



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**USO DE HERRAMIENTA INFORMÁTICA PARA LA OPTIMIZACIÓN DEL
CORTE DE ACERO DE REFUERZO EN LA CONSTRUCCIÓN DE VIVIENDAS**

Proyecto de titulación previo a la obtención del título de:

Tecnólogo superior en construcción

Autor: Luis Guillermo Amaya Vásquez

Tutor: Ing. Erick Bermeo Bonilla

Quito, Ecuador

2025

Dedicatoria

Dedicado a la Hermandad Blanca, a mis Guías a mis Maestros a mi compañero inseparable Lemurin y toda su legión y en especial al maestro Jesús, por brindarme una vez más la posibilidad de obtener el conocimiento y las herramientas para continuar con la misión divina.

Tabla de contenidos

Dedicatoria	ii
Lista de tablas	iv
Lista de figuras	v
Agradecimientos	vii
Introducción.....	1
Definición del problema	1
Objetivo general.....	2
Objetivos específicos	2
Capítulo I.....	4
Estado del arte y la práctica	4
Capítulo II.....	6
Metodología	6
Modelo matemático para la optimización del corte de varillas.	8
Capítulo 3	20
Experimentación y uso del programa de optimización.....	20
Conclusiones.....	40
Recomendaciones.....	41
Referencias bibliográficas	42
Anexos.....	44
Anexo 1.....	44
Manual de uso del programa OptiCorte:.....	44
Anexo 2.....	50
Código del programa OptiCorte:	50

Lista de tablas

Tabla 1 <i>Planilla de aceros de 3 marcas</i>	8
Tabla 2 <i>Patrones de corte</i>	11
Tabla 3 <i>Patrones de corte óptimos</i>	12
Tabla 4 <i>Planilla de aceros de columnas casa Helio</i>	21
Tabla 5 <i>Planilla de aceros de plintos casa Helio</i>	25
Tabla 6 <i>Planilla de aceros de cadenas casa Helio</i>	26
Tabla 7 <i>Planilla de aceros de vigas strap casa Helio</i>	27
Tabla 8 <i>Planilla de aceros de losas casa Helio</i>	27
Tabla 9 <i>Planilla de aceros de losas casa Helio</i>	29
Tabla 10 <i>Tabulación de resultados obtenidos del análisis de la casa Helio</i>	31
Tabla 11 <i>Planilla de aceros cimentación letrero publicitario</i>	33

Lista de figuras

Figura 1	<i>Impacto del estudio sobre problemas de corte a nivel mundial</i>	6
Figura 2	<i>Flujograma de la metodología a usar</i>	7
Figura 3	<i>Función de transferencia y restricciones LP-SOLVE</i>	11
Figura 4	<i>Resultados de la optimización software LP-SOLVE</i>	12
Figura 5	<i>Diagrama de cortes de varilla luego de la optimización</i>	13
Figura 6	<i>Modelado en Excel y Solver</i>	14
Figura 7	<i>Solución encontrada con Excel y Solver</i>	14
Figura 8	<i>Código Python generación de condiciones</i>	15
Figura 9	<i>Código Python optimización con el solver pulp</i>	16
Figura 10	<i>Diagrama de Flujo del código generado para la aplicación OptiCorte</i>	17
Figura 11	<i>Patrones de corte calculados</i>	18
Figura 12	<i>Resultados obtenidos con el software desarrollado</i>	18
Figura 13	<i>Logotipo del programa</i>	19
Figura 14	<i>Fachada casa Helio</i>	20
Figura 15	<i>Optimización 1 planilla de aceros de columnas $\Phi 16\text{mm}$ casa Helio</i>	22
Figura 16	<i>Optimización 2 planilla de aceros de columnas $\Phi 16\text{mm}$ casa Helio</i>	22
Figura 17	<i>Porcentaje óptimo de estimación $\Phi 16\text{mm}$ casa Helio</i>	23
Figura 18	<i>Optimización 1 planilla de aceros de columnas $\Phi 8\text{mm}$ casa Helio</i>	24
Figura 19	<i>Porcentaje óptimo de estimación $\Phi 8\text{ mm}$ casa Helio</i>	24
Figura 20	<i>Optimización 1 planilla de aceros de plintos $\Phi 14\text{mm}$ casa Helio</i>	25
Figura 21	<i>Porcentaje óptimo de estimación $\Phi 14\text{ mm}$ casa Helio</i>	26
Figura 22	<i>Optimización planilla de aceros de losas $\Phi 12\text{ mm}$ casa Helio</i>	28
Figura 23	<i>Porcentaje óptimo de estimación $\Phi 12\text{ mm}$ casa Helio</i>	29
Figura 24	<i>Optimización planilla de gradas $\Phi 10\text{ mm}$ casa Helio</i>	30
Figura 25	<i>Porcentaje óptimo de estimación $\Phi 10\text{ mm}$ casa Helio</i>	30
Figura 26	<i>Vista en corte vertical</i>	32
Figura 27	<i>Vista en corte horizontal 1-1'</i>	32
Figura 28	<i>Vista en corte horizontal 2-2'</i>	32
Figura 29	<i>Optimización planilla de cimentación $\Phi 12$ longitud 12 metros</i>	33
Figura 30	<i>Porcentaje óptimo de estimación $\Phi 12$ longitud 12 metros</i>	34
Figura 31	<i>Optimización planilla de cimentación $\Phi 12$ longitud 6 metros</i>	34
Figura 32	<i>Porcentaje óptimo de estimación $\Phi 12$ longitud 6 metros</i>	35
Figura 33	<i>Optimización planilla de cimentación $\Phi 8$ longitud 12 metros</i>	35
Figura 34	<i>Porcentaje óptimo de estimación $\Phi 8$ longitud 12 metros</i>	36
Figura 35	<i>Optimización planilla de cimentación $\Phi 8$ longitud 6 metros</i>	36
Figura 36	<i>Porcentaje óptimo de estimación $\Phi 8$ longitud 6 metros</i>	37
Figura 37	<i>Diagrama isométrico de tuberías de agua potables de $\frac{1}{2}$"</i>	38
Figura 38	<i>Optimización de tuberías de agua potable de longitud 3 metros</i>	38
Figura 39	<i>Icono de OptiCorte luego de instalar</i>	44
Figura 40	<i>Pantalla inicial de OptiCorte</i>	44
Figura 41	<i>Ingreso de parámetros, longitud de la varilla (m) y cantidad de marcas</i>	45
Figura 42	<i>Ingreso de parámetros, longitudes a cortar (m) y cantidades de marcas</i>	46
Figura 43	<i>Inicio de cálculos del programa</i>	46
Figura 44	<i>Resultados del programa</i>	47
Figura 45	<i>Interpretación de resultados del programa</i>	48
Figura 46	<i>Pantalla inicial para realizar un nuevo cálculo</i>	49

DECLARACIÓN y AUTORIZACIÓN

Yo, LUIS GUILLERMO AMAYA VÁSQUEZ con C.I. 1002882551 autor del trabajo de titulación titulado: “USO DE HERRAMIENTA INFORMÁTICA PARA LA OPTIMIZACIÓN DEL CORTE DE ACERO DE REFUERZO EN LA CONSTRUCCIÓN DE VIVIENDAS”, previo a la obtención del título de TECNÓLOGO SUPERIOR EN CONSTRUCCIÓN, en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, FEBRERO 2025

LUIS GUILLERMO AMAYA VÁSQUEZ

C.I. 100288255-1

Agradecimientos

En primer lugar, doy gracias a Dios y a la hermandad Blanca por haberme guiado y brindado la fortaleza necesaria para culminar este proyecto. Su infinita sabiduría y amor han sido mi sustento en cada paso de este camino.

A mi familia, les expreso mi más profundo agradecimiento por su apoyo incondicional, su paciencia y su aliento en los momentos de mayor desafío. Sin su comprensión y cariño, este logro no habría sido posible.

De manera especial, quiero agradecer a mi tutor de tesis, por su invaluable dirección técnica, sus consejos acertados y su dedicación durante todo el proceso de investigación. Su experiencia y compromiso han sido fundamentales para el desarrollo y la culminación exitosa de este trabajo.

Este logro es el resultado de un esfuerzo colectivo, y por eso, les estaré eternamente agradecido.

Introducción

Este proyecto de fin de carrera analiza la optimización del corte de acero de refuerzo en la construcción de viviendas de hormigón armado, mediante el desarrollo de un modelo matemático basado en optimización lineal entera pura. La finalidad es crear una herramienta informática en Python que determine la cantidad óptima de varillas necesarias y sus patrones de corte, en función de la planilla de aceros.

El uso de técnicas avanzadas para planificar el corte de varillas de acero permite maximizar el aprovechamiento del material. Aplicaciones basadas en programación lineal y algoritmos especializados han mostrado reducciones significativas en los desperdicios de varillas, lo que impacta favorablemente en los costos del proyecto (Coelho et al., 2017; Kwon et al., 2021). Investigaciones recientes han evidenciado que la optimización del corte de acero puede disminuir los costos de material hasta en un 20 % (Kim et al., 2004; Lee et al., 2020).

La metodología del estudio integrará enfoques históricos, experimentales y analíticos, revisando el estado del arte a nivel local e internacional. Se recopilarán datos cuantitativos de planillas de acero de diversos proyectos, los cuales serán analizados con la herramienta informática propuesta. Se evaluarán indicadores que correlacionen el número de marcas y longitudes de la planilla de acero con el porcentaje de desperdicio por diámetro de varilla empleado (Ma et al., 2023; Ren et al., 2023). Además, se incorporarán estrategias de inteligencia artificial para mejorar la precisión de los resultados (Miller et al., 2016).

Definición del problema

El acero de refuerzo representa aproximadamente el 10-15 % del presupuesto de una vivienda de hormigón armado y está sujeto a variaciones por diseño estructural, ubicación del proyecto y gestión del material (Patiño, 2018). Los desperdicios durante el corte de

varillas pueden oscilar entre el 7 % y el 25 % del total del acero de refuerzo (Ochoa, 2014). La implementación de algoritmos de optimización puede reducir estos desperdicios hasta en un 15 % (Widjaja & Kim, 2023).

La planilla de aceros en un plano estructural no especifica la cantidad de varillas necesarias por diámetro, no considera el desperdicio ni optimiza los patrones de corte, lo que conduce a una gestión ineficiente del material (Machado et al., 2020). La falta de herramientas avanzadas de optimización y estandarización en los procesos de corte agrava el problema, aumentando los costos operativos (Wang et al., 2018).

Objetivo general

Desarrollar una herramienta informática que optimice la forma de corte de varillas de refuerzo en la construcción de viviendas, que permita minimizar el desperdicio de material y costos asociados.

Objetivos específicos

- Investigar y revisar plataformas y herramientas existentes para la optimización del corte de acero de refuerzo, con el fin de identificar mejores prácticas y características que puedan ser incorporadas en el desarrollo de la nueva herramienta.
- Crear un algoritmo eficiente en base a un modelo matemático de optimización lineal entera, teniendo en cuenta la dimensión estándar de las varillas de acero y las necesidades específicas de los proyectos de construcción.
- Implementar el algoritmo de optimización mediante el lenguaje de programación Python, usando el entorno de desarrollo integrado (IDE) Spyder en su versión estudiantil.

- Desarrollar una interfaz gráfica de usuario (GUI) con la librería Tkinter, que permita a los ingenieros y operarios ingresar datos de planillas de aceros de manera sencilla y obtener los resultados del corte óptimo de acero.

Capítulo I

Estado del arte y la práctica

Los problemas de corte y embalaje (Cutting and Packing Problems, C&P) pertenecen a la optimización combinatoria y buscan la distribución eficiente de materiales. En el ámbito de la construcción, la optimización del corte de acero de refuerzo es fundamental para reducir desperdicios y maximizar el uso de los recursos (Lee et al., 2020; Ma et al., 2023).

Los problemas de C&P pueden clasificarse en unidimensionales, bidimensionales y tridimensionales. En el caso del acero de refuerzo, los problemas son principalmente unidimensionales debido a la longitud de las varillas (Widjaja & Kim, 2023).

Problemas de Corte Unidimensional

Este tipo de optimización se enfoca en el corte de elementos lineales como varillas de acero, perfiles, tubería. Investigaciones han demostrado que la aplicación de estrategias de optimización puede reducir el desperdicio de material en un 12 % (Ren et al., 2023).

Técnicas y Algoritmos de Optimización.

La resolución de problemas de C&P implica el uso de diferentes enfoques:

Algoritmos Exactos: Incluyen la programación entera y la programación dinámica. Si bien garantizan soluciones óptimas, su aplicabilidad es limitada debido a la alta demanda computacional (Kim et al., 2004).

Algoritmos Heurísticos: Proporcionan soluciones eficientes sin garantizar la óptima. Estrategias como secuencias de corte y heurísticas de patrones se utilizan con frecuencia en la construcción (Machado et al., 2020).

Algoritmos Metaheurísticos: Incorporan técnicas como algoritmos genéticos y búsqueda tabú, que optimizan soluciones en problemas complejos de gran escala (Miller et al., 2016).

Aplicaciones en el Corte de Acero de Refuerzo

La optimización del corte de acero no solo reduce desperdicios, sino que también mejora la eficiencia operativa y disminuye costos (Widjaja & Kim, 2023).

Optimización de Longitudes de Corte: Algoritmos de corte unidimensional determinan patrones óptimos para minimizar los residuos y satisfacer requisitos específicos del proyecto (Ren et al., 2023).

Minimización de Desperdicio: La implementación de estrategias heurísticas y metaheurísticas permite optimizar el uso del material y reducir costos operativos (Lee et al., 2020).

Integración con Sistemas de Gestión: La combinación de modelos de optimización con herramientas de gestión mejora la planificación y la logística en proyectos de gran escala (Kwon et al., 2021).

Programación Lineal Entera Pura (ILP)

El uso de Programación Lineal Entera Pura (ILP) ha permitido generar estrategias de corte eficientes en proyectos de construcción. Investigaciones han demostrado que combinar ILP con estrategias codiciosas mejora la eficiencia y reduce desperdicios (Ren et al., 2023).

En proyectos de gran envergadura, la integración de algoritmos de optimización con sistemas de gestión ha resultado clave para reducir costos y mejorar la sostenibilidad en la construcción (Coelho et al., 2017).

Capítulo II

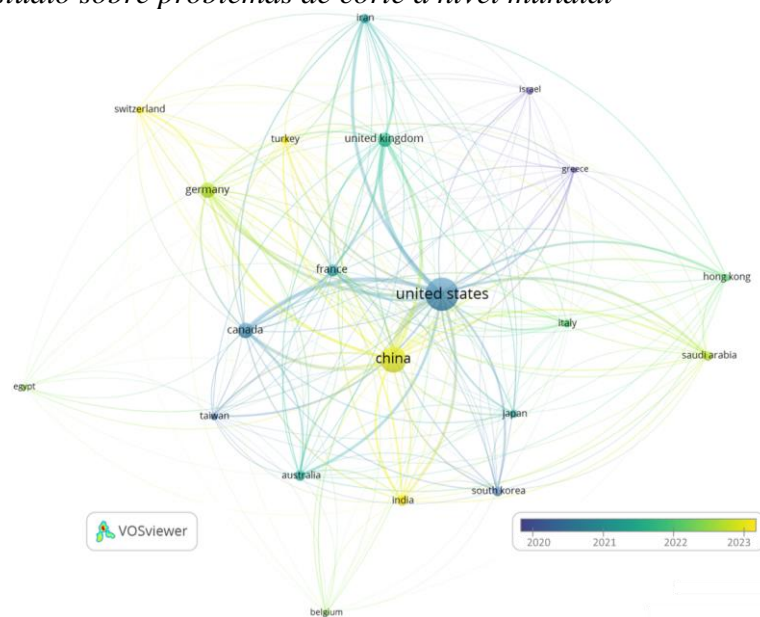
Metodología

En el presente capítulo se presenta la metodología utilizada para abordar el desarrollo del trabajo. El proyecto se basa en un enfoque cuantitativo, dado que implica el análisis y optimización de datos numéricos para minimizar el desperdicio en el corte de varillas. La investigación tiene una orientación aplicada, pues busca resolver un problema práctico en la construcción mediante un modelo matemático y su implementación computacional.

Se inicia con la metodología histórico descriptiva con la revisión inicial del estado del arte a nivel mundial, regional y local. Se verificarán trabajos relacionados para conocer el camino recorrido y lo propuesto en dichos trabajos de acuerdo a la Figura 1, Estados Unidos, China, Canadá e India son los países que lideran la investigación referida al problema de corte (Cutting and packing problems), la figura 1 se obtuvo utilizando la base de datos de “Scopus” y el software VOSviewer.

Figura 1

Impacto del estudio sobre problemas de corte a nivel mundial



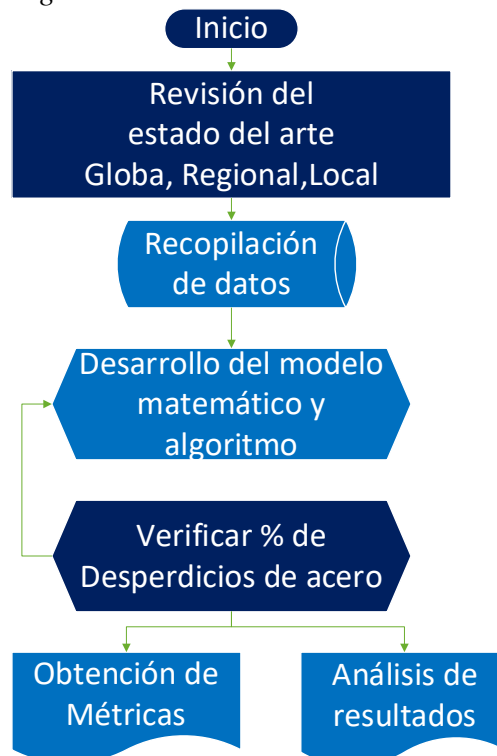
Nota. Fuente. Figura generada en VOSviewer por el autor.

Metodología experimental. Obtención de planillas de aceros, de los planos estructurales de viviendas, de esta manera se tendrá datos cuantitativos, los datos de las planillas de aceros, diámetro de varillas, numero de marca, longitud de la marca y cantidad. Estos datos serán introducidos en el programa de optimización base de este trabajo de investigación.

Metodología deductiva. Uso de las herramientas informáticas para análisis, con los datos recopilados e introducirá la información recopilada. El software generará resultados cuantitativos sobre el porcentaje de desperdicio potencialmente optimizado. Estos resultados también serán datos numéricos que se utilizarán para la comparación.

Metodología analítica sistemática. - Análisis comparativo entre los datos reales obtenidos de las obras y los datos simulados obtenidos mediante la herramienta informática. Este análisis sistemático permitirá describir las diferencias en los porcentajes de desperdicio y evaluar la eficacia de la herramienta informática en términos de reducción del desperdicio, en la Figura 2 se expone un flujograma de la metodología expuesta.

Figura 2
Flujograma de la metodología a usar.



Nota. Figura generada por el autor.

La construcción civil es una industria compleja que involucra la planificación, gestión y ejecución de proyectos de gran envergadura, como la construcción de casas y edificios. En este contexto, la programación lineal (PL) se presenta como una herramienta matemática esencial para optimizar diversos aspectos de estos proyectos, desde la asignación de recursos hasta la planificación de actividades. Este ensayo explora cómo la PL se aplica en la construcción civil, destacando su utilidad en la optimización de costos, la gestión de materiales, la planificación del cronograma de obras y la toma de decisiones estratégicas.

Modelo matemático para la optimización del corte de varillas.

El problema de optimización del corte de varillas se puede modelar utilizando la Programación Lineal Entera (PLE). El objetivo es minimizar el desperdicio de material cuando se cortan varillas de longitudes estandarizadas para satisfacer las necesidades específicas de un proyecto de construcción.

A continuación, se explica el modelo matemático ayudado de un ejemplo numérico. Se tiene la siguiente planilla de aceros:

Tabla 1
Planilla de aceros de 3 marcas

Marcas	Longitud (m)	Cantidad
m100	3	10
m101	5	5
m102	7	8

Para el problema de optimización de corte de varillas, la formulación matemática se enfoca en minimizar el número total de varillas de 12 metros usadas para satisfacer las demandas específicas de la tabla No.1

Definiciones:

Varillas Disponibles: Se dispone de varillas de una longitud estandarizada

$$L = 12 \text{ metros.}$$

Longitudes Requeridas: Se requieren n diferentes longitudes de varillas para el proyecto, denotadas como l_1, l_2, \dots, l_n

Demanda: d_i representa la cantidad de varillas de longitud l_i requeridas para el proyecto.

Objetivo de la optimización.

Determinar cuántas varillas de longitud L se requiere y cómo cortar para minimizar el desperdicio.

Z : Es el total de varillas de 12 metros necesarias.

x_i : Representa la cantidad de veces que se utiliza el patrón i .

n : Es el total de patrones de corte posibles que cumplen con la restricción.

de longitud máxima.

Función objetivo.

La función objetivo se enfoca en minimizar el número total de varillas de 12 metros utilizadas, que se denomina x_i para cada patrón de corte viable i La función objetivo se expresa como:

$$\text{Minimizar } Z = \sum_{i=1}^n x_j \quad (1)$$

Donde:

Z : Es el total de varillas de 12 metros necesarias.

x_i : Representa la cantidad de veces que se utiliza el patrón i .

n : Es el total de patrones de corte posibles que cumplen con la restricción.

Para el ejemplo propuesto se tiene $n = 11$ patrones de corte que mas adelante se describe la forma de obtenerlos. Entonces la función objetivo queda de la siguiente manera:

$$Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \quad (2)$$

Restricciones

Las restricciones se aseguran de que cada demanda de longitud específica se cumpla. Para cada longitud requerida, se establece una restricción que debe satisfacer al menos la cantidad demandada. Si denotamos cada longitud como m_j y su demanda como d_j , las restricciones se formulan de la siguiente manera:

$$\text{Satisfacción de la Demanda: } \sum_{i=1}^n a_{ij} * x_i \geq d_j \quad \text{para cada } j \quad (3)$$

Donde:

a_{ij} : Representa el número de veces que la longitud m_j aparece en el patrón i .

d_j : Es la demanda mínima de longitud m_j .

Las restricciones para el ejemplo numérico propuesto, para la longitud de 3 metros se requiere 10 unidades.

$$\sum_{i=1}^n a_{i3} * x_i \geq 10 \quad (4)$$

Para la longitud de 5 metros se requiere 5 unidades.

$$\sum_{i=1}^n a_{i5} * x_i \geq 5 \quad (5)$$

Para la longitud de 7 metros se requiere 8 unidades.

$$\sum_{i=1}^n a_{i7} * x_i \geq 8 \quad (6)$$

Para calcular la cantidad de patrones de corte posibles del ejemplo numérico, se describe en la Tabla 2:

Tabla 2
Patrones de corte

Patrón de Corte	Longitud (m)			Total (m)	Desperdicio (m)
	3	5	7		
x1	0	1	1	12	0
x2	4	0	0	12	0
x3	2	1	0	11	1
x4	1	0	1	10	2
x5	0	2	0	10	2
x6	3	0	0	9	3
x7	1	1	0	8	4
x8	0	0	1	7	5
x9	2	0	0	6	6
x10	0	1	0	5	7
x11	1	0	0	3	9
	11	5	8		
	Cantidad (u)				

De la tabla No.2 se extraen las restricciones donde: $x_n = \text{numero entero}$

$$0x_1 + 4x_2 + 2x_3 + x_4 + 0x_5 + 3x_6 + x_7 + 0x_8 + 2x_9 + 0x_{10} + x_{11} \geq 10 \quad (7)$$

$$x_1 + 0x_2 + x_3 + 0x_4 + 2x_5 + 0x_6 + 1x_7 + 0x_8 + 2x_9 + x_{10} + 0x_{11} \geq 5 \quad (8)$$

$$x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 0x_6 + 0x_7 + x_8 + 0x_9 + 0x_{10} + 0x_{11} \geq 8 \quad (9)$$

Con la función de transferencia y las restricciones planteadas se introducen en un software de optimización matemática para encontrar la solución, para el ejemplo se introdujo en el programa LP_SOLVE.

Figura 3
Función de transferencia y restricciones LP-SOLVE

```

1 /* Objective function */
2 min:x1 + x2 + x3 + x4 +x5+x6+x7+x8+x9+x10+x11;
3 /* Variable bounds */
4 4*x2+2*x3+x4+3*x6+x7+2*x9+x11 >= 11;
5 x1+x3+2*x5+x7+x10 >= 5;
6 x1+x4+x8>= 8;
7 int x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11;
  
```

Nota. Figura generada por el autor utilizando el software LP Solve IDE 5.5.2.0

Figura 4
Resultados de la optimización software LP-SOLVE

Variables	MILP ...	result
	10	10
x1	5	5
x2	2	2
x3	0	0
x4	3	3
x5	0	0
x6	0	0
x7	0	0
x8	0	0
x9	0	0
x10	0	0
x11	0	0

Nota. Figura generada por el autor utilizando el software LP Solve IDE 5.5.2.0

Los resultados obtenidos se interpretan de la siguiente manera:

$$x_1 = 5, \quad x_2 = 2, \quad x_4 = 3,$$

$$\text{Numero total de varillas (Z)} = x_1 + x_2 + x_4 \quad (10)$$

$$\text{Numero total de varillas (Z)} = 5 + 2 + 3$$

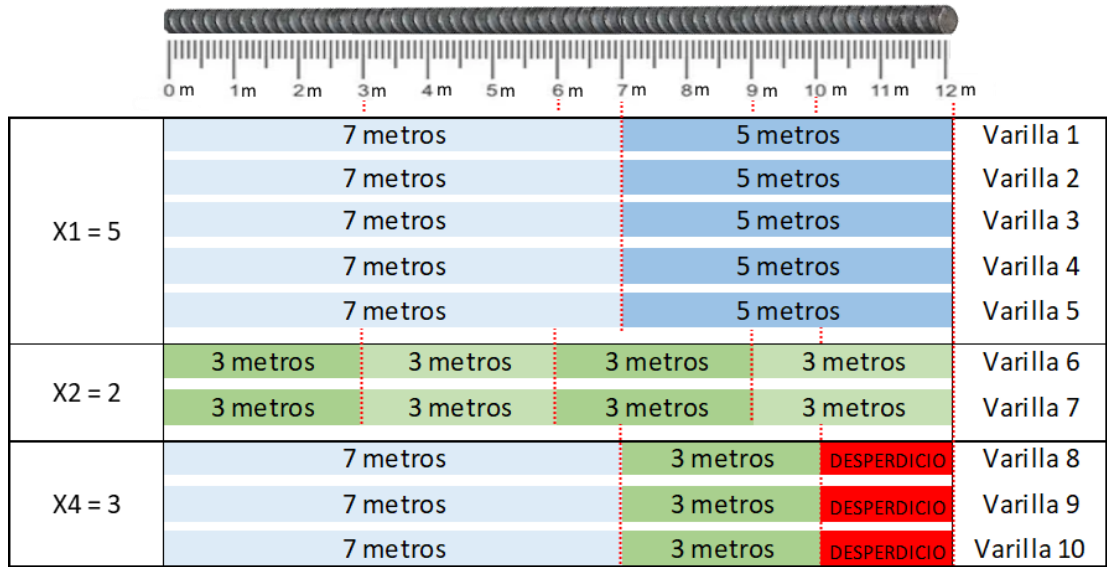
$$\text{Numero total de varillas (Z)} = 10$$

Tabla 3
Patrones de corte óptimos

Patrón de Corte	Longitud (m)			Total (m)	Desperdicio (m)
	3	5	7		
x1	0	1	1	12	0
x2	4	0	0	12	0
x3	2	1	0	11	1
x4	1	0	1	10	2
x5	0	2	0	10	2
x6	3	0	0	9	3
x7	1	1	0	8	4
x8	0	0	1	7	5
x9	2	0	0	6	6
x10	0	1	0	5	7
x11	1	0	0	3	9

En la Figura 5 se describe a detalle los resultados de optimización obtenidos con el uso de LP Solve IDE 5.5.2.0 y Excel (Solver), que en si fueron los mismos resultados.

Figura 5
Diagrama de cortes de varilla luego de la optimización



Nota. Figura generada por el autor

El mismo problema planteado ahora se procede a resolver con el optimizador de Excel. La herramienta Solver en Excel es un complemento que permite resolver problemas de optimización lineal, como el ejemplo de minimización de desperdicio de varillas de 12 metros al cumplir una demanda específica de cortes de distintas longitudes (3, 5, y 7 metros). Lo primero que se debe hacer es identificar la función objetivo y las restricciones que ya se calcularon.

Función Objetivo.

$$Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \quad (11)$$

Restricciones.

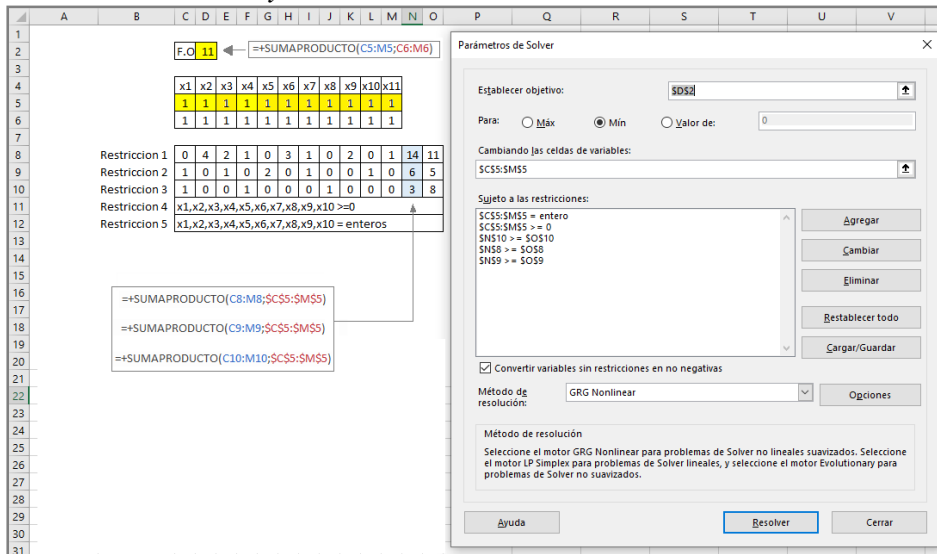
$$0x_1 + 4x_2 + 2x_3 + x_4 + 0x_5 + 3x_6 + x_7 + 0x_8 + 2x_9 + 0x_{10} + x_{11} \geq 10 \quad (12)$$

$$x_1 + 0x_2 + x_3 + 0x_4 + 2x_5 + 0x_6 + 1x_7 + 0x_8 + 2x_9 + x_{10} + 0x_{11} \geq 5 \quad (13)$$

$$x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 0x_6 + 0x_7 + x_8 + 0x_9 + 0x_{10} + 0x_{11} \geq 8 \quad (14)$$

En la Figura 6 Se ilustra como modelar en Excel y Solver la función objetivo y las restricciones:

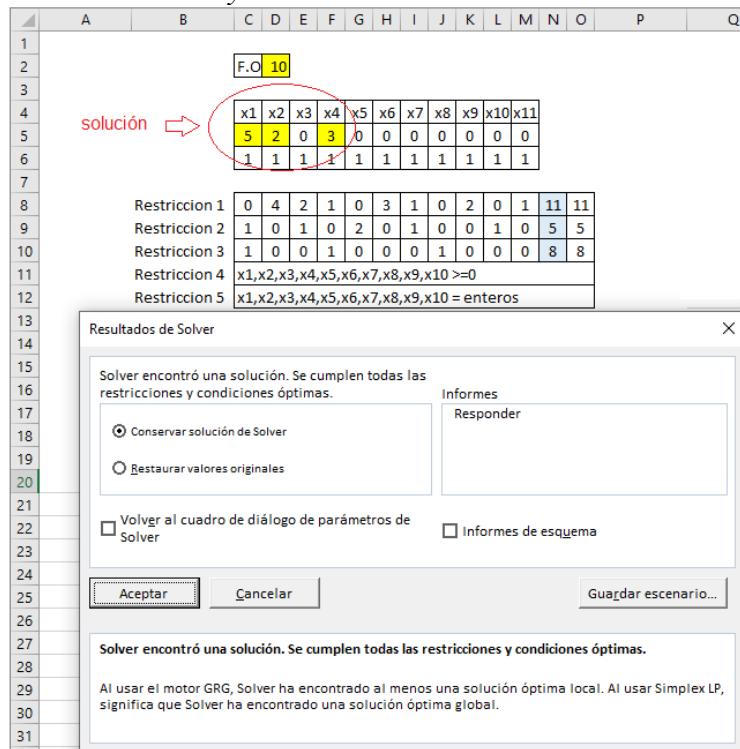
Figura 6
Modelado en Excel y Solver.



Nota. Figura generada por el autor utilizando el software Excel de office.

Una vez modelado la función objetivo e introducidas las restricciones se presiona el botón resolver y se obtiene los resultados en las celdas resaltadas en color amarillo como se ilustra en la Figura 7.

Figura 7
Solución encontrada con Excel y Solver



Nota. Figura generada por el autor utilizando el software Excel de office.

La respuesta encontrada con el optimizador de Excel es la misma que se encontró con el software LP_SOLVE.

Programación en Python para Optimización de Cortes de Varillas. A continuación, se detalla la metodología utilizada en la programación del modelo de optimización de cortes de varillas de acero, se expone las librerías empleadas y su integración en el código. Al programarse de lo denomina como “OPTI_CORTE”.

Interfaz Gráfica con Tkinter.

Entradas de texto, recogen las longitudes totales de las varillas, longitudes específicas y cantidades necesarias. Los botones permiten ingresar datos, realizar cálculos y reiniciar el proceso. Los marcos con scroll, facilitan la navegación por las entradas y resultados cuando el contenido supera el tamaño de la ventana.

Generación de Patrones con Itertools.

Genera todas las combinaciones posibles de longitudes de cortes que no excedan la longitud total disponible. Filtra patrones válidos para reducir el tiempo de procesamiento y evitar combinaciones irrelevantes, a continuación, se presenta el código:

Figura 8
Código Python generación de condiciones

```
python

def generate_patterns(measurements, total_length):
    patterns = []
    for r in range(1, len(measurements) + 20):
        for combo in itertools.combinations_with_replacement(measurements, r):
            if sum(combo) <= total_length:
                patterns.append(combo)
    return patterns
```

Nota. Figura generada por el autor utilizando el software IDE Spider.

Este método garantiza que solo se consideren patrones factibles y evita combinaciones que excedan la longitud total.

Optimización con PuLP.

Define variables de decisión para representar el número de varillas utilizadas en cada patrón. Establece la función objetivo, minimizar el número total de varillas. Agrega restricciones, de esta manera asegura que se cumpla la cantidad mínima requerida para cada longitud, restringe los patrones a la longitud total de las varillas disponibles y resuelve el problema usando el solver integrado de PuLP.

Figura 9

Código Python optimización con el solver pulp

```
python

demand = self.vector2
x = pulp.LpVariable.dicts("pattern", range(len(patterns_with_waste)), lowBound=0,
cat='Integer')

problem = pulp.LpProblem("Minimize_Waste", pulp.LpMinimize)
problem += pulp.lpSum(x[i] for i in range(len(patterns_with_waste))), "TotalBars"

for j in range(len(measurements)):
    problem += pulp.lpSum(pattern.count(measurements[j]) * x[i]
        for i, (pattern, waste) in enumerate(patterns_with_waste)) >= demand[j]
```

Nota. Figura generada por el autor utilizando el software IDE Spider.

La formulación asegura que se minimice el uso total de varillas, respetando las restricciones de demanda.

Presentación de Resultados

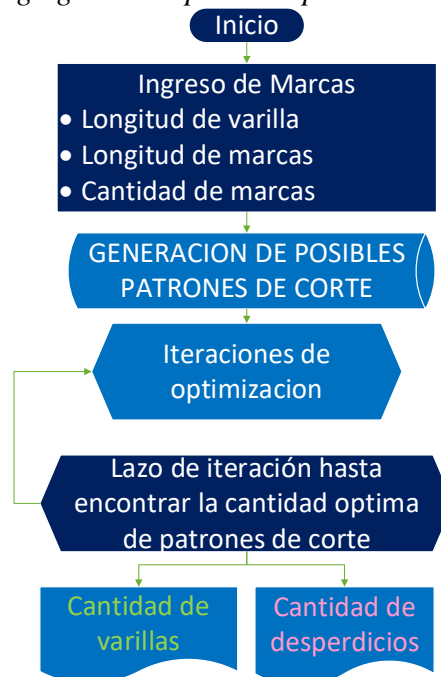
El programa OptiCorte presenta sus resultados en tres cuadros de texto:

1. Resumen general: Total de varillas utilizadas y desperdicio total.
2. Patrones óptimos: Detalle de cómo se deben cortar las varillas.
3. Desperdicio por patrón: Desperdicio asociado a cada combinación

Cada Cuadro de texto proporciona información clara y estructurada de forma fácil y sencilla para que pueda interpretar el usuario final.

En la Figura 10 se ilustra el proceso de optimización del corte de varillas mediante un diagrama de flujo. Este proceso inicia con el ingreso de datos, donde se introduce la longitud de la varilla, la longitud de las marcas a cortar y la cantidad de marcas necesarias. Con esta información, el sistema procede a la generación de posibles patrones de corte, creando combinaciones que buscan minimizar el desperdicio. A continuación, se realizan iteraciones de optimización, en las cuales el programa evalúa y ajusta los patrones propuestos para mejorar la eficiencia en el uso del material. Luego, se ejecuta un lazo de iteración que continúa refinando los patrones hasta encontrar la solución óptima, asegurando un equilibrio entre la cantidad de varillas utilizadas y la reducción del desperdicio. Finalmente, el sistema entrega los resultados, indicando cuántas varillas se requieren y cuánto material se desperdicia, lo que permite una toma de decisiones más eficiente en la planificación del corte.

Figura 10
Diagrama de Flujo del código generado para la aplicación OptiCorte



Nota. Figura generada por el autor.

Con el desarrollo del programa de optimización “OptiCorte”, se continua con la experimentación del ejemplo antes resuelto con LP_SOLVE y EXCECEL.

Ahora los mismos datos serán introducidos al programa “OptiCorte” el programa automáticamente calcula los patrones de corte, por ende, las restricciones y la función objetivo, para el ejemplo numérico en la Figura 10 se muestra los patrones de corte calculados.

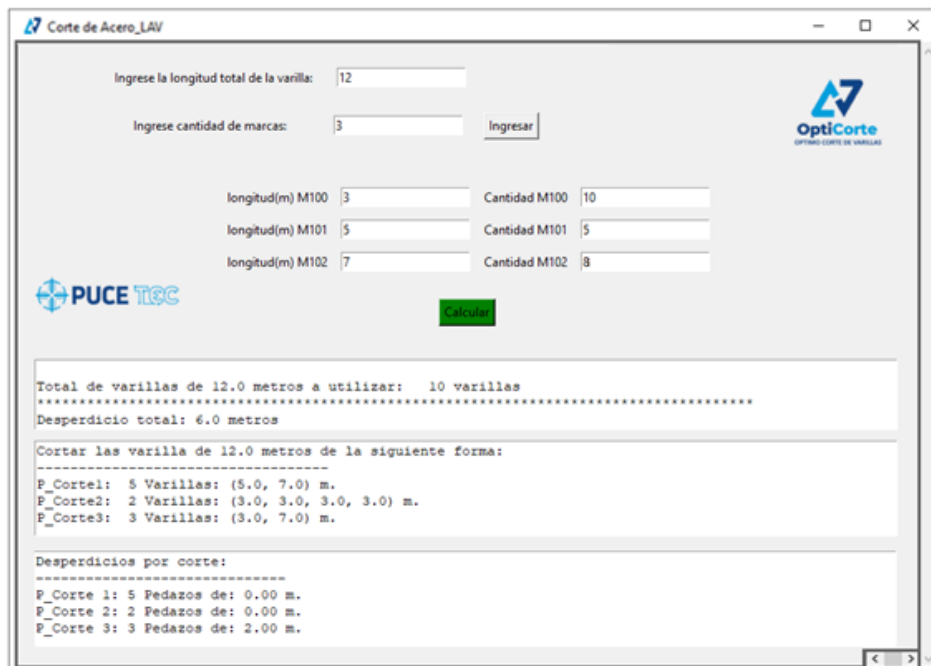
Figura 11
Patrones de corte calculados

```

Patrones generados (cortes válidos) para una varilla de 12 metros:
-----
Patrón: (5.0, 7.0), Desperdicio: 0.0 metros
Patrón: (3.0, 3.0, 3.0, 3.0), Desperdicio: 0.0 metros
Patrón: (3.0, 3.0, 5.0), Desperdicio: 1.0 metros
Patrón: (3.0, 7.0), Desperdicio: 2.0 metros
Patrón: (5.0, 5.0), Desperdicio: 2.0 metros
Patrón: (3.0, 3.0, 3.0), Desperdicio: 3.0 metros
Patrón: (3.0, 5.0), Desperdicio: 4.0 metros
Patrón: (7.0, ), Desperdicio: 5.0 metros
Patrón: (3.0, 3.0), Desperdicio: 6.0 metros
Patrón: (5.0, ), Desperdicio: 7.0 metros
Patrón: (3.0, ), Desperdicio: 9.0 metros
    
```

Nota. Figura generada por el autor utilizando el software OptiCorte.

Figura 12
Resultados obtenidos con el software desarrollado



Nota. Figura generada por el autor utilizando el software OptiCorte.

En la Figura 11 se ilustra cómo se introduce los datos a programa OptiCorte. A diferencia de los dos programas antes usados, en este programa solo se ingresa la longitud de la varilla (12 m) la cantidad de marcas y las longitudes de cada marca, finalmente arroja el resultado optimizado.

Se obtiene los mismos resultados que se obtuvieron con los 2 softwares antes, con OptiCorte el ingreso de los datos de forma dinámica y rápida, no es necesario tener conocimientos de la teoría de optimización lineal entera, a diferencia de los 2 programas anteriores.

Una vez iniciadas las pruebas del programa “OptiCorte”, se identificó la necesidad de crear un logotipo, ya que este es el primer elemento que los usuarios ven y asocian con la aplicación. Un diseño bien elaborado no solo fortalece la identidad de la marca, sino que también ayuda a diferenciar a OptiCorte de otros programas similares en el mercado. Los clientes y usuarios son más propensos a confiar en un programa que presenta una imagen profesional. Un buen logotipo puede reflejar la calidad y el compromiso de OptiCorte con la excelencia en el servicio.

Figura 13
Logotipo del programa



Nota. Figura generada por el autor.

El logotipo creado está basado en los colores símbolos de la PUCE además tiene 2 estribos de columna entrecortados en la mitad, el logotipo creado no es solo un elemento decorativo, sino una inversión estratégica en la marca y el éxito de OptiCorte. Refuerza la identidad, establece confianza, mejora el reconocimiento y facilita la comunicación de los valores del programa, posicionándolo favorablemente en el mercado competitivo de la optimización del corte de varillas de acero.

Capítulo 3

Experimentación y uso del programa de optimización.

En este capítulo se analiza de forma práctica el uso del programa OptiCorte, objetivo de este trabajo de titulación. El uso será introduciendo datos de planillas de acero de viviendas construidas en la ciudad de Quito, de las cuales se extraerá con el uso del programa el número de varillas necesarias por diámetro, se analizará el porcentaje de desperdicio por cada diámetro de varilla, a las viviendas bajo análisis se las llamará con nombres de ficticios por temas de derechos de autor de los planos usados.

Casa Helio.

La casa Helio está construida en la ciudad de Quito en la parroquia de Llano Chico, es una vivienda familiar de 150m² de construcción en 2 plantas, las dimensiones de la construcción son: 7,85m de frente por 9,55m de fondo. De acuerdo a los planos estructurales de esta vivienda, se cuenta con 7 planilla de acero distribuidas de la siguiente manera:

Figura 14
Fachada casa Helio



Nota. Figura generada por el autor

1. Planilla de columnas
2. Planilla de plintos.

3. Planilla de cadenas
4. Planilla de vigas Strap
5. Planilla de losas
6. Planilla de gradas

Los datos de cada planilla serán ingresados a al programa OptiCorte para encontrar el número de varillas optimo por cada diámetro.

Optimización planilla de columnas casa Helio.

Se inicia el análisis de optimización con la planilla de columnas donde se tiene 3 marcas de varillas de un diámetro (Φ) de 16mm, de acuerdo a la Tabla 4

Tabla 4
Planilla de aceros de columnas casa Helio

PLANILLA DE COLUMNAS											
	MARCA	Ø	TIPO	DIMENSIONES (m)			Cantidad	Dimensión (m)	Longitud Total (m)	Peso Específico (Kg/m)	Peso (kg)
				a	b	c					
Acero Principal	800	16	C	0,17	6,41	0,17	40	6,75	270	1,578	426,06
	801	16	L	0,17	4,9		40	5,07	202,8	1,578	320,02
	802	16	L	0,17	2,4		40	2,60	102,8	1,578	162,22
Estribos	400	8	O	0,24	0,14		900	0,38	342	0,395	270,18
	401	8	A	0,14	0,1		900	0,24	306	0,395	120,87

En una primera corrida del programa se analiza que existen un desperdicio total de 23,19 metros de varilla Φ 16mm, en la Figura 15 se observa que entre los desperdicios que se van a obtener existen 10 pedazos de 1,6 metros. Al tener la marca M802 de 2,60 metros se puede formar esta marca traslapando 2 pedazos de 1,6 metros. Con esta consideración se vuelve a correr el programa, pero de aumenta una marca de 1,6 metros de 10 unidades y la marca de 2,6 metros se reduce en 5 unidades, que son las que se va traslapar.

Figura 15

Optimización 1 planilla de aceros de columnas $\Phi 16\text{mm}$ casa Helio

Ingrese la longitud total de la varilla: 12

Ingrese cantidad de marcas: 3

Luis Amaya V. PUCETEC

longitud(m) M100	6.75	Cantidad M100	40
longitud(m) M101	5.07	Cantidad M101	40
longitud(m) M102	2.60	Cantidad M102	40

Total de varillas de 12.0 metros a utilizar: 50 varillas
Desperdicio total: 23.199999999999985 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

P_Cortel: 40 Varillas: (6.75, 5.07) m.
P_Corte2: 10 Varillas: (2.6, 2.6, 2.6, 2.6) m.

Desperdicios por corte:

P_Corte 1: 40 Pedazos de: 0.18 m.
P_Corte 2: 10 Pedazos de: 1.60 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Con el criterio de traslape se logra optimizar de mejor manera reduciendo el número total de varillas que en la primera optimización fue 50 unidades a 49 unidades y la cantidad de desperdicio de 23,19 metros a 8,2 metros.

Figura 16

Optimización 2 planilla de aceros de columnas $\Phi 16\text{mm}$ casa Helio

Ingrese la longitud total de la varilla: 12

Ingrese cantidad de marcas: 4

Luis Amaya V. PUCETEC

longitud(m) M100	6.75	Cantidad M100	40
longitud(m) M101	5.07	Cantidad M101	40
longitud(m) M102	2.6	Cantidad M102	35
longitud(m) M103	1.6	Cantidad M103	10

Total de varillas de 12.0 metros a utilizar: 49 varillas
Desperdicio total: 8.200000000000003 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

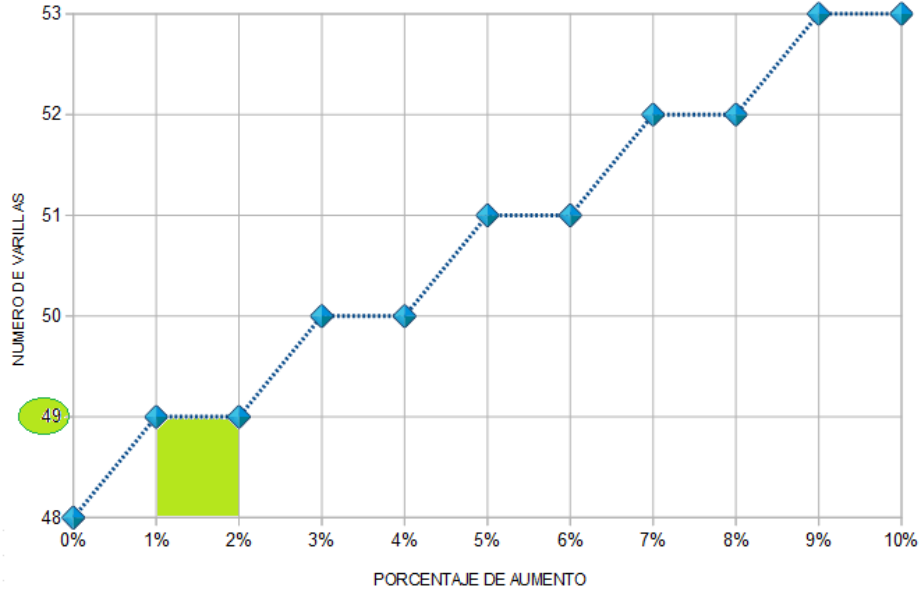
P_Cortel: 16 Varillas: (6.75, 2.6, 2.6) m.
P_Corte2: 24 Varillas: (6.75, 5.07) m.
P_Corte3: 8 Varillas: (5.07, 5.07, 1.6) m.
P_Corte4: 1 Varillas: (2.6, 2.6, 2.6, 1.6, 1.6) m.

Desperdicios por corte:

P_Corte 1: 16 Pedazos de: 0.05 m.
P_Corte 2: 24 Pedazos de: 0.18 m.
P_Corte 3: 8 Pedazos de: 0.26 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Figura 17
Porcentaje óptimo de estimación $\Phi 16\text{mm}$ casa Helio



Nota. Figura generada por el autor

De acuerdo a la Figura 17, si se considera un porcentaje extra del 1% o 2% se estará en el porcentaje óptimo. Si se estima un 3% y 4% se estimará una varilla adicional y se estima el 5% se estimará 2 varillas adicionales siendo esto una pérdida para el constructor.

Ahora se procederá a optimizar las marcas 400 y 401 estas marcas son pertenecientes a los estribos que proporciona refuerzo lateral a una columna, ayudando a mejorar su capacidad de carga y resistencia al pandeo. Además, contribuye a mantener las varillas de refuerzo en su posición durante el vertido del hormigón y mejora la capacidad de la columna para resistir cargas sísmicas y otras fuerzas laterales. En sí, los estribos son esenciales para aumentar la durabilidad y seguridad estructural de las columnas.

Optimización planilla de plintos casa Helio.

Para la planilla de plintos se tiene 3 marca como se muestra en la Tabla 5

Tabla 5
Planilla de aceros de plintos casa Helio

PLANILLA DE PLINTOS											
MARCA	Ø	TIPO	DIMENSIONES (m)				Cantidad	Dimensión (m)	Longitud (m)	Peso Específico (Kg/m)	Peso (kg)
			a	b	c	d					
1412	14	D	0,15	0,9	0,15		33	1,2	39,6	1,208	47,84
1421	14	D	0,15	1,9	0,15		18	2,2	39,6	1,208	47,84
1418	14	D	0,15	1,5	0,15		108	1,8	194,4	1,208	234,84

Una vez corrido el programa se observa en la Figura 20 que se requiere 24 varillas de 12 metros y que existirá un desperdicio de 11.39 metros de varilla de $\Phi 14\text{mm}$.

Figura 20
Optimización 1 planilla de aceros de plintos $\Phi 14\text{mm}$ casa Helio

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:

Luis Amaya V. PUCETEC

longitud(m) M100	<input type="text" value="1.2"/>	Cantidad M100	<input type="text" value="33"/>
longitud(m) M101	<input type="text" value="2.2"/>	Cantidad M101	<input type="text" value="18"/>
longitud(m) M102	<input type="text" value="1.8"/>	Cantidad M102	<input type="text" value="108"/>

Total de varillas de 12.0 metros a utilizar: 24 varillas

Desperdicio total: 11.399999999999983 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

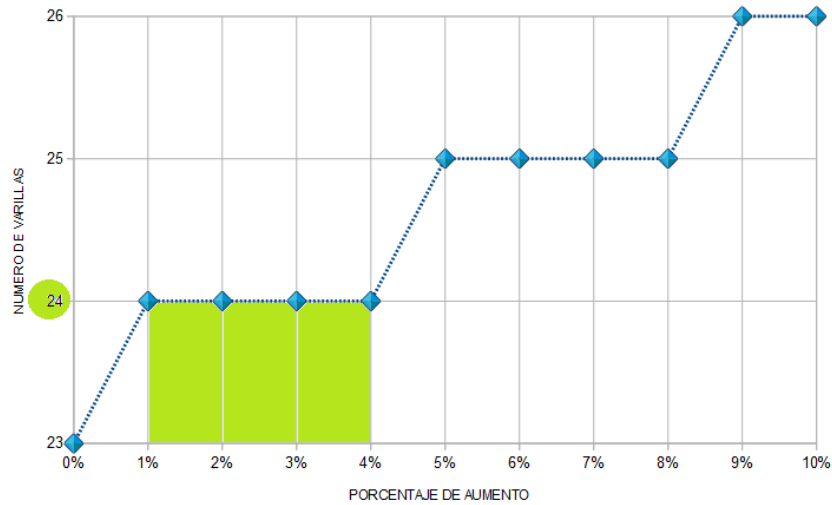
P_Corte1: 17 Varillas: (1.2, 1.2, 2.2, 1.8, 1.8, 1.8, 1.8) m.
P_Corte2: 1 Varillas: (2.2, 1.8, 1.8, 1.8, 1.8, 1.8) m.
P_Corte3: 6 Varillas: (1.8, 1.8, 1.8, 1.8, 1.8, 1.8) m. |

Desperdicios por corte:

P_Corte 1: 17 Pedazos de: 0.20 m.
P_Corte 2: 1 Pedazos de: 0.80 m.
P_Corte 3: 6 Pedazos de: 1.20 m.

Nota. Figura generada por el autor utilizando el software OptiCorte

Figura 21
Porcentaje óptimo de estimación $\Phi 14$ mm casa Helio



Nota. Figura generada por el autor

Optimización planilla de cadenas casa Helio.

Para el caso de la planilla de cadenas al ser una maraca por diámetro $\Phi 12$ mm y $\Phi 8$ mm no aplica correré el programa de optimización solo basta con dividir 12 metros de la varia para la longitud de la marca, resultando 28 varillas $\Phi 12$ mm con cero desperdicios. Para $\Phi 8$ mm se requiere 12 varillas y se tendrá un desperdicio de 12 pedazos de 0.24 metro dando un total de 2.88 metros.

Tabla 6
Planilla de aceros de cadenas casa Helio

PLANILLA DE CADENAS											
MARCA	Ø	TIPO	DIMENSIONES (m)				Cantidad	Dimensión (m)	Longitud total (m)	Peso Específico (Kg/m)	Peso (kg)
			a	b	c	d					
1201	12	I	12				28	12	336	0,888	298,37
1202	8	O	0,14	0,14	0,14	0,14	250	0,56	140	0,395	55,30

Optimización planilla de vigas strap casa Helio.

Para el caso de la planilla de vigas strap al ser una maraca por diámetro $\Phi 14$ mm y $\Phi 10$ mm, no aplica correré el programa de optimización solo basta con dividir 12 metros de la varia para la longitud de la marca, resultando 9 varillas $\Phi 14$ mm con un desperdicio de 3,63 metros

por varilla. Para $\Phi 10\text{mm}$ se requiere 6 varillas y se tendrá un desperdicio de 6 pedazos de 0.40 metro dando un total de 2.40 metros.

Tabla 7
Planilla de aceros de vigas strap casa Helio

PLANILLA VIGAS STRAP											
MARCA	\emptyset	TIPO	DIMENSIONES (m)				Cantidad	Dimensión (m)	Longitud total (m)	Peso	
			a	b	c	d				Especifico (Kg/m)	Peso (kg)
1445	14	C	0,15	3,89	0,15		18	4,19	75,42	1,208	91,11
1034	10	O	0,24	0,34	0,24	0,34	57	1,16	66,12	0,617	40,80

Optimización planilla de losas casa Helio.

La optimización de la planilla de losa es la que más marcas tiene, un total de 13 marcas con dimensiones variadas, como se aprecia en la Tabla 8

Tabla 8
Planilla de aceros de losas casa Helio

PLANILLA DE LOSAS										
MARCA	\emptyset	TIPO	DIMENSIONES (m)			Cantidad	Dimensión (m)	Longitud total (m)	Peso	
			a	b	c				Especifico (Kg/m)	Peso (kg)
700	12	C	0,17	1,32	0,17	12	1,66	22,08	0,888	19,61
701	12	C	0,17	3	0,17	12	3,34	40,08	0,888	35,59
702	12	C	0,17	2,8	0,17	20	3,14	62,8	0,888	55,77
703	12	C	0,17	1,52	0,17	54	1,86	104,76	0,888	93,03
704	12	C	0,17	2,7	0,17	30	3,04	91,2	0,888	80,99
705	12	C	0,17	5,4	0,17	4	5,74	22,96	0,888	20,39
706	12	C	0,17	7,9	0,17	11	8,24	90,64	0,888	80,49
707	12	C	0,17	6,9	0,17	6	7,24	43,44	0,888	38,57
708	12	C	0,17	9,8	0,17	6	10,14	60,84	0,888	54,03
709	12	C	0,17	1,7	0,17	15	2,04	30,6	0,888	27,17
710	12	C	0,17	8,1	0,17	15	8,44	126,6	0,888	112,42
711	12	C	0,17	3,1	0,17	13	3,44	44,72	0,888	39,71
712	12	C	0,17	10	0,17	13	10,34	134,42	0,888	119,36

Se realiza la corrida de la optimización en este caso el tiempo computación si fue considerable de demoró en encontrar una respuesta optima 5:23 minutos el programa fue corrido en una computadora portátil marca HP con un procesador Intel(R) Core (TM) i7-

6500U CPU @ 2.50GHz 2.59 GHz y una memoria RAM de 16 Gigas y un disco solido 500 gigas.

La optimización arroja que se requiere 75 varillas $\Phi 12$ mm, 13 patrones de corte óptimos un desperdicio de 24,43 metros prácticamente 2 varillas de desperdicio.

Figura 22

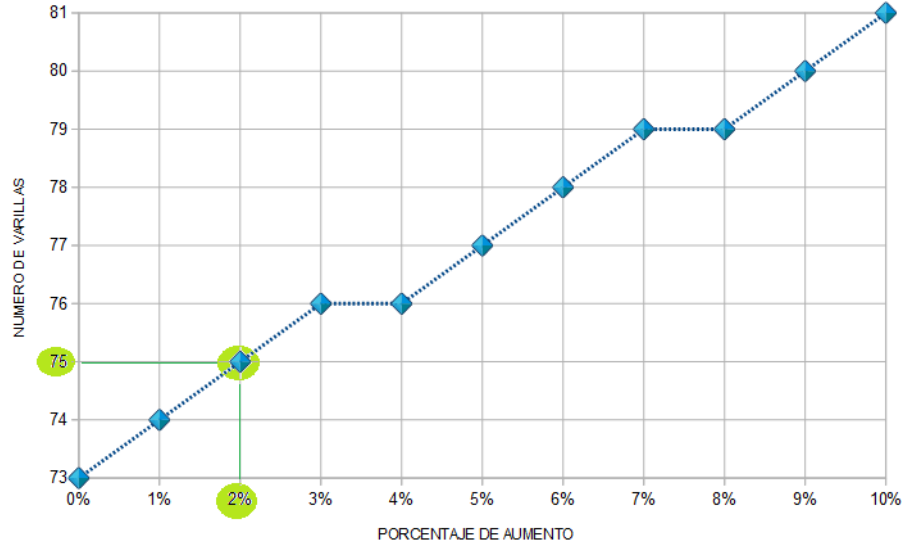
Optimización planilla de aceros de losas $\Phi 12$ mm casa Helio

```
Total de varillas de 12.0 metros a utilizar: 75 varillas
*****
Desperdicio total: 24.439999999999999 metros

Cortar las varilla de 12.0 metros de la siguiente forma:
-----
P_Corte1: 6 Varillas: (1.86, 10.14) m.
P_Corte2: 1 Varillas: (1.86, 1.86, 8.24) m.
P_Corte3: 12 Varillas: (1.86, 1.86, 1.86, 3.04, 3.34) m.
P_Corte4: 4 Varillas: (3.04, 3.04, 5.74) m.
P_Corte5: 10 Varillas: (3.04, 8.24) m.
P_Corte6: 2 Varillas: (1.86, 1.86, 1.86, 1.86, 1.86, 1.86) m.
P_Corte7: 12 Varillas: (1.66, 10.34) m.
P_Corte8: 6 Varillas: (3.44, 8.44) m.
P_Corte9: 6 Varillas: (2.04, 3.14, 3.14, 3.44) m.
P_Corte10: 9 Varillas: (3.14, 8.44) m.
P_Corte11: 5 Varillas: (2.04, 2.04, 7.24) m.
P_Corte12: 1 Varillas: (3.44, 7.24) m.
P_Corte13: 1 Varillas: (10.34,) m.

Desperdicios por corte:
-----
P_Corte 1: 6 Pedazos de: 0.00 m.
P_Corte 2: 1 Pedazos de: 0.04 m.
P_Corte 3: 12 Pedazos de: 0.04 m.
P_Corte 4: 4 Pedazos de: 0.18 m.
P_Corte 5: 10 Pedazos de: 0.72 m.
P_Corte 6: 2 Pedazos de: 0.84 m.
P_Corte 7: 12 Pedazos de: 0.00 m.
P_Corte 8: 6 Pedazos de: 0.12 m.
P_Corte 9: 6 Pedazos de: 0.24 m.
P_Corte 10: 9 Pedazos de: 0.42 m.
P_Corte 11: 5 Pedazos de: 0.68 m.
P_Corte 12: 1 Pedazos de: 1.32 m.
P_Corte 13: 1 Pedazos de: 1.66 m.
```

Figura 23
Porcentaje óptimo de estimación $\Phi 12$ mm casa Helio



Nota. Figura generada por el autor

Optimización planilla de gradas casa Helio.

La optimización de la planilla de gradas tiene 8 marcas con dimensiones menores a 4 metros, como se observa en la Tabla 9.

Tabla 9
Planilla de aceros de losas casa Helio

PLANILLA DE GRADAS														
MARCA	\emptyset	TIP O	DIMENSIONES (m)						Cantidad	Dimensión (m)	Longitud (m)	Peso Específico (Kg/m)	Peso (kg)	
			a	b	c	d	e	f						
Acero	100	10	C	0,17	0,9	0,17				52	1,24	64,48	0,617	39,78
	101	10	C	0,17	1,9	0,17				10	2,24	22,4	0,617	13,82
	102	10	G	0,2	0,15	0,4	2,05	0,53	0,55	6	3,88	23,28	0,617	14,36
	103	10	J	0,35	2,11	0,2				6	2,66	15,96	0,617	9,85
	104	10	K	0,3	0,65	0,2			0,15	12	1,3	13,8	0,617	8,51
	105	10	C	0,4	1,15	0,15				12	1,7	20,4	0,617	12,59
	106	10	J	0,6	2,2	0,2				6	3	18	0,617	11,11
	107	10	J	0,3	0,95	0,15				12	1,4	16,8	0,617	10,37
	109	10	J	0,87	2,2	0,2				6	3,27	19,62	0,617	12,11

Cuando las dimensiones de las marcas son menores a los 4 metros y el número de marcas es mayor a 6 se tiene desperdicios de acero ínfimos. Es el caso de acuerdo a la optimización

ejecutada para las gradas se requiere 19 varillas $\Phi 120$ mm, se tiene 8 patrones de corte y un desperdicio de 0.15 metros.

Figura 24
Optimización planilla de gradas $\Phi 10$ mm casa Helio

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:

OptiCorte
OPTIMIZADO CORTE DE VARILLAS

Luis Amaya V.		PUCETEC	
longitud(m) M100	1.24	Cantidad M100	52
longitud(m) M101	2.24	Cantidad M101	10
longitud(m) M102	3.88	Cantidad M102	6
longitud(m) M103	2.66	Cantidad M103	6
longitud(m) M104	1.3	Cantidad M104	12
longitud(m) M105	1.7	Cantidad M105	12
longitud(m) M106	3	Cantidad M106	6
longitud(m) M107	1.4	Cantidad M107	12
longitud(m) M108	3.27	Cantidad M108	4

PUCE TSC

Total de varillas de 12.0 metros a utilizar: 19 varillas
 Desperdicio total: 0.15999999999999966 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

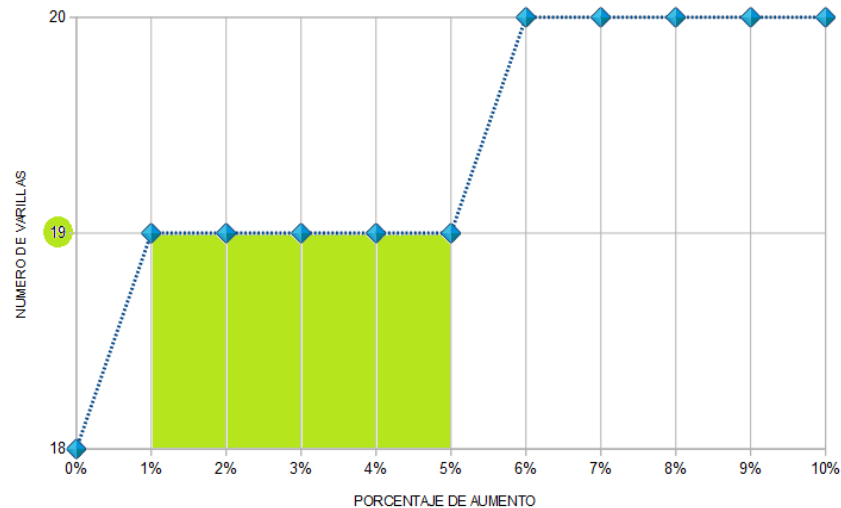
E_Corte1: 3 Varillas: (1.24, 1.24, 2.24, 3.88, 1.7, 1.7) m.
 E_Corte2: 6 Varillas: (1.24, 1.24, 1.24, 1.24, 1.24, 3.0, 1.4, 1.4) m.
 E_Corte3: 1 Varillas: (1.3, 1.3, 1.3, 1.3, 1.7, 1.7, 1.7, 1.7) m.
 E_Corte4: 3 Varillas: (2.24, 3.88, 1.3, 1.3, 3.27) m.
 E_Corte5: 1 Varillas: (2.66, 2.66, 1.7, 1.7, 3.27) m.
 E_Corte6: 2 Varillas: (1.24, 1.24, 1.24, 1.24, 1.24, 2.24, 2.24, 1.3) m.
 E_Corte7: 2 Varillas: (1.24, 1.24, 1.24, 1.24, 2.66, 2.66, 1.7) m.
 E_Corte8: 1 Varillas: (1.24, 1.24, 1.24, 1.7, 3.27, 3.27) m.

Desperdicios por corte:

E_Corte 1: 3 Pedazos de: 0.00 m.
 E_Corte 2: 6 Pedazos de: 0.00 m.
 E_Corte 3: 1 Pedazos de: 0.00 m.
 E_Corte 4: 3 Pedazos de: 0.01 m.
 E_Corte 5: 1 Pedazos de: 0.01 m.
 E_Corte 6: 2 Pedazos de: 0.02 m.
 E_Corte 7: 2 Pedazos de: 0.02 m.
 E_Corte 8: 1 Pedazos de: 0.04 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Figura 25
Porcentaje óptimo de estimación $\Phi 10$ mm casa Helio



Nota. Figura generada por el autor

Realizando un análisis económico de los resultados obtenidos de la optimización del corte de acero para la casa Helio, con el uso del programa OptiCorte se determina que existió un ahorro del 3.48% de acero esto representa 89.7 kg, considerando un costo de 2.8 dólares por kilo de acero se tiene un ahorro de 246.12 dólares como se resume en la Tabla 10.

Tabla 10

Tabulación de resultados obtenidos del análisis de la casa Helio

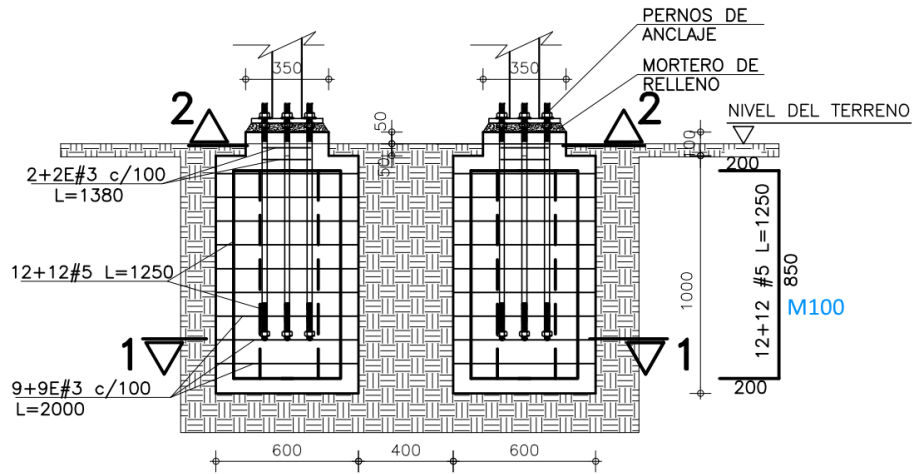
	SIN OPTIMIZAR			ÓPTIMO	
	NO. VARILLAS	PESO (Kg)		NO. VARILLAS	PESO (Kg)
Φ8	50	237		47	222,78
Φ10	19	140,676		19	140,676
Φ12	77	820,512		75	799,2
Φ16	51	965,736		49	927,864
Φ14	25	362,4		24	347,904
	TOTAL (kg)	2526,324		TOTAL (kg)	2438,424

Ahorro	87,9 kg
Ahorro	3,48%
Costo 1Kg Acero FY=4200	\$ 2,80
Ahorro USD	\$ 246,12

Cimentación de hormigón armado para letrero publicitario.

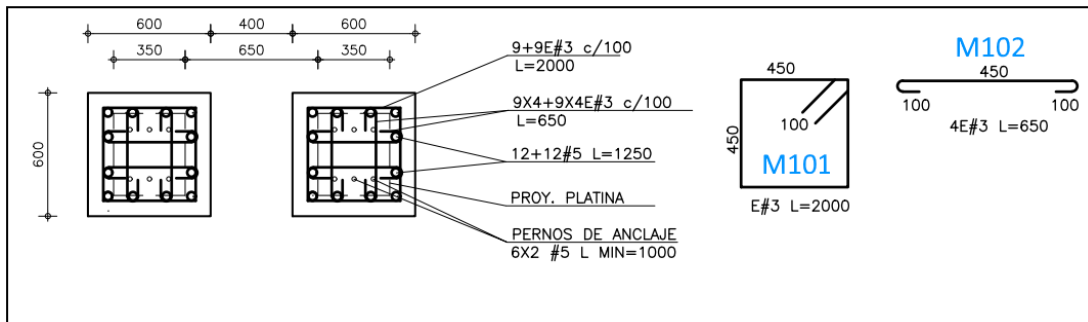
Como segundo ejemplo práctico de la aplicación del programa OptiCorte, se analiza la planilla de aceros de una cimentación de hormigón armado para un letrero publicitario de una estación de servicio de combustible. En este ejemplo se plantea 2 escenarios. El primer escenario se considera varillas de 6 metros de longitud bajo esta precisa se corre las optimizaciones de corte de las varillas. El segundo escenario se considera varillas de 12 metros, se corre las optimizaciones de corte, al final se analiza en que escenario éxito más desperdicio de acero. En las Figuras 26, 27, 28 se muestra el plano estructural con el detalle de las marcas.

Figura 26
Vista en corte vertical



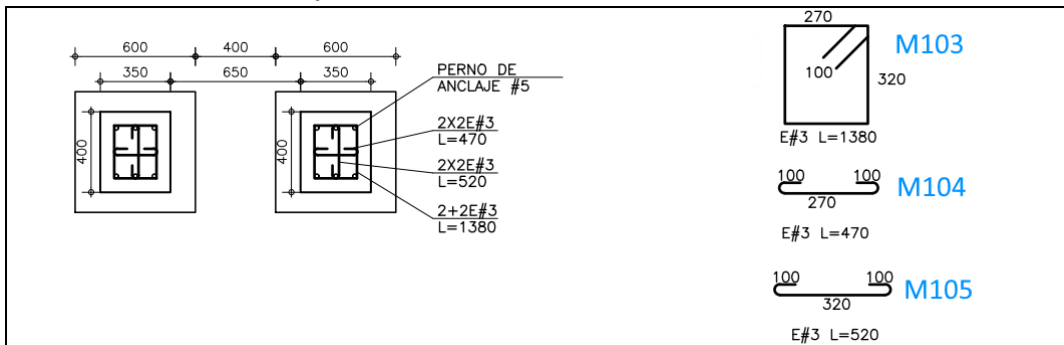
Nota. Figura generada por el autor

Figura 27
Vista en corte horizontal 1-1'



Nota. Figura generada por el autor

Figura 28
Vista en corte horizontal 2-2'



Nota. Figura generada por el autor

Para el primer caso se considera varillas de 6 metros, de acuerdo a la planilla de aceros se cuenta una sola marca para varillas de 12mm. Y se tiene 5 marcas con diferentes dimensiones para varillas de 8mm.

Tabla 11
Planilla de aceros cimentación letrero publicitario

CUADRO DE CIMENTACIÓN LETRERO PUBLICITARIO														
MARCA	Ø	TIPO	DIMENSIONES (m)						Canti- dad	Dimensión (m)	Longitud (m)	Peso Específico (Kg/m)	Peso (kg)	
			a	b	c	d	e	f						
Acero	100	12	C	0,2	0,85	0,2				12	1,25	15	0,617	9,26
	101	8	G	0,45	0,45	0,45	0,45	0,1	0,1	9	2	18	0,617	11,11
	102	8	J	0,1	0,45	0,1				36	0,65	23,4	0,617	14,44
	103	8	G	0,27	0,32	0,27	0,32	0,1	0,1	3	1,38	4,14	0,617	2,55
	104	8	J	0,1	0,27	0,1			0,15	6	0,62	2,82	0,617	1,74
	105	8	J	0,1	0,32	0,1				6	0,52	3,12	0,617	1,93
													Total	41,02

Con los datos de la Tabla 11 se procederá a introducir en el programa OptiCorte para analizar la forma de los cortes con varillas de 6 metros posterior se ejecutará la misma practica para varillas de 12m, finalmente se determinará donde se obtuvo mayor porcentaje de desperdicio de acero de refuerzo.

Figura 29
Optimización planilla de cimentación Ø12 longitud 12 metros

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:

Luis Amaya V. PUCETEC

longitud(m) M100 Cantidad M100

Total de varillas de 12.0 metros a utilizar: 2 varillas

Desperdicio total: 9 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

P_Corte1: 1 Varillas: (1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25) m.
P_Corte2: 1 Varillas: (1.25, 1.25, 1.25) m.

Desperdicios por corte:

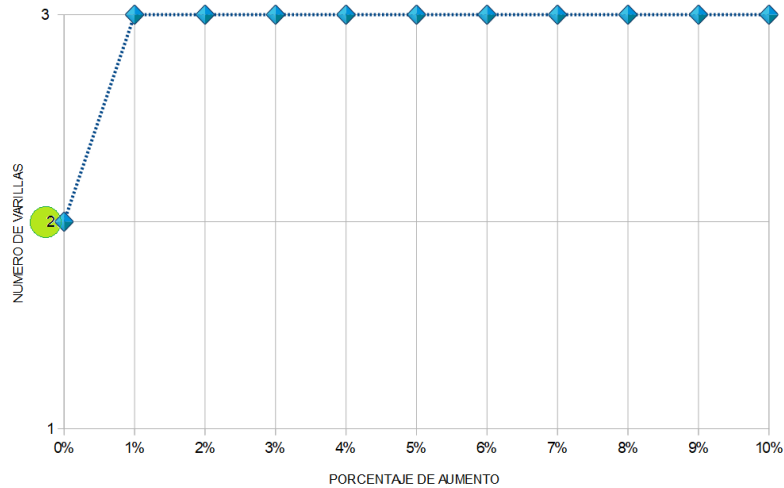
P_Corte 1: 1 Pedazos de: 0.75 m.
P_Corte 2: 1 Pedazos de: 8.25 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Para la marca 100 se requiere 2 varillas de 12 mm. y existe un sobrante de 9m, en este caso no se puede hablar de un desperdicio porque es una longitud considerable de sobrante. Por

ende, si se estima a partir del 1% se estaría adquiriendo una varilla extra, esto se muestra en la Figura 30

Figura 30
Porcentaje óptimo de estimación $\Phi 12$ longitud 12 metros



Nota. Figura generada por el autor

Con los datos de la Tabla No. 11 se procederá a introducir en el programa OptiCorte para analizar la forma de los cortes con varillas de 6 metros para la marca 100.

Figura 31
Optimización planilla de cimentación $\Phi 12$ longitud 6 metros

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:

Luis Amaya V. PUCETEC

longitud(m) M100 Cantidad M100

Total de varillas de 6.0 metros a utilizar: 3 varillas

Desperdicio total: 3.0 metros

Cortar las varilla de 6.0 metros de la siguiente forma:

P_Cortel: 3 Varillas: (1.25, 1.25, 1.25, 1.25) m.

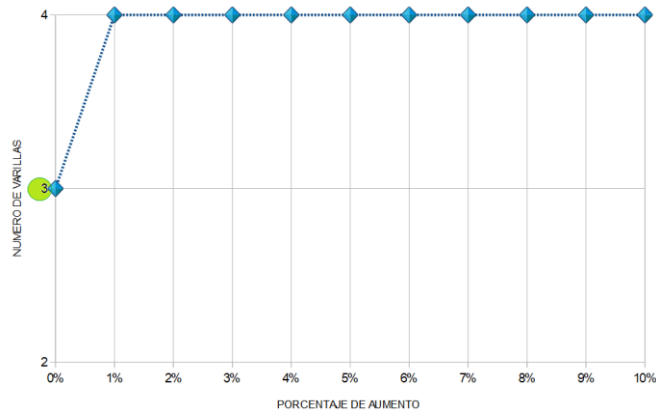
Desperdicios por corte:

P_Corte 1: 3 Pedazos de: 1.00 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Para la marca 100 se requiere 3 varillas de 6 mm. y existe un sobrante de 3, al usar varillas de 6 metros el sobrante es menor. En este caso si se estimas a partir del 1% se estaría adquiriendo una varilla extra, esto se muestra en la Figura 32

Figura 32
Porcentaje óptimo de estimación $\Phi 12$ longitud 6 metros




Nota. Figura generada por el autor

Ahora se analiza las marcas 101 a 105 para varillas de un diámetro de 8mm de sección y una longitud de 12 metros.


Figura 33
Optimización planilla de cimentación $\Phi 8$ longitud 12 metros

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas: Ingresar



Luis Amaya V.



PUCETEC

longitud(m) M100	<input type="text" value="2"/>	Cantidad M100	<input type="text" value="9"/>
longitud(m) M101	<input type="text" value="0.65"/>	Cantidad M101	<input type="text" value="36"/>
longitud(m) M102	<input type="text" value="1.38"/>	Cantidad M102	<input type="text" value="3"/>
longitud(m) M103	<input type="text" value="0.62"/>	Cantidad M103	<input type="text" value="6"/>
longitud(m) M104	<input type="text" value="0.52"/>	Cantidad M104	<input type="text" value="6"/>

Total de varillas de 12.0 metros a utilizar: 5 varillas

Desperdicio total: 0.010000000000005116 metros

Cortar las varilla de 12.0 metros de la siguiente forma:

P_Corte1: 1 Varillas: (2.0, 2.0, 2.0, 2.0, 2.0, 2.0) m.

P_Corte2: 1 Varillas: (2.0, 2.0, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.62, 0.62, 0.52, 0.52, 0.52) m.

P_Corte3: 2 Varillas: (0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.62, 0.62, 0.52, 0.52) m.

Desperdicios por corte:

P_Corte 1: 1 Pedazos de: 0.00 m.

P_Corte 2: 1 Pedazos de: 0.00 m.

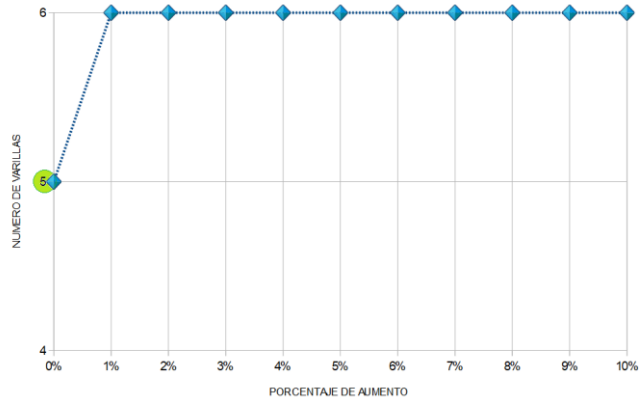
P_Corte 3: 2 Pedazos de: 0.00 m.

P_Corte 4: 1 Pedazos de: 0.01 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Efectuada la corrida se observa que no hay sobrantes. Por lo tanto, la estimación mayor igual al 1% será ineficiente por que se aumentará una varilla de 12 metros de manera innecesaria.

Figura 34
Porcentaje óptimo de estimación $\Phi 8$ longitud 12 metros



Nota. Figura generada por el autor

Ahora se analiza las marcas 101 a 105 para varillas de un diámetro de 8mm de sección y una longitud de 6 metros.

Figura 35
Optimización planilla de cimentación $\Phi 8$ longitud 6 metros

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas: Ingresar

Luis Amaya V.

longitud(m) M100

longitud(m) M101

longitud(m) M102

longitud(m) M103

longitud(m) M104

PUCETEC

Cantidad M100

Cantidad M101

Cantidad M102

Cantidad M103

Cantidad M104

Total de varillas de 6.0 metros a utilizar: 9 varillas

Desperdicio total: 0.38000000000000167 metros

Cortar las varilla de 6.0 metros de la siguiente forma:

P_Corte1: 3 Varillas: (2.0, 2.0, 2.0) m.

P_Corte2: 1 Varillas: (0.62, 0.62, 0.62, 0.62, 0.62, 0.62, 0.62, 0.62, 0.52, 0.52) m.

P_Corte3: 1 Varillas: (0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.52, 0.52, 0.52, 0.52) m.

P_Corte4: 3 Varillas: (0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 1.38) m.

P_Corte5: 1 Varillas: (0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65, 0.65) m.

Desperdicios por corte:

P_Corte 1: 3 Pedazos de: 0.00 m.

P_Corte 2: 1 Pedazos de: 0.00 m.

P_Corte 3: 1 Pedazos de: 0.02 m.

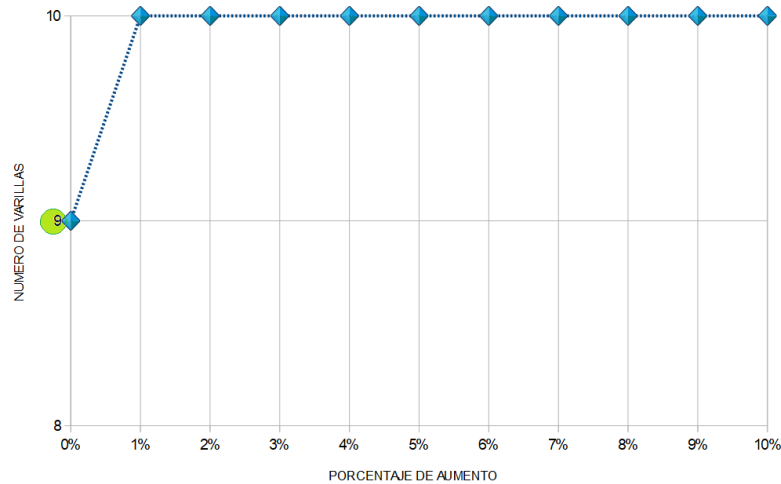
P_Corte 4: 3 Pedazos de: 0.07 m.

P_Corte 5: 1 Pedazos de: 0.15 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Efectuada la corrida se observa que hay un sobrante de 38 centímetros en total sumando varias longitudes. Por lo tanto, la estimación mayor igual al 1% será ineficiente por que se aumentará una varilla de 6 metros de manera innecesaria.

Figura 36
Porcentaje óptimo de estimación $\Phi 8$ longitud 6 metros



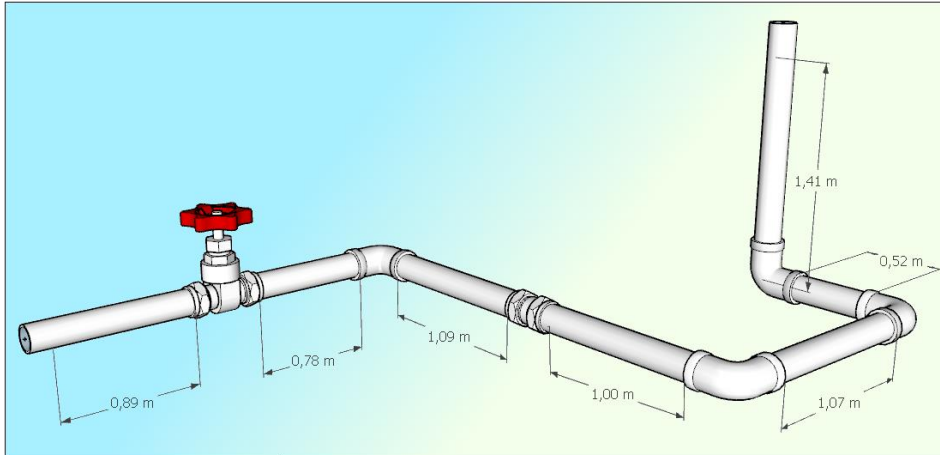
Nota. Figura generada por el autor

Analizando los dos casos de estudio se determina que, con las varillas de 6 metros existe menos desperdicio, se llega a esta conclusión por el número de marcas y la longitud del presente caso. Podrá ser para otros casos que convenga las varillas de 12 metros, todo depende de la longitud de las marcas y el número de las mismas para poder determinar que longitud de varilla (6m o 12m) es la ideal para el caso de estudio.

Optimización de tubería de ½” para agua potable en una vivienda.

El programa OptiCorte también puede ser empleado para optimizar los cortes de tuberías hidrosanitarias en una vivienda, en la Figura 37 se observa un diagrama isométrico de tuberías de agua potable, las longitudes de las tuberías serán ingresadas al programa OptiCorte para obtener el listado de cortes y la cantidad de tubería de 3 metros necesaria.

Figura 37
Diagrama isométrico de tuberías de agua potables de 1/2"




Nota. Figura generada por el autor

Al ser un ejemplo teórico se toma asume que en el mercado existe tuberías de longitud de 3 metros. Para el diagrama isométrico presentado se tiene 7 longitudes (marcas) de las cuales se requiere una unidad de cada una.

Figura 38
Optimización de tuberías de agua potable de longitud 3 metros

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:



	Luis Amaya V.	PUCETEC
longitud(m) M100	<input type="text" value="0.89"/>	Cantidad M100 <input type="text" value="1"/>
longitud(m) M101	<input type="text" value="0.78"/>	Cantidad M101 <input type="text" value="1"/>
longitud(m) M102	<input type="text" value="1.09"/>	Cantidad M102 <input type="text" value="1"/>
longitud(m) M103	<input type="text" value="1"/>	Cantidad M103 <input type="text" value="1"/>
longitud(m) M104	<input type="text" value="1.07"/>	Cantidad M104 <input type="text" value="1"/>
longitud(m) M105	<input type="text" value="0.52"/>	Cantidad M105 <input type="text" value="1"/>
longitud(m) M106	<input type="text" value="1.41"/>	Cantidad M106 <input type="text" value="1"/>

Total de varillas de 3.0 metros a utilizar: 3 varillas
 Desperdicio total: 0.129999999999999945 metros

Cortar las varilla de 3.0 metros de la siguiente forma:

P_Corte1: 1 Varillas: (1.07, 0.52, 1.41) m.
 P_Corte2: 1 Varillas: (0.78, 1.09, 1.07) m.
 P_Corte3: 1 Varillas: (0.89, 1.0, 0.52, 0.52) m.

Desperdicios por corte:

P_Corte 1: 1 Pedazos de: 0.00 m.
 P_Corte 2: 1 Pedazos de: 0.06 m.
 P_Corte 3: 1 Pedazos de: 0.07 m.

Nota. Figura generada por el autor utilizando el software OptiCorte.

Efectuada la corrida se obtiene como resultados que se requiere 3 tubos de 3 metros y que va a existir un desperdicio de 12 centímetros. El programa es muy versátil y la optimización se adapta a la medida que se le ingrese, de esta misma manera se puede optimizar las tuberías sanitarias.

Conclusiones

El programa OptiCorte demostró ser eficaz en diversos escenarios, logrando reducciones significativas. Por ejemplo, en la planilla de columnas de la Casa Helio, el desperdicio de varillas $\Phi 16\text{mm}$ se redujo de 23,19 metros a 8,2 metros, optimizando el uso del material mediante criterios de traslape.

En proyectos con características específicas, como el análisis entre varillas de 6 y 12 metros, se evidenció que la optimización depende de la distribución de las marcas. Este enfoque adaptable permite ajustar la estrategia según las condiciones del proyecto.

El programa es eficiente en la mayoría de los casos, con tiempos de procesamiento aceptables. Aunque en escenarios complejos, como las planillas de losas con múltiples marcas, el tiempo de cómputo puede incrementarse, los resultados siguen siendo robustos.

Además de acero, OptiCorte mostró su capacidad para optimizar otros materiales, como tuberías hidrosanitarias, ampliando su aplicabilidad en la industria de la construcción.

La herramienta ayuda a evitar compras innecesarias al ajustar los porcentajes de desperdicio.

Por ejemplo, valores superiores al 1% pueden derivar en costos adicionales significativos.

En ciertos casos, como cadenas y vigas strap con marcas uniformes, la optimización no es aplicable, siendo más adecuado el cálculo manual. Esto subraya la importancia de identificar cuándo el programa agrega valor.

La reducción del desperdicio no solo implica beneficios económicos, sino que también contribuye a prácticas más sostenibles. Este proyecto aporta herramientas que alinean la eficiencia operativa con el cuidado del medio ambiente, optimizando recursos críticos para la industria.

Estas conclusiones consolidan el valor de OptiCorte como una solución innovadora y práctica, posicionándolo como una herramienta de referencia para la planificación y gestión eficiente de materiales en construcción.

Recomendaciones.

Es importante que los constructores y diseñadores se capaciten en el manejo del programa para maximizar su eficacia y adaptar los criterios de optimización según las necesidades específicas del proyecto.

Se debe priorizar el uso del programa en planillas con múltiples marcas y cortes complejos, como losas y gradas, donde el potencial de reducción de desperdicio es mayor.

Extender el uso del programa a otros elementos, como tuberías y perfiles estructurales, puede generar ahorros adicionales y mejorar la sostenibilidad del proyecto.

Realizar validaciones periódicas de los resultados y actualizar el algoritmo del programa para incorporar nuevas funcionalidades, como la consideración de costos asociados al traslape y la reutilización de sobrantes.

Llevar un registro detallado de los resultados de optimización en diversos proyectos permitirá identificar patrones y mejorar la toma de decisiones en futuros usos.

Referencias bibliográficas

- Coelho, K. R., Cherri, A. C., Baptista, E. C., Jabbour, C. J. C., & Soler, E. M. (2017). Sustainable operations: The cutting stock problem with usable leftovers from a sustainable perspective. *Journal of Cleaner Production*, 167: 545–552. <https://doi.org/10.1016/j.jclepro.2017.08.153>.
- Kim, S. K., Hong, W. K., & Joo, J. K. (2004). Algorithms for Reducing the Waste Rate of Reinforcement Bars. *Journal of Asian Architecture and Building Engineering*, 3(1),17-23. <https://doi.org/10.3130/jaabe.3.17>.
- Kwon, K., Kim, D., & Kim, S. (2021). Cutting waste minimization of rebar for sustainable structural work: A systematic literature review. *Sustainability*, 13(11),5929. <https://doi.org/10.3390/su13115929>.
- Lee, D., Son, S., Kim, D., & Kim, S. (2020). Special-length-priority algorithm to minimize reinforcing bar-cutting waste for sustainable construction. *Sustainability*, 12(15),5950. <https://doi.org/10.3390/SU12155950>.
- Ma, Z., Zhao, Q., Cang, T., Li, Z., Zhu, Y., & Hei, X. (2023). An Intelligent Optimization Method of Reinforcing Bar Cutting for Construction Site. *CMES - Computer Modeling in Engineering and Sciences*, 134(1), 637–655. <https://doi.org/10.32604/cmes.2022.021216>.
- Machado, A. A., Zayatz, J. C., Da Silva, M. M., Melluzzi Neto, G., Leal, G. C. L., y Palma Lima, R. H. (2020). Aluminum bar cutting optimization for door and window manufacturing. *Dyna*, 87(212), 155–162. <https://doi.org/10.15446/dyna.v87n212.82636>.
- Miller, S. A., Horvath, A., & Monteiro, P. J. M. (2016). Readily implementable techniques can cut annual CO2 emissions from the production of concrete by over 20%. *Environmental Research Letters*, 11(7). <https://doi.org/10.1088/1748-9326/11/7/074029>
- Ochoa, G. (2014). *Optimización de corte de varillas de acero de construcción* (Tesis de pregrado). Universidad de Cuenca, Cuenca, Ecuador. <http://dspace.ucuenca.edu.ec/bitstream/123456789/19839/1/tesis.pdf>

- Patiño, J. (2018). *Estudio comparativo en el costo de metro cuadrado de vivienda de interes social, entre el uso de estructuras de hormigón armado y estructura metálica*. <https://dspace.uazuay.edu.ec/bitstream/datos/8498/1/14216.pdf>
- Ren, K., Jia, L., Huang, J., & Wu, M. (2023). Research on cutting stock optimization of rebar engineering based on building information modeling and an improved particle swarm optimization algorithm. *Developments in the Built Environment*, 13(December 2022), 100121. <https://doi.org/10.1016/j.dibe.2023.100121>.
- Wang, W., Shi, T., Shi, L., & Zhao, Q. (2018). Integrated optimisation on flow-shop production with cutting stock. *International Journal of Production Research*, 57(1), 1–17. <https://doi.org/10.1080/00207543.2018.1556823>.
- Widjaja, D. D., & Kim, S. (2023). Reducing Rebar Cutting Waste and Rebar Usage of Beams: A Two-Stage Optimization Algorithm. *Buildings*, 13(9). <https://doi.org/10.3390/buildings13092279>

Anexos.

Anexo 1.

Manual de usa del programa OptiCorte:

Una vez que el programa se ha instalado dar doble clic en el ícono que se crea por defecto en el escritorio, tal como se ilustra en la Figura 39

Figura 39

Icono de OptiCorte luego de instalar

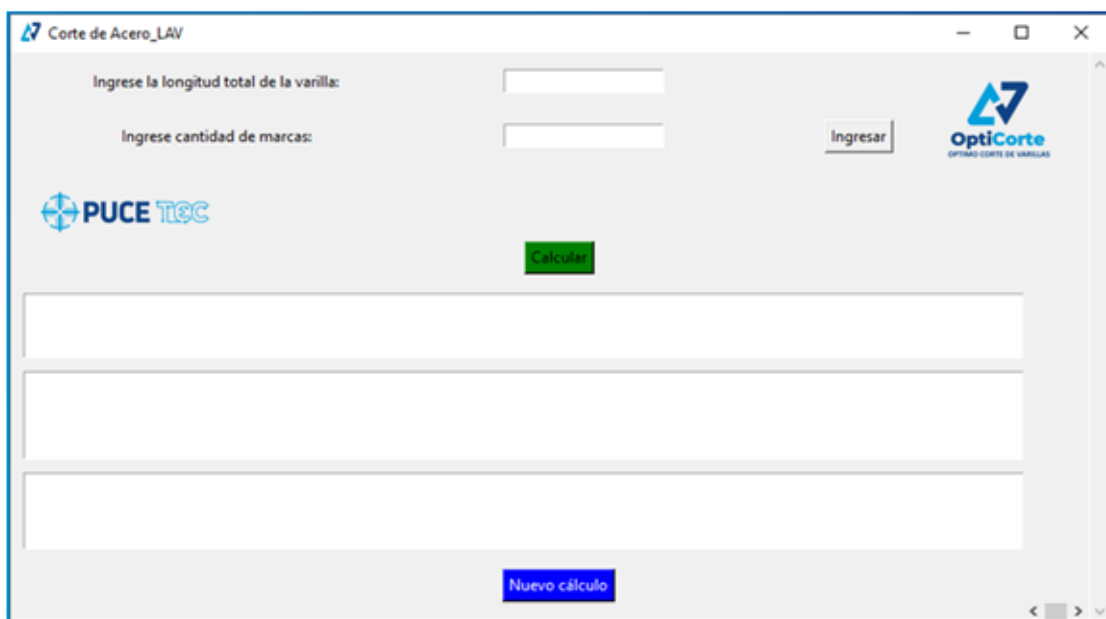


Nota. Figura generada por el autor

Se abrirá el programa como se muestra en la Figura 39.

Figura 40

Pantalla inicial de OptiCorte



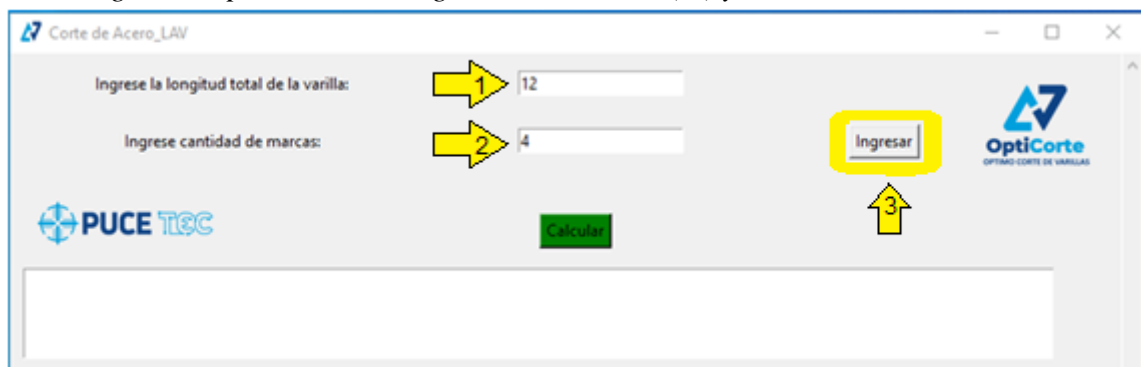
Nota. Figura generada por el autor utilizando el software OptiCorte

Los primeros parámetros a ingresar en el programa son:

1. Longitud de la varilla en metros (en Ecuador comercialmente se tiene varillas de 12 metros de longitud).
2. Ingresar el número de marcas o longitudes requerida a cortar (este dato se obtiene de la planilla de aceros que se encuentra en el plano estructural).
3. Una vez ingresados los datos se presiona el botón ingresar.

Figura 41

Ingreso de parámetros, longitud de la varilla (m) y cantidad de marcas



Nota. Figura generada por el autor utilizando el software OptiCorte

Luego de dar clic en el botón ingresar, aparece una pantalla con las cajas de texto de longitud y cantidad como se muestra en la Figura 42:

4. Ingresar las longitudes de cada marca en metros.
5. Ingresar la cantidad de cada marca en función de la planilla de aceros.

Figura 42

Ingreso de parámetros, longitudes a cortar (m) y cantidades de marcas

Corte de Acero_LAV

Ingrese la longitud total de la varilla: 12

Ingrese cantidad de marcas: 4 Ingresar

PUCE TEC

Luis Amaya V. PUCETEC

longitud(m)	M100	Cantidad	M100
longitud(m)	M101	Cantidad	M101
longitud(m)	M102	Cantidad	M102
longitud(m)	M103	Cantidad	M103

Calcular

Nota. Figura generada por el autor utilizando el software OptiCorte.

Es importante indicar que el programa NO permite ingresar datos de longitudes mayores a la longitud de la varilla, de acuerdo a la Figura 43, si se ingresa una longitud superior a 12 metros el programa alertará un erro.

- Una vez que se han ingresado los datos correctamente de sebera dar clic en el botón calcular.

Figura 43

Inicio de cálculos del programa

Corte de Acero_LAV

Ingrese la longitud total de la varilla: 12

Ingrese cantidad de marcas: 4 Ingresar

PUCE TEC

Luis Amaya V. PUCETEC

longitud(m)	M100	1.5	Cantidad	M100	10
longitud(m)	M101	2	Cantidad	M101	15
longitud(m)	M102	3.5	Cantidad	M102	12
longitud(m)	M103	5	Cantidad	M103	20

Calcular

Nota. Figura generada por el autor utilizando el software OptiCorte.

El programa muestra los resultados de la optimización como se ilustra en la Figura 44

Figura 44
Resultados del programa

Ingrese la longitud total de la varilla:
 Ingrese cantidad de marcas:

PUCE TBC Luis Amaya V. PUCETEC

longitud(m) M100	Cantidad M100
<input type="text" value="1.5"/>	<input type="text" value="10"/>
<input type="text" value="2"/>	<input type="text" value="15"/>
<input type="text" value="3.5"/>	<input type="text" value="12"/>
<input type="text" value="5"/>	<input type="text" value="20"/>

```

Total de varillas de 12.0 metros a utilizar: 16 varillas
.....
Desperdicio total: 1.5 metros
-----
Cortar las varilla de 12.0 metros de la siguiente forma:
P_Corte1: 8 Varillas: (2.0, 5.0, 5.0) m.
P_Corte2: 3 Varillas: (3.5, 3.5, 5.0) m.
P_Corte3: 1 Varillas: (1.5, 2.0, 3.5, 5.0) m.
P_Corte4: 1 Varillas: (2.0, 2.0, 2.0, 2.0, 2.0) m.
P_Corte5: 3 Varillas: (1.5, 1.5, 1.5, 3.5, 3.5) m.
-----
Desperdicios por corte:
P_Corte 1: 8 Pedazos de: 0.00 m.
P_Corte 2: 3 Pedazos de: 0.00 m.
P_Corte 3: 1 Pedazos de: 0.00 m.
P_Corte 4: 1 Pedazos de: 0.00 m.
P_Corte 5: 3 Pedazos de: 0.50 m.
  
```

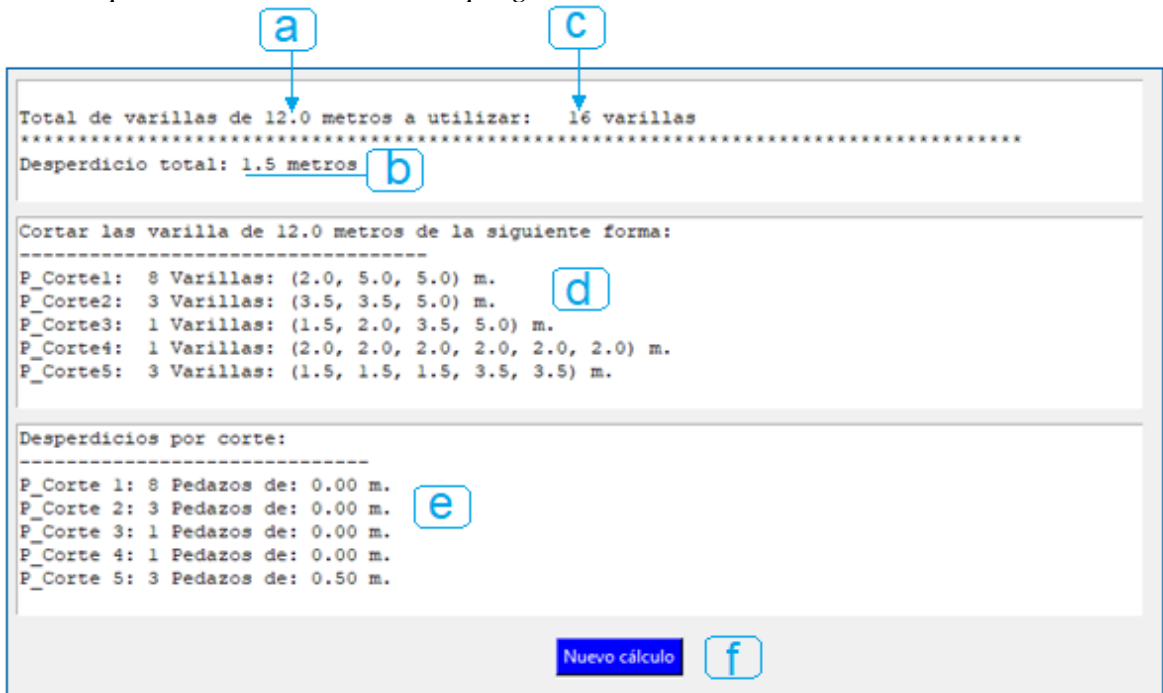
Nota. Figura generada por el autor utilizando el software OptiCorte.

Los resultados obtenidos se muestran en la Figura 45, y se interpretan de la siguiente manera:

- Se muestra la longitud de la varilla bajo análisis en este caso longitud total de la varilla 12 metros.
- Muestra la longitud total de desperdicios en metros
- Indica la cantidad de varillas de 12 metros necesarias para obtener las cantidades y las longitudes ingresada.
- Indica los patrones de corte, la forma en cómo se debe cortar las varillas. Para el caso de ejemplo expone que:
 - 8 varillas de deberán cortar en pesados de 2.0m, 5.0m, 5.0m.
 - 3 varillas de deberán cortar en pesados de 3.5m, 3.5m, 5.0m.
 - 1 varillas de deberán cortar en pesados de 1.5m, 2.0m, 3.5m, 5.0m.

- 1 varillas de deberán cortar 5 pesados de 2.0m.
- 3 varillas de deberán cortar en pesados de 1.5m, 1.5m, 1.5m, 3.5m, 3.5m.

Figura 45
Interpretación de resultados del programa



Nota. Figura generada por el autor utilizando el software OptiCorte.

- Se muestra la cantidad de desperdicio por corte, para el ejemplo en el patrón de corte numero 5 resultará como sobrante 3 pedazos de varilla de 0.5 metros.
- Si se desea realizar un nuevo cálculo deberá dar clic sobre el boto nuevo cálculo y el programa reseteará todos sus cálculos y mostrará la pantalla en blanco como se ilustra en la Figura 46.

Figura 46
Pantalla inicial para realizar un nuevo cálculo

Corte de Acero_LAV

Ingrese la longitud total de la varilla:

Ingrese cantidad de marcas:

Ingresar

OptiCorte
OPTIMO CORTE DE VARILLAS

PUCE TSC

Calcular

Nuevo cálculo

Nota. Figura generada por el autor utilizando el software OptiCorte.

Anexo 2.

Código del programa OptiCorte:

```
1   import tkinter as tk
2   from tkinter import messagebox
3   import itertools
4   import pulp
5   #-----
6   class VectorInputApp:
7       def __init__(self, root):
8           self.root = root
9           self.root.title("Corte de Acero_LAV")
10          self.root.geometry("850x850")
12          # Crear frame principal con scrollbars
13          self.main_frame = tk.Frame(root)
14          self.main_frame.pack(fill=tk.BOTH, expand=1)
16          self.canvas = tk.Canvas(self.main_frame)
17          self.canvas.pack(side=tk.LEFT,
18                          fill=tk.BOTH, expand=1)
19          self.scrollbar_y = tk.Scrollbar(self.main_frame, orient=tk.VERTICAL,
20                                         command=self.canvas.yview)
21          self.scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
22          self.scrollbar_x = tk.Scrollbar(self.main_frame, orient=tk.HORIZONTAL,
23                                         command=self.canvas.xview)
24          self.scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)
25          self.canvas.configure(yscrollcommand=self.scrollbar_y.set,
26                               xscrollcommand=self.scrollbar_x.set)
27          self.canvas.bind('<Configure>', lambda e:
28                          self.canvas.configure(scrollregion=self.canvas.bbox("all")))
29          #-----
30          # Frame dentro del canvas
31          self.second_frame = tk.Frame(self.canvas)
32          self.canvas.create_window((0, 0), window=self.second_frame,
33                                   anchor="nw")
34          #-----
35          # Entrada de datos
36          self.total_length_label = tk.Label(self.second_frame, text="Ingrese la
37                                          longitud total de la varilla:")
38          self.total_length_label.grid(row=0, column=0, padx=10, pady=10)
39          self.total_length_entry = tk.Entry(self.second_frame)
40          self.total_length_entry.grid(row=0, column=1, padx=10, pady=10)
41          self.intro_label = tk.Label(self.second_frame, text=
42                                     "Ingrese cantidad de marcas:")
43          self.intro_label.grid(row=1, column=0, padx=10, pady=10)
44          self.length_entry = tk.Entry(self.second_frame)
```

```

43     self.length_entry.grid(row=1, column=1, padx=10, pady=10)
44     self.submit_length_button = tk.Button(self.second_frame, text="Ingresar",
45         command=self.submit_length)
46     self.submit_length_button.grid(row=1, column=2, padx=10, pady=10)
47     self.entries_frame = tk.Frame(self.second_frame)
48     self.entries_frame.grid(row=2, column=0, columnspan=4, padx=10,
49         pady=10)
50     self.submit_vectors_button = tk.Button(self.second_frame, text=
51         "Calcular", command=self.submit_vectors, bg="green")
52     self.submit_vectors_button.grid(row=3, column=0, columnspan=4,
53         pady=10)
54     self.result_text = tk.Text(self.second_frame, wrap=tk.WORD)
55     self.result_text.grid(row=4, column=0, padx=10, pady=5, columnspan=4)
56
57     self.new_calculation_button = tk.Button(self.second_frame, text="Nuevo
58         cálculo", command=self.new_calculation, bg="blue", fg="white")
59     self.new_calculation_button.grid(row=5, column=0, columnspan=4,
60         pady=(10, 10))
61 #-----
62     def submit_length(self):
63         try:
64             self.vector_length = int(self.length_entry.get())
65             if self.vector_length <= 0:
66                 raise ValueError
67             self.show_vector_input()
68         except ValueError:
69             messagebox.showerror
70                 ("Error", "Por favor, ingrese un número entero positivo.")
71 #-----
72     def show_vector_input(self):
73         for widget in self.entries_frame.winfo_children():
74             widget.destroy()
75         self.vector1_entries = [ ]
76         self.vector2_entries = [ ]
77
78         for i in range(self.vector_length):
79             tk.Label(self.entries_frame, text=f"longitud(m) M{i+100}").grid(row=i,
80                 column=0, padx=5, pady=5)
81             entry1 = tk.Entry(self.entries_frame)
82             entry1.grid(row=i, column=1, padx=5, pady=5)
83             self.vector1_entries.append(entry1)
84
85             tk.Label(self.entries_frame, text=f"Cantidad M{i+100}").grid(row=i,
86                 column=2, padx=5, pady=5)
87             entry2 = tk.Entry(self.entries_frame)

```

```

84         entry2.grid(row=i, column=3, padx=5, pady=5)
85         self.vector2_entries.append(entry2)
86     #-----
87     def submit_vectors(self):
88         try:
89             self.vector1 = [float(entry.get()) for entry in self.vector1_entries]
90             self.vector2 = [float(entry.get()) for entry in self.vector2_entries]
91             self.run_optimization()
92         except ValueError:
93             messagebox.showerror
94                 ("Error", "Por favor, ingrese números válidos en todos los campos.")
95     #-----
96     def run_optimization(self):
97         try:
98             total_length = float(self.total_length_entry.get())
99         except ValueError:
100             messagebox.showerror("Error", "Por favor, ingrese una longitud total
101 válida.")
102         return
103     measurements = self.vector1
104     #-----
105     def generate_patterns(measurements, total_length):
106         patterns = []
107         for r in range(1, len(measurements) + 1):
108             for combo in itertools.combinations_with_replacement
109                 (measurements, r):
110                 if sum(combo) <= total_length:
111                     patterns.append(combo)
112         return patterns
113     #-----
114     def calculate_waste(pattern, total_length):
115         return total_length - sum(pattern)
116     #-----
117     patterns = generate_patterns(measurements, total_length)
118     patterns_with_waste = [(pattern, calculate_waste(pattern, total_length))
119                             for pattern in patterns]
120     patterns_with_waste.sort(key=lambda x: x[1])
121     #-----
122     demand = self.vector2
123     x = pulp.LpVariable.dicts("pattern", range(len(patterns_with_waste)),
124                               lowBound=0, cat='Integer')
125     problem = pulp.LpProblem("Minimize_Waste", pulp.LpMinimize)
126     problem += pulp.lpSum(x[i] for i in range(len(patterns_with_waste))),
127                          "TotalBars"
128     #-----

```

```

124     for j in range(len(measurements)):
125         problem += pulp.lpSum(pattern.count(measurements[j]) * x[i] for i,
            (pattern, waste) in enumerate(patterns_with_waste)) >= demand[j],
            f"Demand_{measurements[j]}m"
126     #-----
127     problem.solve()
128     #-----
129     result_text = f"Total de varillas de {total_length} metros a utilizar:
            {int(pulp.value(problem.objective))} varillas\n"
130     result_text += f"Desperdicio total: {sum(patterns_with_waste[i][1] *
            x[i].varValue for i in range(len(patterns_with_waste)):.2f} metros\n"
131     result_text += "Patrones de corte:\n"
132     #-----
133     for i, (pattern, waste) in enumerate(patterns_with_waste):
134         if x[i].varValue > 0:
136             result_text += f"Patrón {i+1}: {pattern}, usado {int(x[i].varValue)} veces,
            desperdicio: {waste:.2f} m\n"
138     self.display_results(result_text)
139     #-----
140     def display_results(self, result_text):
141         max_width = max(len(line) for line in result_text.split("\n"))
142         self.result_text.config(width=max(50, max_width),
            height=len(result_text.split("\n")) + 2)
143         self.result_text.delete(1.0, tk.END)
144         self.result_text.insert(tk.END, result_text)
145     #-----
146     def new_calculation(self):
147         self.total_length_entry.delete(0, tk.END)
148         self.length_entry.delete(0, tk.END)
149         for widget in self.entries_frame.winfo_children():
150             widget.destroy()
151         self.result_text.delete(1.0, tk.END)
152     #-----
153     if __name__ == "__main__":
154         root = tk.Tk()
155         app = VectorInputApp(root)
156         root.mainloop()

```