        off_x+=d_max

    diam_off = 0

    diametros_sort = sorted(diametros,key=lambda x: x["e"], reverse=True)

    for idx,diametro in enumerate(diametros_sort):

        radio = diametro["e"] / 2.0

        milimetros = diametro["spe"] # Grosor de línea en milímetros

        escala_milimetros_a_puntos = transforms.ScaledTranslation(0, 0,
fig.dpi_scale_trans)

        ancho_linea = milimetros / 25.4 # Convertir milímetros a pulgadas

        ancho_linea_puntos =
escala_milimetros_a_puntos.transform((ancho_linea, 0))[0]

        ty=y

        if idx ==0:

            ty=y+diam_off if fila_max_r==radio or fila_max_r==0 else
y+diam_off-fila_max_r+radio

```

```
else:

    ty=y-r_max+radio

    circulo = plt.Circle((x, ty), radio, edgecolor=get_color(diametro["e"])-
diametro["i"]), facecolor='none', linewidth=ancho_linea_puntos)

    ax.add_artist(circulo)

    diam_off+=diametro["spe"]

if off_x >= off_xmax:

    off_xmax = off_x

print(off_xmax,off_ymax)

circulo = plt.Circle((0, 0), 0, edgecolor="red",facecolor='none', label="8mm")

ax.add_artist(circulo)

circulo = plt.Circle((0, 0), 0, edgecolor="green",facecolor='none',
label="5mm")

ax.add_artist(circulo)

circulo = plt.Circle((0, 0), 0, edgecolor="blue",facecolor='none',
label="10mm")

ax.add_artist(circulo)

circulo = plt.Circle((0, 0), 0, edgecolor="pink",facecolor='none',
label="15mm")
```



```
ax.add_artist(circulo)

ax.set_aspect('equal', adjustable='box')

plt.xlim(0, ancho)

plt.ylim(0, alto)

ax.legend(loc='upper left', bbox_to_anchor=(1, 1))

plt.title("Distribucion de tubos containers (20 pies)")

plt.ylabel("Alto containers (mm)")

plt.xlabel("Ancho containers (mm)")

plt.show()

grafic_carga()

if self.elegirContainer.get()=="20 pies":

    ancho=2430

    alto=2590

    prepared =[]

    for x in range(len(list_tubos["Diametro externo"])):

        for i in range(int(list_tubos["Cantidad"][x])):

            prepared.append({
```

```
        "e":float(list_tubos["Diametro externo"][x]),  
        "i":float(list_tubos["Diametro interno"][x]),  
        "spe":float(list_tubos["espesor"][x])  
    })
```

```
colors = {  
    "1": "red",  
    "2": "blue"  
}
```

```
def get_color(spe):  
    if spe < 10:  
        return "blue"  
  
    if spe >= 10 and spe < 50:  
        return "red"  
  
    if spe >= 50 and spe < 100:  
        return "green"  
  
    return "black"
```

```
def get_linewidth(spe):  
    if spe < 10:  
        return "blue"  
  
    if spe >= 10 and spe < 15:  
        return "red"  
  
    if spe >= 15 and spe < 25:  
        return "green"  
  
    return "black"  
  
max_x = ancho  
  
max_y = alto  
  
diam_array = get_diametros(prepared)  
  
fig, ax = plt.subplots()  
  
off_xmax = 0  
  
off_ymax = 0  
  
off_x = 0  
  
off_y = 0  
  
restantes = []
```

```
fila_max_r=0

for idx,diametros in enumerate(diam_array):

    d_max = max(diametro["e"] for diametro in diametros)

    r_max = d_max/2

    if idx == 0:

        y=r_max

        off_ymax = d_max

    if (off_x+d_max) > max_x:

        if (off_y+d_max) > max_y:

            print("Ha alcanzado el limite maximo de altura")

            restantes.append(diametros)

            continue

        print("nueva fila")

        fila_max_r=r_max

        y=off_y+r_max

        x=r_max

        off_x = d_max

        off_y+=d_max

        off_ymax+=d_max
```

```

else:

    x=off_x+r_max

    off_x+=d_max

diam_off = 0

diametros_sort = sorted(diametros,key=lambda x: x["e"], reverse=True)

for idx,diametro in enumerate(diametros_sort):

    radio = diametro["e"] / 2.0

    milimetros = diametro["spe"] # Grosor de línea en milímetros

    escala_milimetros_a_puntos = transforms.ScaledTranslation(0, 0,
fig.dpi_scale_trans)

    ancho_linea = milimetros / 25.4 # Convertir milímetros a pulgadas

    ancho_linea_puntos =
escala_milimetros_a_puntos.transform((ancho_linea, 0))[0]

    ty=y

    if idx ==0:

        ty=y+diam_off if fila_max_r==radio or fila_max_r==0 else
y+diam_off-fila_max_r+radio

    else:

```

```
        ty=y-r_max+radio

        circulo = plt.Circle((x, ty), radio, edgecolor=get_color(diametro["e"])-
diametro["i"]), facecolor='none', linewidth=ancho_linea_puntos)

        ax.add_artist(circulo)

        diam_off+=diametro["spe"]

    if off_x >= off_xmax:

        off_xmax = off_x

    print(off_xmax,off_ymax)

    circulo = plt.Circle((0, 0), 0, edgecolor="red",facecolor='none', label="8mm")

    ax.add_artist(circulo)

    circulo = plt.Circle((0, 0), 0, edgecolor="green",facecolor='none',
label="5mm")

    ax.add_artist(circulo)

    circulo = plt.Circle((0, 0), 0, edgecolor="blue",facecolor='none',
label="10mm")

    ax.add_artist(circulo)

    circulo = plt.Circle((0, 0), 0, edgecolor="pink",facecolor='none',
label="15mm")

    ax.add_artist(circulo)
```

```
ax.set_aspect('equal', adjustable='box')

plt.xlim(0, ancho)

plt.ylim(0, alto)

ax.legend(loc='upper left', bbox_to_anchor=(1, 1))

plt.title("Distribucion de tubos containers (20 pies)")

plt.ylabel("Alto containers (mm)")

plt.xlabel("Ancho containers (mm)")

plt.show()

grafic_carga()
```

```
def delete():
```

```
    selected_item = tabla.selection()[0]
```

```
    tabla.delete(selected_item)
```

```
del_btn = ttk.Button(fx1,width=8,text="Delete", command=delete)
```

```
del_btn.place(x=425,y=320)
```

```
btnAgregar=Button(fx1,text="Agregar" a
lista",width=15,command=Agregar,fg="white",bg="green")
```

```
btnAgregar.grid(row=6,column=2,padx=2,pady=8,columnspan=1)
```

```
btnDOanalisis=Button(fframe1,text="Hacer  
Analisis",width=15,fg="white",bg="green",command=Analisis)
```

```
btnDOanalisis.place(x=200,y=550)
```

```
entradaDatos()
```

```
def main ():
```

```
vent = Tk()
```

```
vent.title("volume optimizer software")
```

```
vent.geometry("500x1000")
```

```
ventana=ventanaprincipal(vent)
```

```
ventana.grid(row=0, column=0, sticky="NSEW")
```

```
vent.rowconfigure(0, weight=1)
```

```
vent.columnconfigure(0, weight=1)
```

```
ventana.mainloop()
```

```
if __name__ == "__main__":
```


main()