

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y
MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELÉCTRICA



TESIS

**DESARROLLO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN
DE COSTOS Y PRESUPUESTOS EN SISTEMAS DE DISTRIBUCIÓN, APLICADO
AL PROYECTO DE ELECTRIFICACIÓN CHALLABAMBA ALTA**

PRESENTADO POR:

Br. DEYVID HUAYTA CONTRERAS

Br. EDUARDO FRANCO ZUÑIGA NAVARRO

**PARA OPTAR AL TÍTULO PROFESIONAL
DE INGENIERO ELECTRICISTA**

ASESOR:

Dr. JOSÉ WILFREDO CALLASI QUISPE

CUSCO – PERÚ
2025



Universidad Nacional de San Antonio Abad del Cusco

INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el Asesor Dr. José Wilfredo Callas Quispe

..... quien aplica el software de detección de similitud al trabajo de investigación/tesis titulada:

Desarrollo e implementación de un software para la gestión de costos y presupuestos en sistemas de distribución aplicada al proyecto de electrificación Challabamba Alta

Presentado por: Br. Huanta Centenas David DNI N° 48355769;

presentado por: Br. Zuñiga Navarro Eduardo Franco DNI N°: 76882515

Para optar el título Profesional/Grado Académico de

Ingeniero Electricista

Informo que el trabajo de investigación ha sido sometido a revisión por 2 veces, mediante el Software de Similitud, conforme al Art. 6° del **Reglamento para Uso del Sistema Detección de Similitud en la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 3 %.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	<input checked="" type="checkbox"/>
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	<input type="checkbox"/>
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	<input type="checkbox"/>

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 05 de Enero de 2026



Firma

Post firma José Wilfredo Callas Quispe

Nro. de DNI 23812797

ORCID del Asesor 0000-0003-0714-4499

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema de Detección de Similitud: oid: 27259:543262829

VOLUMEN DE TESIS ROTULADO.pdf

 Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trnoid::27259:543262829

144 páginas

Fecha de entrega

27 dic 2025, 9:26 a.m. GMT-5

22.178 palabras

Fecha de descarga

27 dic 2025, 9:32 a.m. GMT-5

126.708 caracteres

Nombre del archivo

VOLUMEN DE TESIS ROTULADO.pdf

Tamaño del archivo

11,6 MB




3% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 20 palabras)

Fuentes principales

- 2%  Fuentes de Internet
- 0%  Publicaciones
- 2%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión



Caracteres reemplazados

64 caracteres sospechosos en N.º de páginas

Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

DEDICATORIA

A nuestras familias, cuyo amor incondicional y apoyo constante han sido la base sobre la que hemos construido este sueño. Su confianza en nosotros nos ha dado la fuerza para superar cada obstáculo y seguir adelante con determinación. Gracias por estar siempre a nuestro lado, celebrando nuestros logros y sosteniéndonos en los momentos difíciles.

A los amigos que han sido compañeros en esta etapa universitaria, con quienes compartimos desafíos, madrugadas de estudio y momentos inolvidables.

A nuestros docentes, por guiarnos con paciencia, entrega y sabiduría. Su enseñanza ha sido esencial para nuestra formación, brindándonos no solo conocimiento, sino también inspiración y herramientas para afrontar con confianza los desafíos profesionales que están por venir.

Los tesistas

PRESENTACIÓN

Señor Decano de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica de la Universidad Nacional de San Antonio Abad del Cusco, y Señores Docentes miembros del Jurado de la Escuela Profesional de Ingeniería Eléctrica:

En cumplimiento con las disposiciones del Reglamento de Grados y Títulos vigente, y con el propósito de optar al título profesional de Ingeniero Electricista, presentamos ante usted la tesis titulada: **“DESARROLLO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN DE COSTOS Y PRESUPUESTOS EN SISTEMAS DE DISTRIBUCIÓN, APLICADO AL PROYECTO DE ELECTRIFICACIÓN CHALLABAMBA ALTA”**.

Este trabajo se enfoca en el desarrollo y la implementación de una herramienta de software libre diseñada para optimizar la gestión de costos y presupuestos en el ámbito de los sistemas de distribución eléctrica. El objetivo principal es mejorar la eficiencia en la planificación y el control financiero de proyectos, tomando como caso de estudio el proyecto de electrificación de Challabamba Alta. La aplicación de este software no solo busca facilitar la administración de recursos, sino también potenciar la toma de decisiones económicas en proyectos de infraestructura eléctrica, promoviendo la accesibilidad y la colaboración a través de su naturaleza de código abierto.

INTRODUCCIÓN

En el contexto actual de la ingeniería, la eficiencia y precisión en la gestión de costos y presupuestos son pilares fundamentales para el éxito de cualquier proyecto. La industria de la ingeniería eléctrica, y en particular los proyectos de electrificación rural, no son la excepción. La complejidad inherente a la planificación y ejecución de sistemas de distribución eléctrica demanda herramientas robustas que permitan la optimización de recursos y la toma de decisiones informadas, minimizando los errores que pueden surgir de métodos manuales o de soluciones de software restrictivas.

El mercado peruano ofrece diversas herramientas especializadas en la elaboración de presupuestos para proyectos de ingeniería, tales como S10 y Delphin Express. Si bien estos programas brindan funcionalidades avanzadas y una interfaz intuitiva, su elevado costo de adquisición limita significativamente su accesibilidad, especialmente para profesionales emergentes y la comunidad estudiantil. Esta barrera económica obstaculiza el desarrollo de competencias cruciales en la gestión de costos y presupuestos para proyectos de distribución eléctrica, creando una necesidad insatisfecha en el ámbito académico y profesional.

Ante esta problemática, la presente tesis se centra en el Desarrollo e Implementación de un Software para la Gestión de Costos y Presupuestos en Sistemas de Distribución, Aplicado al Proyecto de Electrificación Challabamba Alta. Este trabajo de investigación, propone una solución basada en los principios del software libre.

RESUMEN

La presente tesis tiene como objetivo principal el diseño, desarrollo e implementación de un software de código abierto destinado a la gestión de costos y presupuestos en proyectos de sistemas de distribución eléctrica, con aplicación específica en el proyecto de electrificación del sector Challabamba Alta. La investigación parte del análisis teórico de la gestión de costos en obras eléctricas, considerando los procesos y normativas vigentes, y propone una solución tecnológica accesible y adaptable, especialmente orientada a profesionales y estudiantes del área de ingeniería.

El software ha sido desarrollado utilizando el lenguaje de programación Python, elegido por su versatilidad, facilidad de aprendizaje y una amplia comunidad de soporte, lo cual garantiza su sostenibilidad y mejora continua. La herramienta incluye una interfaz gráfica intuitiva que facilita el cálculo de presupuestos, así como la integración de una base de datos editable con insumos, montaje electromecánico y partidas, permitiendo su actualización según los requerimientos del usuario.

Asimismo, el análisis de costos unitarios y suministros ha sido desarrollado tomando como referencia los armados estándar utilizados por la empresa ELSE, lo que proporciona mayor precisión y aplicabilidad a contextos reales. El resultado de la investigación es una solución que contribuye significativamente a mejorar la eficiencia en la elaboración de presupuestos eléctricos, promoviendo además el uso de herramientas digitales de libre acceso en el ámbito de la electricidad.

Palabras Clave: Lenguaje de Programación, Software, Gestión de Costos y Presupuestos, Sistemas de Distribución.

GLOSARIO DE TÉRMINOS

- **API:** Interfaz de Programación de Aplicaciones. Permite que diferentes programas se comuniquen entre sí.
- **APU:** Análisis de Precios Unitarios. Desglose detallado del costo por unidad de una actividad o partida en un proyecto.
- **BT:** Baja Tensión. La parte del sistema de distribución eléctrica que lleva la energía directamente a los consumidores finales.
- **CSV:** Comma Separated Values (Valores Separados por Comas). Un formato de archivo de texto simple donde los datos se organizan en columnas separadas por comas.
- **DF:** Abreviatura común para DataFrame. Es la estructura de datos fundamental de la biblioteca Pandas en Python, similar a una hoja de cálculo o tabla de base de datos.
- **GUI:** Graphical User Interface (Interfaz Gráfica de Usuario). Es el entorno visual con el que los usuarios interactúan con un software, como ventanas, botones y menús.
- **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos. Una organización global que promueve la innovación tecnológica relacionada con la electricidad.
- **kV:** Kilovoltios. Una unidad de medida para la tensión eléctrica, equivalente a mil voltios.
- **SUM:** Suministro. Se refiere a la provisión o entrega de los materiales, equipos o recursos necesarios para la ejecución de un proyecto.
- **MON:** Montaje. Se refiere al proceso de ensamblar e instalar componentes y equipos en un proyecto, por ejemplo, el montaje de estructuras eléctricas.
- **MT:** Media Tensión. La sección del sistema de distribución eléctrica que opera con niveles de tensión intermedios, entre la alta tensión y la baja tensión.

- **Pandas:** Una potente biblioteca de Python esencial para la manipulación y análisis de datos. Facilita el trabajo con datos estructurados como tablas y series de tiempo.
- **Python:** Un lenguaje de programación de alto nivel, muy valorado por su sintaxis clara y legible, lo que lo hace fácil de aprender y usar. Es extremadamente versátil, ideal para el desarrollo web, análisis de datos, inteligencia artificial y, como en este proyecto, para la creación de software.
- **RP:** Redes Primarias. La parte del sistema de distribución eléctrica encargada de transportar la energía a niveles de media tensión desde las subestaciones hasta los puntos de transformación.
- **RS:** Redes Secundarias. Son las redes de distribución de energía eléctrica que operan a baja tensión, desde los transformadores de distribución hasta los puntos de conexión de los usuarios finales.
- **Tkinter:** La biblioteca estándar de Python para desarrollar interfaces gráficas de usuario (GUI) de escritorio. Permite construir aplicaciones con elementos visuales interactivos.
- **Ttkbootstrap:** Una extensión de Tkinter que mejora significativamente la apariencia y el estilo de las interfaces gráficas. Utiliza temas modernos inspirados en el framework Bootstrap para un diseño más atractivo y consistente.
- **Treeview:** Un widget (componente visual de GUI) proporcionado por Tkinter, que se utiliza para mostrar datos en una estructura jerárquica de árbol y, al mismo tiempo, en formato tabular, lo que es útil para organizar y presentar información compleja con subcategorías.
- **ELSE:** Empresa concesionaria Electro Sur Este S.A.A.

INDICE GENERAL

RESUMEN.....	viii
ÍNDICE DE FIGURAS.....	xv
ÍNDICE DE TABLAS.....	xvi
CAPÍTULO I: GENERALIDADES.....	17
1.1. Ámbito De Estudio	17
1.2. Problemática	18
1.2.1. Planteamiento del Problema	18
1.2.2. Formulación del Problema.....	19
1.3. Objetivos.....	19
1.3.1. Objetivo General.....	19
1.3.2. Objetivos Específicos.....	19
1.4. Justificación del Trabajo	20
1.5. Hipótesis	21
1.5.1. Hipótesis General.....	21
1.5.2. Hipótesis Específicas.	21
1.6. Operacionalización de Variables de estudio	22
1.6.1. Variable Independiente	22
1.6.2. Variable Dependiente.....	22
1.7. Alcances y Limitaciones	23
1.7.1. Alcances de la Investigación.....	23
1.7.2. Limitaciones de la Investigación	23
1.8. Metodología	24
1.9. Población y Muestra	24
1.9.1. Población.....	24

1.9.2. Muestra	24
1.10. Técnicas de Recolección de Datos.....	25
1.11. Procesamiento de Datos.....	25
1.12. Matriz de consistencia	25
MATRIZ DE CONSISTENCIA	26
TÍTULO: DESARROLLO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN DE COSTOS Y PRESUPUESTOS EN SISTEMAS DE DISTRIBUCIÓN, APLICADO AL PROYECTO DE ELECTRIFICACIÓN CHALLABAMBA ALTA ...	26
CAPÍTULO II: MARCO TEÓRICO.....	27
2.1. Antecedentes del trabajo	27
2.1.1. Antecedentes Locales.....	27
2.1.2. Antecedentes Nacionales	28
2.1.3. Antecedentes Internacionales.....	28
2.1.4. Programas Comerciales	31
2.2. Bases Teóricas	31
2.2.1. Presupuesto	31
2.2.2. Metrados	34
2.2.3. Costos directos	35
2.2.4. Costos indirectos.....	35
2.2.5. Clasificación de los costos indirectos	36
2.2.6. Aporte unitario de materiales.....	36
2.2.7. Estructura de un sistema de análisis unitarios.....	37
2.2.8. Proyectos de construcción.....	37
2.2.9. Sistema de distribución	38
2.2.10. Clasificación de las Redes de Distribución según su Ubicación	39

2.2.11. El lenguaje de programación Python	41
2.3. Base Conceptual	43
2.3.1. Media tensión eléctrica	43
2.3.2. Red de Baja Tensión	44
2.3.3. Presupuesto de obra	44
2.3.4. Software libre.....	44
2.3.5. Proyecto	44
2.4. Base Normativa.....	44
CAPÍTULO III: DISEÑO DE LA INTERFAZ GRÁFICA Y ESTRUCTURA	
FUNCIONAL DEL SOFTWARE	48
3.1. Introducción	48
3.2. Requisitos del Sistema	48
3.2.1. Requisitos Funcionales	48
3.2.2. Requisitos No Funcionales	49
3.3. Arquitectura del Software	49
3.3.1. Componentes principales	50
3.3.2. Librerías Clave.....	51
3.3.3. Flujo de Datos Principal.....	52
3.3.4. Diagrama de Clases.....	54
3.4. Flujo de Datos (Diagrama de secuencia)	55
3.5. Funcionalidades Clave	55
3.5.1. Gestión Dinámica de Partidas	55
3.5.2. Cálculo en Tiempo Real.....	56
3.5.3. Visualización de Datos	56
3.6. Integración con DIREDCAD.....	56

3.6.1. Descripción General del Sistema de Importación.....	56
3.6.2. Tipos de Archivos y Formatos Soportados	57
3.6.3. Funcionalidades de la Interfaz de Usuario	59
3.6.4. Proceso Detallado de Importación y Procesamiento	61
3.7. Limitaciones Identificadas	63
3.8. Conclusiones del capítulo	63
CAPÍTULO IV: IMPLEMENTACIÓN Y VALIDACIÓN DEL SOFTWARE EN EL PROYECTO CHALLABAMBA ALTA.....	65
4.1. Introducción	65
4.2. Aplicación del software al proyecto de electrificación Challabamba Alta.....	65
4.3. Descripción general del proyecto Challabamba Alta.....	65
4.4. Datos técnicos empleados para la validación	70
4.5. Resultados obtenidos con el software desarrollado	72
4.6. Presupuesto total generado automáticamente	72
4.7. Análisis comparativo con otras herramientas de cálculo de presupuesto	83
4.7.1. Comparación con Microsoft Excel	83
4.7.2. Comparación con S10	85
4.7.3. Análisis de Tiempos de Elaboración.....	87
4.7.4. Análisis de errores en la Elaboración.....	88
4.8. Conclusión del capítulo	88
CONCLUSIONES Y SUGERENCIAS.....	90
BIBLIOGRAFÍA.....	93
ANEXOS.....	95

ÍNDICE DE FIGURAS

Figura 1 Ubicación del proyecto Challabamba Alta	17
Figura 2 Formato de aplicación general.....	35
Figura 3 Diagrama de bloques del sistema de análisis de precios unitarios	37
Figura 4 Fases de un proyecto	38
Figura 5 Sistema Eléctrico Peruano	39
Figura 6 Arquitectura modular de tres capas	49
Figura 7 Ventana principal del programa	50
Figura 8 Sistema de pestañas dinámicas	50
Figura 9 Código para Cargar Datos de Partida	52
Figura 10 Código para el procesamiento	53
Figura 11 Diagrama de clases	54
Figura 12 Flujo de Datos.....	55
Figura 13 Importación de datos desde archivos generados en DirecCAD	57
Figura 14 Interfaz del boton Explorar.....	60
Figura 15 Interfaz del boton Actualizar metrado del Direcdad	60
Figura 16 Diagrama de Flujo del Proceso de Importación	62
Figura 14 Plano Red primaria proyecto Challabamba Alta	68
Figura 15 Plano Red secundaria proyecto Challabamba Alta	69
Figura 16 Relación de insumos RP	71
Figura 17 Relación de insumos RS	71
Figura 18 Resumen General del presupuesto del proyecto Challamba Alta generada por el software.....	73
Figura 19 Reporte generado por excel para el mismo proyecto	84
Figura 20 Reporte generado en S10 para el mismo proyecto	85

ÍNDICE DE TABLAS

Tabla 1	<i>Variables e indicadores Dependientes</i>	22
Tabla 2	<i>Variable Independiente</i>	22
Tabla 3	<i>Librerías clave para la elaboración del software</i>	52
Tabla 4	<i>Especificaciones Técnicas Detalladas del Proyecto Challabamba Alta</i>	58
Tabla 5	<i>Estructura de Datos del Archivo DIREDCAD RS (Planilla de Estructuras)</i>	59
Tabla 6	<i>Ejemplo de Datos Procesados del Archivo RS</i>	59
Tabla 7	<i>Especificaciones Técnicas Detalladas del Proyecto Challabamba Alta</i>	66
Tabla 8	<i>Características del Transformador Monofásico 10 kVA</i>	67
Tabla 9	<i>Gestión Presupuestal del Proyecto Challabamba Alta</i>	74
Tabla 10	<i>Presupuesto Suministro Red Primaria</i>	75
Tabla 11	<i>Presupuesto Montaje Red Primaria</i>	78
Tabla 12	<i>Presupuesto Suministro Red Secundaria</i>	80
Tabla 13	<i>Presupuesto Montaje Red Secundaria</i>	82
Tabla 14	<i>Síntesis del análisis comparativo</i>	86
Tabla 15	<i>Comparación de errores detectados según el método de elaboración</i>	88

CAPÍTULO I: GENERALIDADES

1.1. Ámbito De Estudio

Este trabajo presenta el desarrollo e implementación de un software de código abierto para la gestión de costos y presupuestos en sistemas de distribución, aplicado al proyecto de electrificación en Challabamba Alta, ubicado en la provincia de Paucartambo, departamento de Cusco. Además, su diseño está orientado a facilitar el acceso y la adaptación a distintos requerimientos del proyecto.

Figura 1

Ubicación del proyecto Challabamba Alta



Nota. Mapviewer, Arcgis 2025.

1.2. Problemática

1.2.1. Planteamiento del Problema

En la actualidad, la rapidez y precisión en la gestión de costos y presupuestos son fundamentales para la toma de decisiones en proyectos de ingeniería. Por ello, es necesario contar con herramientas informáticas que permitan realizar cálculos complejos de manera eficiente y precisa, minimizando el riesgo de errores asociados a los métodos manuales. Esto es especialmente crucial en la planificación y ejecución de proyectos en sistemas de distribución eléctrica, ya que un cálculo inexacto del presupuesto puede comprometer la viabilidad y rentabilidad del proyecto.

En el mercado peruano existen diversos programas especializados en la elaboración de presupuestos en ingeniería, como S10 y Delphin Express, que ofrecen funcionalidades avanzadas y una interfaz intuitiva. Sin embargo, estos softwares tienen un alto costo de adquisición, lo que limita su acceso a muchos profesionales y, especialmente, a estudiantes universitarios interesados en la gestión de costos y presupuestos para proyectos de distribución eléctrica.

Ante esta problemática, surge la necesidad de desarrollar e implementar un software de gestión de costos y presupuestos para sistemas de distribución, aplicado al proyecto de electrificación Challabamba Alta, ubicado en la provincia de Paucartambo, departamento de Cusco. Este software se basará en los principios del software libre, permitiendo su uso sin restricciones económicas y con la posibilidad de adaptación y mejora por parte de los usuarios. Este enfoque garantizará mayor accesibilidad y flexibilidad, optimizando recursos y fomentando el aprendizaje y especialización en el área.

Por ello, el presente trabajo tiene como finalidad diseñar y desarrollar una herramienta informática que facilite la gestión de costos y la elaboración de presupuestos en sistemas de

distribución eléctrica, ofreciendo una solución accesible y adaptable para profesionales y estudiantes involucrados en proyectos de electrificación.

1.2.2. Formulación del Problema

1.2.2.1. Problema General

¿En qué medida el desarrollo e implementación de un software de código abierto reduce el tiempo de elaboración y la cantidad de errores en la gestión de costos y presupuestos de proyectos de sistemas de distribución en el sector Challabamba Alta?

1.2.2.2. Problemas Específicos

- 1) ¿Cuáles son los fundamentos teóricos más relevantes relacionados con la gestión de costos y presupuestos en sistemas de distribución eléctrica?
- 2) ¿Cómo se puede diseñar una interfaz gráfica y los procesos necesarios para optimizar la interacción del usuario y mejorar su experiencia en la gestión de costos y presupuestos?
- 3) ¿Cómo la implementación del software diseñado valida su funcionalidad al aplicarse en el proyecto de electrificación del sector Challabamba Alta?

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un software de código abierto para la gestión de costos y presupuestos en proyectos de sistemas de distribución, aplicado al proyecto de electrificación en el sector Challabamba Alta.

1.3.2. Objetivos Específicos

- 1) Analizar los fundamentos teóricos relacionados con la gestión de costos y presupuestos en sistemas de distribución.
- 2) Diseñar la interfaz gráfica como los procesos necesarios para su operatividad, con el fin de facilitar la interacción del usuario y optimizar su experiencia.

- 3) Implementar el software diseñado para el cálculo de costos y presupuestos en proyectos de distribución eléctrica, con el fin de validar su funcionalidad mediante su aplicación en el proyecto de electrificación del sector Challabamba Alta.

1.4. Justificación del Trabajo

En la actualidad existe una variedad de herramientas informáticas que permiten llevar a cabo el presupuesto de un proyecto de electrificación, pero estos trabajan bajo licencia de pago, es por ello que el presente trabajo implica aplicar los conocimientos de los cursos de costos y presupuesto, con la finalidad de diseñar una herramienta informática bajo el concepto de software libre. Haciendo posible que los usuarios o personas puedan utilizar esta herramienta de manera gratuita, y que tengan las facilidades para poder mejorarlo o adaptarlo según las necesidades que se requieran. Para algún día tener un programa hecho por la comunidad electricista casi perfecta y que compita o este a la altura con otros programas comerciales.

La herramienta informática permitirá realizar el cálculo del presupuesto en sistemas de distribución de manera precisa y coherente, minimizando los errores que se pueden cometer al realizarlo de manera manual. Esto es importante para evitar errores o desviaciones del presupuesto, que pueden generar pérdidas económicas en los proyectos. Así también, al diseñarlo bajo el concepto de software libre el costo para la adquisición será de manera gratuita y de libre distribución.

La herramienta informática propuesta permitirá a los estudiantes y académicos aprender o entender los conceptos y metodologías relacionadas a la elaboración de presupuestos en redes eléctricas BT, lo que será fundamental para su desarrollo profesional. Pueden también utilizar la herramienta informática para la elaboración de presupuestos como parte de proyectos académicos o futuras tesis, lo que les permite abordar problemas reales y contribuir al conocimiento en su campo de estudio.

1.5. Hipótesis

1.5.1. Hipótesis General

La implementación de un software de código abierto para la gestión de costos y presupuestos en proyectos de sistemas de distribución reduce significativamente los tiempos de procesamiento y la cantidad de errores en el cálculo de costos, como se evidencia en su aplicación al proyecto de electrificación del sector Challabamba Alta.

1.5.2. Hipótesis Específicas.

- 1) Los fundamentos teóricos clave en la gestión de costos y presupuestos en sistemas de distribución eléctrica incluyen control de costos, estimación de recursos y optimización de presupuestos.
- 2) Un diseño eficiente de la interfaz gráfica y procesos bien definidos mejoran la interacción del usuario y optimizan la gestión de costos y presupuestos.
- 3) Con la implementación del software diseñado, se valida su funcionalidad al aplicarse en el proyecto de electrificación del sector Challabamba Alta, evidenciando que cumple con los requerimientos técnicos y mejora la precisión en la gestión de costos y presupuestos.

1.6. Operacionalización de Variables de estudio

1.6.1. Variable Independiente

Tabla 1

Variable Independiente

Variable Independiente	Definición Conceptual	Dimensión	Indicador	Unidad
Desarrollo e implementación del software de código abierto	“Programa informático libremente accesible y modificable, diseñado para automatizar el proceso de estimación, planificación y control de presupuestos en proyectos de sistemas eléctricos”.	Funcionalidad del software	Lenguaje de programación Python	Python

Nota. Elaboración propia.

1.6.2. Variable Dependiente

Tabla 2

Variable Dependiente

Variable Dependiente	Definición Conceptual	Dimensión	Indicador	Unidad
Eficiencia en la gestión de costos y presupuestos	“Es la capacidad de optimizar recursos, reducir tiempos y minimizar errores en el cálculo, planificación y control de los costos y presupuestos de un proyecto, mediante herramientas tecnológicas”.	Tiempo de elaboración del presupuesto	Tiempo promedio en realizar un presupuesto	Horas
		Precisión en cálculos	Cantidad de errores detectados	Número

Nota. Elaboración propia.

1.7. Alcances y Limitaciones

1.7.1. Alcances de la Investigación

- El trabajo de tesis pretende:
- Diseñar, desarrollar e implementar un software de código abierto y de fácil acceso para la gestión de costos y presupuestos en sistemas de distribución eléctrica.
- Utilizar el lenguaje de programación Python, seleccionado por su versatilidad, amplia comunidad de soporte y facilidad de aprendizaje, lo que permitirá futuras mejoras y adaptaciones por parte de los usuarios.
- Revisar los conceptos teóricos relacionados con la gestión de costos y presupuestos en proyectos de sistemas de distribución.
- Diseñar una interfaz gráfica intuitiva y funcional, que facilite al usuario el cálculo de presupuestos de manera eficiente.
- Realizar el análisis de costos unitarios y suministros del proyecto, tomando como referencia los armados utilizados por la empresa ELSE.
- Integrar una base de datos personalizable que incluya insumos, montaje electromecánico y partidas, permitiendo su actualización según las necesidades específicas del usuario.
- El análisis de costos unitarios, así como los suministros del proyecto se harán en base a los armados de la empresa ELSE.

1.7.2. Limitaciones de la Investigación

Dado que el desarrollo de software es un proceso que demanda tiempo, personal calificado y recursos específicos, este trabajo se ha enfocado exclusivamente en la gestión de costos y presupuestos para redes aéreas de distribución. Esto implica que el software desarrollado no abarca, en su actual versión, las particularidades y complejidades asociadas a proyectos de redes subterráneas.

1.8. Metodología

Por ser un trabajo de estudio de ingeniería, de acuerdo al fin que se persigue es de tipo Aplicada y Tecnológica.

La investigación es de tipo aplicada, ya que se basa en conocimientos teóricos sobre la gestión de costos y presupuestos en sistemas de distribución, utilizando teoría previamente desarrollada. Además, el enfoque principal de este trabajo es la aplicación inmediata de estos fundamentos teóricos a través del desarrollo de un software especializado.

Asimismo, es una investigación tecnológica, ya que implica la implementación de nuevos procedimientos para el diseño y desarrollo del software. En este caso, el programa automatizará los cálculos y generará informes con los datos de salida, lo que permitirá optimizar el proceso de elaboración de presupuestos, reduciendo el tiempo y los recursos empleados.

1.9. Población y Muestra

1.9.1. Población

La población de este estudio está compuesta por la concesionaria Electro Sur Este, encargada de la distribución eléctrica en el sur y sureste del Perú, específicamente en las regiones donde se lleva a cabo el proyecto de electrificación. Dado que el software de gestión de costos y presupuestos se basará en una base de datos relacionada con los proyectos y operaciones de esta concesionaria, se considera que la población es representativa de los proyectos de distribución en áreas bajo su jurisdicción.

1.9.2. Muestra

La muestra de este estudio corresponde al sector de Challabamba Alta, una zona específica dentro de la concesionaria Electro Sur Este, donde se aplicará el software para la gestión de costos y presupuestos del proyecto de electrificación.

1.10. Técnicas de Recolección de Datos

Para garantizar una recolección de información precisa para el análisis de costos y presupuestos, se utilizarán fuentes bibliográficas, trabajos referenciales y la experiencia de profesionales especializados en la elaboración de presupuestos en sistemas de distribución.

1.11. Procesamiento de Datos

Para el desarrollo del software de gestión de costos y presupuestos en sistemas de distribución, se utilizará Python 3.11, un lenguaje que ofrece una amplia gama de módulos y bibliotecas que facilitan el procesamiento y análisis de datos. Entre los módulos clave que se emplearán están:

- NumPy: Para el manejo eficiente de arrays y operaciones matemáticas.
- Pandas: Para la manipulación y análisis de datos en estructuras tabulares.
- Matplotlib y Seaborn: Para la visualización gráfica de los datos procesados.

Además, se implementará ttkbootstrap para la creación de la interfaz gráfica, proporcionando una experiencia visual moderna y funcional. La estructura del software incluirá múltiples notebooks organizados que facilitarán la visualización y manipulación de la información, asegurando un flujo de trabajo eficiente.

El sistema estará diseñado con funciones específicas para la carga de datos, gestión de pestañas y la ejecución de procesos internos, lo que permitirá un manejo óptimo de la información. Además, se integrarán herramientas como:

Microsoft Office Word: Para la creación y edición de documentos.

Microsoft Office Excel: Para la manipulación de datos y la creación de informes detallados.

1.12. Matriz de consistencia

La matriz de consistencia se muestra en la siguiente página

MATRIZ DE CONSISTENCIA TÍTULO: DESARROLLO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN DE COSTOS Y PRESUPUESTOS EN SISTEMAS DE DISTRIBUCIÓN, APLICADO AL PROYECTO DE ELECTRIFICACIÓN CHALLABAMBA ALTA																											
FORMULACIÓN DEL PROBLEMA	OBJETIVOS	HIPÓTESIS	TÉCNICAS DE RECOLECCIÓN Y PROCESAMIENTO DE DATOS	CONCLUSIONES DEL TRABAJO																							
<p><u>a) Problema General</u> ¿En qué medida el desarrollo e implementación de un software de código abierto reduce el tiempo de elaboración y la cantidad de errores en la gestión de costos y presupuestos de proyectos de sistemas de distribución en el sector Challabamba Alta?</p> <p><u>b) Problema Específicos</u> 1. ¿Cuáles son los fundamentos teóricos más relevantes relacionados con la gestión de costos y presupuestos en sistemas de distribución eléctrica?</p> <p>2. ¿Cómo se puede diseñar una interfaz gráfica y los procesos necesarios para optimizar la interacción del usuario y mejorar su experiencia en la gestión de costos y presupuestos?</p> <p>3. ¿Cómo la implementación del software diseñado valida su funcionalidad al aplicarse en el proyecto de electrificación del sector Challabamba Alta?</p>	<p><u>a) Objetivo General</u> Desarrollar un software de código abierto para la gestión de costos y presupuestos en proyectos de sistemas de distribución, aplicado al proyecto de electrificación en el sector Challabamba Alta.</p> <p><u>b) Objetivos Específicos</u> 1. Analizar los fundamentos teóricos relacionados con la gestión de costos y presupuestos en sistemas de distribución.</p> <p>2. Diseñar la interfaz gráfica como los procesos necesarios para su operatividad, con el fin de facilitar la interacción del usuario y optimizar su experiencia.</p> <p>3. Implementar el software diseñado para el cálculo de costos y presupuestos en proyectos de distribución eléctrica, con el fin de validar su funcionalidad mediante su aplicación en el proyecto de electrificación del sector Challabamba Alta.</p>	<p><u>a) Hipótesis General</u> La implementación de un software de código abierto para la gestión de costos y presupuestos en proyectos de sistemas de distribución reduce significativamente los tiempos de procesamiento y la cantidad de errores en el cálculo de costos, como se evidencia en su aplicación al proyecto de electrificación del sector Challabamba Alta.</p> <p><u>b) Hipótesis Específicas</u> 1. El análisis de los fundamentos teóricos permite identificar principios clave que orientan el diseño funcional del software para la gestión de costos en sistemas de distribución eléctrica.</p> <p>2. Un diseño eficiente de la interfaz gráfica y procesos bien definidos mejoran la interacción del usuario y optimizan la gestión de costos y presupuestos.</p> <p>3. Con la implementación del software diseñado, se valida su funcionalidad al aplicarse en el proyecto de electrificación del sector Challabamba Alta, evidenciando que cumple con los requerimientos técnicos y mejora la precisión en la gestión de costos y presupuestos.</p>	<p>Para asegurar una recolección precisa de información para el análisis de costos y presupuestos, se emplearán fuentes bibliográficas, trabajos referenciales y la experiencia de profesionales especializados en presupuestos de sistemas de distribución. El procesamiento de los datos se realizará utilizando Python 3.11, un lenguaje de programación que ofrece diversas bibliotecas y módulos para facilitar el desarrollo del software y el análisis de la información de manera eficiente.</p> <p>MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES</p> <table> <tr> <th>Variable Independiente</th><th>Definición Conceptual</th><th>Dimensión</th><th>Indicador</th><th>Unidad</th></tr> <tr> <td>Desarrollo e implementación del software de código abierto</td><td>“Programa informático libremente accesible y modificable, diseñado para automatizar el proceso de estimación, planificación y control de presupuestos en proyectos de sistemas eléctricos”.</td><td>Funcionalidad del software</td><td>Lenguaje de programación Python</td><td>Python</td></tr> </table> <table> <tr> <th>Variable Dependiente</th><th>Definición Conceptual</th><th>Dimensión</th><th>Indicador</th><th>Unidad</th></tr> <tr> <td rowspan="2">Eficiencia en la gestión de costos y presupuestos</td><td rowspan="2">“Es la capacidad de optimizar recursos, reducir tiempos y minimizar errores en el cálculo, planificación y control de los costos y presupuestos de un proyecto, mediante herramientas tecnológicas”.</td><td>Tiempo de elaboración del presupuesto</td><td>Tiempo promedio en realizar un presupuesto</td><td>Horas</td></tr> <tr> <td>Precisión en cálculos</td><td>Cantidad de errores detectados</td><td>Número</td></tr> </table>	Variable Independiente	Definición Conceptual	Dimensión	Indicador	Unidad	Desarrollo e implementación del software de código abierto	“Programa informático libremente accesible y modificable, diseñado para automatizar el proceso de estimación, planificación y control de presupuestos en proyectos de sistemas eléctricos”.	Funcionalidad del software	Lenguaje de programación Python	Python	Variable Dependiente	Definición Conceptual	Dimensión	Indicador	Unidad	Eficiencia en la gestión de costos y presupuestos	“Es la capacidad de optimizar recursos, reducir tiempos y minimizar errores en el cálculo, planificación y control de los costos y presupuestos de un proyecto, mediante herramientas tecnológicas”.	Tiempo de elaboración del presupuesto	Tiempo promedio en realizar un presupuesto	Horas	Precisión en cálculos	Cantidad de errores detectados	Número	<p>1. El software desarrollado demuestra ser una solución eficiente, accesible y adaptable para la gestión de costos y presupuestos en proyectos de sistemas de distribución eléctrica. Su arquitectura de código abierto garantiza libre disponibilidad, posibilidad de personalización y escalabilidad, lo que facilita su implementación en diversos contextos. Estas cualidades lo consolidan como una herramienta valiosa para instituciones, profesionales y estudiantes del sector eléctrico, contribuyendo a una gestión más precisa, ágil y pertinente en proyectos de electrificación rural.</p> <p>2. El análisis de los fundamentos teóricos permitió establecer una base sólida para el diseño e implementación del software propuesto. Se identificaron los principios clave de la gestión de costos en proyectos de distribución eléctrica, como la estructuración jerárquica de partidas, el análisis de precios unitarios y la importancia del metrado. Este marco conceptual sirvió de guía para el desarrollo del sistema, asegurando que las funcionalidades implementadas respondan a los requerimientos técnicos y prácticos de proyectos de electrificación rural.</p> <p>3. El diseño de la interfaz del software se centró en la simplicidad, usabilidad y eficiencia. Se logró una estructura intuitiva que facilita la navegación y el uso del sistema sin requerir conocimientos avanzados en programación. Los procesos complejos, como la importación de metrados desde DIREDCAD o la asignación automática de partidas, fueron optimizados para ejecutarse en pocos pasos. Esta orientación al usuario final constituye una mejora frente a herramientas tradicionales más complejas, especialmente en entornos académicos y profesionales donde la accesibilidad tecnológica es clave. Además, el desarrollo del sistema se realizó íntegramente en Python, un lenguaje de programación de código abierto, lo que favorece su adaptabilidad, escalabilidad y distribución libre sin costos de licenciamiento.</p> <p>4. La aplicación del software desarrollado en el proyecto Challabamba Alta permitió validar su funcionalidad mediante la comparación con el método manual (excel) y con el software S10. Los resultados evidencian una mejora significativa en la eficiencia de elaboración de presupuestos: el tiempo promedio de procesamiento se redujo de 8 horas con el método manual y 3 horas con S10, a solo 1.5 horas con el sistema propuesto, gracias a la automatización de cálculos y a la importación directa de metrados desde DIREDCAD. En cuanto a precisión, los errores detectados pasaron de 6 en el método manual y 3 en S10 a únicamente 1 en el software desarrollado, asociado a un ajuste visual de la interfaz. Estas mejoras confirman que la implementación del sistema optimiza de manera sustancial la gestión de costos y presupuestos en proyectos de electrificación rural.</p>
Variable Independiente	Definición Conceptual	Dimensión	Indicador	Unidad																							
Desarrollo e implementación del software de código abierto	“Programa informático libremente accesible y modificable, diseñado para automatizar el proceso de estimación, planificación y control de presupuestos en proyectos de sistemas eléctricos”.	Funcionalidad del software	Lenguaje de programación Python	Python																							
Variable Dependiente	Definición Conceptual	Dimensión	Indicador	Unidad																							
Eficiencia en la gestión de costos y presupuestos	“Es la capacidad de optimizar recursos, reducir tiempos y minimizar errores en el cálculo, planificación y control de los costos y presupuestos de un proyecto, mediante herramientas tecnológicas”.	Tiempo de elaboración del presupuesto	Tiempo promedio en realizar un presupuesto	Horas																							
		Precisión en cálculos	Cantidad de errores detectados	Número																							

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes del trabajo

Se toma en consideración los siguientes trabajos que guardan relación con el tema y se detallan a continuación:

2.1.1. Antecedentes Locales

- ❖ El trabajo titulado “Evaluación comparativa del S10, Delphin Express, Arquímedes y Sistemas RW7”, Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), Cusco – Perú 2021, presentado por: Medina Velasque, E. R., que obtuvo las siguientes aportaciones:
 - “La presente tesis realiza la evaluación comparativa de los programas de costos y presupuestos: S10, Delphin Express, Arquímedes y Sistemas RW7, para determinar cuál de este software se adapta mejor a las necesidades de elaboración de presupuestos en proyectos de construcción” (Medina Velasque, 2021).
 - “El estudio comparativo permitió identificar las ventajas y desventajas de cada software analizado, proporcionando criterios técnicos objetivos para la selección de la herramienta más adecuada según los requerimientos específicos de cada proyecto y las capacidades técnicas del usuario” (Medina Velasque, 2021).
 - “La investigación demostró que cada programa presenta características diferenciadas en cuanto a interfaz de usuario, capacidad de generación de reportes, manejo de bases de datos de costos y adaptabilidad a los formatos requeridos por las entidades peruanas, factores determinantes en la eficiencia del proceso de elaboración de costos y presupuestos” (Medina Velasque, 2021).

2.1.2. Antecedentes Nacionales

❖ El trabajo titulado “Eficiencia Del Software Delphin Express Para La Elaboración De Costos Y Presupuestos”, Universidad José Carlos Mariátegui, Moquegua – Perú 2023, presentado por: Chino Fur Daniel Aldo, que obtuvo las siguientes aportaciones:

- “Con el uso del, al elaborar los costos y presupuestos de un proyecto es eficiente en las tareas que realiza, a su vez obtendremos una gran cantidad de reportes finales de una manera ordenada y se mantendrá una correlación de los ítems, que muchas veces al momento de la elaboración del presupuesto se pasan por alto” (Chino Fur, 2023).
- “Con el software Delphin Express BIM 2019, no necesitas ser un entendido o tener mucha experiencia en manejo de este software para realizar un presupuesto, porque su entorno es muy amigable, fácil de entender, son muy similares a las ventanas y herramientas de Office, esto nos permite encontrar de una manera muy fáciles la herramienta que estamos buscando, lo que lo convierte en un software para todos todo público muy fácil de entender y manipular” (Chino Fur, 2023).

2.1.3. Antecedentes Internacionales

1. El trabajo titulado “Diseño De Una Herramienta Informática Para El Cálculo De Precios Unitarios, Presupuestos Y Fórmulas De Reajuste”, Universidad Internacional Del Ecuador, Quito – Ecuador 2015, presentado por: Edison Fernando Suarez Morales, que obtuvo las siguientes aportaciones:

- “Que la herramienta informática ayuda a crear formularios y reportes de precios unitarios que optimizar el tiempo a proyectistas y constructores poque tendrán una base de datos estable con costos de equipo y maquinaria, mano de obra y materiales” (Suárez Morales, 2015).
- “El usuario de la herramienta informática puede hacer uso con toda la seguridad y confianza porque los cálculos programados se han comprobado y validado a través de la realización de los mismos en la hoja de cálculo” (Suárez Morales, 2015).
- “La herramienta informática ayudara a la efectividad de los proyectistas y constructores a tener una base de datos estable con precios unitarios y presupuestos y por ende disminución del tiempo de trabajo” (Suárez Morales, 2015).

2. El trabajo titulado “Software Para La Gestión Presupuestal De Obras De Construcción En Pequeñas Y Medianas Empresas”, Universidad Piloto De Colombia, Bogotá – Colombia 2021, presentado por: Diana Catalina Contreras López y Sergio Esteban Maldonado Rodríguez, que obtuvo las siguientes aportaciones:

- “El diseño de un software que permite controlar la ejecución del presupuesto de las obras que adelantan pequeñas y medianas empresas de construcción. En el momento en que se hace la redacción de estas conclusiones, ya el software se encuentra publicado en Internet y pueden ingresar las personas que reciben el nombre de usuario y código de acceso que les asignan los autores de este trabajo, que fueron los encargos de su desarrollo y son sus

actuales administradores” (Contreras López y Maldonado Rodríguez, 2021).

- “Para el desarrollo del software, en primer lugar, se definieron los parámetros necesarios para controlar la ejecución del presupuestal de las obras adelantadas por pequeñas y medianas empresas de construcción, tal como estaba previsto en el primer objetivo específico del trabajo. Esos parámetros se definieron con base en el marco de referencia que se presenta en el segundo capítulo de este trabajo, así como en la experiencia directa de los autores en empresas de este tipo” (Contreras López y Maldonado Rodríguez, 2021).
- “A diferencia de otras opciones de software que existen en el mercado, que están orientados a la gestión presupuestal, dado que una de las características que se definieron antes de desarrollarlo, fue su bajo costo. Desde el punto de vista técnico y de desarrollo de software, se pueden desarrollar otros módulos enfocados en otros aspectos de un proyecto de ingeniería, como situación de avance de la obra frente al cronograma inicial, control de calidad, administración de personal, etc. Esos módulos podrían operar de manera autónoma, con lo cual también serían de bajo costo si se comercializan de manera independiente. Si se unificaran o integraran en uno solo todos los módulos desarrollados de manera separada, se tendría un software mucho más robusto, aunque seguramente el costo de un desarrollo con esas características haría

que resultara inalcanzable para empresas medianas y pequeñas”
(Contreras López & Maldonado Rodríguez, 2021).

2.1.4. Programas Comerciales

1. Sistema S10, “es un software de ingeniería que calcula el costo que involucra construir cualquier obra de ingeniería, así también determina las cantidades de mano de obra, material y equipo que se necesitaría para dicha obra. El programa es interactivo y visual en su totalidad, que permite calcular otros parámetros como formula polinómica, gastos generales (gastos de oficina), tiempos de duración de las actividades, etc.” (S10peru.com, 2023)
2. “Delphin Express es un software muy completo que trabaja con la normativa peruana. Presenta una interfaz muy amigable e intuitiva que permite la elaboración de presupuestos, metrados, especificaciones técnicas, programación de obra, valorizaciones, control y seguimiento de obra y además posee la herramienta de cuaderno de obra digital” (ddexpress.com, Delphin Express, 2022).

2.2. Bases Teóricas

2.2.1. Presupuesto

2.2.1.1. ¿Qué es gestión de costos y presupuestos?

La gestión de costos y presupuestos es una disciplina fundamental dentro de la administración de proyectos que se encarga de la planificación, estimación, asignación, control y supervisión de los recursos financieros de un proyecto u organización. Su objetivo principal es garantizar que los costos finales no excedan los fondos aprobados, optimizando el uso de los recursos para cumplir con los objetivos estratégicos dentro de los límites establecidos.

“La gestión de un presupuesto se refiere al proceso de planificación, control y supervisión de los recursos financieros asignados a un proyecto, asegurando su uso eficiente y alineación con los objetivos establecidos. Este proceso implica la estimación de costos, la asignación de fondos, el monitoreo del gasto y la toma de decisiones para optimizar los recursos disponibles” (Kerzner, 2017).

2.2.1.2. ¿Qué es un presupuesto?

“Presupuesto es la estimación o valoración económica de un proyecto o servicio determinado tomando en cuenta sus datos técnicos y tiempo de ejecución. El presupuesto de obra es el elemento primordial en la planeación y ejecución de una obra y se procede a la elaboración del mismo en el momento en que se tome la decisión de construir. Su objetivo es determinar los costos requeridos para completar un proyecto de acuerdo a unos planos y unas especificaciones técnicas brindadas por el especialista. Puede definirse como la presentación ordenada de los resultados previstos de un plan o un proyecto” (Suárez Morales, 2015).

2.2.1.3. Importancia del presupuesto

“La estimación de costos y la elaboración de presupuestos, representa uno de los pasos más importantes en lo que se refiere la planificación de una obra. En cada etapa de la construcción, el presupuesto representa la base para la toma de decisiones y, en los que se refiere en obras de carácter público (licitaciones), es el factor más importante en la adjudicación de contratos. Actualmente, la gran competitividad en el sector de la construcción, hace que la estimación de costos sea una de las causas de éxito o fracaso de empresas. La elaboración de un presupuesto, tiene su base en la asignación de un precio unitario a cada una de las actividades que se desarrollan representadas por un volumen de obra. El costo total es la sumatoria de la multiplicación de los precios unitarios y la cantidad de cada ítem” (Padilla Carreño, 2012).

2.2.1.4. Característica del presupuesto

“Todo presupuesto tiene cuatro características fundamentales: es aproximado, es singular, es temporal y es una herramienta de control. El presupuesto es aproximado, sus previsiones se acercarán más o menos al costo real de la obra, dependiendo de la habilidad (uso correcto de técnicas presupuestales), el criterio (visualización correcta del desarrollo de la obra) y experiencia del encargado. El presupuesto es singular, como lo es cada obra, sus condiciones de localización, clima y medio ambiente, calidad de la mano de obra características del constructor, etc. Cada obra requiere un presupuesto propio, así como cada persona o empresa tiene su forma particular de presupuestar.

El presupuesto es temporal, los costos que en él se establecen sólo son válidos mientras tengan vigencia los precios que sirvieron de base para su elaboración. Los principales factores de variación son: Incremento del costo de los insumos y servicios; utilización de nuevos productos y técnicas; desarrollo de nuevos equipos, herramientas, materiales, tecnología, etc.; descuentos por volumen; reducción en ofertas de insumos por situaciones especiales, cambios estacionales” (Padilla Carreño, 2012).

2.2.1.5. Etapas en el estudio de un presupuesto

“Generalmente, cuando se realiza un presupuesto, se tiene un tiempo definido para realizarlo y desde el punto de vista de una empresa constructora, se tiene que cumplir con una serie de aspectos técnicos para la presentación de la propuesta, por lo tanto, se deben tomar los siguientes aspectos” (Padilla Carreño, 2012).

- a) “Se debe analizar el calendario para la presentación de la propuesta, es decir tomar en cuenta cuando se terminará el análisis de los precios de los materiales, el tiempo en que se terminará de elaborar los aspectos técnicos de la propuesta, tiempo que se requerirá la compaginación de la propuesta, etc” (Padilla Carreño, 2012).

- b) Posteriormente se debe realizar un exhaustivo análisis de las bases de la licitación plasmado en el pliego de condiciones otorgado por la empresa contratante (Padilla Carreño, 2012).
- c) Se debe preparar un listado de cotizaciones de los materiales a utilizar en la obra, para esto se debe tener claramente identificadas las exigencias y especificaciones técnicas que pide la entidad contratante. En el caso de cotizaciones de subcontratos se debe procurar entregar el máximo de información disponible al cotizador (Padilla Carreño, 2012).

2.2.2. Metrados

“En términos generales podemos definir a los metrados como el cálculo o cuantificación por partidas, de la cantidad de obra a ejecutar. El metrado debe realizarse con un proceso ordenado y sistemático de cálculo, en base a partidas” (Salinas Seminario, 2012).

2.2.2.1. Recomendaciones para realizar un metrado

“Precisar la zona de estudio o de metrado y los trabajos que se van a ejecutar. El orden para elaborar el metrado en forma manual es primordial dado que brindará la secuencia en que se toman las medidas o lecturas de los planos, enumerándose las páginas en las cuales se escriben las cantidades incluidas las observaciones pertinentes. El ordenamiento dará la pauta para realizar una verificación rápida y reducir errores” (Ramos Salazar, 2015).

2.2.2.2. Formatos

En la siguiente figura se muestra un formato general para la elaboración de metrados según el requerimiento.

Figura 2
Formato de aplicación general

METRADO								
Obra:				Hoja N.º: de				
Propietario:				Plano N.º:				
Fecha:				Hecho por:				
				Revisado:				
Partida N.º	Especificaciones	N.º de veces	Medidas			Parcial	Total	Unidad
			Largo	Ancho	Altura			

Nota. (Ramos Salazar, 2015).

2.2.3. Costos directos

“El costo directo es la suma de costos de materiales, mano de obra (incluyendo leyes sociales), equipos, herramientas, y todos los elementos requeridos para la ejecución de una obra. Estos costos directos analizados de cada una de las partidas conformantes de una obra pueden tener diversos grados de aproximación de acuerdo al interés propuesto; sin embargo, al efectuar un mayor refinamiento de los mismos no siempre conduce a una mayor exactitud dado que siempre existirán diferencias entre los diversos estimados de costos de la misma partida. Ello debido a los diferentes criterios que se pueden asumir, así como a la experiencia del ingeniero que los elabore” (Ramos Salazar, 2015).

2.2.4. Costos indirectos

“La construcción es una actividad de un variado y heterogéneo espectro de obras que puede abarcar desde la ejecución de una vivienda hasta la de una central hidroeléctrica. Una de sus principales características es desarrollarse en un determinado ejemplo, de acuerdo a la obra, la cual la hace vulnerable a los efectos de la economía del medio en que se desenvuelve. En general, se puede definir los costos directos como gastos

que se pueden aplicar a una partida determinada y los costos indirectos como todos aquellos gastos que no pueden aplicarse a una partida determinada, sino al conjunto de la obra; estos se detallarán a continuación” (Ramos Salazar, 2015).

2.2.5. Clasificación de los costos indirectos

2.2.5.1. Gastos generales

“Están establecidos como aquellos gastos que debe efectuar el contratista durante la construcción, derivados de su propia actividad empresarial, por lo cual no pueden ser incluidos dentro de las partidas de la obra. Cabe indicar que los gastos generales son cantidades que se calculan analíticamente, por lo que, a pesar que se expresan como un porcentaje del costo directo, no son un porcentaje sino una parte del costo indirecto” (cersa.org.pe, 2023).

2.2.5.2. Utilidades

La utilidad es la diferencia entre los ingresos y todos los costos y gastos en los cuales se incurrió durante el período. Por tanto, la utilidad y no los ingresos son lo que realmente gana la empresa (cersa.org.pe, 2023).

2.2.6. Aporte unitario de materiales

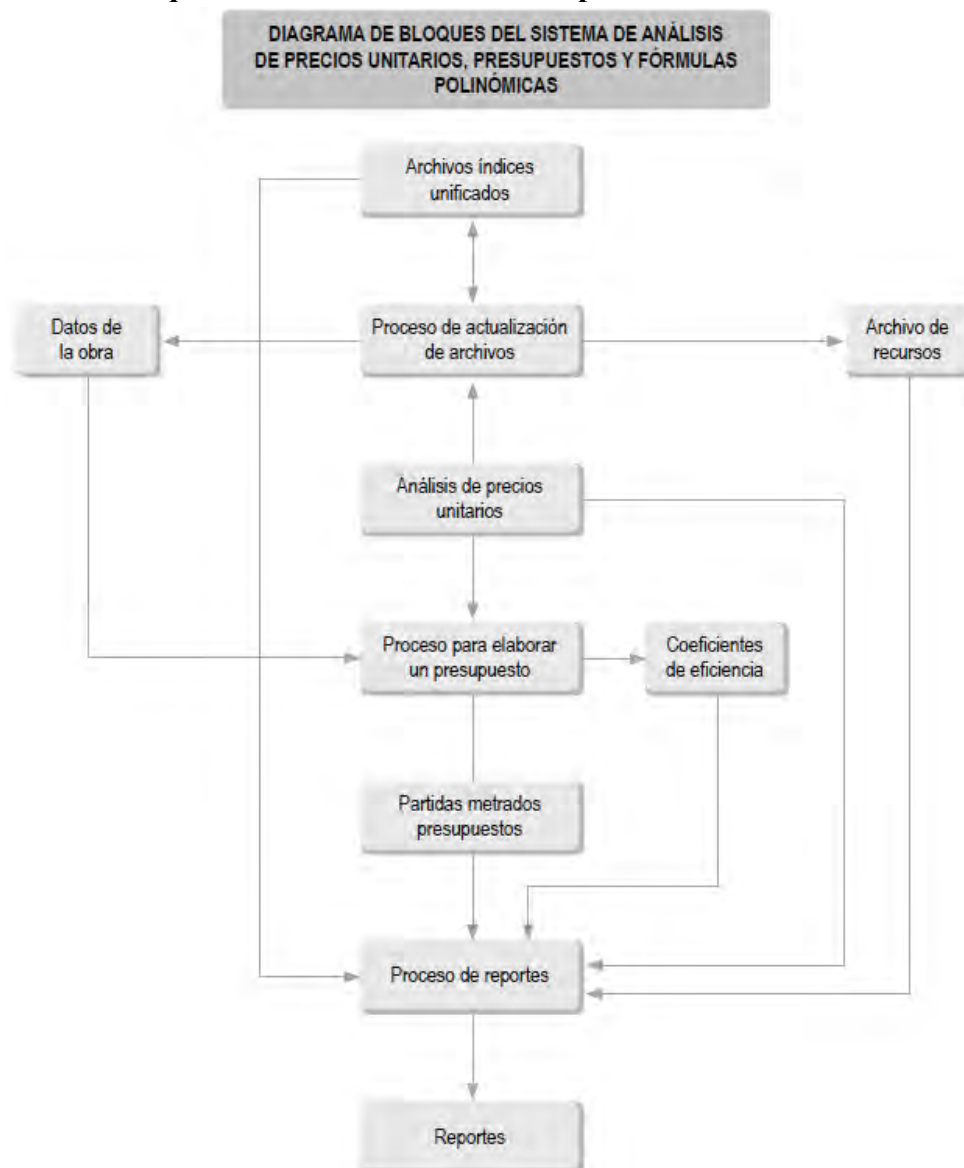
“La cantidad de material se establece de acuerdo a condiciones preestablecidas físicas o geométricas dadas según el estudio técnico del mismo, y que consideran las publicaciones especializadas o, mejor aún, que elaboran los análisis con registros directos de la obra, considerando en razón a ello que estos análisis de costos responden a un proceso dinámico de confección. Los insumos de materiales son expresados en unidades de comercialización: bolsa de cemento, metro cúbico de arena o piedra chancada, pie cuadrado de madera, kilogramo o varillas de fierros, etc” (Ramos Salazar, 2015).

2.2.7. Estructura de un sistema de análisis unitarios

“Todo sistema de análisis de precios unitarios, presupuestos y fórmulas polinómicas debe estructurarse de acuerdo al Diagrama de Bloque que se presenta y describe a continuación” (Ramos Salazar, 2015).

Figura 3

Diagrama de bloques del sistema de análisis de precios unitarios



Nota. (Ramos Salazar, 2015).

2.2.8. Proyectos de construcción

“En general el término proyecto de construcción se refiere a diferentes tipos de obra que pueden ser de origen público o privado y que se adelantan con el propósito de

satisfacer necesidades relacionadas con infraestructura vial, servicios públicos, educación, salud, vivienda, comercio, turismo, recreación, etc.” (Araya, 2019).

Figura 4
Fases de un proyecto



Nota. (Contreras López y Maldonado Rodríguez, 2021).

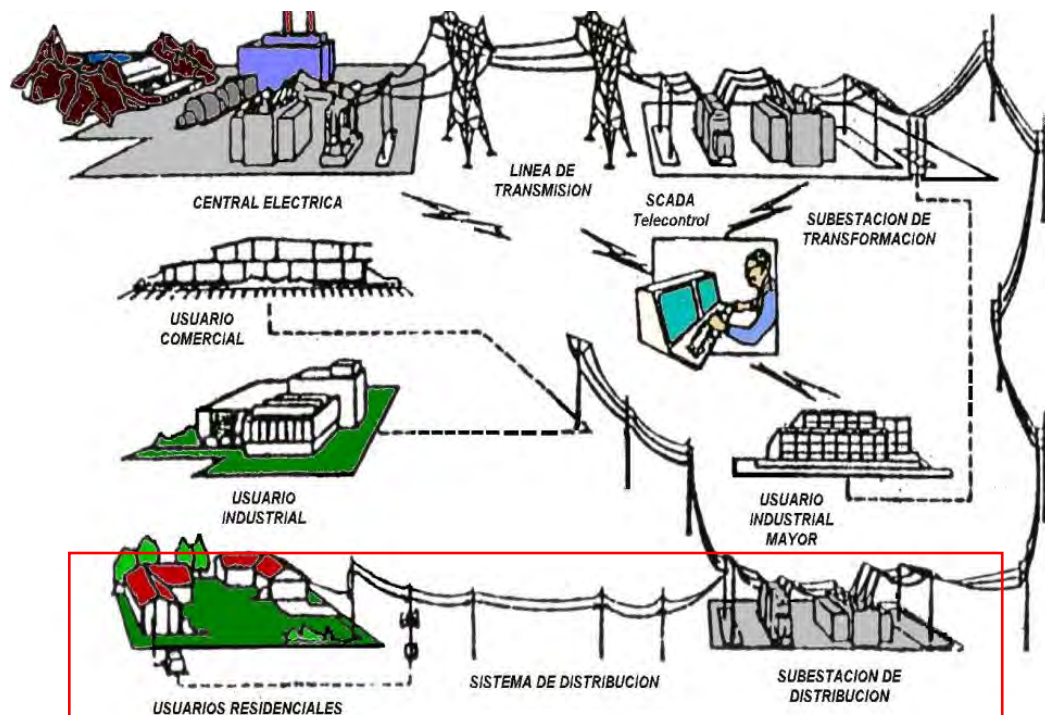
2.2.9. Sistema de distribución

El sistema de distribución eléctrica en el Perú representa una parte significativa de la inversión total del sistema eléctrico nacional, requiriendo un trabajo cuidadoso en el planeamiento, diseño, construcción y operación. Esto implica el manejo de información voluminosa y la toma de numerosas decisiones técnicas y económicas, constituyendo una tarea compleja pero de gran importancia para el desarrollo nacional.

“Los sistemas de distribución incluyen todos los elementos de transporte de energía eléctrica comprendidos entre las subestaciones primarias, donde la transmisión de potencia se reduce a niveles de distribución, y las reducciones de voltaje en los clientes. Un sistema de distribución normalmente consta de redes de, subestaciones de distribución, que transforman la energía a una tensión más baja, adecuada para la distribución local, alimentadores, los cuales alimentan un área definida” (Ministerio de Energía y Minas, 2011).

Según el Código Nacional de Electricidad - Suministro 2011, "el punto de entrega, para los suministros en baja tensión, como la conexión eléctrica entre la acometida y las instalaciones del concesionario" define claramente la interfaz entre el sistema de distribución y las instalaciones del usuario. "En los casos de media tensión, el concesionario establecerá el punto de entrega en forma coordinada con el usuario" (CNE-Suministro, 2011).

Figura 5
Sistema Eléctrico Peruano



Nota. www.minem.gob.pe.

2.2.10. Clasificación de las Redes de Distribución según su Ubicación

2.2.10.1. Redes de distribución urbana

Según la Norma Técnica de Calidad de los Servicios Eléctricos (NTCSE), aprobada mediante Decreto Supremo N° 020-97-EM, las redes de distribución urbana en el Perú se caracterizan por atender a usuarios concentrados en zonas de alta densidad poblacional. "El presente Decreto Supremo aprueba la Norma técnica de calidad de los servicios eléctricos, cuyo objetivo establecer los niveles mínimos de calidad de los

servicios eléctricos" (Decreto Supremo N° 020-97-EM, 1997), incluyendo tolerancias de tensión más estrictas que los sistemas rurales.

Las principales características de un sistema de distribución urbano según la normatividad peruana son:

- Facilidad de acceso: Infraestructura desarrollada que permite el mantenimiento y operación eficiente del sistema eléctrico.
- Usuarios concentrados: Alta densidad de consumidores por kilómetro de red, lo que optimiza la inversión en infraestructura.
- Cargas generalmente trifásicas: Predominio de consumidores industriales y comerciales que requieren suministro trifásico.
- Calidad de suministro: Estándares más exigentes de continuidad y calidad del servicio eléctrico según la NTCSE.

2.2.10.2. Redes de distribución rurales

"Para los servicios calificados como Urbano-Rurales y/o Rurales, las tolerancias de tensión son de hasta el $\pm 7.5\%$ " (Código Nacional de Electricidad - Suministro, 2011). Esta clasificación reconoce las particularidades de las zonas rurales del Perú, donde la dispersión geográfica y las condiciones topográficas complejas requieren criterios técnicos diferenciados.

OSINERGMIN, definido como "una institución pública encargada de regular y supervisar que las empresas del sector eléctrico cumplan las disposiciones legales de las actividades que desarrollan" (OSINERGMIN, 2024), supervisa el cumplimiento de estos estándares diferenciados para las redes rurales.

Las redes de distribución rurales en el Perú, reguladas por la normatividad nacional, presentan las siguientes características específicas:

Las redes de distribución rurales en el Perú, reguladas por la normatividad nacional, presentan las siguientes características específicas:

- **Extensas distancias:** Grandes vanos entre estructuras debido a la dispersión de usuarios en el territorio nacional.
- **Topografía compleja:** Adaptación a las condiciones geográficas andinas y de selva del país, con altitudes que pueden superar los 3,800 msnm.
- **Baja densidad de carga:** Menor número de usuarios por kilómetro de red, lo que impacta en la viabilidad económica de los proyectos.
- **Tolerancias técnicas diferenciadas:** Compensaciones por mala calidad de suministro con criterios específicos para servicio urbano-rural y rural.
- **Programas especiales de electrificación:** Desarrollo bajo marcos normativos específicos para la electrificación rural.

2.2.10.3. Redes de distribución urbano-rurales

La normatividad peruana, establecida en la NTCSE, reconoce una clasificación intermedia denominada "urbano-rural" (Decreto Supremo N° 020-97-EM, 1997). Esta categoría atiende a localidades que presentan características mixtas, donde coexisten zonas de concentración poblacional con áreas dispersas, requiriendo criterios técnicos que consideren ambas realidades.

2.2.11. El lenguaje de programación Python

“Python es un lenguaje de programación de alto nivel creado a finales de los 80/principios de los 90 por Guido van Rossum, holandés que trabajaba por aquella época en el Centro para las Matemáticas y la Informática de los Países Bajos. Sus instrucciones están muy cercanas al lenguaje natural en inglés y se hace hincapié en la legibilidad del código. Toma su nombre de los Monty Python, grupo humorista de los 60 que gustaban mucho a Guido. Python fue creado como sucesor del lenguaje” (Delgado Quintero, 2023).

2.2.11.1. Características del lenguaje

“Python es un lenguaje de programación interpretado y multiplataforma cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional” (Delgado Quintero, 2023).

2.2.11.2. Entornos virtuales

Cuando se trabaja en distintos proyectos, no todos ellos requieren los mismos paquetes ni siquiera la misma versión de Python. La gestión de estas situaciones no es sencilla si únicamente instalamos paquetes y manejamos configuraciones a nivel global (a nivel de máquina). Es por ello que surge el concepto de entornos virtuales. Como su propio nombre indica se trata de crear distintos entornos en función de las necesidades de cada proyecto, y esto nos permite establecer qué versión de Python usaremos y qué paquetes instalaremos (Delgado Quintero, 2023).

2.2.11.3. Principales librerías

“Python es ampliamente utilizado en la comunidad de ciencia de datos. Las librerías de Python son esenciales para el análisis de datos, ya que proporcionan un conjunto de funciones y herramientas que se pueden usar para manipular y analizar los datos. Hay diferentes tipos de librerías de Python disponibles que se clasifican según su caso de uso” (Pontia, 2023).

2.2.11.4. NumPy

“NumPy es una librería de Python para trabajar con arreglos, matrices y objetos de dimensiones superiores. Es uno de los paquetes fundamentales para la computación científica con Python” (Pontia, 2023).

2.2.11.5. Pandas

“Pandas es una librería de herramientas de análisis de datos escrita en Python. Proporciona estructuras de datos y funciones que facilitan el trabajo con diferentes tipos de conjuntos de datos. Los pandas se pueden utilizar para diversas tareas, como: manipulación de datos, visualización de datos, modelado estadístico, aprendizaje automático y análisis de los datos” (Pontia, 2023).

2.2.11.6. SciPy

“SciPy es una librería de Python para matemáticas, ciencias e ingeniería. Proporciona herramientas para el cálculo numérico, la optimización, la integración y otras operaciones matemáticas” (Pontia, 2023).

2.2.11.7. Matplotlib

“Matplotlib es una librería de trazado 2D de Python que produce gráficos en una variedad de formatos. Se puede utilizar en otros idiomas a través de una interfaz. La versión actual de matplotlib utiliza el paradigma de la programación orientada a objetos, pero también brinda acceso a la mayoría de sus funciones a través de una interfaz de procedimiento. Ofrece modos interactivos y no interactivos, en los que los usuarios pueden crear gráficos y guardarlos en formatos de archivo como JPEG o PNG (PNG solo para el modo no interactivo)” (Pontia, 2023).

2.3. Base Conceptual

2.3.1. Media tensión eléctrica

“Con abreviatura M.T. se define como cualquier conjunto de niveles de tensión comprendidos entre la alta tensión y la baja tensión. Los límites son $1 \text{ kV} < U \leq 35 \text{ kV}$, siendo U la Tensión Nominal” (CNE-Suministro, 2011).

2.3.2. Red de Baja Tensión

“Con abreviatura B.T. se define como: conjunto de niveles de tensión utilizados para la distribución de la electricidad. Su límite superior generalmente es $U \leq 1 \text{ kV}$, siendo U la Tensión Nominal” (CNE-Suministro, 2011).

2.3.3. Presupuesto de obra

“Es el presupuesto de la obra según la ubicación de las partidas en los metrados con recomendación de la información de análisis de precios unitarios, costos y presupuestos de la Cámara Peruana de la Construcción” (CAPECO, 2023).

2.3.4. Software libre

“El software libre o software de fuentes abiertas es un software cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier finalidad y redistribuido con cambios o mejoras sobre él ” (Robles, 2018).

2.3.5. Proyecto

“Se define un proyecto como el conjunto de documentos técnicos que contienen las especificaciones y características de una obra de construcción, así como los planos, memorias descriptivas, especificaciones técnicas, cálculos estructurales y demás documentos necesarios para su ejecución y supervisión, con el objetivo de garantizar su adecuada planificación, diseño y construcción de acuerdo con las normas y estándares establecidos” (RNE, 2006).

2.4. Base Normativa

La norma:

Código Nacional de Electricidad - CNE 2011

Esta norma es esencial para garantizar que el desarrollo e implementación del software se realice dentro de un marco normativo actualizado y conforme a los requerimientos establecidos para los sistemas eléctricos en el Perú. El **CNE 2011** regula

las condiciones y los procedimientos técnicos para la instalación, operación y mantenimiento de redes eléctricas, y su inclusión en el marco normativo del proyecto asegura que la gestión de costos y presupuestos en sistemas de distribución cumpla con las normativas actuales para la electrificación de áreas como Challabamba Alta.

Norma DGE 091

La Norma DGE 091, emitida por la Dirección General de Electricidad, establece las condiciones y procedimientos técnicos necesarios para asegurar que la infraestructura eléctrica en las zonas de concesión de distribución cumpla con las normativas de seguridad y calidad establecidas. Esta norma es esencial para garantizar que todas las instalaciones eléctricas, incluyendo las subestaciones y redes de distribución, se ejecuten conforme a las mejores prácticas del sector y estén alineadas con los requisitos regulatorios nacionales.

PROCEDIMIENTOS PARA LA ELABORACIÓN DE PROYECTOS Y EJECUCIÓN DE OBRAS EN SISTEMAS DE DISTRIBUCIÓN Y SISTEMAS DE UTILIZACIÓN EN MEDIA TENSIÓN EN ZONAS DE CONCESIÓN DE DISTRIBUCIÓN (RESOLUCION DIRECTORAL N° 018-2002-EM-DGE), menciona que:

Suministro Eléctrico (suministro), “es el bastecimiento regular de energía eléctrica del Concesionario al usuario dentro del régimen establecido por la Ley de Concesiones Eléctricas y su Reglamento” (DGE, 2002).

Usuario, “persona natural o jurídica que ocupa un predio y está en capacidad de hacer uso legal del suministro eléctrico correspondiente; es el responsable de cumplir con las obligaciones técnicas y económicas que se derivan de la utilización de la electricidad” (DGE, 2002)..

Consideraciones de Diseño:

- Subsistema de Distribución Primaria.
- Subsistema de Distribución Secundaria.
- Sistemas de Utilización en Media Tensión.
- Instalaciones de Alumbrado de Vías Públicas.

La norma técnica:

METRADOS PARA OBRAS DE EDIFICACIÓN Y HABILITACIONES URBANAS (SECCIÓN HU.4), Ministerio de Vivienda, Construcción y Saneamiento:

Obras Preliminares, Se considera obras preliminares a las actividades, el suministro y ejecución de labores correspondientes a la preparación del terreno para la ejecución de obras referente a habilitaciones urbanas (MVCS, 2010).

Cables De Energía, “Esta partida considera los cables de energía que se han de instalar fijados, adosados a los postes, así como en los que pasan de instalación subterránea a aéreo a través de sistemas de conductos o tuberías, correspondiente a los alimentadores, circuitos derivados, que salen de las subestaciones, se considerara a partir de las barras colectoras, seccionadores verticales u horizontales hasta el punto de alimentación del último usuario en cada circuito o punta muerta” (MVCS, 2010).

Estructura De Soporte, “Se considera postes y otros soportes que estén normados y cuyos fabricantes estén aceptados por la compañía eléctrica de distribución, así como los accesorios y ferretería de fijación de los cables de energía y de la propia estructura, tanto para media tensión y baja tensión” (MVCS, 2010).

Postes, “Son elementos de soporte de los cables aéreos que son instalados en lugar donde no moleste el tránsito vehicular y cuya distancia de separación con cables de energía debe ser de acuerdo al CNE-Suministro” (MVCS, 2010).

Redes Aéreas, “Esta partida considera los cables de energía que se han de instalar fijados, adosados a los postes, así como en los que pasan de instalación subterránea a aéreo a través de sistemas de conductos o tuberías, correspondiente a los alimentadores, circuitos derivados, que salen de las subestaciones, se considerara a partir de las barras colectoras, seccionadores verticales u horizontales hasta el punto de alimentación del último usuario en cada circuito o punta muerta” (MVCS, 2010).

Subestación De Distribución, “Mediante las subestaciones de distribución se logra reducir la media tensión a baja tensión para efectuar el suministro de energía eléctrica a cada usuario y al alumbrado público. La ejecución de las subestaciones de distribución en general, implican Movimiento de Tierras, Obras de concreto simple, obras de concreto armado, muros de albañilería, ductos, bases para equipos, electrodos de puesta a tierra, fabricaciones metálicas y acabados” (MVCS, 2010).

CAPÍTULO III: DISEÑO DE LA INTERFAZ GRÁFICA Y ESTRUCTURA FUNCIONAL DEL SOFTWARE

3.1. Introducción

Este capítulo detalla el diseño e implementación de la interfaz gráfica del software desarrollado para la gestión de costos y presupuestos en sistemas de distribución eléctrica. Se abordarán la arquitectura del sistema, los componentes visuales, la interacción con el usuario y la estructura funcional basada en el lenguaje Python y la librería ttkbootstrap para una experiencia moderna y accesible.

3.2. Requisitos del Sistema

Los requisitos del sistema definen las capacidades y condiciones que el software debe cumplir para garantizar su funcionalidad, usabilidad y eficiencia en el contexto del proyecto de electrificación Challabamba Alta. Se clasifican en funcionales (qué debe hacer el sistema) y no funcionales (cómo debe hacerlo).

3.2.1. Requisitos Funcionales

Los requisitos funcionales describen las acciones específicas que el software debe realizar:

3.2.1.1. Gestión de Partidas Presupuestarias

- ✓ El sistema debe permitir la creación, edición y eliminación de partidas de costos (MonRP y MonRS) desde archivos CSV.
- ✓ Debe calcular automáticamente subtotales, totales y rendimientos al modificar cantidades o precios unitarios.

3.2.1.2. Cálculo de Costos

- ✓ Costos directos (materiales, mano de obra, equipos).
- ✓ Gastos generales (gastos administrativos).
- ✓ Utilidad e impuestos.

3.2.1.3. Visualización de Datos

- ✓ Panel de resumen con totales.
- ✓ Panel de resumen con subtotales.

3.2.1.4. Integración con Herramientas Externas

Compatibilidad con DIREDCAD para actualización automática de metrados.

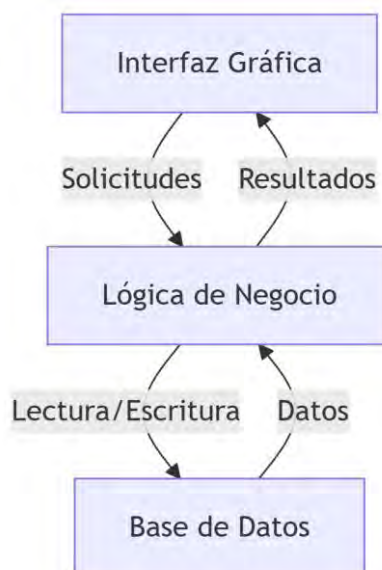
3.2.2. Requisitos No Funcionales

- ✓ Estos requisitos definen los criterios de calidad y rendimiento del sistema:
- ✓ Soporte para archivos CSV con hasta 10,000 registros sin degradación de performance.
- ✓ Funcionamiento en sistemas operativos Windows 10.
- ✓ Validación de datos para evitar inyección de código en archivos CSV.
- ✓ Soporte para escalado de fuentes en pantallas de alta resolución.

3.3. Arquitectura del Software

El sistema sigue una arquitectura modular de tres capas, separando la lógica de presentación, procesamiento y almacenamiento:

Figura 6
Arquitectura modular de tres capas



Nota. Elaboración propia.

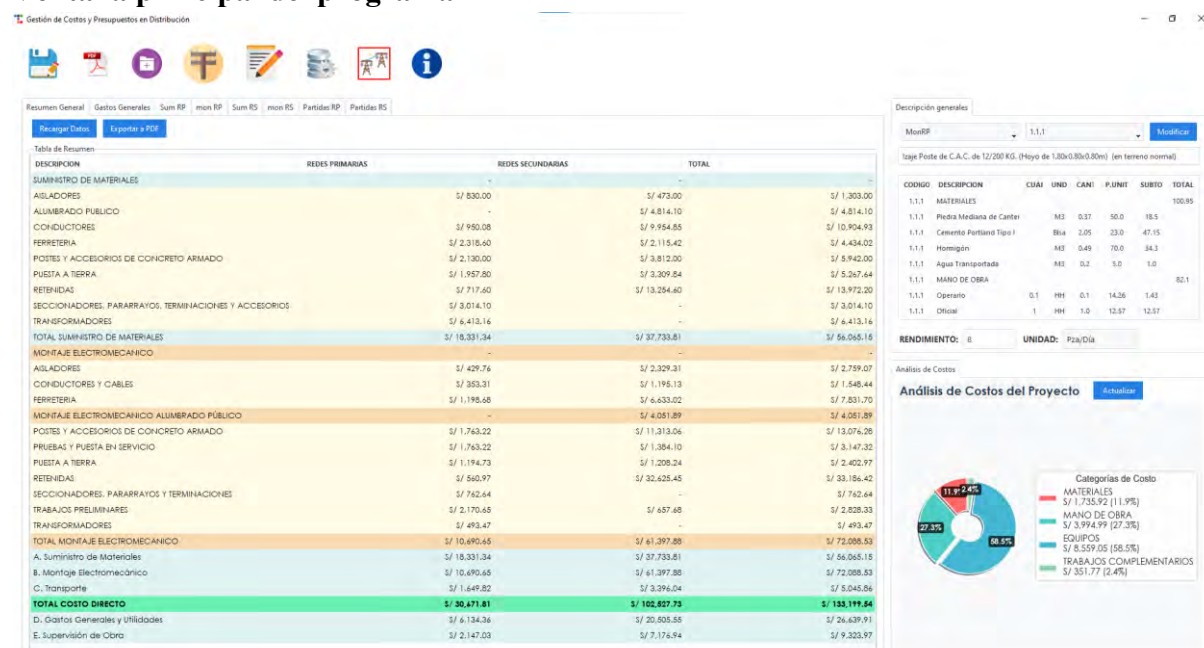
3.3.1. Componentes principales

3.3.1.1. Capa de Presentación (GUI)

Desarrollada con ttkbootstrap, una modernización de Tkinter, esta aplicación contiene:

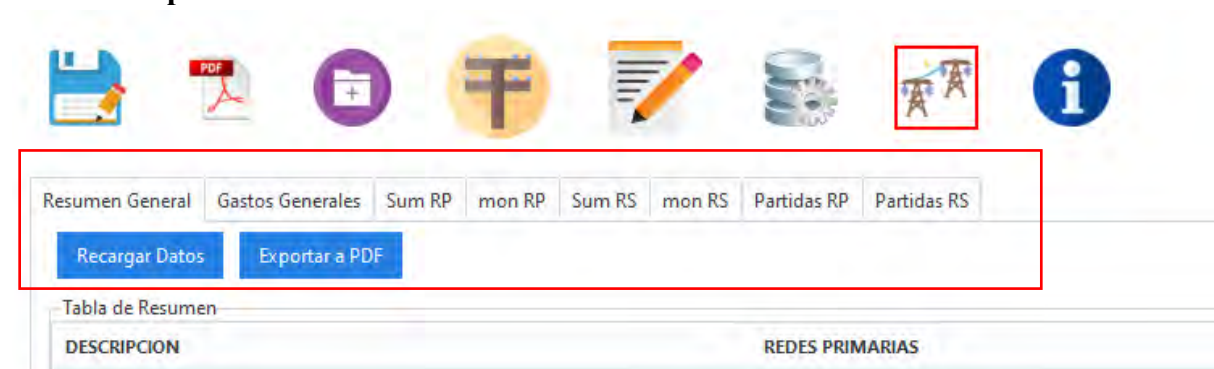
- ✓ Ventana principal con paneles acoplables
- ✓ Sistema de pestañas dinámicas
- ✓ Widgets especializados (Treeviews, Combobox inteligentes)

Figura 7
Ventana principal del programa



Nota. Elaboración propia.

Figura 8
Sistema de pestañas dinámicas



Nota. Elaboración propia.

3.3.1.2. Capa de Lógica

Módulos independientes por funcionalidad:

- ✓ B1_sum_rp.py: Cálculo de suministros en media tensión
- ✓ B2_mon_rp.py: Montos para partidas de media tensión
- ✓ K5_ACCS_automatizar_diredcad.py: Integración con DIREDCAD

3.3.1.3. Capa de Datos

Almacenamiento en archivos CSV estructurados:

- ✓ BBDD_partidasRP.csv: Partidas de red primaria
- ✓ BBDD_Relacion_de_Costos.csv: Precios unitarios

3.3.1.4. Ventajas

- ✓ Portabilidad
- ✓ Fácil interoperabilidad con Excel
- ✓ Bajo requerimiento de recursos

3.3.2. Librerías Clave

El sistema de gestión de costos y presupuestos se construye en Python 3.10+, sirviendo como el lenguaje de programación principal. Para una interfaz de usuario moderna y atractiva, se emplea ttkbootstrap 1.10, que mejora la apariencia de Tkinter con estilos inspirados en Bootstrap. La visualización de datos es posible gracias a Matplotlib 3.7+, permitiendo la generación de gráficos analíticos para una comprensión clara de los costos y presupuestos. Finalmente, Pandas 2.0+ es crucial para el procesamiento eficiente y estructurado de datos, especialmente aquellos importados desde archivos CSV, facilitando la manipulación y análisis de grandes volúmenes de información.

Tabla 3*Librerías clave para la elaboración del software*

Tecnología	Versión	Uso en el Sistema
Python	3.10+	Lenguaje base del desarrollo
ttkbootstrap	1.1	Modernización de la interfaz gráfica
Matplotlib	3.7+	Generación de gráficos analíticos
Pandas	2.0+	Procesamiento eficiente de datos CSV

Nota. Elaboración propia.

3.3.3. Flujo de Datos Principal

3.3.3.1. Entrada de Datos:

Cuando el usuario selecciona una partida específica desde un combobox en la interfaz, el sistema activa una función en Python diseñada para cargar los datos relevantes. Esta función, `cargar_partidas(archivo)`, abre el archivo CSV correspondiente a la partida seleccionada. Utiliza `csv.DictReader` para leer el contenido del archivo, interpretando cada fila como un diccionario donde las claves son los encabezados de las columnas. Finalmente, retorna una lista de estos diccionarios, lo que permite que el sistema acceda y utilice la información de manera estructurada y eficiente para la partida elegida.

Figura 9

Código para Cargar Datos de Partida

```
def cargar_partidas(archivo):
    with open(archivo, encoding='utf-8-sig') as f:
        return list(csv.DictReader(f))
```

Nota. Elaboración propia.

3.3.3.2. Procesamiento

El sistema realiza el cálculo de costos en tiempo real a medida que el usuario ingresa o modifica datos. Esto se logra mediante una función en Python llamada `calcular_subtotal`. Esta función toma dos parámetros, cantidad y precio, y multiplica estos valores para obtener el subtotal. Es importante destacar que los valores se convierten a tipo float para asegurar la

precisión en la operación y, finalmente, el resultado se redondea a dos decimales, lo que es crucial para manejar valores monetarios de forma exacta.

Figura 10

Código para el procesamiento

```
def calcular_subtotal(cantidad, precio):  
    return round(float(cantidad) * float(precio), 2)
```

Nota. Elaboración propia.

3.3.3.3. Salida

El sistema garantiza una actualización de los datos en varios componentes clave de la interfaz, lo que permite al usuario visualizar los cambios en tiempo real y mantener un seguimiento preciso de los costos y presupuestos.

Esta actualización se refleja de inmediato en:

- El **Treeview de partidas**: Aquí se muestra una tabla detallada con cada partida del proyecto, sus cantidades, precios y subtotales. Cualquier modificación o nuevo cálculo se verá reflejado instantáneamente en esta vista.
- El **Gráfico circular**: Esta representación visual ofrece un resumen porcentual de la distribución de los costos entre las diferentes partidas. A medida que los valores cambian, el gráfico se ajusta automáticamente, brindando una perspectiva rápida de la composición del presupuesto.
- El **Panel de resumen**: Este panel centraliza los totales y subtotales generales del proyecto, como el costo total y otros indicadores clave. Al igual que los demás componentes, se actualiza en tiempo real para reflejar cualquier cambio en los cálculos.

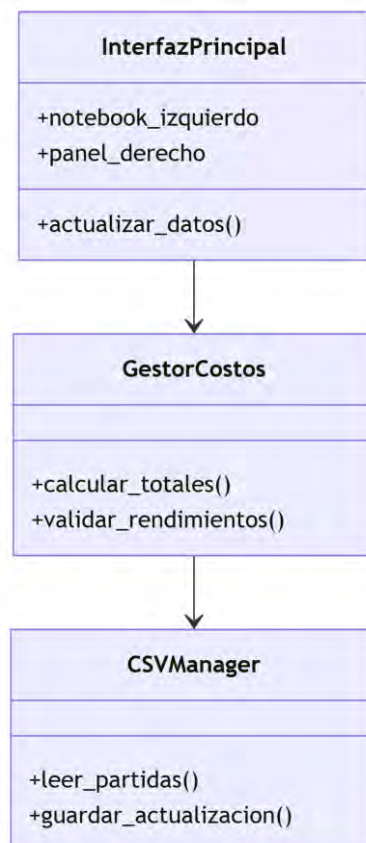
Esta sincronización en la actualización de la información es crucial para la toma de decisiones informadas, ya que permite a los usuarios observar el impacto de cada modificación de forma inmediata y comprender la situación económica del proyecto en todo momento.

3.3.4. Diagrama de Clases

El diagrama de clases del sistema se compone de tres clases principales interrelacionadas: InterfazPrincipal (gestión de la ventana y componentes visuales usando ttk.Notebook),

Con métodos como cargar_interfaz() y exportar_a_excel()), GestorCostos (núcleo lógico con atributos como partidas_activas y métodos como calcular_totales() y validar_rendimientos()), y CSVManager (manejo de datos con operaciones como leer_partidas() y validar_formato()). Estas clases siguen un flujo MVC: la Interfaz Principal solicita acciones al GestorCostos, que a su vez interactúa con CSVManager para leer/escribir archivos CSV, garantizando modularidad y separación de responsabilidades.

Figura 11
Diagrama de clases

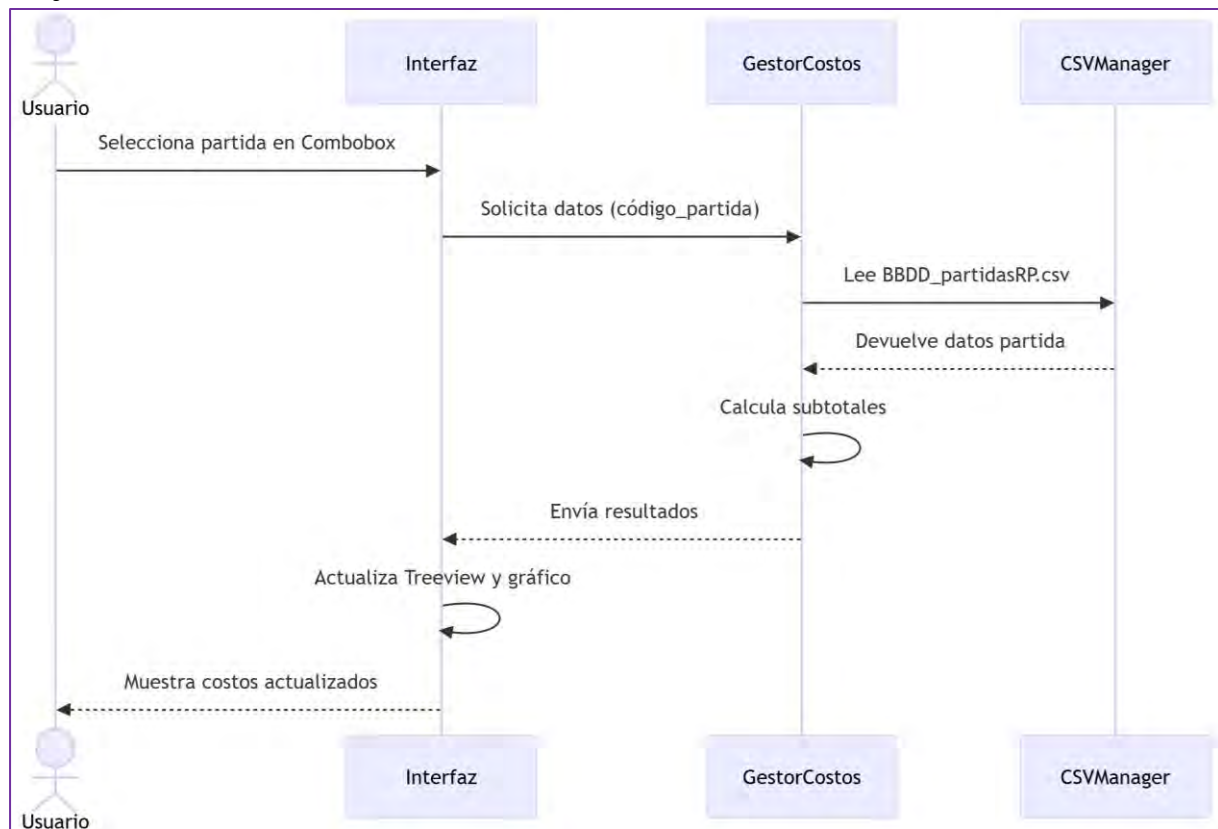


Nota. Elaboración propia.

3.4. Flujo de Datos (Diagrama de secuencia)

El sistema sigue un flujo bien definido cuando el usuario interactúa con la interfaz. Al seleccionar una partida en el Combobox, la interfaz envía una solicitud al módulo GestorCostos, que a su vez recupera los datos del archivo CSV correspondiente a través del CSVManager. Una vez obtenidos los datos, se realizan los cálculos necesarios y los resultados se envían de vuelta a la interfaz para su visualización. Este proceso garantiza una respuesta ágil, manteniendo actualizados los valores mostrados al usuario en todo momento.

Figura 12
Flujo de Datos



Nota. Elaboración propia.

3.5. Funcionalidades Clave

3.5.1. Gestión Dinámica de Partidas

La interfaz cuenta con un Combobox inteligente que filtra automáticamente las partidas disponibles según el tipo seleccionado (MonRP o MonRS). Este componente no solo facilita

la búsqueda, sino que también incorpora validaciones en tiempo real para prevenir errores comunes, como ingresar valores negativos o caracteres no numéricos en campos críticos.

3.5.2. Cálculo en Tiempo Real

Cada modificación en los campos de cantidad, rendimiento o precio unitario desencadena una serie de recálculos inmediatos. El sistema actualiza los subtotales, totales por categoría y el monto general, proporcionando retroalimentación instantánea al usuario. Esta característica es fundamental para agilizar el proceso de presupuestación.

3.5.3. Visualización de Datos

La representación gráfica de los costos mediante gráficos interactivos permite una comprensión rápida de la distribución del presupuesto. Los usuarios pueden identificar visualmente las partidas más significativas y cómo contribuyen al costo total del proyecto.

3.6. Integración con DIREDCAD

3.6.1. Descripción General del Sistema de Importación

Una de las mayores ventajas de este software libre es su capacidad para integrarse no solo con DIREDCAD, sino también con otras herramientas de diseño como REDCAD, AutoCAD, DLTCAD y más, según los requerimientos específicos de cada usuario. A diferencia de los sistemas comerciales cerrados, cuya funcionalidad es limitada y rígida, esta solución permite modificar el código fuente para adaptarse a distintos formatos y flujos de trabajo.

Por ejemplo, si un usuario necesita importar metrados desde AutoCAD, puede desarrollarse un módulo adicional que lea archivos .DWG mediante librerías como pyautocad. De igual manera, si otro proyecto requiere compatibilidad con REDCAD, basta con ajustar el sistema para interpretar sus archivos nativos. Esta flexibilidad es clave en entornos donde se utilizan múltiples herramientas de diseño, permitiendo que el software se adapte a las necesidades particulares en lugar de obligar al usuario a cambiar su metodología de trabajo.

Además, al ser de código abierto, estas adaptaciones pueden ser compartidas y mejoradas por la comunidad, enriqueciendo continuamente las capacidades del sistema sin incurrir en costos adicionales de licencias o desarrollos propietarios. Esto lo convierte en una solución verdaderamente escalable y personalizable, superando las limitaciones de los softwares comerciales que ofrecen poca o ninguna capacidad de modificación.

La integración con DIREDCAD es solo el punto de partida: la arquitectura del sistema está diseñada para extenderse y conectarse con otras herramientas según sea necesario, garantizando que cada usuario pueda optimizar su proceso de presupuesto de acuerdo con sus propias herramientas y flujos de trabajo.

Figura 13
Importación de datos desde archivos generados en DiredCAD



Nota. Elaboración propia.

3.6.2. Tipos de Archivos y Formatos Soportados

El sistema está configurado para trabajar con archivos Excel en formatos .xlsx y .xls, lo que garantiza la compatibilidad con las versiones más recientes y anteriores de Microsoft Excel. Los archivos RP (Recursos y Presupuestos) contienen información técnica detallada

sobre equipos eléctricos y sus especificaciones. Por ejemplo, incluyen datos sobre transformadores monofásicos con especificaciones como capacidad (10 kVA), tensiones primarias y secundarias (22.9 kV / 0.46 kV / 0.23 kV), frecuencia (60 Hz), nivel de aislamiento (150 kV BIL), tipo de refrigeración (ONAN), altitud de diseño (3800 msnm).

Tabla 4

Especificaciones Técnicas Detalladas del Proyecto Challabamba Alta

Parámetro	Especificación	Descripción
Capacidad	10 kVA monofásico	Potencia nominal del transformador
Tensión Primaria	22.9 kV ($\pm 2 \times 2.5\%$)	Voltaje de entrada con tolerancias
Tensión Secundaria	0.46 kV / 0.23 kV	Voltajes de salida del sistema
Frecuencia	60 Hz	Frecuencia de operación estándar
Aislamiento	150 kV BIL	Nivel Básico de Impulso
Refrigeración	ONAN	Oil Natural Air Natural
Altitud	3800 msnm	Altitud máxima de operación
Aplicación	Distribución rural	Uso en zonas de alta montaña

Nota. Elaboración propia.

Por otro lado, los archivos RS (Resumen de Suministros) contienen planillas estructuradas de elementos constructivos, específicamente "PLANILLA DE ESTRUCTURAS TIPO". Estos archivos incluyen información detallada sobre circuitos eléctricos, soportes, distancias entre postes, tipos de conductores, clasificación de postes, sistemas de retenidas, puestas a tierra (PAT), puntos de apoyo (AP) y diferentes tipos de armado. Los datos se organizan en columnas específicas que incluyen números de circuito, números de soporte, distancias en metros, tipos de conductor (como 8/200, 8/300), clasificaciones de postes (E1, E3, E4, E5, E6), cantidades de retenidas (RI, RV, RC), y especificaciones de armados tanto cortos como largos. Esta estructura tabular permite un control preciso de cada elemento de la infraestructura eléctrica del proyecto.

Tabla 5*Estructura de Datos del Archivo DIREDCAD RS (Planilla de Estructuras)*

Campo	Descripción	Valores Típicos	Función
Circuito	Número del circuito eléctrico	1, 2	Identificación del sistema
Soporte	Número del poste/soporte	1, 2, 3...	Secuencia de estructuras
Distancia	Separación entre postes (m)	20.93, 30.95, 41.73	Cálculo de vanos
Conductor	Tipo de cable utilizado	8/200, 8/300	Especificación técnica
Nº Postes BT/MT	Cantidad de postes por tipo	0, 1	Conteo de estructuras
Clase Poste	Clasificación estructural	E1, E3, E4, E5, E6	Tipo de armado
Retenidas	Sistemas de anclaje	RI, RV, RC	Estabilización
PAT	Puesta a tierra	0, 1, 2	Sistema de seguridad
AP	Puntos de apoyo	0, 1	Elementos de soporte
Armado	Configuración de montaje	Corta, Larga	Tipo de instalación

Nota. Elaboración propia.

Tabla 6*Ejemplo de Datos Procesados del Archivo RS*

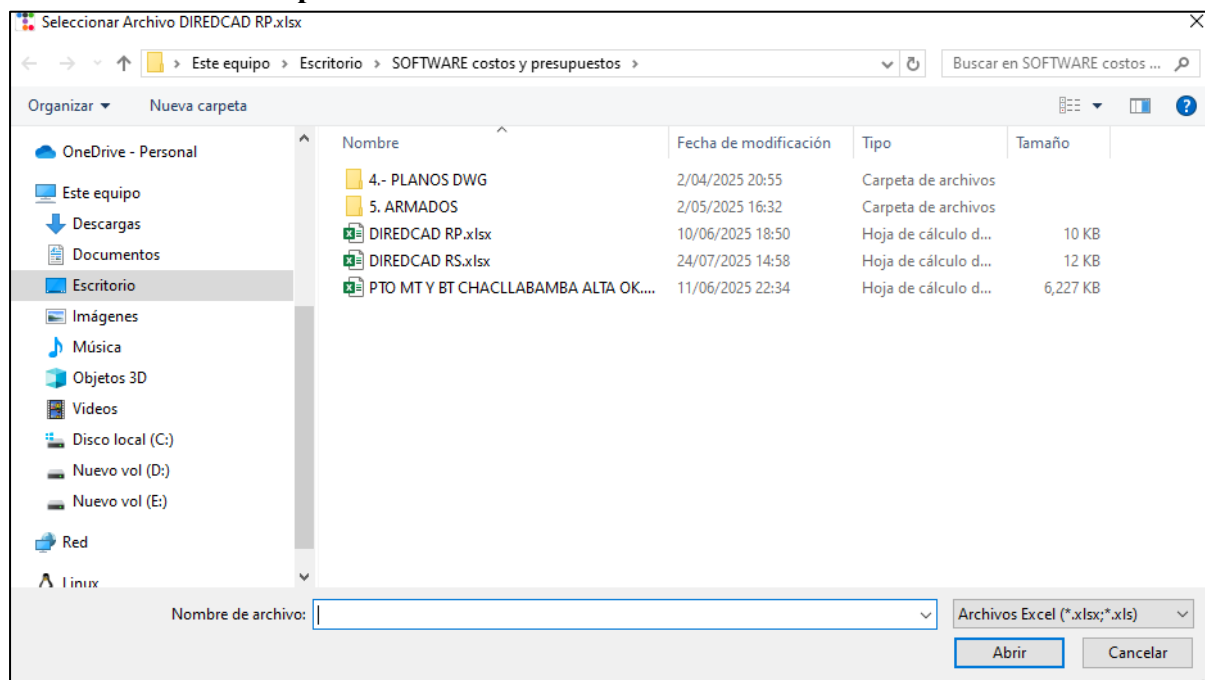
Circuito	Soporte	Distancia (m)	Conductor	Clase Poste	Retenidas	PAT	Armado
1	1	20.93	8/200	E1	1	0	Corta
1	2	30.95	8/300	E4	0	0	Larga
1	3	30.24	8/200	E1	2	0	Corta
1	4	34.85	8/300	E4	1	0	Larga
2	1	25.89	8/300	E4	0	0	Corta
2	2	44.14	8/300	E4	0	0	Larga

Nota. Elaboración propia.

3.6.3. Funcionalidades de la Interfaz de Usuario

Los botones "Explorar" para cada tipo de archivo funcionan como interfaces de selección que abren diálogos específicos del sistema operativo. Estos diálogos están configurados con filtros que restringen la selección únicamente a archivos Excel, evitando errores de formato durante la importación. Cuando el usuario selecciona un archivo, el sistema automáticamente valida su existencia y actualiza la interfaz con el nombre del archivo seleccionado, proporcionando retroalimentación visual inmediata.

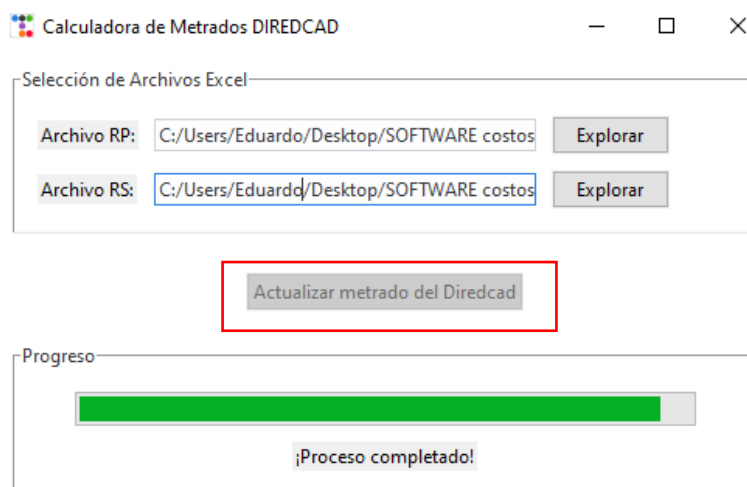
Figura 14
Interfaz del boton Explorar



Nota. Elaboración propia.

El botón "Actualizar metrado del Direcad" representa el núcleo funcional del sistema de importación. Este botón inicia un proceso multifase que incluye validaciones previas, procesamiento de datos y generación de resultados. Antes de comenzar el procesamiento, el sistema verifica que ambos archivos hayan sido seleccionados y muestra advertencias específicas si alguno falta.

Figura 15
Interfaz del boton Actualizar metrado del Direcad



Nota. Elaboración propia.

Una vez iniciado el proceso, el botón se deshabilita temporalmente para prevenir ejecuciones múltiples que podrían causar conflictos en el procesamiento.

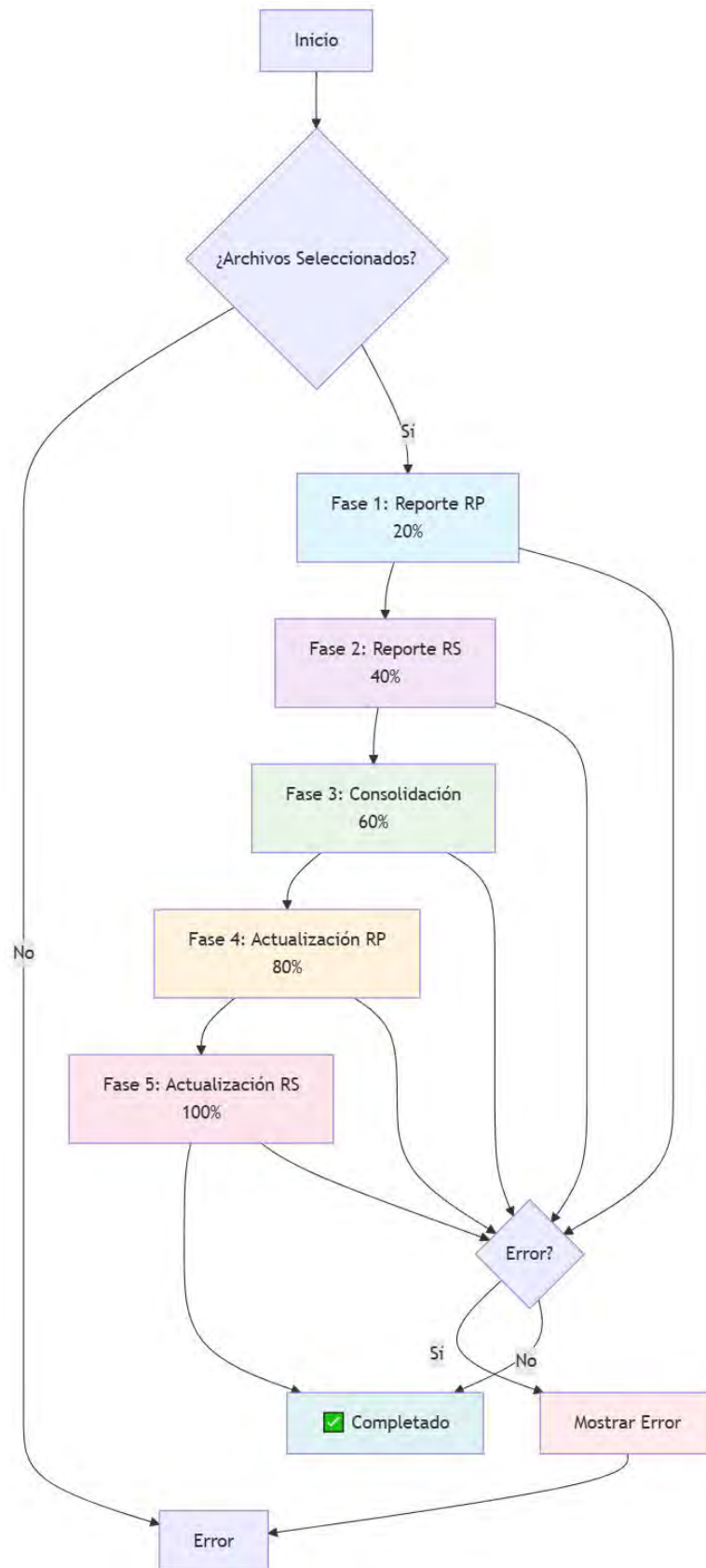
3.6.4. Proceso Detallado de Importación y Procesamiento

El proceso de importación se ejecuta en cinco fases distintas, cada una con objetivos específicos e indicadores de progreso visual, tal como se ilustra en la siguiente figura. Este diagrama de flujo muestra la secuencia completa del proceso, incluyendo las validaciones, el manejo de errores y los puntos de control de calidad.

La primera fase se enfoca en la generación del reporte RP, donde el sistema lee y analiza los datos técnicos de equipos eléctricos contenidos en el archivo Excel seleccionado. Durante esta fase, que representa el 20% del proceso total, el sistema procesa especificaciones técnicas como parámetros de transformadores, incluyendo capacidades, tensiones, frecuencias, niveles de aislamiento y condiciones de operación. El sistema convierte estos datos del formato Excel a estructuras de datos internas más eficientes, organizando la información técnica de manera que pueda ser procesada conjuntamente con los datos estructurales.

La segunda fase involucra la generación del reporte RS, alcanzando el 40% del progreso total. Aquí, el sistema procesa las planillas de estructuras eléctricas, analizando información detallada sobre circuitos, tipos de postes, distancias, conductores, sistemas de retenidas y configuraciones de armado. Esta fase es crucial porque organiza datos estructurales complejos que incluyen elementos como postes clase E1, E3, E4, E5 y E6, conductores tipo 8/200 y 8/300, sistemas de puesta a tierra (PAT), y especificaciones de retenidas en diferentes configuraciones. El sistema establece las bases para la integración de la información técnica de equipos con los datos estructurales de la infraestructura eléctrica.

Figura 16
Diagrama de Flujo del Proceso de Importación



Nota. Elaboración propia.

La tercera fase, que lleva el progreso al 60%, se centra en la consolidación y exportación de los datos procesados. Durante esta etapa, el sistema fusiona la información de ambos archivos, eliminando duplicados y resolviendo conflictos de datos. El resultado es un archivo CSV consolidado denominado 'BBDD_metrado_DIREDCAD.csv' que contiene toda la información integrada y lista para su uso posterior.

Las fases cuarta y quinta se dedican a las actualizaciones y refinamientos finales de los datos. Estas incluyen cálculos adicionales, actualizaciones monetarias y verificaciones de integridad que aseguran la precisión de los resultados finales. El progreso avanza del 70% al 100% durante estas fases, proporcionando al usuario una indicación clara del estado del procesamiento.

3.7. Limitaciones Identificadas

A pesar de sus ventajas, el sistema presenta algunas limitaciones importantes. La dependencia de archivos CSV, aunque sencilla de implementar, puede generar problemas con formatos inconsistentes o cuando múltiples usuarios necesitan acceder simultáneamente a los datos. Además, el rendimiento puede verse afectado al trabajar con proyectos muy grandes que superen los 50,000 registros.

3.8. Conclusiones del capítulo

El desarrollo de este software de gestión de costos y presupuestos para sistemas de distribución eléctrica ha demostrado que una arquitectura modular y de código abierto ofrece ventajas significativas frente a soluciones comerciales. A través de un diseño centrado en la usabilidad y adaptabilidad, se logró:

1. **Interoperabilidad Efectiva:** La integración nativa con DIREDCAD y la capacidad de extenderse a otras herramientas (REDCAD, AutoCAD, DLTCAD, etc.) mediante modificaciones al código permiten que el sistema se ajuste a diversos flujos de trabajo, eliminando la dependencia de formatos cerrados y costosos.

2. **Automatización y Precisión:** La automatización de procesos como el cálculo de costos en tiempo real y la importación de metrados desde software de diseño reduce errores humanos y agiliza la elaboración de presupuestos, optimizando tiempos que antes requerían horas de trabajo manual.
3. **Flexibilidad y Escalabilidad:** Al ser software libre, el sistema puede ser modificado y mejorado según las necesidades específicas de cada proyecto o institución, algo imposible en alternativas comerciales. Esto incluye desde la conexión con nuevas herramientas hasta la optimización de algoritmos para grandes volúmenes de datos.
4. **Costo Accesible y Adaptación Local:** A diferencia de los sistemas propietarios, esta solución evita licencias costosas y está diseñada para trabajar con estándares y normativas peruanas desde su concepción, lo que la hace ideal para proyectos públicos y privados en el país.

CAPÍTULO IV: IMPLEMENTACIÓN Y VALIDACIÓN DEL SOFTWARE EN EL PROYECTO CHALLABAMBA ALTA

4.1. Introducción

El presente capítulo documenta la implementación práctica del software de gestión de costos en el proyecto de electrificación rural de Challabamba Alta. Este caso de estudio permitió validar la eficacia de la herramienta en condiciones de trabajo, donde todos los metrados se realizaron a partir de planos definitivos tanto para red primaria (RP) como red secundaria (RS). Esta implementación demostró la capacidad del sistema para interpretar diseños técnicos complejos y convertirlos automáticamente en estructuras de costos precisas.

4.2. Aplicación del software al proyecto de electrificación Challabamba Alta

La implementación del software se basó completamente en los planos del proyecto, asegurando la precisión en los metrados.

4.3. Descripción general del proyecto Challabamba Alta

El proyecto de electrificación rural Challabamba Alta, ubicado en el distrito de Challabamba, provincia de Paucartambo, Cusco, busca ampliar la cobertura eléctrica en zonas de difícil acceso para mejorar la calidad de vida y fomentar el desarrollo socioeconómico de las comunidades. Este proyecto implica la instalación de infraestructura eléctrica de media y baja tensión para suministrar energía continua a hogares y locales comunales en un terreno accidentado con accesos limitados.

Este proyecto servirá como caso piloto para validar el software desarrollado en esta tesis. La nueva herramienta permite estructurar partidas presupuestales y generar salidas compatibles con hojas de cálculo, optimizando así la formulación de proyectos eléctricos rurales.

Tabla 7*Especificaciones Técnicas Detalladas del Proyecto Challabamba Alta*

PARÁMETRO TÉCNICO	MEDIA TENSIÓN (MT)	BAJA TENSIÓN (BT)	DETALLES TÉCNICOS
EXTENSIÓN DE REDES	0.442 km	1.193 km	Total proyecto: 1.635 km lineales
CANTIDAD DE POSTES	2 postes	32 postes	Total: 34 postes de concreto armado centrifugado
ESPECIFICACIÓN POSTES MT	Postes C.A.C. 12/300/150/330 kg	-	Altura: 12m, Peso: 300kg, Diámetro base: 150mm
ESPECIFICACIÓN POSTES BT	-	14 postes 8/200 kg + 18 postes 8/300 kg	Altura: 8m, Pesos: 200kg y 300kg respectivamente
CONDUCTORES MT	AAAC 35 mm ² , 7 hilos	-	Aleación aluminio 6201-T81, para 22.9 kV
CONDUCTORES BT	-	CAAI 2x25+16/N25 mm ²	Cable autoportante con neutro portante aislado
TRANSFORMADOR	1 unidad monofásico 10 kVA	-	Relación: 22.9 kV/0.46-0.23 kV
			Aislación: 150 kV BIL
			Refrigeración: ONAN
			Altitud operación: 3800 msnm
PROTECCIONES MT	4 seccionadores Cut-Out 27 kV	-	Capacidad corte: 10 kA
	4 pararrayos 21 kV		Nivel básico de impulso: 150 kV
SISTEMA DE PUESTA A TIERRA	4 PAT (3 PAT-1 + 1 PAT-0)	6 PAT-1	Varillas de cobre: 19mm Ø x 2.40m
			Cemento conductivo: 25 kg por PAT
			Resistencia requerida: <10 Ω
AISLADORES MT	2 PIN ANSI 56-4 + 4 suspensión RPP-25	-	Tensión nominal: 25-36 kV
			Línea de fuga: 660-685 mm
AISLADORES BT	-	15 PIN ANSI 56-2	Para retenidas y apoyos secundarios
SISTEMA DE RETENIDAS	2 retenidas aisladas	15 retenidas (11 simples + 4 verticales)	Cable acero: Ø10 mm galvanizado
			Templadores: 255x16mm

PARÁMETRO TÉCNICO	MEDIA TENSIÓN (MT)	BAJA TENSIÓN (BT)	DETALLES TÉCNICOS
			Planchas anclaje: 300x300x6.4mm
ALUMBRADO PÚBLICO	-	9 luminarias LED 50W	Tensión: 120-277V
			IP66: Protección total contra polvo y agua
			Pastorales: acero galvanizado
ARMADOS Y FERRETERIA	3 tipos de armados (DT-2, AB4, AB5)	7 tipos de armados (E0 a E6)	Material: Acero galvanizado
			Crucetas: perfil angular 64x64x2500mm
TOPOGRAFÍA Y ACCESOS	Terreno mixto (normal/rocoso)	Terreno mixto (normal/rocoso)	Accesos: Limitados
			Altitud promedio: 3800 msnm
ESTUDIOS Y PROYECTOS	Ingeniería de detalle MT	Ingeniería de detalle BT	
	Replanteo topográfico MT	Replanteo topográfico BT	

Nota. Elaboración propia en base a las especificaciones y requerimientos del proyecto.

Tabla 8

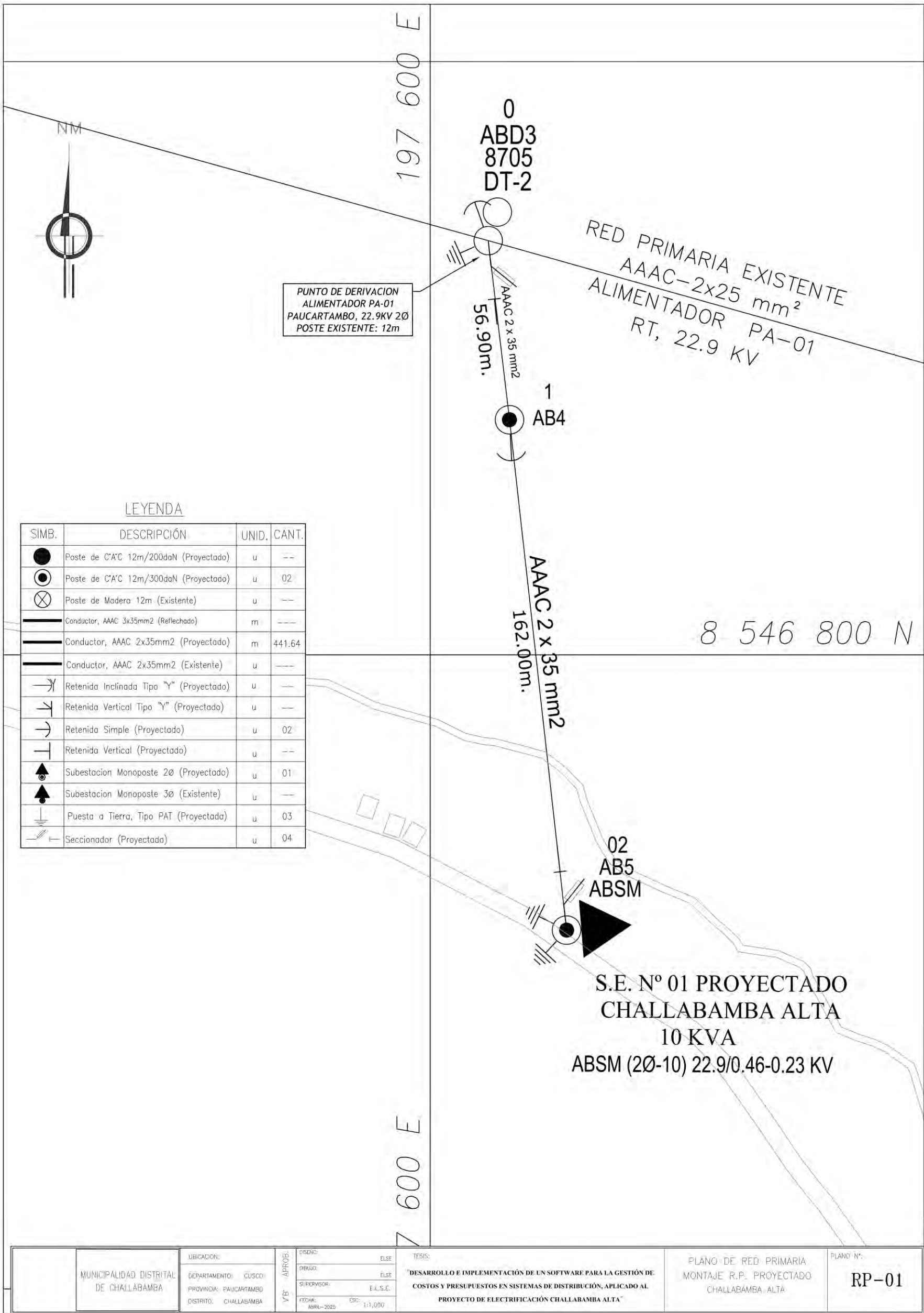
Características del Transformador Monofásico 10 kVA

Parámetro	Especificación
Capacidad	10 kVA monofásico
Tensiones	Primario: 22.9 kV ($\pm 2 \times 2.5\%$)
	Secundario: 0.46 kV / 0.23 kV
Frecuencia	60 Hz
Aislamiento	150 kV BIL (Nivel Básico de Impulso)
Refrigeración	ONAN (Oil Natural Air Natural)
Altitud	Diseñado para 3800 msnm
Aplicación	Distribución rural en zonas de alta montaña

Nota. Elaboración propia en base a las especificaciones y requerimientos del proyecto

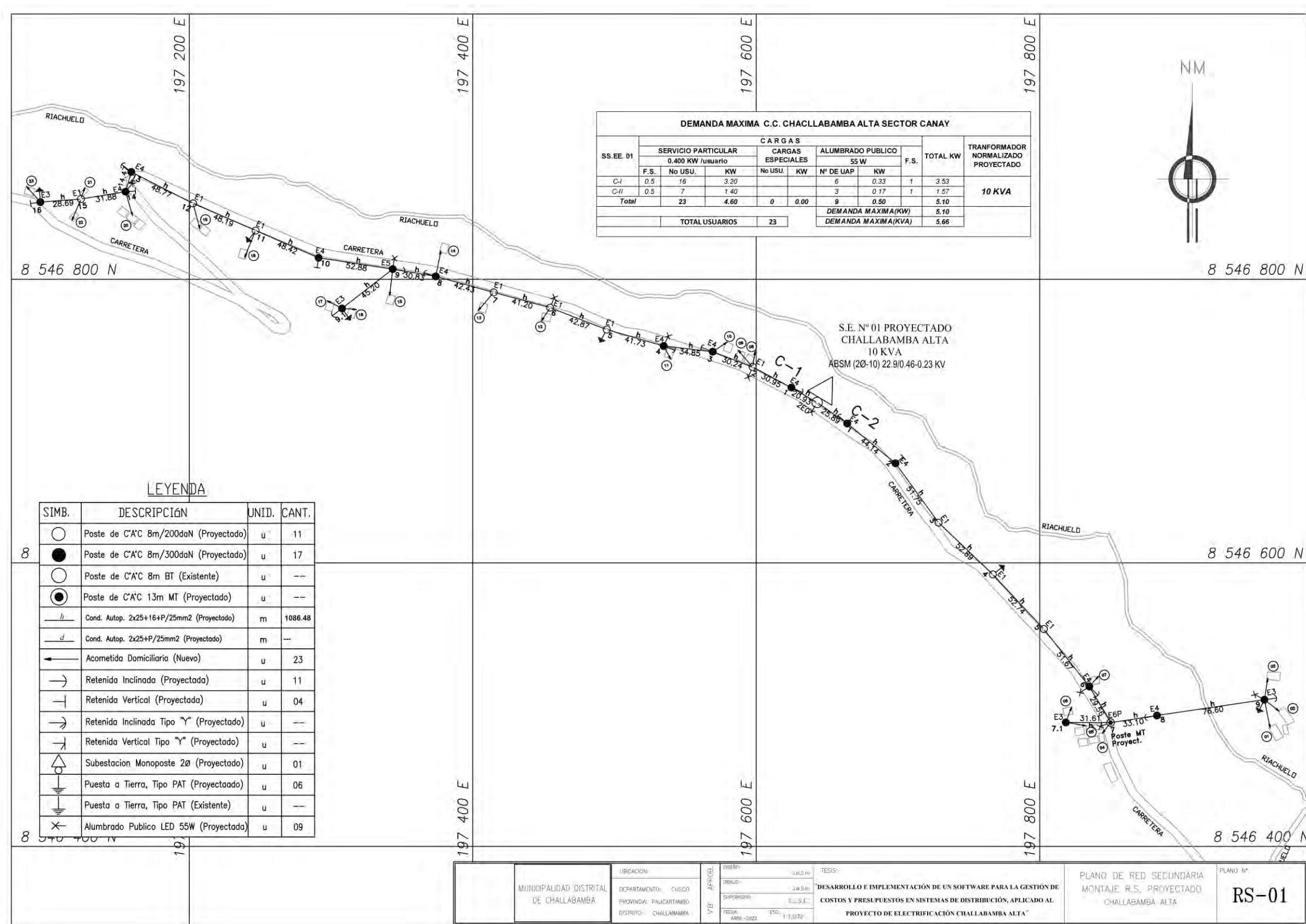
Estas tablas resumen los componentes técnicos que justifican los costos del proyecto. Servirán como base para elaborar el presupuesto y, consecuentemente, como referencia para validar el software desarrollado en la tesis.

Plano Red primaria proyecto Challabamba Alta



Nota. Elaboración propia.

Figura 18
Plano Red secundaria proyecto Challabamba Alta



Nota. Elaboración propia.

4.4. Datos técnicos empleados para la validación

Para validar la funcionalidad del software desarrollado, se emplearon datos reales del proyecto de electrificación rural Challabamba Alta, ubicado en la provincia de Paucartambo, región Cusco. Este proyecto incluyó la instalación de redes primarias y secundarias, así como sistemas de alumbrado público, puesta a tierra, y otros componentes fundamentales en un sistema de distribución eléctrica.

Los datos técnicos se obtuvieron principalmente a través de los planos eléctricos elaborados en AutoCAD y validados en DIREDCAD, los cuales fueron exportados en formatos compatibles para su procesamiento. Se consideraron los siguientes elementos:

Componentes eléctricos: postes, conductores, aisladores, transformadores, pararrayos, seccionadores, retenidas, etc.

Actividades: montaje electromecánico, suministro de materiales, trabajos preliminares, pruebas y puesta en servicio.

Categorías de red: clasificación por redes primarias y redes secundarias.

Unidades y precios: costos unitarios por partida, extraídos de bases de datos presupuestales estandarizadas.

Figura 19
Relación de insumos RP

CODIGO	DESCRIPCION	UNIDAD	CANTIDAD	PRECIO UNITARIO	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				\$/ 2,130.00
▣ __1.1.0	POSTES CONCRETO ARMADO CENTRIFUGADO DE:				
_ 1.1.1	12/200/140/320	u	0.0	\$/ 900.00	\$/ 0.00
_ 1.1.2	12/300/150/330	u	2.0	\$/ 990.00	\$/ 1,980.00
▣ __1.2.0	ACCESORIOS DE C.A				
_ 1.2.1	Caja de Concreto Para P.T. (0.30 x 0.40 x 0.40) m, con tapa de espesor de 0.05 m de C.V.	u	3.0	\$/ 50.00	\$/ 150.00
2.0.0	AISLADORES				\$/ 830.00
▣ __2.1.0	AISLADORES DE PORCELANA, CLASE ANSI:				
_ 2.1.1	TIPO PIN CLASE ANSI 56-4, 25 KV, 685 mm LINEA DE FUGA	u	2.0	\$/ 65.00	\$/ 130.00
▣ __2.2.0	AISLADORES POLIMÉRICOS TIPO SUSPENSIÓN :				
_ 2.2.1	RPP-25, Tipo Clevis-Ojo, Vtípico 25kV, Distancia de Fuga Mínima 660mm	u	4.0	\$/ 120.00	\$/ 480.00
_ 2.2.2	Tipo Clevis-Ojo, Vtípico 36kV, Distancia de Fuga Mínima 660 mm (para Retenida Aislada)	u	2.0	\$/ 110.00	\$/ 220.00
3.0.0	CONDUCTORES				\$/ 950.08
▣ __3.1.0	CONDUCTOR DE COBRE DESNUDO, CABLEADO, TEMPLE BLANDO, DE :				
_ 3.1.1	25mm2, 7 HILOS (Para PAT-1)	m	54.0	\$/ 12.50	\$/ 675.00
_ 3.1.2	25mm2, 7 HILOS (Para PAT-0)	m	15.0	\$/ 12.50	\$/ 187.50
▣ __3.2.0	CONDUCTOR DE ALEACION DE ALUMINIO 6201-T81 [AAAC] CABLEADO DE :				
_ 3.2.1	35mm2, 7 HILOS	m	6.0	\$/ 1.93	\$/ 11.58
▣ __3.3.0	ALAMBRE DE ALUMINIO, PARA AMARRE DE :				
_ 3.3.1	10mm2, 7 HILOS	m	5.0	\$/ 1.20	\$/ 6.00
▣ __3.4.0	CONDUCTOR DE ALUMINIO, TEMPLE BLANDO AISLADO DE :				
_ 3.4.1	35mm2, 7 HILOS	m	6.0	\$/ 8.00	\$/ 48.00
▣ __3.5.0	PLANCHA DE COBRE :				
_ 3.5.1	PLANCHA DE COBRE TIPO 'J'	m	4.0	\$/ 5.50	\$/ 22.00
4.0.0	FERRETERIA				\$/ 2,316.60
▣ __4.1.0	ARMADO DT-2	Cjto	1.0		
_ 4.1.1	Abrazadera Tipo CAS Simple de 64mm, E=6.4mm, D=200mm, C/4P/4T/4C/8A/4AP.	Und	1.0	\$/ 51.00	\$/ 51.00
_ 4.1.2	Abrazadera de A"G" P/Postes de CAC de Fijación Tipo U de 1800X75X6.4mm.	Und	1.0	\$/ 45.00	\$/ 45.00

Actualizar y Guardar

Nota. Visualización generada por el software.

Figura 20
Relación de insumos RS

CODIGO	DESCRIPCION	UNIDAD	CANTIDAD	PRECIO UNITARIO	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				\$/ 3,812.00
▣ __1.1.0	POSTES CONCRETO ARMADO CENTRIFUGADO DE:				
_ 1.1.1	POSTE DE C.A.C. 8/200/140/260	u	2.0	\$/ 441.00	\$/ 882.00
_ 1.1.2	POSTE DE C.A.C. 8/300/150/270	u	5.0	\$/ 526.00	\$/ 2,630.00
▣ __1.2.0	ACCESORIOS DE C.A				
_ 1.2.1	Caja de Concreto Para P.T. (0.30 x 0.40 x 0.40) m, con tapa de espesor de 0.05 m de C.V.	u	6.0	\$/ 50.00	\$/ 300.00
2.0.0	AISLADORES				\$/ 473.00
▣ __2.1.0	AISLADORES DE PORCELANA, CLASE ANSI:				
_ 2.1.1	54-2, PARA (RETENIDAS)	u	43.0	\$/ 11.00	\$/ 473.00
3.0.0	CONDUCTORES				\$/ 9,954.85
▣ __3.1.0	CONDUCTOR DE COBRE DESNUDO, CABLEADO, TEMPLE BLANDO, DE :				
_ 3.1.1	25mm2, 7 HILOS (Para puesta a tierra)	m	60.0	\$/ 9.28	\$/ 556.80
▣ __3.2.0	CABLE AUTOPORTANTE DE ALUMINIO TIPO CAAI, C/ PORTANTE DE AL-AL AISLADO:				
_ 3.2.1	2x25+16/N25 mm2	m	1086.4799999999999	\$/ 8.65	\$/ 9,398.05
_ 3.2.2	2x25/N25 mm2	m	0.0	\$/ 6.14	\$/ 0.00
4.0.0	FERRETERIA				\$/ 2,115.42
▣ __4.1.0	ARMADO E0	Cjto	2.0		
_ 4.1.1	GRAPA DE ANCLAJE CONICA TIPO COCODRILO	Und	2.0	\$/ 18.00	\$/ 36.00
_ 4.1.2	HEBILLA PARA FLEJE DE ACERO INOXIDABLE 19 mm	Und	4.0	\$/ 2.15	\$/ 8.60
_ 4.1.3	FLEJE DE ACERO INOXIDABLE 19mm	Und	4.0	\$/ 6.50	\$/ 26.00
_ 4.1.4	PLANCHA GANCHO DE SUSPENSIÓN ESFUERZO DE CARGA=15 kn	Und	2.0	\$/ 22.00	\$/ 44.00
▣ __4.2.0	ARMADO E1	Cjto	11.0		
_ 4.2.1	GRAPA DE SUSPENSIÓN DE ALUMINIO CUBIERTA DE PINTURA PLASTIFICADA (SO 14.1 O SIMILAR)	Und	11.0	\$/ 18.00	\$/ 198.00
_ 4.2.2	PERNO CON GANCHO DE SUSPENSIÓN D=16 mm, L=228 mm, C/T/CT/2A/AP (SOT 15.8 O SIMILAR)	Und	11.0	\$/ 15.00	\$/ 165.00
_ 4.2.3	ARANDELA CUADRADA CURVA DE 37X37X3MM, D=17.5MM	Und	22.0	\$/ 4.00	\$/ 88.00
▣ __4.3.0	ARMADO E3	Cjto	4.0		
_ 4.3.1	GRAPA DE ANCLAJE CÓNICA TIPO COCODRILO (SO 3.25 O SIMILAR)	Und	4.0	\$/ 18.00	\$/ 72.00
_ 4.3.2	PERNO CON OJAL D=16 mm, L=228 mm, C/T/CT/2A/AP	Und	4.0	\$/ 18.00	\$/ 72.00

Nota. Visualización generada por el software.

4.5. Resultados obtenidos con el software desarrollado

Al aplicar el software de acuerdo a los datos técnicos del proyecto Challabamba Alta, se obtuvieron los siguientes resultados:

- **Generación automática de partidas presupuestales**, organizadas jerárquicamente por tipo de red (primaria o secundaria) y tipo de actividad (suministro, montaje, transporte).
- **Cálculo inmediato de subtotales**, discriminados por componente (conductores, alumbrado, puesta a tierra, etc.).
- **Visualización estructurada del presupuesto**, de acuerdo a la actividad, código de ítem, o nivel jerárquico.
- **Reducción significativa del tiempo de elaboración presupuestal**, debido a la automatización de la codificación, lectura y estructuración de datos.

Los resultados coinciden con los valores obtenidos por otros métodos y softwares comerciales, validando la precisión y funcionalidad del sistema propuesto.

4.6. Presupuesto total generado automáticamente

El software permitió calcular y consolidar automáticamente el presupuesto del proyecto Challabamba Alta, diferenciando los costos según actividad y tipo de red.

Figura 21

Resumen General del presupuesto del proyecto Challamba Alta generada por el software

DESCRIPCION	REDES PRIMARIAS	REDES SECUNDARIAS	TOTAL
SUMINISTRO DE MATERIALES	-	-	-
AISLADORES	S/ 830.00	S/ 473.00	S/ 1,303.00
ALUMBRADO PUBLICO	-	S/ 4,814.10	S/ 4,814.10
CONDUCTORES	S/ 950.08	S/ 9,954.85	S/ 10,904.93
FERRETERIA	S/ 2,318.60	S/ 2,115.42	S/ 4,434.02
POSTES Y ACCESORIOS DE CONCRETO ARMADO	S/ 2,130.00	S/ 3,812.00	S/ 5,942.00
PUESTA A TIERRA	S/ 1,957.80	S/ 3,309.84	S/ 5,267.64
RETENIDAS	S/ 717.60	S/ 13,254.60	S/ 13,972.20
SECCIONADORES, PARARRAYOS, TERMINACIONES Y ACCESORIOS	S/ 3,014.10	-	S/ 3,014.10
TRANSFORMADORES	S/ 6,413.16	-	S/ 6,413.16
TOTAL SUMINISTRO DE MATERIALES	S/ 18,331.34	S/ 37,733.81	S/ 56,065.15
MONTAJE ELECTROMECHANICO	-	-	-
AISLADORES	S/ 429.76	S/ 2,329.31	S/ 2,759.07
CONDUCTORES Y CABLES	S/ 353.31	S/ 1,195.13	S/ 1,548.44
FERRETERIA	S/ 1,198.68	S/ 6,633.02	S/ 7,831.70
MONTAJE ELECTROMECHANICO ALUMBRADO PÚBLICO	-	S/ 4,051.89	S/ 4,051.89
POSTES Y ACCESORIOS DE CONCRETO ARMADO	S/ 1,763.22	S/ 11,313.06	S/ 13,076.28
PRUEBAS Y PUESTA EN SERVICIO	S/ 1,763.22	S/ 1,384.10	S/ 3,147.32
PUESTA A TIERRA	S/ 1,194.73	S/ 1,208.24	S/ 2,402.97
RETENIDAS	S/ 560.97	S/ 32,625.45	S/ 33,186.42
SECCIONADORES, PARARRAYOS Y TERMINACIONES	S/ 762.64	-	S/ 762.64
TRABAJOS PRELIMINARES	S/ 2,170.65	S/ 657.68	S/ 2,828.33
TRANSFORMADORES	S/ 493.47	-	S/ 493.47
TOTAL MONTAJE ELECTROMECHANICO	S/ 10,690.65	S/ 61,397.88	S/ 72,088.53
A. Suministro de Materiales	S/ 18,331.34	S/ 37,733.81	S/ 56,065.15
B. Montaje Electromecánico	S/ 10,690.65	S/ 61,397.88	S/ 72,088.53
C. Transporte	S/ 1,649.82	S/ 3,396.04	S/ 5,045.86
TOTAL COSTO DIRECTO	S/ 30,671.81	S/ 102,527.73	S/ 133,199.54
D. Gastos Generales y Utilidades	S/ 6,134.36	S/ 20,505.55	S/ 26,639.91
E. Supervisión de Obra	S/ 2,147.03	S/ 7,176.94	S/ 9,323.97
SUB TOTAL	S/ 38,953.20	S/ 130,210.22	S/ 169,163.42
IGV	S/ 7,011.58	S/ 23,437.84	S/ 30,449.42
TOTAL GENERAL (SOLES)	S/ 45,964.78	S/ 153,648.06	S/ 199,612.84

Nota. Visualización generada por el software.

Tabla 9*Gestión Presupuestal del Proyecto Challabamba Alta*

Conceptos	Red Primaria (S/)	Red Secundaria (S/)	Total (S/)
COSTOS DIRECTOS			
• Suministro de materiales	18,331.34	37,733.81	56,065.15
• Montaje electromecánico	10,690.65	61,397.88	72,088.53
• Transporte	1,649.82	3,396.04	5,045.86
Subtotal Costo Directo	30,671.81	102,527.73	133,199.54
Gastos Generales y Utilidades	6,134.36	20,505.55	26,639.91
Supervisión de Obra	2,147.03	7,176.94	9,323.97
Subtotal (sin IGV)	38,953.20	130,210.22	169,163.42
IGV (18%)	7,011.58	23,437.84	30,449.42
TOTAL GENERAL	45,964.78	153,648.06	199,612.84

Nota. Resultados obtenidos del software.

La Tabla anterior muestra la gestión presupuestal realizada automáticamente por el software, donde se evidencia el flujo completo desde los costos directos hasta el total general, incluyendo la aplicación de gastos generales, utilidad, supervisión de obra e IGV.

A continuación, se presentan los resultados organizados en función del resumen general de gastos:

Costo directo total:

- ✓ Redes primarias: S/ 30,671.81
- ✓ Redes secundarias: S/ 102,527.73

Total: S/ 133,199.54

Gastos generales y utilidades:

- ✓ Redes primarias: S/ 6,134.36
- ✓ Redes secundarias: S/ 20,505.55

Total: S/ 26,639.91

Supervisión de obra:

- ✓ Redes primarias: S/ 2,147.03
- ✓ Redes secundarias: S/ 7,176.94

Total: S/ 9,323.97

Subtotal sin IGV: S/ 169,163.42

IGV (18%): S/ 30,449.42

Total general del proyecto: S/ 199,612.84

El sistema permitió que este resultado se obtuviera de manera eficiente al integrar los datos estructurados desde los planos del proyecto, procesarlos internamente, y generar la tabla consolidada con los valores detallados. La validación mostró que los montos generados por el software coincidieron con los presupuestos establecidos por métodos convencionales, confirmando su eficacia y confiabilidad.

A continuación se presentan los resultados desglosados por los siguientes conceptos:

Suministro de media tensión (MT)

Montaje de media tensión (MT)

Suministro de baja tensión (BT)

Montaje de baja tensión (BT)

En las partidas de media tensión (suministro y montaje) se incluyen los componentes de la subestación de distribución debido a que este componente es parte de la media tensión (SED).

Tabla 10

Presupuesto Suministro Red Primaria

CODIGO	DESCRIPCION	UNIDAD	CANT	P.U.	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				S/ 2,130.00
1.1.0	POSTES CONCRETO ARMADO CENTRIFUGADO DE:				
1.1.1	12/200/140/320	u	0	S/ 900.00	S/ 0.00
1.1.2	12/300/150/330	u	2	S/ 990.00	S/ 1,980.00
1.2.0	ACCESORIOS DE C.A				
1.2.1	Caja de Concreto Para P.T. (0.30 x 0.40 x 0.40) m, con tapa de espesor de 0.05 m de C.V.	u	3	S/ 50.00	S/ 150.00
2.0.0	AISLADORES				S/ 830.00
2.1.0	AISLADORES DE PORCELANA, CLASE ANSI:				
2.1.1	TIPO PIN CLASE ANSI 56-4, 25 KV, 685 mm LINEA DE FUGA	u	2	S/ 65.00	S/ 130.00

2.2.0	AISLADORES POLIMÉRICOS TIPO SUSPENSIÓN :				
2.2.1	RPP-25, Tipo Clevis-Ojo, Vtípico 25kV, Distancia de Fuga Mínima 660mm	u	4	S/ 120.00	S/ 480.00
2.2.2	Tipo Clevis-Ojo, Vtípico 36kV, Distancia de Fuga Mínima 660 mm (para Retenida Aislada)	u	2	S/ 110.00	S/ 220.00
3.0.0	CONDUCTORES				S/ 950.08
3.1.0	CONDUCTOR DE COBRE DESNUDO, CABLEADO, TEMPLE BLANDO, DE :				
3.1.1	25mm ² , 7 HILOS (Para PAT-1)	m	54	S/ 12.50	S/ 675.00
3.1.2	25mm ² , 7 HILOS (Para PAT-0)	m	15	S/ 12.50	S/ 187.50
3.2.0	CONDUCTOR DE ALEACION DE ALUMINIO 6201-T81 (AAAC) CABLEADO DE :				
3.2.1	35mm ² , 7 HILOS	m	6	S/ 1.93	S/ 11.58
3.3.0	ALAMBRE DE ALUMINIO, PARA AMARRE DE :				
3.3.1	10mm ² , 7 HILOS	m	5	S/ 1.20	S/ 6.00
3.4.0	CONDUCTOR DE ALUMINIO, TEMPLE BLANDO AISLADO DE :				
3.4.1	35mm ² , 7 HILOS	m	6	S/ 8.00	S/ 48.00
3.5.0	PLANCHA DE COBRE :				
3.5.1	PLANCHA DE COBRE TIPO 'J'	m	4	S/ 5.50	S/ 22.00
4.0.0	FERRETERIA				
4.1.0	ARMADO DT-2	Cjto	1		S/ 2,318.60
4.1.1	Abrazadera Tipo CAS Simple de 64mm, E=6.4mm, D=200mm, C/4P/4T/4C/8A/4AP.	Und	1	S/ 51.00	S/ 51.00
4.1.2	Abrazadera de A°G° P/Postes de CAC de Fijación Tipo U de 180ØX75X6.4mm.	Und	1	S/ 45.00	S/ 45.00
4.1.3	Cruceta de Perfil Angular de F°G° de 64x64x2,500mm, E=6.4mm, con 3 dados de 100mm	Und	1	S/ 250.00	S/ 250.00
4.1.4	Perno Maquinado L=50mm, D=13mm, C/T/CT/2A/AP	Und	2	S/ 2.50	S/ 5.00
4.1.5	Perno con Ojal, D=16mm, L=178mm, C/T/CT/2A/AP	Und	2	S/ 18.00	S/ 36.00
4.1.6	Riostra de Perfil Angular de F°G° de 38x38x1,000mm, E=5mm	Und	2	S/ 22.00	S/ 44.00
4.2.0	ARMADO AB1	Cjto	0		
4.2.1	Abrazadera Tipo CAS Simple de 64mm, E=6.4mm, D=160mm, C/3P/3T/3CT/6A/3AP.	Und	0	S/ 48.00	S/ 0.00
4.2.2	Abrazadera de Fijacion Tipo U de 64mm, E=6.4mm, D=150mm, C/2P/2T/2CT/4A/2AP.	Und	0	S/ 45.00	S/ 0.00
4.2.3	Cruceta de Perfil Angular de F°G° de 64x64x2,500mm, E=6.4mm, con 3 dados de 100mm	Und	0	S/ 250.00	S/ 0.00
4.2.4	Perno Maquinado L=50mm, D=13mm, C/T/CT/2A/AP	Und	0	S/ 2.50	S/ 0.00
4.2.5	Arandela Cuadrada Curva de A°G° de 57x57x5 mm, agujero de 17.5mmØ	Und	0	S/ 2.80	S/ 0.00
4.2.6	Riostra de Perfil Angular de F°G° de 38x38x1,000mm, E=5mm	Und	0	S/ 22.00	S/ 0.00
4.3.0	ARMADO AB4	Cjto	1		
4.3.1	Cruceta de Perfil Angular de F°G° de 64x64x2,500 mm, e=6.4 mm, con 02 dados segun Detalle de Armado	Und	2	S/ 250.00	S/ 500.00
4.3.2	Riostra de Perfil Angular de F°G° de 38x38x1,000 mm, e=5 mm	Und	4	S/ 22.00	S/ 88.00
4.3.3	Abrazadera Tipo CAS Doble de A°G° de 160mmØx64x6.4 mm, C/04P+04T+04CT+08A+04AP, Ø Según Req.	Und	1	S/ 48.00	S/ 48.00
4.3.4	Perno Maquinado de A°G° de 13mmØx50 mm, C/T+CT+02A+AP	Und	4	S/ 2.50	S/ 10.00

4.3.5	Perno Doble Armado de A°G° de 16mmØx457 mm, Rosca Completa, C/02T+02CT+04A+02AP	Und	4	S/ 15.00	S/ 60.00
4.3.6	Arandela Cuadrada Curva de A°G° de 57x57x5 mm, agujero de 17.5mmØ	Und	2	S/ 2.80	S/ 5.60
4.3.7	Tuerca Ojo, D=16mm	Und	3	S/ 9.20	S/ 27.60
4.4.0	ARMADO AB5	Cjto	1		
4.4.1	Abrazadera Tipo CAS Doble de 64mm, E=6.4mm, D=170mm, C/4P/4T/4CT/8A/4AP, Ø Según Req.	Und	1	S/ 50.00	S/ 50.00
4.4.2	Cruceta de Perfil Angular de F°G° de 64x64x2,500mm, E=6.4mm, con 2 dados de 100mm	Und	2	S/ 250.00	S/ 500.00
4.4.3	Perno Doble Armado 16mm, L=457mm, CR=55kN, C/2T/2CT/4A/2AP	Und	4	S/ 15.00	S/ 60.00
4.4.4	Perno Maquinado L=50mm, D=13mm, C/T/CT/2A/AP	Und	4	S/ 2.50	S/ 10.00
4.4.5	Riostra de Perfil Angular de F°G° de 38x38x1,000mm, E=5mm	Und	4	S/ 22.00	S/ 88.00
4.4.6	Tuerca Ojo, D=16mm	Und	2	S/ 9.20	S/ 18.40
4.5.0	SUMINISTROS VARIOS				
4.5.1	Espiga para Cruceta L=254+178mm, D=28.6mm, DR=35mm, P/Aislador PIN ANSI 56-4, C/T/CT/A/AP	Und	2	S/ 35.00	S/ 70.00
4.5.2	Hebillas para cinta bandit de 19mm para puesta a tierra	Und	9	S/ 1.00	S/ 9.00
4.5.3	Espiga para Vertice Punta de Poste L=609mm, DR=35mm, P/Aislador PIN ANSI 56-4, C/T/CT/A/AP	Und	2	S/ 30.00	S/ 60.00
4.5.4	Grapa de Anclaje Tipo Pistola de 02 Pernos de AL, P/Conductor 16-50 mm ²	Und	4	S/ 32.00	S/ 128.00
4.5.5	Cinta Plana de Armar de AL Recocido para Conductor AAAC, de 1.40x7.60 mm	Und		S/ 2.50	
4.5.6	Grillete de Anclaje Tipo Recto de A°G°, 16 mm Øx19mm abertura, 77mm con pasador de seguridad, 18000 Libras	Und	4	S/ 13.00	S/ 52.00
4.5.7	Varilla de Armar Preformada Simple de AL de 1117 mm, 8 hilos, P/Conductor de AL 35mm ²	Und	2	S/ 16.00	S/ 32.00
4.5.8	Tubo F.G. de 19mm diam x 1.5mm de 4m para bajada de Cu para puesta a tierra	Und	3	S/ 22.00	S/ 66.00
4.5.9	Cinta Bandit de 19mm. de ancho para puesta a tierra	m	1	S/ 5.00	S/ 5.00
5.0.0	RETENIDAS				S/ 717.60
5.1.0	RETENIDA INCLINADA AISLADA	Cjto	2		
5.1.1	Cable de Acero de Grado Siemens Martin D=10.0mm	m	36	S/ 4.00	S/ 144.00
5.1.10	Guardacabo P/Cable de Acero Ø 10mm, 1.3mm de espesor	Und	6	S/ 2.00	S/ 12.00
5.1.11	Grapa Doble Vía A°G°, 3 Pernos, 152.4 mm Long P/Cable de Acero 10mm Ø	Und	8	S/ 12.00	S/ 96.00
5.1.2	Alambre Galvanizado N° 14 AWG	m	12	S/ 0.80	S/ 9.60
5.1.3	Arandela Cuadrada de L=102x102mm, e=6.4mm, Diámetro de Agujero=19 mm	Und	2	S/ 7.00	S/ 14.00
5.1.4	Abrazadera para Soporte de Retenida, D=180 mm, e=6.4mm, L=330mm, H=75mm, C/3P/3T/3C/6A/3AP	Und	2	S/ 38.00	S/ 76.00
5.1.5	Varilla de Anclaje con Ojal L=2400mm, D=19mm	Und	2	S/ 56.00	S/ 112.00
5.1.6	Plancha de Fierro Galvanizado 400mm x 400mm, e=6.4mm	Und	2	S/ 40.00	S/ 80.00
5.1.7	Grillete de Anclaje Tipo Recto D=16mm, C/Pasador de Seguridad	Und	2	S/ 12.00	S/ 24.00
5.1.8	Templador de F°G° de 300x19 mmØ	Und	2	S/ 40.00	S/ 80.00
5.1.9	Canaleta Guardacable L=2,400mm, E=1.3mm	Und	2	S/ 35.00	S/ 70.00
6.0.0	PUESTA A TIERRA				S/ 1,957.80

6.1.0	SUMINISTROS VARIOS PAT-1	Cjto	3		
6.1.1	Plancha Antirrobo de Bronce para Varilla de Puesta a Tierra de 200x200mm, e=6mm, C/Agujero de 20mmØ	Und	3	S/ 45.00	S/ 135.00
6.1.2	Varilla de cobre con rosca, tuerca y contratuerca en un extremo de 19mm Ø x 2.40m	Und	3	S/ 311.00	S/ 933.00
6.1.3	Conector de Cobre Tipo AB P/Varilla 19 mm Ø - Cable 25mm ²	Und	3	S/ 11.50	S/ 34.50
6.1.4	Cemento Conductivo x 25Kg	Bls	6	S/ 81.30	S/ 487.80
6.1.5	Tierra Negra Cernida - Arcilla	m3	10.5	S/ 35.00	S/ 367.50
7.0.0	SECCIONADORES, PARARRAYOS, TERMINACIONES Y ACCESORIOS				S/ 3,014.10
7.1.0	SECCIONADORES CUT-OUT				
7.1.1	27KV, 150 KV BIL, 100 A, 10 KA	Und	4	S/ 370.00	S/ 1,480.00
7.2.0	PARARRAYOS POLIMÉRICO DE ÓXIDO METÁLICO, CLASE DISTRIBUCIÓN				
7.2.1	21 kV; 150 KV BIL, 10kA	Und	4	S/ 350.00	S/ 1,400.00
7.3.0	CONECTOR DE COBRE TIPO PERNO PARTIDO (SPLIT BOLT)				
7.3.1	25-35/6-35mm ²	Und	3	S/ 4.50	S/ 13.50
7.4.0	CONECTOR DE AL-AL, TIPO DOBLE VÍA				
7.4.1	16-120mm ² , Dos Pernos	Und	2	S/ 6.00	S/ 12.00
7.5.0	MANGUITO DE REPARACIÓN DE ALUMINIO A COMPRESIÓN				
7.5.1	Sección de 35mm ²	Und	2	S/ 54.30	S/ 108.60
8.0.0	TRANSFORMADORES				S/ 6,413.16
8.1.0	TRANSFORMADORES MONOFASICOS 02 AISLADORES M.T				
8.1.1	10 KVA	Und	1	S/ 6,413.16	S/ 6,413.16
	Total				S/ 18,331.31

Nota. Resultados obtenidos del software.

Tabla 11

Presupuesto Montaje Red Primaria

CODIGO	DESCRIPCION	UNIDAD	CANT	P.U.	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				
1.1.0	INSTALACIÓN DE POSTES DE CONCRETO EN TERRENO NORMAL				
1.1.1	Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)	Pza	0	S/ 261.64	S/ 0.00
1.1.2	Izaje Poste de C.A.C. de 12/300 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)	Pza	1	S/ 282.92	S/ 282.92
1.2.0	INSTALACIÓN DE POSTES DE CONCRETO EN TERRENO ROCOSO				
1.2.1	Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno rocoso)	Pza	0	S/ 284.58	S/ 0.00
1.2.2	Izaje Poste de C.A.C. de 12/300 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno rocoso)	Pza	1	S/ 353.33	S/ 353.33
1.3.0	TRANSPORTE DE POSTES				
1.3.1	Transporte de postes de MT de almacen en a pie de carretera en obra	Pza	2	S/ 26.32	S/ 52.64
1.3.2	Transporte de postes de MT de pie de carretera a punto de izaje	Pza	2	S/ 113.15	S/ 226.30
2.0.0	AISLADORES				

CODIGO	DESCRIPCION	UNIDAD	CANT	P.U.	TOTAL
2.1.0	AISLADORES DE PORCELANA, CLASE ANSI:				
2.1.1	Instalación de Aislador de Porcelana Tipo PIN Clase ANSI 56-4, Incl. Accesorios	Cjto	2	S/ 60.23	S/ 120.46
2.2.0	AISLADORES POLIMÉRICOS TIPO SUSPENSIÓN :				
2.2.1	Instalación de Aislador Polimérico Tipo Suspensión RPP-25, Incl. Accesorios	Cjto	4	S/ 77.44	S/ 309.76
3.0.0	CONDUCTORES Y CABLES				
3.1.0	MONTAJE DE CONDUCTOR DE ALEACION DE ALUMINIO 6201-T81 (AAAC) CABLEADO :				
3.1.1	Instalación de Conductor AAAC 35 mm2, 7 Hilos, P/Fase	m	441.64	S/ 0.80	S/ 353.31
4.0.0	FERRETERIA				
4.1.0	MONTAJE DE ARMADOS QUE COMPRENDE: INSTALACION Y FIJACION EN EL POSTE				
4.1.1	ARMADO DT-2	Cjto	1	S/ 374.88	S/ 374.88
4.1.2	ARMADO AB1	Cjto	0	S/ 374.88	S/ 0.00
4.1.3	ARMADO AB4	Cjto	1	S/ 449.84	S/ 449.84
4.1.4	ARMADO AB5	Cjto	1	S/ 374.88	S/ 374.88
5.0.0	RETENIDAS				
5.1.0	INSTALACIÓN DE RETENIDAS EN TERRENO NORMAL				
5.1.1	Retenida Inclinada Aislada (en terreno normal)	Cjto	1	S/ 266.21	S/ 266.21
5.2.0	INSTALACIÓN DE RETENIDAS EN TERRENO ROCOSO				
5.2.1	Retenida Inclinada Aislada (en terreno rocoso)	Cjto	1	S/ 294.98	S/ 294.98
6.0.0	PUESTA A TIERRA				
6.1.0	PAT-1 PARA M.T. EN TERRENO NORMAL CON EXCAVACIÓN, ARMADO, COMPACTACIÓN, CAJA DE REGISTRO Y CONECTORES TIPO PERNO PARTIDO.				
6.1.1	Puesta a Tierra en Terreno (PAT-1)	Cjto	3	S/ 341.33	S/ 1,023.99
6.2.0	PAT-0 PARA M.T. EN TERRENO NORMAL CON EXCAVACIÓN, ARMADO, COMPACTACIÓN, CAJA DE REGISTRO Y CONECTORES TIPO PERNO PARTIDO.				
6.2.1	Puesta a Tierra en Terreno (PAT-0)	Cjto	1	S/ 171.30	S/ 171.30
7.0.0	SECCIONADORES, PARARRAYOS Y TERMINACIONES				
7.1.0	SECCIONADORES CUT-OUT				
7.1.1	Instalación de seccionador CUT-OUT 27 kV con fusible chicote, conexión a red y suministro de cintas.	Pza	4	S/ 95.42	S/ 381.68
7.2.0	PARARRAYOS POLIMÉRICO DE ÓXIDO METÁLICO, CLASE DISTRIBUCIÓN				
7.2.1	Instalación de Pararrayos de 21 KV. Incluye el Conexión a la Red, el Suministro de Cinta Autofundente y Vinilica	Pza	4	S/ 95.42	S/ 381.68
8.0.0	TRANSFORMADORES				
8.1.0	MONTAJE DE TRANSFORMADOR SOBRE EL POSTE				
8.1.1	Transformador Monofasico Monoposte de 10 KVA	u	1	S/ 494.73	S/ 494.73
8.1.2	Transformador Monofasico Monoposte de 15 KVA	km	0	S/ 494.73	S/ 0.00
9.0.0	TRABAJOS PRELIMINARES				
9.1.0	ACTIVIDADES PREVIAS A LA CONSTRUCCIÓN				
9.1.1	Cartel Para Obra	u	1	S/ 1,876.15	S/ 1,876.15
9.1.2	Estudios de Ingeniería de Detalle de las Redes Primarias	km	0.44164	S/ 319.46	S/ 141.08
9.1.3	Replanteo Topográfico, Ubicación de Estructuras de Acuerdo a la Ingeniería de Detalle de las Redes Primarias	km	0.44164	S/ 354.77	S/ 156.82
10.0.0	PRUEBAS Y PUESTA EN SERVICIO				

CODIGO	DESCRIPCION	UNIDAD	CANT	P.U.	TOTAL
10.1.0	ACTIVIDADES PREVIAS A LA CONSTRUCCIÓN				
10.1.1	Señalización, pintado de Codigos (numeración otorgada por ELSE).	Estr.	2	S/ 39.59	S/ 79.18
10.1.2	Corte y Reconexión de la Red de MT de Servicio Público	u	1	S/ 196.48	S/ 196.48
10.1.3	Pruebas y Puesta en Servicio de las Redes Primarias	Km	0.44164	S/ 974.79	S/ 430.52
10.1.4	Expediente Técnico Finales Conforme a Obra , Incluye la Presentación Digitalizada del Expediente	Km	0.44164	S/ 320.73	S/ 141.68
	Total				S/ 37,733.81

Nota. Visualización generada por el software.

Tabla 12

Presupuesto Suministro Red Secundaria

CODIGO	DESCRIPCIÓN	UNIDAD	CANT	PU	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				S/ 6,942.00
1.1.0	POSTES CONCRETO ARMADO CENTRIFUGADO:				
1.1.1	POSTE DE C.A.	UND	17	S/ 441.00	S/ 7,497.00
1.1.2	POSTE DE C.A.	UND	18	S/ 526.00	S/ 9,468.00
1.2.0	ACCESORIOS DE C.A.				
1.2.1	Caja de concu	UND	6	S/ 50.00	S/ 300.00
1.2.2	ACCESORIOS				S/ 165.00
2.0.0	AISLADORES DE PORCELANA, CLASE ANSI:				S/ 11.00
2.1.1	54-2, PARA TU	UND	15	S/ 11.00	S/ 165.00
3.0.0	CONDUCTORES				S/ 10,675.00
3.1.0	CONDUCTOR DE COBRE DESNUDO, CABLEADO, SIMPLE BLANDO:				S/ 6,561.00
3.1.1	25mm2, 7 HILOS	M	680	S/ 9.65	S/ 6,562.00
3.2.0	CABLE AUTOPORTANTE DE ALUMINIO TIPO CAA, C/PORTANTE DE				S/ 10,318.00
3.2.1	2x25/16mm2	M	1180.2	S/ 8.74	S/ 10,318.00
3.2.2	2x25/N25 mm2	UND	0	S/ 6.14	S/ 0.00
4.0.0	FERRETERIA				S/ 2,282.42
4.1.0	ARMADO E0				
4.1.1	GRAPA DE AN	UND	4	S/ 18.00	S/ 72.00
4.1.2	HEBILLA PAR	UND	4	S/ 2.15	S/ 8.60
4.1.3	FLEJE DE ACE	UND	4	S/ 6.50	S/ 26.00
4.1.4	ARANDELA CA	UND	4	S/ 15.00	S/ 60.00
4.2.0	ARMADO E1				
4.2.1	GRAPA DE SU	UND	14	S/ 18.00	S/ 252.00
4.2.2	PERNO CON	UND	14	S/ 15.00	S/ 210.00
4.2.3	ARANDELA CU	UND	28	S/ 4.00	S/ 112.00
4.3.0	ARMADO E3				
4.3.1	GRAPA DE AN	UND	5	S/ 18.00	S/ 90.00
4.3.2	PERNO CON	UND	5	S/ 18.00	S/ 90.00
4.3.3	ARANDELA CI	UND	10	S/ 4.00	S/ 40.00
4.4.0	ARMADO E4				
4.4.1	GRAPA DE AN	UND	22	S/ 18.00	S/ 396.00
4.4.2	PERNO CON	UND	11	S/ 18.00	S/ 198.00

CODIGO	DESCRIPCIÓN	UNIDAD	CANT	PU	TOTAL
4.4.3	TUERCA O/O	UND	11	S/ 9.30	S/ 102.30
4.4.4	ARANDELA CI	UND	22	S/ 4.00	S/ 88.00
4.5.0	ARMADO E5	CJTO	1		
4.5.1	GRAPA DE AN	UND	1	S/ 18.00	S/ 18.00
4.5.2	GRAPA DE SU	UND	1	S/ 18.00	S/ 18.00
4.5.3	ARANDELA CA	UND	1	S/ 15.00	S/ 15.00
4.5.4	PERNO CON	UND	1	S/ 18.00	S/ 18.00
4.5.5	ARANDELA CU	UND	2	S/ 4.00	S/ 8.00
4.6.0	ARMADO E6	CJTO	3		
4.6.1	GRAPA DE AN	UND	3	S/ 18.00	S/ 54.00
4.6.2	PERNO CON	UND	2	S/ 18.00	S/ 36.00
4.6.3	TUERCA O/O	UND	1	S/ 9.30	S/ 9.30
4.6.4	ARANDELA CU	UND	4	S/ 4.00	S/ 16.00
4.7.0	ARMADO O/C	CJTO	9		
4.7.1	HEBILLA PAR	UND	18	S/ 2.15	S/ 38.70
4.7.2	FLEJE DE ACE	UND	18	S/ 6.50	S/ 117.00
4.7.3	ARANDELA CU	UND	18	S/ 4.00	S/ 72.00
4.7.4	PERNO MAQ	UND	9	S/ 6.08	S/ 54.72
5.0.0	RETENIDAS				S/ 4,793.00
5.1.0	RETENIDA	CJTO	10		
5.1.1	Cable de Acero	M	10	S/ 44.00	S/ 440.00
5.1.2	Alambre galv	UND	16.5	S/ 8.00	S/ 132.00
5.1.3	Varilla de Retenida	UND	11	S/ 56.00	S/ 616.00
5.1.4	Grapas	UND	11	S/ 55.00	S/ 605.00
5.1.5	Arandela Cu	UND	11	S/ 3.65	S/ 40.15
5.1.6	Abrazadera Cu	UND	11	S/ 38.00	S/ 418.00
5.1.7	Grapa Parall	UND	44	S/ 12.00	S/ 528.00
5.1.8	Cinta Autol	UND	11	S/ 25.00	S/ 275.00
5.1.9	Guardacabo	UND	11	S/ 2.00	S/ 22.00
5.1.10	Templador	UND	11	S/ 35.00	S/ 385.00
5.2.0	RETENIDA	CJTO	4		
5.2.1	Cable de Acero	M	40	S/ 40.00	S/ 1,600.00
5.2.2	Alambre galv	UND	4	S/ 8.00	S/ 32.00
5.2.3	Varilla de Retenida	UND	4	S/ 56.00	S/ 224.00
5.2.4	Grapas	UND	4	S/ 55.00	S/ 220.00
5.2.5	Arandela Cu	UND	4	S/ 3.50	S/ 14.00
5.2.6	Abrazadera Cu	UND	4	S/ 38.00	S/ 152.00
5.2.7	Grapa Parall	UND	16	S/ 12.00	S/ 192.00
5.2.8	Cinta Autol	UND	4	S/ 25.00	S/ 100.00
5.2.9	Guardacabo	UND	4	S/ 2.00	S/ 8.00
5.2.10	Templador	UND	4	S/ 35.00	S/ 140.00
5.2.11	Contrapunta	UND	4	S/ 65.00	S/ 260.00
6.0.0	PUESTA A TIERRA				S/ 3,309.84

CODIGO	DESCRIPCIÓN	UNIDAD	CANT	PU	TOTAL
6.1.0	SUMINISTRO POZO	UND	6		
6.1.1	Plancha Anod	UND	6	S/ 455.00	S/ 2,730.00
6.1.2	Varilla de Co	UND	6	S/ 213.54	S/ 1,281.24
6.1.3	Conector de	UND	6	S/ 8.00	S/ 48.00
6.1.4	Cemento con	BLS	12	S/ 27.20	S/ 326.40
6.1.5	Tierra Negra	M3	21	S/ 35.00	S/ 735.00
7.0.0	ALUMBRADO PUBLICO				S/ 4,414.10
7.1.0	SUMINISTRO ALUMBRADO PUBLICO				
7.1.1	50 W, 120/27	LUMIN	8	S/ 491.50	S/ 3,932.00
7.1.2	Aislador de lu	UND	9	S/ 35.00	S/ 315.00
7.1.3	Abrazadera	UND	9	S/ 8.00	S/ 72.00
7.1.4	Conector de	UND	9	S/ 8.00	S/ 72.00
7.1.5	Tubo de fier	UND	9	S/ 1.00	S/ 9.00
7.1.6	Cable de cobr	M	20	S/ 6.50	S/ 130.00
8.0.0	SUBESTACIONES				S/ 1,402.75
8.1.0	SUBESTACION DE				
8.1.1	Transformador	UND	1	S/ 1,402.75	S/ 1,402.75
	Total				S/ 37,733.81

Nota. Visualización generada por el software.

Tabla 13

Presupuesto Montaje Red Secundaria

CODIGO	DESCRIPCIÓN	UNIDAD	CANT	PU	TOTAL
1.0.0	POSTES Y ACCESORIOS DE CONCRETO ARMADO				S/ 12,892.36
1.1.0	INSTALACIÓN DE POSTES DE CONCRETO EN TERRENO NORMAL				
1.1.1	Izaje Poste de C.A. Pza	8	S/ 217.80	S/ 1,742.40	
1.1.2	Izaje Poste de C.A. Pza	12	S/ 227.83	S/ 2,733.96	
1.2.0	INSTALACIÓN DE POSTES DE CONCRETO EN TERRENO ROCOSO				
1.2.1	Izaje Poste de C.A. Pza	6	S/ 284.58	S/ 1,707.48	
1.2.2	Izaje Poste de C.A. Pza	7	S/ 353.33	S/ 2,473.31	
1.3.0	TRANSPORTE DE POSTES				
1.3.1	Transporte de Pza	32	S/ 26.32	S/ 842.24	
1.3.2	Transporte de Pza	32	S/ 113.15	S/ 3,620.80	
2.0.0	AISLADORES				S/ 813.15
2.1.0	AISLADORES DE PORCELANA, CLASE ANSI:				
2.1.1	Instalación de Cjto	15	S/ 54.21	S/ 813.15	
3.0.0	CONDUCTORES Y CABLES				S/ 1,312.18
3.1.0	INSTALACION DE CABLE AUTOPORTANTE				
3.1.1	2x25+16/N25 m	1192.89	S/ 1.10	S/ 1,312.18	
3.1.2	2x25/N25 m m	0	S/ 1.10	S/ 0.00	
4.0.0	FERRETERIA				S/ 7,123.69
4.1.0	MONTAJE DE ARMADOS QUE COMPRENDE: INSTALACION Y FIJACION				
4.1.1	Armado E0 Cjto	2	S/ 170.97	S/ 341.94	
4.1.2	Armado E1 Cjto	14	S/ 113.98	S/ 1,595.72	

CODIGO	DESCRIPCIÓN	UNIDAD	CANT	PU	TOTAL
4.1.3	Armado E2 Cjto	5	S/ 113.98	S/ 569.90	
4.1.4	Armado E4 Cjto	11	S/ 155.43	S/ 1,709.73	
4.1.5	Armado E5 Cjto	1	S/ 170.97	S/ 170.97	
4.1.6	Armado E6 Cjto	1	S/ 170.97	S/ 170.97	
4.1.7	Armado CAJ Cjto	9	S/ 284.94	S/ 2,564.46	
5.0.0	RETENIDAS				S/ 11,407.33
5.1.0	INSTALACIÓN DE RETENIDAS EN TERRENO NORMAL				
5.1.1	Retenida Sim Cjto	11	S/ 757.79	S/ 8,335.69	
5.1.2	Retenida Ver Cjto	4	S/ 767.91	S/ 3,071.64	
6.0.0	PUESTA A TIERRA				S/ 1,210.72
6.1.0	INSTALACIÓN DE PUESTA A TIERRA TIPO VARILLA VERTICAL				
6.1.1	Puesta a Tier Cjto	4	S/ 50.45	S/ 201.80	
6.1.2	Puesta a Tier Cjto	2	S/ 512.50	S/ 1,025.00	
7.0.0	TRABAJOS PRELIMINARES				S/ 722.19
7.1.0	ACTIVIDADES PREVIAS A LA CONSTRUCCIÓN				
7.1.1	Ingeniería de km	1.19289	S/ 124.48	S/ 148.49	
7.1.2	Replanteo Td km	1.19289	S/ 480.93	S/ 573.70	
8.0.0	PRUEBAS Y PUESTA EN SERVICIO				S/ 1,550.60
8.1.0	ACTIVIDADES PREVIAS A LA CONSTRUCCIÓN				
8.1.1	Señalización Estr.	32	S/ 24.60	S/ 787.20	
8.1.2	Pruebas y Pu Km	1.19289	S/ 189.75	S/ 226.35	
8.1.3	Expediente T. Km	1.19289	S/ 450.21	S/ 537.05	
9.0.0	INSTALACION ELECTROMECHANICO ALUMBRADO PÚBLICO				S/ 4,051.81
9.1.0	LUMINARIAS				
9.1.1	Instalación de Cjto	9	S/ 450.21	S/ 4,051.89	
				TOTAL	S/ 61,397.68

Nota. Visualización generada por el software.

4.7. Análisis comparativo con otras herramientas de cálculo de presupuesto

Para evaluar la eficacia del software desarrollado en el presente trabajo, se realizó un análisis comparativo con dos herramientas comúnmente utilizadas en la elaboración de presupuestos de obras: Microsoft Excel y el software S10. Este análisis consideró aspectos clave como accesibilidad, compatibilidad con herramientas de diseño, estructura de partidas, facilidad de uso, nivel de automatización y tiempo de elaboración del presupuesto.

4.7.1. Comparación con Microsoft Excel

Microsoft Excel es una herramienta ampliamente utilizada por profesionales para el análisis de datos, cálculos presupuestales y presentación de información. Su fortaleza radica en

su versatilidad y capacidad de personalización, lo que permite al usuario crear estructuras jerárquicas de partidas, aplicar fórmulas y generar tablas dinámicas para la visualización de resultados.

No obstante, Excel carece de integración directa con plataformas de diseño eléctrico como DIREDCAD, lo que obliga al usuario a ingresar los datos manualmente. Esta limitación no solo incrementa el tiempo de procesamiento, sino que también introduce un mayor riesgo de errores humanos. Además, el uso eficiente de Excel para presupuestos requiere conocimientos intermedios o avanzados en fórmulas, macros y organización de hojas de cálculo.

Figura 22

Reporte generado por excel para el mismo proyecto

DESCRIPCION	REDES PRIMARIAS (\$/)	REDES SECUNDARIAS (\$/)	TOTAL (\$/)
AISLADORES	830.00	473.00	1,303.00
ALUMBRADO PÚBLICO	0.00	4,814.10	4,814.10
CONDUCTORES	950.08	9,954.85	10,904.93
FERRETERIA	2,318.60	2,115.42	4,434.02
POSTES Y ACCESORIOS DE CONCRETO	2,130.00	3,812.00	5,942.00
PUESTA A TIERRA	1,957.80	3,309.84	5,267.64
RETENIDAS	717.60	13,254.60	13,972.20
SECCIONADORES, PARARRAYOS, ETC.	3,014.10	0.00	3,014.10
TRANSFORMADORES	6,413.16	0.00	6,413.16
TOTAL SUMINISTROS	18,331.34	37,733.81	56,065.15
AISLADORES (MONTAJE)	429.76	2,329.31	2,759.07
CONDUCTORES Y CABLES	353.31	1,195.13	1,548.44
FERRETERIA (MONTAJE)	1,198.68	6,633.02	7,831.70
ALUMBRADO PÚBLICO (MONTAJE)	0.00	4,051.89	4,051.89
POSTES Y ACCESORIOS (MONTAJE)	1,763.22	11,313.06	13,076.28
PRUEBAS Y PUESTA EN SERVICIO	1,763.22	1,384.10	3,147.32
PUESTA A TIERRA (MONTAJE)	1,194.73	1,206.24	2,402.97
RETENIDAS (MONTAJE)	560.97	32,625.45	33,186.42
SECCIONADORES, PARARRAYOS, ETC. (MONTAJE)	762.64	0.00	762.64
TRABAJOS PRELIMINARES	2,170.65	657.68	2,828.33
TRANSFORMADORES (MONTAJE)	493.47	0.00	493.47
TOTAL MONTAJE	10,690.65	61,397.88	72,098.53
TRANSPORTE	1,649.82	3,396.04	5,045.86
TOTAL COSTO DIRECTO	30,671.81	102,527.73	133,199.54
GASTOS GENERALES Y UTILIDADES	6,134.36	20,505.55	26,639.91
SUPERVISION DE OBRA	2,147.03	7,176.34	9,323.97
SUB TOTAL	38,953.20	130,210.22	169,163.42
ICV	7,011.58	23,437.84	30,449.42
TOTAL GENERAL (SOLES)	45,964.78	153,648.06	199,612.84

Nota. Reporte generado en microsoft excel.

4.7.2. Comparación con S10

El software S10 es una herramienta especializada y reconocida en el ámbito de la ingeniería para la elaboración de presupuestos de obras civiles, saneamiento y, en menor medida, eléctricas. Posee una base de datos de partidas, análisis de precios unitarios, e informes conforme a los formatos exigidos por entidades públicas, como el Ministerio de Economía y Finanzas a través del sistema Invierte.pe.

Figura 23
Reporte generado en S10 para el mismo proyecto

CÓDIGO	DESCRIPCIÓN	UNIDAD	CANTIDAD	P.U. (S/)	PARCIAL (S/)
01	SUMINISTRO DE MATERIALES				
01.01	AISLADORES	unid			1,303.00
01.02	ALUMBRADO PÚBLICO	unid			4,814.10
01.03	CONDUCTORES	m			10,904.93
01.04	FERRETERIA	kg			4,434.02
01.05	POSTES Y ACCESORIOS DE CONCRETO	unid			5,942.00
01.06	PUESTA A TIERRA	unid			5,267.64
01.07	RETENIDAS	unid			13,972.20
01.08	SECCIONADORES, PARARRAYOS, ETC.	unid			3,014.10
01.09	TRANSFORMADORES	unid			6,413.16
01.10	TOTAL SUMINISTROS				56,065.15
02	MONTAJE ELECTROMECÁNICO				
02.01	AISLADORES (MONTAJE)	unid			2,759.07
02.02	CONDUCTORES Y CABLES	m			1,548.44
02.03	FERRETERIA (MONTAJE)	kg			7,831.70
02.04	ALUMBRADO PÚBLICO (MONTAJE)	unid			4,051.89
02.05	POSTES Y ACCESORIOS (MONTAJE)	unid			13,076.28
02.06	PRUEBAS Y PUESTA EN SERVICIO	serv			3,147.32
02.07	PUESTA A TIERRA (MONTAJE)	unid			2,402.97
02.08	RETENIDAS (MONTAJE)	unid			33,186.42
02.09	SECCIONADORES, PARARRAYOS, ETC. (MONTAJE)	unid			762.64
02.10	TRABAJOS PRELIMINARES	serv			2,828.33
02.11	TRANSFORMADORES (MONTAJE)	unid			493.47
02.12	TOTAL MONTAJE				72,098.53
03	TRANSPORTE	viaje			5,045.86
04	GASTOS GENERALES Y UTILIDADES	%			26,639.91
05	SUPERVISIÓN DE OBRA	%			9,323.97
06	IGV	%			30,449.42
TOTAL	TOTAL GENERAL				199,612.84

Nota. Reporte generado en S10.

Sin embargo, S10 presenta varias limitaciones en el contexto de proyectos de electrificación rural. En primer lugar, no cuenta con integración directa con DIREDCAD, por lo que el metrado debe ser cargado manualmente. En segundo lugar, su enfoque está dirigido a

obras de infraestructura civil, por lo que la adaptación de partidas eléctricas requiere experiencia y ajustes específicos. Finalmente, el costo de adquisición de la licencia representa una barrera para estudiantes, investigadores o pequeñas entidades locales.

El software desarrollado, por otro lado, está diseñado específicamente para proyectos eléctricos en zonas rurales. Su enfoque de código abierto y gratuito lo convierte en una herramienta accesible para profesionales y estudiantes. Permite importar directamente los datos técnicos del diseño eléctrico, genera automáticamente las partidas necesarias, y estructura el presupuesto final de forma clara, precisa y rápida. A pesar de que no incluye formatos oficiales del Estado como S10, su utilidad práctica es muy alta para el diseño técnico y la evaluación económica preliminar de proyectos de distribución eléctrica.

A continuación, se presenta una síntesis de la comparación realizada:

Tabla 14
Síntesis del análisis comparativo

Criterio	Software Desarrollado	Microsoft Excel	S10
Accesibilidad	Gratuito, código abierto	Licencia pagada	Licencia pagada
Integración con DIRECAD u otros programas	Total	Parcial	No disponible
Especialización en proyectos eléctricos	Alta	Depende del usuario	Media
Visualización del presupuesto	Clara, por niveles y redes	Alta, pero requiere configuración	Estructurada para informes oficiales
Tiempo de elaboración	Rápido	Medio	Rápido
Adaptabilidad	Alta (modificable por el usuario)	Alta	Limitada

Nota. Elaboración propia.

4.7.3. Análisis de Tiempos de Elaboración

El tiempo requerido para la elaboración de presupuestos es un factor importante en la gestión de proyectos de electrificación, especialmente en contextos rurales donde los plazos de ejecución suelen ser ajustados. Para evaluar la eficiencia del software desarrollado, se realizó un análisis comparativo del tiempo empleado en la generación del presupuesto del proyecto Challabamba Alta utilizando tres métodos:

1. Método manual (Excel):

- **Tiempo promedio:** 8 horas.
- **Procesos involucrados:**
 - Ingreso manual de metrados desde planos.
 - Cálculo individual de partidas (subtotales, totales).
 - Revisión de fórmulas y consistencia de datos.

2. Software S10:

- **Tiempo promedio:** 3 horas.
- **Procesos involucrados:**
 - Carga manual de metrados (no integra DIREDCAD).
 - Configuración de partidas eléctricas (requiere adaptación).
 - Generación de reportes estructurados.

3. Software desarrollado (esta investigación):

- **Tiempo promedio:** 1.5 horas.
- **Procesos optimizados:**
 - Importación automática de metrados desde DIREDCAD.
 - Cálculo en tiempo real de costos (subtotales, IGV, utilidad).
 - Generación instantánea de reportes y gráficos.

Esto demuestra una mejora significativa en la eficiencia operativa, lo que favorece una gestión más ágil de los procesos de presupuestación en proyectos de electrificación rural.

4.7.4. Análisis de errores en la Elaboración

Durante la fase de validación del software desarrollado, se identificó un error menor, no relacionado con cálculos ni con la lógica del código, sino con la interfaz gráfica de usuario (GUI).

Tabla 15

Comparación de errores detectados según el método de elaboración

Método	Total de errores detectados	Tipo de error predominante
Elaboración manual	6	Transcripción, cálculos incorrectos, omisiones
Software S10	3	Adaptación inadecuada de partidas eléctricas
Software desarrollado	1	Desajuste en interfaz gráfica según la resolución de la pantalla

Nota. Resultados obtenidos durante la etapa de validación experimental del software.

Este único error en el software desarrollado corresponde a una variación en la disposición de los botones y campos de entrada en pantallas con diferentes resoluciones o versiones de Windows. Esta variabilidad no afecta el funcionamiento del software ni la precisión de los resultados, pero puede dificultar la interacción del usuario si no se ajusta manualmente la ventana.

Este caso refleja una mejora significativa en comparación con otros métodos:

- 83.3 % menos errores que el método manual.
- 66.7% menos errores que el uso del software S10.

4.8. Conclusión del capítulo

La implementación y validación del software desarrollado permitió demostrar su efectividad en la gestión de costos y presupuestos para proyectos de electrificación rural, como el caso aplicado al sector Challabamba Alta. La integración con herramientas de diseño eléctrico, como DIREDCAD, y la automatización en la generación de presupuestos jerárquicos

(organizados en red primaria y secundaria) confirman su funcionalidad práctica y su enfoque técnico especializado.

El análisis comparativo evidenció mejoras sustanciales respecto a métodos tradicionales. En cuanto a eficiencia temporal, el software redujo el tiempo de elaboración de presupuestos en un 81.25% respecto al método manual (Excel) y en un 50% respecto a S10. Esto se debe principalmente a la automatización de procesos clave como la importación de metrados, el cálculo de costos y la generación de reportes estructurados.

Asimismo, el análisis de errores mostró que el software desarrollado presenta una precisión superior, con solo un error menor relacionado con la visualización de la interfaz en diferentes resoluciones de pantalla, sin impacto en los cálculos ni en la integridad de los datos. En comparación, el método manual presentó 6 errores y S10 un total de 3, reflejando una reducción de errores del 83.3% y 66.7% respectivamente.

Además, el sistema incorpora una base de datos interna con partidas, insumos y estructuras típicas de proyectos de electrificación rural, lo que facilita su implementación inmediata en contextos similares. A diferencia de S10, cuyo costo y complejidad limitan su uso académico o institucional, el software propuesto es libre, accesible, escalable y adaptable, lo cual lo convierte en una alternativa estratégica tanto para profesionales como para estudiantes y técnicos en formación.

En resumen, los resultados demuestran que el software no solo satisface las exigencias técnicas para la elaboración de presupuestos en redes eléctricas rurales, sino que también agiliza el trabajo, reduce los errores y facilita el acceso a herramientas especializadas. De esta manera, contribuye de forma significativa a planificar proyectos de electrificación en zonas de difícil acceso de manera más eficiente y sostenible.

CONCLUSIONES Y SUGERENCIAS

CONCLUSIONES

1. El software desarrollado demuestra ser una solución eficiente, accesible y adaptable para la gestión de costos y presupuestos en proyectos de sistemas de distribución eléctrica. Su arquitectura de código abierto garantiza libre disponibilidad, posibilidad de personalización y escalabilidad, lo que facilita su implementación en diversos contextos. Estas cualidades lo consolidan como una herramienta valiosa para instituciones, profesionales y estudiantes del sector eléctrico, contribuyendo a una gestión más precisa, ágil y pertinente en proyectos de electrificación rural.
2. El análisis de los fundamentos teóricos permitió establecer una base sólida para el diseño e implementación del software propuesto. Se identificaron los principios clave de la gestión de costos en proyectos de distribución eléctrica, como la estructuración jerárquica de partidas, el análisis de precios unitarios y la importancia del metrado. Este marco conceptual sirvió de guía para el desarrollo del sistema, asegurando que las funcionalidades implementadas respondan a los requerimientos técnicos y prácticos de proyectos de electrificación rural.
3. El diseño de la interfaz del software se centró en la simplicidad, usabilidad y eficiencia. Se logró una estructura intuitiva que facilita la navegación y el uso del sistema sin requerir conocimientos avanzados en programación. Los procesos complejos, como la importación de metrados desde DIREDCAD o la asignación automática de partidas, fueron optimizados para ejecutarse en pocos pasos. Esta orientación al usuario final constituye una mejora frente a herramientas tradicionales más complejas, especialmente en entornos académicos y profesionales donde la accesibilidad tecnológica es clave. Además, el desarrollo del sistema se realizó íntegramente en

Python, un lenguaje de programación de código abierto, lo que favorece su adaptabilidad, escalabilidad y distribución libre sin costos de licenciamiento.

4. La aplicación del software desarrollado en el proyecto Challabamba Alta permitió validar su funcionalidad mediante la comparación con el método manual (excel) y con el software S10. Los resultados evidencian una mejora significativa en la eficiencia de elaboración de presupuestos: el tiempo promedio de procesamiento se redujo de 8 horas con el método manual y 3 horas con S10, a solo 1.5 horas con el sistema propuesto, gracias a la automatización de cálculos y a la importación directa de metrados desde DIREDCAD. En cuanto a precisión, los errores detectados pasaron de 6 en el método manual y 3 en S10 a únicamente 1 en el software desarrollado, asociado a un ajuste visual de la interfaz. Estas mejoras confirman que la implementación del sistema optimiza de manera sustancial la gestión de costos y presupuestos en proyectos de electrificación rural.

SUGERENCIAS

1. Se recomienda incorporar una base de datos más amplia y actualizada de insumos eléctricos, considerando diferentes regiones del país, costos logísticos, y variaciones en normativas técnicas regionales, a fin de mejorar la precisión del presupuesto generado.
2. El software puede extenderse para ser aplicado en proyectos de alumbrado público, sistemas fotovoltaicos, centros de transformación y redes urbanas, lo que incrementaría su aplicabilidad y valor dentro del sector eléctrico.
3. Se sugiere desarrollar funciones que permitan exportar automáticamente los presupuestos a formatos oficiales requeridos por entidades públicas (como el F-6, F-7 o formatos de Invierte.pe), facilitando su uso en procesos de licitación o evaluación estatal.
4. Para aumentar la confiabilidad del software, se recomienda incorporar mecanismos de validación automática de datos de entrada (por ejemplo, niveles incorrectos, códigos duplicados o métricas fuera de rango), así como alertas de errores en tiempo real.
5. A fin de mejorar la accesibilidad, sería beneficioso portar el sistema a versiones web o móviles que puedan ejecutarse desde cualquier dispositivo con conexión a Internet, lo cual favorecería su uso en campo o en comunidades remotas.
6. Como línea futura, se propone la integración del software con sistemas de información geográfica para georreferenciar postes, transformadores y líneas, permitiendo así una planificación espacial más precisa del proyecto de electrificación.

BIBLIOGRAFÍA

- Araya, F. (2019). *Estado Del Arte Del Uso De BIM Para La Resolución De Demandas En Proyectos*.
- CAPECO. (2023). *Cámara Peruana de la Construcción*.
cersa.org.pe. (2023).
- Charles , H. T. (2019). *Contabilidad de costos: Un enfoque gerencial*.
- Chino Fur, D. A. (2023). *Eficiencia Del Software Delphin Express Para La Elaboración De Costos Y Presupuestos*. Moquegua.
- CNE-Suministro. (2011). *Código Nacional de Electricidad*.
- Contreras López, D. C., & Maldonado Rodríguez, S. E. (2021). *Software Para La Gestión Presupuestal De Obras De Construcción En Pequeñas Y Medianas Empresas*. Bogotá.
ddexpress.com, Delphin Express. (2022).
- Delgado Quintero, S. (2023). *Aprende Python*.
- DGE. (2002). *Procedimientos Para La Elaboración De Proyectos Y Ejecución De Obras En Sistemas De Distribución Y Sistemas De Utilización En Media Tensión En Zonas De Concesión De Distribución*.
- IEEE - Instituto de Ingenieros Eléctricos y Electrónicos. (2000).
- Kerzner, H. (2017). *Gestión de Proyectos: Un Enfoque Sistémico para la Planificación, Programación y Control*.
- Medina Velasque, E. R. (2021). *Evaluación comparativa del S10, Delphin Express, Arquímedes y Sistemas RW7*.
microsoft.com. (2024). <https://support.microsoft.com/es-es/office/tareas>
- MVCS. (2010). *Metrados Para Obras De Edificación Y Habilitaciones Urbanas*.
- Norma Técnica Peruana. (2017).
- OSINERGMIN. (2016). *Informe Técnico N° DSE-CT-130-2016*. Cusco.

Padilla Carreño, U. (2012). *Análisis De Costos Unitarios*.

Pontia, T. (2023). *Librerías De Python Para El Análisis De Datos*.

Ramírez Castaño, S. (2004). *Redes de distribución de energía*.

Ramos Salazar, J. (2015). *Costos Y Presupuestos En Edificaciones*.

Riofrío, C. (2014). *Apuntes de distribución de energía eléctrica*.

RNE. (2006). *Reglamento Nacional de Edificaciones*.

Robles, G. (2018). *Introducción al Software Libre*.

Rodríguez, J., & Gómez, L. (2019). *Aplicación de Herramientas Digitales en la Gestión de Costos*.

S10peru.com. (2023).

Salinas Seminario, M. (2012). *Costos Y Presupuestos De Obra*.

Sallam, A. A., & Malik, O. P. (2011). *Sistemas eléctricos De Distribución*.

Suárez Morales, E. F. (2015). *Diseño De Una Herramienta Informática Para El Cálculo De Precios Unitarios, Presupuestos Y Fórmulas De Reajuste*.

twenergy.com. (2013). *twenergy.com: twenergy.com*

www.edpproyectos.com. (2022).

ANEXOS

ANEXO A

INTERFAZ GRÁFICA DEL SOFTWARE



Resumen General Gastos Generales Sum RP mon RP Sum RS mon RS Partidas RP Partidas RS

Recargar Datos Exportar a PDF

Tabla de Resumen

DESCRIPCION	REDES PRIMARIAS	REDES SECUNDARIAS	TOTAL
SUMINISTRO DE MATERIALES	-	-	-
AI SLADORES	\$/ 830.00	\$/ 473.00	\$/ 1,303.00
ALUMBRADO PUBLICO	-	\$/ 4,814.10	\$/ 4,814.10
CONDUCTORES	\$/ 950.08	\$/ 9,954.85	\$/ 10,904.93
FERRETERIA	\$/ 2,318.60	\$/ 2,115.42	\$/ 4,434.02
POSTES Y ACCESORIOS DE CONCRETO ARMADO	\$/ 2,130.00	\$/ 3,812.00	\$/ 5,942.00
PUESTA A TIERRA	\$/ 1,957.80	\$/ 3,309.84	\$/ 5,267.64
RETENIDAS	\$/ 717.60	\$/ 13,972.60	\$/ 13,972.20
SECCIONADORES, PARARRAYOS, TERMINACIONES Y ACCESORIOS	\$/ 3,014.10	-	\$/ 3,014.10
TRANSFORMADORES	\$/ 6,413.16	-	\$/ 6,413.16
TOTAL SUMINISTRO DE MATERIALES	\$/ 18,331.34	\$/ 37,733.81	\$/ 56,065.15
MONTAJE ELECTROMECHANICO	-	-	-
AI SLADORES	\$/ 429.76	\$/ 2,329.31	\$/ 2,759.07
CONDUCTORES Y CABLES	\$/ 353.31	\$/ 1,195.13	\$/ 1,548.44
FERRETERIA	\$/ 1,198.68	\$/ 6,633.02	\$/ 7,831.70
MONTAJE ELECTROMECHANICO ALUMBRADO PÚBLICO	-	\$/ 4,051.89	\$/ 4,051.89
POSTES Y ACCESORIOS DE CONCRETO ARMADO	\$/ 1,763.22	\$/ 11,313.06	\$/ 13,076.28
PRUEBAS Y PUESTA EN SERVICIO	\$/ 1,763.22	\$/ 1,384.10	\$/ 3,147.32
PUESTA A TIERRA	\$/ 1,194.73	\$/ 1,208.24	\$/ 2,402.97
RETENIDAS	\$/ 560.97	\$/ 32,625.45	\$/ 33,186.42
SECCIONADORES, PARARRAYOS Y TERMINACIONES	\$/ 762.64	-	\$/ 762.64
TRABAJOS PRELIMINARES	\$/ 2,170.65	\$/ 657.68	\$/ 2,828.33
TRANSFORMADORES	\$/ 493.47	-	\$/ 493.47
TOTAL MONTAJE ELECTROMECHANICO	\$/ 10,690.65	\$/ 61,397.88	\$/ 72,088.53
A. Suministro de Materiales	\$/ 18,331.34	\$/ 37,733.81	\$/ 56,065.15
B. Montaje Electromecánico	\$/ 10,690.65	\$/ 61,397.88	\$/ 72,088.53
C. Transporte	\$/ 1,649.82	\$/ 3,396.04	\$/ 5,045.86
TOTAL COSTO DIRECTO	\$/ 30,671.81	\$/ 102,527.73	\$/ 133,199.54
D. Gastos Generales y Utilidades	\$/ 6,134.36	\$/ 20,505.55	\$/ 26,639.91
E. Supervisión de Obra	\$/ 2,147.03	\$/ 7,176.94	\$/ 9,323.97
SUB TOTAL	\$/ 38,953.20	\$/ 130,210.22	\$/ 169,163.42

Descripción generales

MonRP 1.1.1 Modificar

Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)

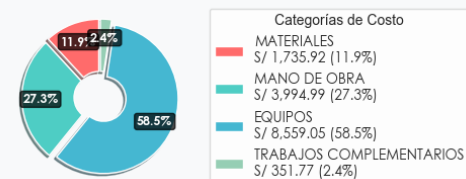
CODIGO	DESCRIPCION	CUAI	UND	CAN1	P.UNIT	SUBTO	TOTAL
1.1.1	MATERIALES						100.95
1.1.1	Piedra Mediana de Canter		M3	0.37	50.0	18.5	
1.1.1	Cemento Portland Tipo I		Blsa	2.05	23.0	47.15	
1.1.1	Hormigón		M3	0.49	70.0	34.3	
1.1.1	Agua Transportada		M3	0.2	5.0	1.0	
1.1.1	MANO DE OBRA						82.1
1.1.1	Operario		0.1	HH	0.1	14.26	1.43
1.1.1	Oficial		1	HH	1.0	12.57	12.57
1.1.1	Peón		6	HH	6.0	11.35	68.1

RENDIMIENTO: 8 UNIDAD: Pza/Día

Análisis de Costos

Análisis de Costos del Proyecto

Actualizar



Selección de Armado

Código: 4 2

Seleccione un armado:

ARMADO AB1

Actualizar

Insertar Armado

Componentes del Armado

CODIGO	DESCRIPCION	MULTIPLICADOR	UNIDAD	CANTIDAD	PRECIO UNITARIO	TOTAL
4.2.1	Abrazadera Tipo CAS Simple de 64mm, E=6.4mm, D=160mm, C/3P/3T/3CT/6A/3AI	1.0	Und	16	48	768
4.2.1	GRAPA DE SUSPENSIÓN DE ALUMINIO CUBIERTA DE PINTURA PLASTIFICADA (SO	1.0	Und	101	S/ 18.00	S/ 51.00
4.2.2	Abrazadera de Fijación Tipo U de 64mm, E=6.4mm, D=150mm, C/2P/2T/2CT/4A/2	1.0	Und	16	45	720
4.2.2	PERNO CON GANCHO DE SUSPENSIÓN D=16 mm, L=228 mm, C/T/CT/2A/AP (SOT	1.0	Und	101	S/ 15.00	S/ 45.00
4.2.3	Cruceta de Perfil Angular de F°G° de 64x64x2,500mm, E=6.4mm, con 3 dados de	1.0	Und	16	250	4000
4.2.3	ARANDELA CUADRADA CURVA DE 57X57X5MM, D=17.5MM	2.0	Und	202	S/ 4.00	S/ 250.00
4.2.4	Perno Maquinado L=50mm, D=13mm, C/T/CT/2A/AP	2.0	Und	32	2.5	80
4.2.5	Arandela Cuadrada Curva de A°G° de 57x57x5 mm, agujero de 17.5mmØ	2.0	Und	32	2.8	89.6
4.2.6	Riostra de Perfil Angular de F°G° de 38x38x1,000mm, E=5mm	2.0	Und	32	22	704

Imagen del Armado



Formulario de Datos

Nivel:

Código:

Descripción:

Unidad:

Cantidad:

Precio Unitario:

Mostrar

Limpiar

Guardar y Ajustar

Registros

NIVEL	CODIGO	DESCRIPCION	UNIDAD	CANTIDAD	PRECIO UNITARIO	TOTAL

Seleccione los CSV a exportar

- ☐ Resumen
- ☐ Gastos Generales
- ☐ Suministro RP
- ☐ Montaje RP
- ☐ Suministro RS
- ☐ Montaje RS
- ☐ Partidas RP
- ☐ Partidas RS
- ☐ Relación de Costos

Seleccionar carpeta destino

Destino:

Nombre archivo base (sin extensión): Reporte_Consolidado

Exportar a PDF

Descripción generales

MonRP

1.1.1

Modificar

Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)

CODIGO	DESCRIPCION	CUANTIA	UND	CANTIDAD	P.UNIT	SUBTOTAL	TOTAL
1.1.1	MATERIALES						100.95
1.1.1	Piedra Mediana de Cantera		M3	0.37	50.0	18.5	
1.1.1	Cemento Portland Tipo I		Blsa	2.05	23.0	47.15	
1.1.1	Hormigón		M3	0.49	70.0	34.3	
1.1.1	Agua Transportada		M3	0.2	5.0	1.0	
1.1.1	MANO DE OBRA						82.1
1.1.1	Operario	0.1	HH	0.1	14.26	1.43	
1.1.1	Oficial	1	HH	1.0	12.57	12.57	
1.1.1	Peón	6	HH	6.0	11.35	68.1	
1.1.1	EQUIPOS						59.6
1.1.1	Camion Grua 6 Tn.	0.5	HM	0.5	104.01	52.01	
1.1.1	Camioneta Rural 4 x 4 130 HP 5 Pasa	0.15	HM	0.15	50.59	7.59	
1.1.1	EQUIPOS DE SEGURIDAD Y HERRAMIE						8.2
1.1.1	Equipos de Seguridad	5	% MO			4.1	

RENDIMIENTO:

8

UNIDAD:

Pza/Día

Guardar Cambios

Agregar

Eliminar

Detalle	Costo (\$/.)
Detalle	Costo (\$/.)
Operario	14.26
Oficial	12.57
Peón	11.35
Ingeniero de Redes Primarias	26.14
Ingeniero de Redes Secundarias	26.14
Ingeniero Especialista	26.14
Profesional licenciado	26.14
Técnico Especialista en Dibujo por Computadora	11.88
Topógrafo Operador de Estación Total	17.11
Arqueologo	21.38
Abono foliar	3.6
Abono, potacio y fosforo	3.8
Agua Transportada	5
Aprobación del Informe Final del PMA	1071.2
Autorización para realizar el PMA con Infraestructura Nueva	926.8
Balon de Gas mas Soplete	15
Barrenos 5' x 1/8"	22

Datos cargados: 77 registros

ANEXO B

CODIGO ELABORADO EN PYTHON

Selected files

1 printable files

a_main.py

a_main.py

```

1  import ttkbootstrap as ttk
2
3  from ttkbootstrap.constants import *
4  from tkinter import PhotoImage
5
6  from C_funciones import on_button_click
7  from B1_sum_rp import crear_pestana_sum_rp
8  from B2_mon_rp import crear_pestana_mon_rp
9  from B3_sum_rs import crear_pestana_sum_rs
10 from B4_mon_rs import crear_pestana_mon_rs
11 from B5_gastos_generales import crear_pestana_gastos_generales
12 from B6_resumen import crear_pestana_resumen
13 from B7_partidas_mon_rp import crear_pestana_partidasRP
14 from B8_partidas_mon_rs import crear_pestana_partidasRS
15
16
17 from A1_panel_derecho_superior import crear_panel_derecho_superior
18 from A2_panel_derecho_inferior import crear_panel_derecho_inferior
19
20 #accesos directos
21 from K1_ACCS_guardar import guardar_excel
22 from K2_ACCS_guardar_pdf import guardar_pdf
23
24 from F1_ACCS_Ingresar_insumo import ingresar_insumo
25 from H1_ACCS_agregar_armado import open_add_armado_window
26 from K3_ACCS_modificar_partidas import run_main_gui
27 from K4_ACCS_Relacion_costos import abrir_relacion_costos
28 from K5_ACCS_automatizar_diredcad import Actualizar_metrado_diredcad
29 from K6_Info import info
30 #Función para los botones
31 from D_funcion_iconos import mostrar_carga
32
33
34 def main():
35     # Crear la ventana principal con el tema "cosmo"
36     root = ttk.Window(themename="cosmo")
37     root.title("Gestión de Costos y Presupuestos en Distribución")
38     root.geometry("1920x1080") # Establecer el tamaño de la ventana
39
40     # Crear un marco principal para dividir la ventana en 3 secciones
41     frame_principal = ttk.Frame(root)
42     frame_principal.pack(expand=True, fill="both", padx=10, pady=10)
43
44     # Crear un marco superior para los botones
45     frame_botones = ttk.Frame(frame_principal)
46     frame_botones.pack(fill="x", pady=5)

```

localhost:63616/83e53adc-5c76-47e9-9c13-3e246b08e299/

1/3

```

47
48 # Cargar imágenes de los botones
49 iconos = {
50     "Icono_guardar": PhotoImage(file="Icono_guardar.png"),
51     "pdf": PhotoImage(file="Icono_pdf.png"),
52     "Ingresar_Insumo": PhotoImage(file="Icono_Agregar_Insumo.png"),
53     "Agregar_Armado": PhotoImage(file="Icono_Agregar_Armado.png"),
54     "Modificar_Partida": PhotoImage(file="Icono_modificar_partida.png"),
55     "Ver_costos": PhotoImage(file="Icono_baseDatos.png"),
56     "Icono_Diredcad": PhotoImage(file="Icono_Diredcad.png"),
57     "Info": PhotoImage(file="Icono_info.png")
58 }
59
60 # Modificar el diccionario de botones para asignar la función correcta
61 botones_funciones = {
62     "Icono_guardar": lambda _: guardar_excel(),
63     "pdf": lambda _: guardar_pdf(),
64     "Ingresar_Insumo": lambda _: ingresar_insumo(),
65     "Agregar_Armado": lambda _: open_add_armado_window(),
66     "Modificar_Partida": lambda _: run_main_gui(),
67     "Ver_costos": lambda _: abrir_relacion_costos(),
68     "Icono_Diredcad": lambda _: Actualizar_metrado_diredcad(),
69     "Info": lambda _: info(),
70     "Icono_ejecutar": lambda _: mostrar_carga() # Aquí llamamos a la nueva función
71 }
72
73 # Crear botones con imágenes
74 for texto, icono in iconos.items():
75     btn = ttk.Button(
76         frame_botones, image=icono, command=lambda t=texto: botones_funciones[t](t),
77         bootstyle="link", cursor="hand2"
78     )
79     btn.image = icono
80     btn.pack(side="left", padx=5, pady=5)
81
82 # Crear un marco para los notebooks con `grid()` para manejar tamaños
83 frame_contenido = ttk.Frame(frame_principal)
84 frame_contenido.pack(expand=True, fill="both")
85
86 frame_contenido.columnconfigure(0, weight=1) # Notebook1 (izquierda, ancho dinámico)
87 frame_contenido.columnconfigure(1, weight=4) # Notebook2 y Notebook3 (derecha, 50%
del ancho total)
88 frame_contenido.rowconfigure(0, weight=3) # Notebook2 (arriba)
89 frame_contenido.rowconfigure(1, weight=1) # Notebook3 (abajo)
90
91 # Primer Notebook (izquierda, grande) → **Tiene pestañas**
92 notebook1 = ttk.Notebook(frame_contenido)
93 notebook1.grid(row=0, column=0, rowspan=2, sticky="nsew", padx=5, pady=5) # Ocupa
toda la izquierda
94
95 # Agregar pestañas al primer notebook
96 crear_pestana_resumen(notebook1)
97 crear_pestana_gastos_generales(notebook1)
98 crear_pestana_sum_rp(notebook1)

```


11/12/25, 12:02 p.m.

Selected files

```
99     crear_pestana_mon_rp(notebook1)
100     crear_pestana_sum_rs(notebook1)
101     crear_pestana_mon_rs(notebook1)
102     crear_pestana_partidasRP(notebook1)
103     crear_pestana_partidasRS(notebook1)
104
105
106     # Segundo Notebook (derecha, superior)
107     notebook2 = ttk.Notebook(frame_contenido)
108     notebook2.grid(row=0, column=1, sticky="nsew", padx=5, pady=5)
109
110
111     tab_derecha = crear_panel_derecho_superior(notebook2)
112
113     # Tercer Notebook (derecha, abajo) → **Sin pestañas, solo un marco**
114     notebook3 = ttk.Notebook(frame_contenido)
115     notebook3.grid(row=1, column=1, sticky="nsew", padx=5, pady=5) # 50% derecha, abajo
116
117     tab_derecha = crear_panel_derecho_inferior(notebook3)
118
119     # Iniciar la interfaz gráfica
120     root.mainloop()
121
122 if __name__ == "__main__":
123     main()
124
```

localhost:63616/83e53adc-5c76-47e9-9c13-3e246b08e299/

3/3

A1_panel_derecho_superior.py

```

1 import ttkbootstrap as ttk
2 from ttkbootstrap.constants import *
3 import os
4 import csv
5 from tkinter import messagebox
6 from I1_actualizar_partida import corregir_cantidades_y_subtotales_por_partida
7
8 opciones_archivos = {
9     "MonRP": "BBDD_monRP.csv",
10    "MonRS": "BBDD_monRS.csv"
11 }
12
13 def extraer_codigos_desde_csv(archivo):
14     """Extrae códigos de archivos CSV con encabezados de diccionario."""
15     codigos = []
16     if os.path.exists(archivo):
17         with open(archivo, "r", encoding="utf-8-sig") as f: # Changed to utf-8-sig
18             reader = csv.DictReader(f)
19             codigos = [fila["CODIGO"] for fila in reader if "CODIGO" in fila and
20 fila["CODIGO"].strip()]
21     return codigos if codigos else ["(sin códigos)"]
22
23 def extraer_codigos_unicos_mon(archivo="BBDD_partidasRP.csv"):
24     """Extrae todos los códigos únicos de la primera columna de un archivo de partidas,
25 preservando el orden original del CSV."""
26     codigos_vistos = set()
27     codigos_orden = []
28     if os.path.exists(archivo):
29         # Prefer utf-8-sig, then fall back to utf-8 if needed
30         for encoding in ("utf-8-sig", "utf-8", "latin1"):
31             try:
32                 with open(archivo, "r", encoding=encoding) as f:
33                     reader = csv.reader(f)
34                     for fila in reader:
35                         if fila and fila[0].strip():
36                             codigo = fila[0].strip()
37                             if codigo not in codigos_vistos:
38                                 codigos_vistos.add(codigo)
39                                 codigos_orden.append(codigo)
40
41                     break
42             except UnicodeDecodeError:
43                 continue
44     return codigos_orden if codigos_orden else ["(sin códigos)"]
45
46 def obtener_descripcion_por_codigo(archivo, codigo):
47     """Obtiene la descripción de un código de un archivo CSV con encabezados de
48 diccionario."""
49     if os.path.exists(archivo):
50         with open(archivo, "r", encoding="utf-8-sig") as f: # Changed to utf-8-sig
51             reader = csv.DictReader(f)
52             for fila in reader:
53                 if fila.get("CODIGO") == codigo:

```



```

51         return fila.get("DESCRIPCION", "(sin descripción)").strip()
52     return "(código no encontrado o sin descripción)"
53
54 def obtener_descripcion_general_por_codigo(archivo, codigo):
55     """Obtiene la descripción general de un código de un archivo CSV con encabezados de
    diccionario."""
56     if os.path.exists(archivo):
57         with open(archivo, "r", encoding="utf-8-sig") as f: # Changed to utf-8-sig
58             reader = csv.DictReader(f)
59             for fila in reader:
60                 if fila.get("CODIGO") == codigo:
61                     return fila.get("DESCRIPCION GENERAL", "(sin descripción
    general)").strip()
62         return "(código no encontrado o sin descripción general)"
63
64 def obtener_descripcion_partida(codigo_seleccionado, archivo_partidas):
65     """Obtiene la descripción de la partida buscando después de la fila que contiene
    'PARTIDA'"""
66     descripcion = "(sin descripción partida)"
67
68     if not os.path.exists(archivo_partidas):
69         return descripcion
70
71     # Prefer utf-8-sig, then fall back to utf-8 if needed
72     for encoding in ("utf-8-sig", "utf-8", "latin1"):
73         try:
74             with open(archivo_partidas, "r", encoding=encoding) as f:
75                 reader = csv.reader(f)
76                 filas = [fila for fila in reader if fila and
    fila[0].startswith(codigo_seleccionado)]
77
78                 for i, fila in enumerate(filas):
79                     if len(fila) > 1 and fila[1].strip().upper() == "PARTIDA":
80                         if i + 1 < len(filas) and len(filas[i + 1]) > 1:
81                             descripcion = filas[i + 1][1].strip()
82                             break
83                 break
84         except UnicodeDecodeError:
85             continue
86
87     return descripcion
88
89 def modificar_descripcion_general_en_csv(archivo, codigo, nueva_descripcion):
90     """Modifica la descripción general para un código específico en un archivo CSV."""
91     if not os.path.exists(archivo):
92         return False
93
94     filas = []
95     exito = False
96
97     with open(archivo, "r", encoding="utf-8-sig") as f: # Changed to utf-8-sig
98         reader = csv.DictReader(f)
99         campos = reader.fieldnames
100         for fila in reader:

```

```

101         if fila.get("CODIGO") == codigo:
102             fila["DESCRIPCION GENERAL"] = nueva_descripcion
103             exito = True
104             filas.append(fila)
105
106     if exito:
107         with open(archivo, "w", encoding="utf-8-sig", newline='') as f: # Changed to utf-
8-sig
108             writer = csv.DictWriter(f, fieldnames=campos)
109             writer.writeheader()
110             writer.writerows(filas)
111
112     return exito
113
114 def modificar_rendimiento_en_csv(archivo_partidas, codigo, nuevo_rendimiento,
nueva_unidad):
115     """Modifica el rendimiento y la unidad para un código específico en el archivo de
partidas."""
116     if not os.path.exists(archivo_partidas):
117         return False
118
119     filas = []
120     exito = False
121
122     # Prefer utf-8-sig for reading, then fall back if needed
123     for encoding in ("utf-8-sig", "utf-8", "latin1"):
124         try:
125             with open(archivo_partidas, "r", encoding=encoding) as f:
126                 reader = csv.reader(f)
127                 for fila in reader:
128                     if (len(fila) > 0 and fila[0].strip() == codigo and
129                         len(fila) > 1 and
130                         fila[1].strip().upper().startswith("RENDIMIEN")):
131
132                         if len(fila) > 2:
133                             fila[2] = nuevo_rendimiento
134                         else:
135                             fila.extend([""] * (3 - len(fila)))
136                             fila[2] = nuevo_rendimiento
137
138                         if len(fila) > 3:
139                             fila[3] = nueva_unidad
140                         else:
141                             fila.extend([""] * (4 - len(fila)))
142                             fila[3] = nueva_unidad
143                         exito = True
144                         filas.append(fila)
145                         break
146         except UnicodeDecodeError:
147             continue
148         except Exception as e:
149             messagebox.showerror("Error de lectura", f"Error al leer el archivo
{archivo_partidas} con codificación {encoding}: {e}")
150     return False

```

```

150
151     if exito:
152         try:
153             with open(archivo_partidas, "w", encoding="utf-8-sig", newline='') as f: #
Changed to utf-8-sig
154                 writer = csv.writer(f)
155                 writer.writerows(filas)
156                 return True
157         except Exception as e:
158             messagebox.showerror("Error de escritura", f"Error al escribir en el archivo
{archivo_partidas}: {e}")
159             return False
160     return False
161
162 def cargar_partidas_relacionadas(codigo_base, archivo_partidas):
163     """Carga todas las filas relacionadas con un código base de MonRP o MonRS."""
164     resultados = []
165     if not os.path.exists(archivo_partidas):
166         return resultados
167
168     # Prefer utf-8-sig, then fall back to utf-8 if needed
169     for encoding in ("utf-8-sig", "utf-8", "latin1"):
170         try:
171             with open(archivo_partidas, "r", encoding=encoding) as f:
172                 reader = csv.reader(f)
173
174                 dentro_de_partida_base = False
175
176                 for fila in reader:
177                     if len(fila) > 0 and fila[0].strip() == codigo_base:
178                         dentro_de_partida_base = True
179                     elif len(fila) > 0 and fila[0].strip() != codigo_base and
dentro_de_partida_base:
180                         break
181
182                     if dentro_de_partida_base:
183                         num_expected_cols = 8 # Changed from 7 to 8 for the new column
184                         while len(fila) < num_expected_cols:
185                             fila.append("")
186
187                         resultados.append({
188                             "CODIGO": fila[0].strip(),
189                             "DESCRIPCION": fila[1].strip(),
190                             "CUANTIA": fila[2].strip(),
191                             "UND": fila[3].strip(),
192                             "CANTIDAD": fila[4].strip(),
193                             "P.UNIT": fila[5].strip(),
194                             "SUBTOTAL": fila[6].strip(),
195                             "TOTAL": fila[7].strip()
196                         })
197                 break
198         except UnicodeDecodeError:
199             continue
200

```



```

201     return resultados
202
203 def obtener_rendimiento_por_codigo(codigo, archivo_partidas):
204     """Busca el rendimiento y la unidad para un código específico."""
205     rendimiento = "(no encontrado)"
206     unidad = "(no encontrada)"
207
208     if not os.path.exists(archivo_partidas):
209         return rendimiento, unidad
210
211     # Prefer utf-8-sig, then fall back to utf-8 if needed
212     for encoding in ("utf-8-sig", "utf-8", "latin1"):
213         try:
214             with open(archivo_partidas, "r", encoding=encoding) as f:
215                 reader = csv.reader(f)
216
217                 for fila in reader:
218                     if (len(fila) > 0 and fila[0].strip() == codigo and
219                         len(fila) > 1 and
220                         fila[1].strip().upper().startswith("RENDIMIENTO")):
221
222                         if len(fila) > 2 and fila[2].strip():
223                             rendimiento = fila[2].strip()
224                         if len(fila) > 3 and fila[3].strip():
225                             unidad = fila[3].strip()
226
227                         return rendimiento, unidad
228
229                     break
230         except UnicodeDecodeError:
231             continue
232
233     return rendimiento, unidad
234
235 def guardar_partidas_en_csv(partidas, archivo="BBDD_partidasRP.csv"):
236     """Guarda una lista de partidas (diccionarios) en un archivo CSV."""
237     if not partidas:
238         return
239
240     campos = partidas[0].keys()
241     with open(archivo, "w", encoding="utf-8-sig", newline='') as f: # Changed to utf-8-
242         sig
243         writer = csv.DictWriter(f, fieldnames=campos)
244         writer.writeheader()
245         writer.writerows(partidas)
246
247 def crear_panel_derecho_superior(notebook):
248     """Crea y configura el panel superior derecho de la interfaz gráfica."""
249     # Crear estilo para el Treeview
250     estilo = ttk.Style()
251     estilo.configure("Treeview",
252                     font=('Segoe UI', 8),
253                     rowheight=25)
254     estilo.configure("Treeview.Heading",
255                     font=('Segoe UI', 8, 'bold'))

```

```

253
254     tab = ttk.Frame(notebook, width=300, height=800)
255     notebook.add(tab, text="Descripción generales")
256
257     frame_controles = ttk.Frame(tab)
258     frame_controles.pack(side="top", fill="x", padx=10, pady=5)
259
260     combo1 = ttk.Combobox(frame_controles, values=list(opciones_archivos.keys()),
261 state="readonly")
262     try:
263         monrp_index = list(opciones_archivos.keys()).index("MonRP")
264         combo1.current(monrp_index)
265     except ValueError:
266         combo1.current(0)
267     combo1.pack(side="left", padx=5, pady=5, expand=True, fill="x")
268
269     combo2 = ttk.Combobox(frame_controles, values=["(esperando selección)"],
270 state="readonly")
271     combo2.current(0)
272     combo2.pack(side="left", padx=5, pady=5, expand=True, fill="x")
273
274     btn_modificar = ttk.Button(frame_controles, text="Modificar", bootstyle="primary")
275     btn_modificar.pack(side="left", padx=5, pady=5)
276
277     descripcion_texto = ttk.Text(tab, height=1, wrap="word", state="disabled")
278     descripcion_texto.pack(fill="x", padx=10, pady=(0, 5))
279
280     descripcion_general_texto = ttk.Text(tab, height=10, wrap="word", font=("Century
281 Gothic", 10, "italic"))
282     descripcion_general_texto.pack_forget()
283
284     frame_rendimiento = ttk.Frame(tab)
285     frame_rendimiento.pack_forget()
286
287     label_rendimiento = ttk.Label(frame_rendimiento, text="RENDIMIENTO:", font=("Segoe
288 UI", 10, "bold"))
289     label_rendimiento.pack(side="left", padx=(0, 5))
290
291     entry_rendimiento = ttk.Entry(frame_rendimiento, width=10, font=("Segoe UI", 10),
292 state="normal")
293     entry_rendimiento.pack(side="left", padx=(0, 10))
294
295     label_unidad = ttk.Label(frame_rendimiento, text="UNIDAD:", font=("Segoe UI", 10,
296 "bold"))
297     label_unidad.pack(side="left", padx=(0, 5))
298
299     entry_unidad = ttk.Entry(frame_rendimiento, width=15, font=("Segoe UI", 10),
300 state="readonly")
301     entry_unidad.pack(side="left")
302
303     widgets_actuales = {"text": None, "tabla": None}
304
305     ANCHO_CODIGO_MON = 55
306     ANCHO_DESCRIPCION_MON = 140

```

```

300     ANCHO_CUANTIA_MON = 40
301     ANCHO_UND_MON = 40
302     ANCHO_CANTIDAD_MON = 40
303     ANCHO_P_UNIT_MON = 60
304     ANCHO_SUBTOTAL_MON = 50
305     ANCHO_TOTAL_MON = 50
306
307     def actualizar_descripcion(event=None):
308         seleccion1 = combo1.get()
309         seleccion2 = combo2.get()
310         archivo = opciones_archivos.get(seleccion1)
311
312         descripcion_general_texto.pack_forget()
313         frame_rendimiento.pack_forget()
314
315         if widgets_actuales["tabla"]:
316             widgets_actuales["tabla"].destroy()
317             widgets_actuales["tabla"] = None
318
319         if seleccion1 in ["MonRP", "MonRS"]:
320             archivo_partidas = "BBDD_partidasRP.csv" if seleccion1 == "MonRP" else
"BBDD_partidasRS.csv"
321
322             descripcion_partida = obtener_descripcion_partida(seleccion2,
archivo_partidas)
323
324             descripcion_texto.configure(state="normal")
325             descripcion_texto.delete("1.0", "end")
326             descripcion_texto.insert("1.0", descripcion_partida)
327             descripcion_texto.configure(state="disabled")
328
329             todas_las_partidas_relacionadas = cargar_partidas_relacionadas(seleccion2,
archivo_partidas)
330
331             indice_inicio_datos_tabla = -1
332             for i, row_dict in enumerate(todas_las_partidas_relacionadas):
333                 if row_dict.get("DESCRIPCION", "").strip().upper() == "DESCRIPCION DE
RECURSO":
334                     indice_inicio_datos_tabla = i
335                     break
336
337             tabla_datos_a_mostrar = []
338             if indice_inicio_datos_tabla != -1:
339                 tabla_datos_a_mostrar = todas_las_partidas_r-
elacionadas[indice_inicio_datos_tabla + 1:]
340
341             if tabla_datos_a_mostrar:
342                 columns = ["CODIGO", "DESCRIPCION", "CUANTIA", "UND", "CANTIDAD",
"P.UNIT", "SUBTOTAL", "TOTAL"]
343                 tree = ttk.Treeview(tab, columns=columns, show="headings", height=8)
344                 estilo = ttk.Style()
345                 estilo.configure('highlight_row.Treeview',
font=('Segoe UI', 8, 'bold'),
346                 background='#f0f0f0')
347

```



```

348
349 tree.heading("CODIGO", text="CODIGO", anchor=CENTER)
350 tree.column("CODIGO", width=ANCHO_CODIGO_MON, anchor=CENTER)
351
352 tree.heading("DESCRIPCION", text="DESCRIPCION", anchor=W)
353 tree.column("DESCRIPCION", width=ANCHO_DESCRIPCION_MON, anchor=W)
354
355 tree.heading("CUANTIA", text="CUANTIA", anchor=CENTER)
356 tree.column("CUANTIA", width=ANCHO_CUANTIA_MON, anchor=CENTER)
357
358 tree.heading("UND", text="UND", anchor=CENTER)
359 tree.column("UND", width=ANCHO_UND_MON, anchor=CENTER)
360
361 tree.heading("CANTIDAD", text="CANTIDAD", anchor=CENTER)
362 tree.column("CANTIDAD", width=ANCHO_CANTIDAD_MON, anchor=CENTER)
363
364 tree.heading("P.UNIT", text="P.UNIT", anchor=CENTER)
365 tree.column("P.UNIT", width=ANCHO_P_UNIT_MON, anchor=CENTER)
366
367 tree.heading("SUBTOTAL", text="SUBTOTAL", anchor=CENTER)
368 tree.column("SUBTOTAL", width=ANCHO_SUBTOTAL_MON, anchor=CENTER)
369
370 tree.heading("TOTAL", text="TOTAL", anchor=CENTER)
371 tree.column("TOTAL", width=ANCHO_TOTAL_MON, anchor=CENTER)
372
373 highlight_phrases = [
374     "MATERIALES", "MANO DE OBRA", "EQUIPOS",
375     "EQUIPOS DE SEGURIDAD Y HERRAMIENTAS",
376     "TRABAJO COMPLEMENTARIOS", "COSTO UNITARIO"
377 ]
378
379 for row_dict in tabla_datos_a_mostrar:
380     values = [row_dict.get(col, "") for col in columns]
381     if row_dict.get("DESCRIPCION", "").strip().upper() in
highlight_phrases:
382         tree.insert("", "end", values=values, tags=('highlight_row',))
383     else:
384         tree.insert("", "end", values=values)
385
386 tree.pack(fill="both", expand=True, padx=10, pady=10)
387 widgets_actuales["tabla"] = tree
388
389 rendimiento, unidad = obtener_rendimiento_por_codigo(seleccion2,
archivo_partidas)
390
391 entry_rendimiento.configure(state="normal")
392 entry_rendimiento.delete(0, "end")
393 entry_rendimiento.insert(0, rendimiento)
394 entry_rendimiento.configure(state="normal") # Keep it editable
395
396 entry_unidad.configure(state="normal")
397 entry_unidad.delete(0, "end")
398 entry_unidad.insert(0, unidad)
399 entry_unidad.configure(state="readonly")

```

```

400
401         frame_rendimiento.pack(side="bottom", fill="x", padx=10, pady=5)
402     else:
403         msg = ttk.Label(tab, text="No se encontraron partidas relacionadas para el
código seleccionado.", font=("Segoe UI", 10, "italic"))
404         msg.pack(padx=10, pady=10)
405         widgets_actuales["tabla"] = msg
406
407     else:
408         descripcion_general_texto.pack(fill="both", expand=True, padx=10, pady=(0,
10))
409
410         descripcion = obtener_descripcion_por_codigo(archivo, seleccion2)
411         descripcion_general = obtener_descripcion_general_por_codigo(archivo,
seleccion2)
412
413         descripcion_texto.configure(state="normal")
414         descripcion_texto.delete("1.0", "end")
415         descripcion_texto.insert("1.0", descripcion)
416         descripcion_texto.configure(state="disabled")
417
418         descripcion_general_texto.configure(state="normal")
419         descripcion_general_texto.delete("1.0", "end")
420         descripcion_general_texto.insert("1.0", descripcion_general)
421         descripcion_general_texto.configure(state="disabled")
422
423     def actualizar_contenido(event=None):
424         seleccion = combo1.get()
425         archivo = opciones_archivos.get(seleccion)
426
427         if widgets_actuales["tabla"]:
428             widgets_actuales["tabla"].destroy()
429             widgets_actuales["tabla"] = None
430
431         descripcion_general_texto.pack_forget()
432         frame_rendimiento.pack_forget()
433
434         if seleccion == "MonRP":
435             codigos = extraer_codigos_unicos_mon(archivo="BBDD_partidasRP.csv")
436             descripcion_general_texto.configure(state="disabled")
437         elif seleccion == "MonRS":
438             codigos = extraer_codigos_unicos_mon(archivo="BBDD_partidasRS.csv")
439             descripcion_general_texto.configure(state="disabled")
440         else:
441             codigos = extraer_codigos_desde_csv(archivo)
442             descripcion_general_texto.configure(state="normal")
443
444         combo2["values"] = codigos
445         if codigos:
446             combo2.current(0)
447             actualizar_descripcion()
448
449     def modificar_datos_en_csv():
450         seleccion1 = combo1.get()

```



```

451     seleccion2 = combo2.get()
452
453     if seleccion1 in ["MonRP", "MonRS"]:
454         archivo_partidas = "BBDD_partidasRP.csv" if seleccion1 == "MonRP" else
"BBDD_partidasRS.csv"
455         nuevo_rendimiento = entry_rendimiento.get().strip()
456         nueva_unidad = entry_unidad.get().strip() # Although this is readonly, we can
still get its value if needed.
457
458         if modificar_rendimiento_en_csv(archivo_partidas, seleccion2,
nuevo_rendimiento, nueva_unidad):
459             messagebox.showinfo("Éxito", "Rendimiento y/o unidad modificados
correctamente.")
460             # After modification, re-read and update the displayed values
461             rendimiento, unidad = obtener_rendimiento_por_codigo(seleccion2,
archivo_partidas)
462             entry_rendimiento.configure(state="normal")
463             entry_rendimiento.delete(0, "end")
464             entry_rendimiento.insert(0, rendimiento)
465             entry_rendimiento.configure(state="normal") # Keep it editable
466
467             #calcular las partidas de acuerdo al nuevo rendimiento
468             csv_partidas = 'BBDD_partidasRS.csv'
469             csv_costos = 'BBDD_Relacion_de_Costos.csv'
470             corregir_cantidades_y_subtotales_por_partida(csv_partidas, csv_costos)
471
472             #calcular las partidas de acuerdo al nuevo rendimiento
473             csv_partidas = 'BBDD_partidasRP.csv'
474             csv_costos = 'BBDD_Relacion_de_Costos.csv'
475             corregir_cantidades_y_subtotales_por_partida(csv_partidas, csv_costos)
476
477         else:
478             messagebox.showerror("Error", "No se pudo modificar el rendimiento y/o
unidad. Asegúrese de que el código exista y el archivo sea modificable.")
479             return
480
481     else:
482         descripcion_general_texto.configure(state="normal")
483         nueva_desc = descripcion_general_texto.get("1.0", "end").strip()
484         descripcion_general_texto.configure(state="disabled")
485
486         archivo = opciones_archivos.get(seleccion1)
487         exito = modificar_descripcion_general_en_csv(archivo, seleccion2, nueva_desc)
488         if exito:
489
490             messagebox.showinfo("Éxito", "Descripción general modificada
correctamente.")
491
492         else:
493             messagebox.showerror("Error", "No se pudo modificar la descripción
general. Asegúrese de que el código exista y el archivo sea modificable.")
494
495
496
497     combo1.bind("<<ComboboxSelected>>", actualizar_contenido)

```

```
498     combo2.bind("<<ComboboxSelected>>", actualizar_descripcion)
499     btn_modificar.config(command=modificar_datos_en_csv)
500
501     actualizar_contenido()
502
503     return tab
504
505 if __name__ == "__main__":
506     root = ttk.Window(themename="cosmo")
507     root.title("Prueba: Panel Derecho Superior")
508
509     ancho_ventana = 1000
510     alto_ventana = 400
511     x = (root.winfo_screenwidth() // 2) - (ancho_ventana // 2)
512     y = (root.winfo_screenheight() // 2) - (alto_ventana // 2)
513     root.geometry(f"{ancho_ventana}x{alto_ventana}+{x}+{y}")
514
515     notebook = ttk.Notebook(root)
516     notebook.pack(expand=True, fill="both", padx=10, pady=10)
517
518     crear_panel_derecho_superior(notebook)
519
520     root.mainloop()
```

A2_panel_derecho_inferior.py

```

1 import csv
2 import tkinter as tk
3 from tkinter import ttk, messagebox
4 from matplotlib.figure import Figure
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6 import matplotlib.pyplot as plt
7 import matplotlib.font_manager as fm
8
9 # Importar ttkbootstrap si lo estás usando para el estilo general de la app
10 try:
11     import ttkbootstrap as ttkb
12     from ttkbootstrap.constants import *
13 except ImportError:
14     print("ttkbootstrap no está instalado. Se usará tkinter.ttk estándar.")
15     ttkb = None # Establecer a None si no está disponible
16
17 # --- Configurar la fuente "Century Gothic" globalmente ---
18 FONT_FAMILY = "Century Gothic"
19
20 # Añadir "Century Gothic" a las fuentes de Matplotlib
21 try:
22     fm.findfont(FONT_FAMILY, rebuild_cache=True)
23     plt.rcParams['font.family'] = FONT_FAMILY
24     plt.rcParams['font.sans-serif'] = [FONT_FAMILY] + plt.rcParams['font.sans-serif']
25 except Exception as e:
26     print(f"Advertencia: No se pudo configurar la fuente '{FONT_FAMILY}' en Matplotlib. Se usará la fuente por defecto. Error: {e}")
27
28 # Configurar el estilo de matplotlib para un aspecto más moderno
29 plt.style.use('seaborn-v0_8-darkgrid')
30
31 # --- Color de fondo deseado para los frames ---
32 BACKGROUND_COLOR = '#f5f5f5' # Un gris claro y suave
33
34 # --- Funciones Auxiliares (sin cambios en la lógica, solo para consistencia) ---
35 def sumar_categorias_totales_csv(nombres_archivos):
36     totales_categorias_global = {
37         "MATERIALES": 0.0, "MANO DE OBRA": 0.0, "EQUIPOS": 0.0,
38         "EQUIPOS DE SEGURIDAD Y HERRAMIENTAS": 0.0, "TRABAJO COMPLEMENTARIOS": 0.0
39     }
40     codificaciones_a_probar = ['latin-1', 'cp1252', 'utf-8']
41     delimitadores_a_probar = [',', '\t', ';', ' ']
42     archivos_procesados_exitosamente = 0
43     for nombre_archivo in nombres_archivos:
44         lector_exitoso_para_este_archivo = False
45         for encoding_test in codificaciones_a_probar:
46             for delimiter_test in delimitadores_a_probar:
47                 try:
48                     with open(nombre_archivo, mode='r', encoding=encoding_test,
49 errors='replace') as archivo_csv:
50                         lector_csv = csv.reader(archivo_csv, delimiter=delimiter_test)

```



```

50         totales_categorias_actual_archivo = {k: 0.0 for k in
totales_categorias_global.keys()}
51         for fila in lector_csv:
52             for categoria_nombre in totales_categorias_global.keys():
53                 if any(categoria_nombre.upper() in cell.upper() for cell
in fila):
54                     try:
55                         valor_str = fila[-1].strip()
56                         if valor_str:
57                             totales_categorias_a-
ctual_archivo[categoria_nombre] += float(valor_str)
58                     except:
59                         for i in range(len(fila) - 2, -1, -1):
60                             valor_str_prev = fila[i].strip()
61                             if valor_str_prev and
valor_str_prev.replace('.', '', 1).isdigit():
62                                 totales_categorias_a-
ctual_archivo[categoria_nombre] += float(valor_str_prev)
63                                 break
64                     except (ValueError, IndexError):
65                         pass
66                     break
67                 lector_exitoso_para_este_archivo = True
68                 for categoria, total in totales_categorias_actual_archivo.items():
69                     totales_categorias_global[categoria] += total
70                 archivos_procesados_exitosamente += 1
71                 break
72             except UnicodeDecodeError: continue
73             except FileNotFoundError:
74                 print(f"Error: El archivo '{nombre_archivo}' no fue encontrado.
Asegúrate de que los archivos están en la ruta correcta.")
75                 lector_exitoso_para_este_archivo = True; break
76             except Exception as e: continue
77             if lector_exitoso_para_este_archivo: break
78         if archivos_procesados_exitosamente == 0:
79             print("\n¡ADVERTENCIA!: No se pudo procesar ningún archivo. Asegúrate de que las
rutas son correctas y los archivos tienen el formato esperado.")
80         return {}
81     return totales_categorias_global
82
83 def print_console_summary(totales_por_categoria, no_data=False):
84     print("\n" + " " * 25)
85     print("      ANALISIS DE COSTOS POR CATEGORIA")
86     print(" " * 25)
87     if not no_data and totales_por_categoria:
88         total_general_impresion = sum(totales_por_categoria.values())
89         if total_general_impresion > 0:
90             for categoria, total in totales_por_categoria.items():
91                 porcentaje = (total / total_general_impresion * 100)
92                 print(f"      {categoria}: S/ {total:,.2f} ({porcentaje:.1f}%)")
93                 print(f"\n      TOTAL GENERAL: S/ {total_general_impresion:,.2f}")
94             else:
95                 print("      No se encontraron costos positivos para las categorías.")
96     else:

```

```

97     print(" No se pudieron obtener los totales de costos por categoría.")
98     print(" " * 25 + "\n")
99
100
101 # --- La función principal del dashboard ---
102
103 def crear_panel_derecho_inferior(notebook):
104     """
105     Crea una nueva pestaña en el notebook para el dashboard de Análisis de Costos
106     y lo puebla con el contenido generado.
107
108     Args:
109         notebook (ttk.Notebook): El widget ttk.Notebook al que se añadirá la nueva
110         pestaña.
111     """
112     tab_analisis_costos = ttk.Frame(notebook, style='Custom.TFrame' )
113     notebook.add(tab_analisis_costos, text="Análisis de Costos")
114
115     file_paths = ['BBDD_partidasRP.csv', 'BBDD_partidasRS.csv']
116
117     def update_dashboard_content():
118         totales_por_categoria = sumar_categorias_totales_csv(file_paths)
119
120         for widget in tab_analisis_costos.winfo_children():
121             if widget != control_buttons_frame:
122                 widget.destroy()
123
124         chart_frame = ttk.Frame(tab_analisis_costos, style='Custom.TFrame')
125         chart_frame.pack(padx=5, pady=3, fill=tk.BOTH, expand=True)
126
127         fig = None
128
129         if totales_por_categoria and sum(totales_por_categoria.values()) > 0:
130             data_para_grafico = {k: v for k, v in totales_por_categoria.items() if v > 0}
131             etiquetas = list(data_para_grafico.keys())
132             valores = list(data_para_grafico.values())
133             total_general = sum(valores)
134
135             colores_modernos = [
136                 '#FF6B6B', '#4ECDC4', '#45B7D1', '#96CEB4', '#FFEAA7',
137                 '#DDA0DD', '#F4A460', '#87CEEB', '#C7CEEA', '#FFD700',
138             ]
139             colores_a_usar = colores_modernos[:len(valores)]
140
141             fig = Figure(figsize=(5, 4), dpi=100, facecolor='#f8f9fa')
142             ax = fig.add_subplot(111)
143
144             wedges, texts, autotexts = ax.pie(
145                 valores,
146                 labels=None,
147                 autopct=lambda pct: f'{pct:.1f}%',
148                 startangle=90,
149                 pctdistance=0.75,
150                 colors=colores_a_usar,

```

```

150         explode=[0.05] * len(valores),
151         shadow=True,
152         wedgeprops=dict(width=0.7, edgecolor='white', linewidth=1)
153     )
154
155     for autotext in autotexts:
156         autotext.set_color('white')
157         autotext.set_weight('bold')
158         autotext.set_fontsize(9)
159         autotext.set_bbox(dict(boxstyle="round,pad=0.2", facecolor='black',
alpha=0.7))
160         autotext.set_fontfamily(FONT_FAMILY)
161
162     ax.axis('equal')
163
164     legend_labels = []
165     for etiq, val in zip(etiquetas, valores):
166         porcentaje_categoria = (val / total_general * 100) if total_general > 0
else 0
167         legend_labels.append(f"{etiq}\nS/ {val:,.2f}
({porcentaje_categoria:.1f}%)")
168
169     # --- CAMBIO: Configurar el fondo de la leyenda a blanco, sin sombra ---
170     ax.legend(wedges, legend_labels,
171             title="Categorías de Costo",
172             title_fontsize=10,
173             fontsize=9,
174             loc="center left",
175             bbox_to_anchor=(1, 0, 0.5, 1),
176             frameon=True, # Mantener el marco
177             facecolor='white', # Fondo blanco
178             edgecolor='lightgray', # Borde sutil (opcional, puedes quitarlo)
179             shadow=False, # Sin sombra
180             prop={'family': FONT_FAMILY})
181
182     fig.tight_layout(rect=[0, 0.03, 1, 0.98])
183
184     canvas = FigureCanvasTkAgg(fig, master=chart_frame)
185     canvas_widget = canvas.get_tk_widget()
186     canvas_widget.pack(fill=tk.BOTH, expand=True)
187     canvas.draw()
188
189     tab_analisis_costos.matplotlib_figure = fig
190     print_console_summary(totales_por_categoria)
191     else:
192         messagebox.showinfo("Gráfico de Costos", "No hay datos válidos o los costos
son cero para generar el gráfico.")
193         print_console_summary(totales_por_categoria, no_data=True)
194         tab_analisis_costos.matplotlib_figure = None
195
196     control_buttons_frame = ttk.Frame(tab_analisis_costos, style='Custom.TFrame')
197     control_buttons_frame.pack(fill=tk.X, pady=(5, 0), padx=10)
198
199     title_label = ttk.Label(control_buttons_frame, text="Análisis de Costos del Proyecto",

```



```

200         font=(FONT_FAMILY, 14, "bold"), foreground='#2c3e50',
background=BACKGROUND_COLOR)
201     title_label.pack(side=tk.LEFT, padx=(0, 20))
202
203     btn_actualizar = ttk.Button(control_buttons_frame, text=" Actualizar",
command=update_dashboard_content)
204     btn_actualizar.pack(side=tk.LEFT, padx=(0, 5))
205
206     update_dashboard_content()
207
208     # # --- Bloque de Ejecución Principal ---
209
210     # if __name__ == "__main__":
211     #     if ttkb:
212     #         root = ttkb.Window(themename="cosmo")
213     #         style = ttkb.Style()
214     #         style.configure('.', font=(FONT_FAMILY, 9))
215     #         style.configure('TButton', font=(FONT_FAMILY, 9))
216     #         style.configure('TLabel', font=(FONT_FAMILY, 9))
217     #         style.configure('Custom.TFrame', background=BACKGROUND_COLOR)
218     #     else:
219     #         root = tk.Tk()
220     #         root.configure(bg=BACKGROUND_COLOR)
221
222     #         default_font = tk.font.nametofont("TkDefaultFont")
223     #         default_font.configure(family=FONT_FAMILY, size=9)
224     #         text_font = tk.font.nametofont("TkTextFont")
225     #         text_font.configure(family=FONT_FAMILY, size=9)
226     #         heading_font = tk.font.nametofont("TkHeadingFont")
227     #         heading_font.configure(family=FONT_FAMILY, size=9)
228
229     #         root.title("Análisis de Costos del Proyecto")
230     #         root.geometry("750x550")
231
232     #         root.update_idletasks()
233     #         x = (root.winfo_screenwidth() // 2) - (root.winfo_width() // 2)
234     #         y = (root.winfo_screenheight() // 2) - (root.winfo_height() // 2)
235     #         root.geometry(f'+{x}+{y}')
236
237     #         main_notebook = ttk.Notebook(root)
238     #         main_notebook.pack(pady=5, expand=True, fill="both")
239
240     #         crear_panel_derecho_inferior(main_notebook)
241
242     #         root.mainloop()

```

B1_sum_rp.py

```

1 # modulo_sum_rp.py
2 import ttkbootstrap as ttk
3 import pandas as pd
4 from ttkbootstrap.constants import *
5 from C_funciones import ajustar_columnas, editar_celda
6 from E1_BBDD_RP_SUM import procesar_data_sumRP
7
8 # Cargar el archivo CSV en un DataFrame
9 df = pd.read_csv("BBDD_sumRP.csv", encoding="utf-8-sig")
10
11 # Actualizar el DataFrame original (df) desde la tabla
12 def actualizar_dataframe_y_guardar(tabla, archivo_csv="BBDD_sumRP.csv"):
13     global df # Asegurar que estamos trabajando con el DataFrame global original
14
15     # Crear un diccionario para actualizar las filas del DataFrame original
16     nuevos_datos = []
17
18     # Función recursiva para recorrer todos los nodos de la tabla
19     def recorrer_nodos(item):
20         valores = list(tabla.item(item, "values")) # Convertir valores de la tabla a
21         lista
22         if valores[0].startswith("|__"): # Limpiar el prefijo '|__' si existe
23             valores[0] = valores[0].replace("|__", "")
24         nuevos_datos.append(valores) # Agregar los valores al nuevo dataset
25         for hijo in tabla.get_children(item): # Recorrer hijos de cada nodo
26             recorrer_nodos(hijo)
27
28     # Recorrer la tabla para obtener los nuevos datos
29     for item in tabla.get_children():
30         recorrer_nodos(item)
31
32     # Crear un DataFrame temporal con las columnas visibles en la tabla
33     columnas_tabla = ["CODIGO", "DESCRIPCION", "UNIDAD", "CANTIDAD", "PRECIO UNITARIO",
34 "TOTAL"]
35     df_tabla = pd.DataFrame(nuevos_datos, columns=columnas_tabla)
36
37     # Actualizar las columnas editadas del DataFrame original (por ejemplo, CANTIDAD y
38     # PRECIO UNITARIO)
39     for index, row in df_tabla.iterrows():
40         codigo = row["CODIGO"] # Usar el código como identificador único
41         # Buscar la fila correspondiente en el DataFrame original y actualizar sus valores
42         if codigo in df["CODIGO"].values:
43             df.loc[df["CODIGO"] == codigo, ["CANTIDAD", "PRECIO UNITARIO", "TOTAL"]] =
44             row[["CANTIDAD", "PRECIO UNITARIO", "TOTAL"]].values
45
46     # Guardar los cambios en el archivo CSV original
47     df.to_csv(archivo_csv, index=False, encoding="utf-8-sig")
48
49     # Llamar a la función para procesar el DataFrame cargado
50     df_procesado = procesar_data_sumRP(df, archivo_csv)
51
52     # Opcional: Imprimir el resultado para inspección

```



```

49     print(df_procesado)
50
51     print(f" DataFrame actualizado (sin perder columnas originales) y cambios guardados en
{archivo_csv}")
52
53     # Actualizar la tabla y el total en la pestaña actual
54     actualizar_tabla_y_total(tabla)
55
56 def actualizar_tabla_y_total(tabla):
57     """
58     Actualiza la tabla y el total en la pestaña actual sin recrearla.
59     """
60     global df
61
62     # Limpiar la tabla actual
63     for item in tabla.get_children():
64         tabla.delete(item)
65
66     # Insertar elementos desde el CSV con jerarquía visual
67     nodos = {}
68     for _, row in df.iterrows():
69         codigo = str(row["CODIGO"]) # Mantener el código sin cambios
70         descripcion = row["DESCRIPCION"]
71         unidad = row["UNIDAD"] if pd.notna(row["UNIDAD"]) else ""
72         cantidad = row["CANTIDAD"] if pd.notna(row["CANTIDAD"]) else ""
73         precio_unitario = row["PRECIO UNITARIO"] if pd.notna(row["PRECIO UNITARIO"]) else ""
74
75         total = row["TOTAL"] if pd.notna(row["TOTAL"]) else ""
76
77         # Determinar el nivel de jerarquía
78         partes = codigo.split(".")
79         nivel = len([p for p in partes if p != "0"])
80
81         # Aplicar formato visual a la jerarquía
82         espaciado = "|_" * (nivel - 1) # Representación visual de niveles
83         codigo_mostrar = f"{espaciado}{codigo}"
84
85         # Determinar el nodo padre
86         if nivel == 1:
87             nodo_padre = ""
88             tag = "Nivel1"
89         elif nivel == 2:
90             nodo_padre = nodos.get(".".join(partes[:1]) + ".0", "")
91             tag = "Nivel2"
92         elif nivel == 3:
93             nodo_padre = nodos.get(".".join(partes[:2]) + ".0", "")
94             tag = "Nivel3"
95         else:
96             nodo_padre = ""
97             tag = ""
98
99         # Insertar en la tabla y asignar etiquetas
100        nodos[codigo] = tabla.insert(
            nodo_padre,

```

```

101         "end",
102         text="",
103         values=(codigo_mostrar, descripcion, unidad, cantidad, precio_unitario,
total),
104         tags=(tag,),
105         open=True # open=(nivel == 1) , para que se contraiga la lista
106     )
107     # Definir el ancho de las columnas
108     tabla.column("#0", width=30, minwidth=80, stretch=False)
109     tabla.column("CODIGO", width=80, minwidth=70, stretch=False)
110     tabla.column("DESCRIPCION", width=800, minwidth=80, stretch=False)
111     tabla.column("UNIDAD", width=80, minwidth=80, stretch=False)
112     tabla.column("CANTIDAD", width=100, minwidth=80, stretch=False)
113     tabla.column("PRECIO UNITARIO", width=130, minwidth=100, stretch=False)
114     tabla.column("TOTAL", width=130, minwidth=80, stretch=False)
115
116
117     # Actualizar el total en la pestaña actual
118     total_nivel_1 = df.loc[df["NIVEL"] == 1, "TOTAL NIVEL 1"].values[0]
119     total_formateado = f"S/ {total_nivel_1:,.2f}"
120     total_label.config(text=f"Total: {total_formateado}")
121
122 #####
123 def recargar_csv_en_tabla(tabla):
124     """
125     Recarga el archivo CSV original y actualiza la tabla en pantalla.
126     """
127     global df
128     df = pd.read_csv("BBDD_sumRP.csv", encoding="utf-8-sig")
129     actualizar_tabla_y_total(tabla)
130     print("CSV recargado y tabla actualizada.")
131
132 #####
133 def crear_pestana_sum_rp(notebook):
134     # Crear una nueva pestaña dentro del Notebook
135     tab1 = ttk.Frame(notebook)
136     notebook.add(tab1, text="Sum RP") # Agregar la pestaña con el nombre "Sum RP"
137
138     # Extraer el primer valor de "TOTAL NIVEL 1" donde el nivel es 1 y formatearlo
139     total_nivel_1 = df.loc[df["NIVEL"] == 1, "TOTAL NIVEL 1"].values[0]
140     # Formatear con coma de miles y punto decimal
141     total_formateado = f"S/ {total_nivel_1:,.2f}"
142
143     # Contenedor para título y total
144     frame_superior = ttk.Frame(tab1)
145     frame_superior.pack(fill="x", padx=10, pady=5)
146
147     # Crear y mostrar el título
148     titulo = ttk.Label(frame_superior, text="SUMINISTRO DE MATERIALES - REDES PRIMARIAS",
149                        font=("Century Gothic", 14, "italic"), foreground="red")
150     titulo.pack(side="left", padx=10)
151
152     # Mostrar el total como un mensaje de texto
153     global total_label

```

```

154     total_label = ttk.Label(frame_superior, text=f"Total: {total_formateado}",
155                             font=("Century Gothic", 13, "italic"), foreground="red")
156     total_label.pack(side="right", padx=10)
157
158     # Definir las columnas de la tabla
159     columnas = ("CODIGO", "DESCRIPCION", "UNIDAD", "CANTIDAD", "PRECIO UNITARIO", "TOTAL")
160
161     # Crear la tabla con las columnas definidas
162     tabla = ttk.Treeview(tab1, columns=columnas, show="tree headings", style="Treeview")
163
164     # Definir el ancho de las columnas
165     # Definir el ancho de las columnas
166     tabla.column("#0", width=30, minwidth=80, stretch=False)
167     tabla.column("CODIGO", width=80, minwidth=70, stretch=False)
168     tabla.column("DESCRIPCION", width=800, minwidth=80, stretch=False)
169     tabla.column("UNIDAD", width=80, minwidth=80, stretch=False)
170     tabla.column("CANTIDAD", width=100, minwidth=80, stretch=False)
171     tabla.column("PRECIO UNITARIO", width=130, minwidth=100, stretch=False)
172     tabla.column("TOTAL", width=130, minwidth=80, stretch=False)
173
174     # Configurar los encabezados de las columnas y su alineación
175     for col in columnas:
176         tabla.heading(col, text=col, anchor="w") # Encabezados centrados
177         tabla.column(col, anchor="w", stretch=True) # Contenido alineado a la izquierda
178
179     # Insertar elementos desde el CSV con jerarquía visual
180     nodos = {}
181     for _, row in df.iterrows():
182         codigo = str(row["CODIGO"]) # Mantener el código sin cambios
183         descripcion = row["DESCRIPCION"]
184         unidad = row["UNIDAD"] if pd.notna(row["UNIDAD"]) else ""
185         cantidad = row["CANTIDAD"] if pd.notna(row["CANTIDAD"]) else ""
186         precio_unitario = row["PRECIO UNITARIO"] if pd.notna(row["PRECIO UNITARIO"]) else ""
187         total = row["TOTAL"] if pd.notna(row["TOTAL"]) else ""
188
189     # Determinar el nivel de jerarquía
190     partes = codigo.split(".")
191     nivel = len([p for p in partes if p != "0"])
192
193     # Aplicar formato visual a la jerarquía
194     espaciado = "|__" * (nivel - 1) # Representación visual de niveles
195     codigo_mostrar = f"{espaciado}{codigo}"
196
197     # Determinar el nodo padre
198     if nivel == 1:
199         nodo_padre = ""
200         tag = "Nivel1"
201     elif nivel == 2:
202         nodo_padre = nodos.get("." + partes[:1]) + ".0", ""
203         tag = "Nivel2"
204     elif nivel == 3:
205         nodo_padre = nodos.get("." + partes[:2]) + ".0", ""
206         tag = "Nivel3"

```



```

207         else:
208             nodo_padre = ""
209             tag = ""
210
211         # Insertar en la tabla y asignar etiquetas
212         nodos[codigo] = tabla.insert(
213             nodo_padre,
214             "end",
215             text="",
216             values=(codigo_mostrar, descripcion, unidad, cantidad, precio_unitario,
total),
217             tags=(tag,),
218             open=True # open=(nivel == 1) , para que se contraiga la lista
219         )
220
221         # Aplicar los estilos a los niveles
222         tabla.tag_configure("Nivel1", background="#00796B", foreground="white", font=("Century
Gothic", 10)) # Verde turquesa con texto blanco
223         tabla.tag_configure("Nivel2", background="#E0F2F1", font=("Century Gothic", 10)) #
Nivel 2 amarillo
224         tabla.tag_configure("Nivel3", background="white", foreground="#0000FF") # Nivel 3
fondo blanco, texto azul
225
226         # Permitir la edición de celdas mediante doble clic
227         tabla.bind("<Double-1>", lambda event: editar_celda(event, tabla, tab1))
228
229         # Agregar la tabla a la pestaña y expandirla para ocupar todo el espacio disponible
230         tabla.pack(expand=True, fill="both", padx=10, pady=10)
231
232         # Crear un Frame para el botón y colocarlo debajo de la tabla
233         frame_boton = ttk.Frame(tab1)
234         frame_boton.pack(fill="x", padx=10, pady=10)
235
236         boton_actualizar_guardar = ttk.Button(
237             frame_boton,
238             text="Actualizar y Guardar",
239             cursor="hand2",
240             command=lambda: actualizar_dataframe_y_guardar(tabla, "BBDD_sumRP.csv")
241         )
242         boton_actualizar_guardar.pack(side="left", padx=5, pady=5)
243
244         boton_recargar_csv = ttk.Button(
245             frame_boton,
246             text="Actualizar CSV",
247             cursor="hand2",
248             command=lambda: recargar_csv_en_tabla(tabla),
249             bootstyle="info"
250         )
251         boton_recargar_csv.pack(side="left", padx=5, pady=5)
252
253         return tab1 # Retornar la pestaña creada

```

B2_mon_rp.py

```

1 # modulo_smon_rp.py
2 import ttkbootstrap as ttk
3 import pandas as pd
4 from ttkbootstrap.constants import *
5 from C_funciones import ajustar_columnas, editar_celda
6 from E2_BBDD_RP_MON import procesar_data_monRP
7 from I1_actualizar_partida import corregir_cantidades_y_subtotales_por_partida
8 from I2_Act_Unitario_monRP import actualizar_todos_precios
9
10 # Cargar el archivo CSV en un DataFrame
11 df = pd.read_csv("BBDD_monRP.csv", encoding="utf-8-sig")
12
13 # Actualizar el DataFrame original (df) desde la tabla
14 def actualizar_dataframe_y_guardar(tabla, archivo_csv="BBDD_monRP.csv"):
15     global df # Asegurar que estamos trabajando con el DataFrame global original
16
17     # --- 1) Crear un diccionario para actualizar las filas del DataFrame original ---
18     nuevos_datos = []
19
20     # Función recursiva para recorrer todos los nodos de la tabla
21     def recorrer_nodos(item):
22         valores = list(tabla.item(item, "values")) # Convertir valores de la tabla a
23         lista
24         if valores[0].startswith("|_"): # Limpiar el prefijo '|_' si existe
25             valores[0] = valores[0].replace("|_", "")
26         nuevos_datos.append(valores) # Agregar los valores al nuevo dataset
27         for hijo in tabla.get_children(item): # Recorrer hijos de cada nodo
28             recorrer_nodos(hijo)
29
30     for item in tabla.get_children():
31         recorrer_nodos(item)
32
33     columnas_tabla = ["CODIGO", "DESCRIPCION", "UNIDAD", "CANTIDAD", "PRECIO UNITARIO",
34     "TOTAL"]
35     df_tabla = pd.DataFrame(nuevos_datos, columns=columnas_tabla)
36
37     # --- 2) Actualizar las columnas editadas del DataFrame original ---
38     for index, row in df_tabla.iterrows():
39         codigo = row["CODIGO"]
40         if codigo in df["CODIGO"].values:
41             df.loc[df["CODIGO"] == codigo, ["CANTIDAD", "PRECIO UNITARIO", "TOTAL"]] =
42             row[["CANTIDAD", "PRECIO UNITARIO", "TOTAL"]].values
43
44     # --- 3) Guardar los cambios en el archivo CSV antes de recalculer ---
45     df.to_csv(archivo_csv, index=False, encoding="utf-8-sig")
46
47     # --- 4) Corregir cálculos y actualizar precios ---
48     csv_partidas = 'BBDD_partidasRP.csv'
49     csv_costos = 'BBDD_Relacion_de_Costos.csv'
50     corregir_cantidades_y_subtotales_por_partida(csv_partidas, csv_costos)
51     actualizar_todos_precios()

```

```

50 # --- 5) Volver a cargar el DataFrame actualizado ---
51 df = pd.read_csv(archivo_csv, encoding="utf-8-sig")
52
53 # --- 6) Procesar el DataFrame final (opcional) ---
54 df_procesado = procesar_data_monRP(df, archivo_csv)
55
56 # --- 7) Imprimir resultados e informar ---
57 print(df_procesado)
58 print(f" DataFrame actualizado (sin perder columnas originales) y cambios guardados en
{archivo_csv}")
59
60 # --- 8) Actualizar la tabla y el total en la pestaña actual ---
61 actualizar_tabla_y_total(tabla)
62
63
64 def actualizar_tabla_y_total(tabla):
65     """
66     Actualiza la tabla y el total en la pestaña actual sin recrearla.
67     """
68     global df
69
70     # Limpiar la tabla actual
71     for item in tabla.get_children():
72         tabla.delete(item)
73
74     # Insertar elementos desde el CSV con jerarquía visual
75     nodos = {}
76     for _, row in df.iterrows():
77         codigo = str(row["CODIGO"]) # Mantener el código sin cambios
78         descripcion = row["DESCRIPCION"]
79         unidad = row["UNIDAD"] if pd.notna(row["UNIDAD"]) else ""
80         cantidad = row["CANTIDAD"] if pd.notna(row["CANTIDAD"]) else ""
81         precio_unitario = row["PRECIO UNITARIO"] if pd.notna(row["PRECIO UNITARIO"]) else ""
82
83         total = row["TOTAL"] if pd.notna(row["TOTAL"]) else ""
84
85         # Determinar el nivel de jerarquía
86         partes = codigo.split(".")
87         nivel = len([p for p in partes if p != "0"])
88
89         # Aplicar formato visual a la jerarquía
90         espaciado = "|__" * (nivel - 1) # Representación visual de niveles
91         codigo_mostrar = f"{espaciado}{codigo}"
92
93         # Determinar el nodo padre
94         if nivel == 1:
95             nodo_padre = ""
96             tag = "Nivel1"
97         elif nivel == 2:
98             nodo_padre = nodos.get(".".join(partes[:1]) + ".0", "")
99             tag = "Nivel2"
100         elif nivel == 3:
101             nodo_padre = nodos.get(".".join(partes[:2]) + ".0", "")
102             tag = "Nivel3"

```



```

102         else:
103             nodo_padre = ""
104             tag = ""
105
106             # Insertar en la tabla y asignar etiquetas
107             nodos[codigo] = tabla.insert(
108                 nodo_padre,
109                 "end",
110                 text="",
111                 values=(codigo_mostrar, descripcion, unidad, cantidad, precio_unitario,
total),
112                 tags=(tag,),
113                 open=True # open=(nivel == 1) , para que se contraiga la lista
114             )
115
116             # Definir el ancho de las columnas
117             tabla.column("#0", width=30, minwidth=80, stretch=False)
118             tabla.column("CODIGO", width=80, minwidth=70, stretch=False)
119             tabla.column("DESCRIPCION", width=800, minwidth=80, stretch=False)
120             tabla.column("UNIDAD", width=80, minwidth=80, stretch=False)
121             tabla.column("CANTIDAD", width=100, minwidth=80, stretch=False)
122             tabla.column("PRECIO UNITARIO", width=130, minwidth=100, stretch=False)
123             tabla.column("TOTAL", width=130, minwidth=80, stretch=False)
124
125             # Actualizar el total en la pestaña actual
126             total_nivel_1 = df.loc[df["NIVEL"] == 1, "TOTAL NIVEL 1"].values[0]
127             total_formateado = f"$/{total_nivel_1:,.2f}"
128             total_label.config(text=f"Total: {total_formateado}")
129
130             #####
131             def recargar_csv_en_tabla(tabla):
132                 """
133                 Recarga el archivo CSV original y actualiza la tabla en pantalla.
134                 """
135                 global df
136                 df = pd.read_csv("BBDD_monRP.csv", encoding="utf-8-sig")
137                 actualizar_tabla_y_total(tabla)
138                 print("CSV recargado y tabla actualizada.")
139
140             #####
141             def crear_pestana_mon_rp(notebook):
142                 # Crear una nueva pestaña dentro del Notebook
143                 tab2 = ttk.Frame(notebook)
144                 notebook.add(tab2, text="mon RP") # Agregar la pestaña con el nombre "Sum RP"
145
146                 # Extraer el primer valor de "TOTAL NIVEL 1" donde el nivel es 1 y formatearlo
147                 total_nivel_1 = df.loc[df["NIVEL"] == 1, "TOTAL NIVEL 1"].values[0]
148                 # Formatear con coma de miles y punto decimal
149                 total_formateado = f"$/{total_nivel_1:,.2f}"
150
151                 # Contenedor para título y total
152                 frame_superior = ttk.Frame(tab2)
153                 frame_superior.pack(fill="x", padx=10, pady=5)
154

```

```

155     # Crear y mostrar el titulo
156     titulo = ttk.Label(frame_superior, text="MONTAJE DE MATERIALES - REDES PRIMARIAS",
157                        font=("Century Gothic", 14, "italic"), foreground="red")
158     titulo.pack(side="left", padx=10)
159
160     # Mostrar el total como un mensaje de texto
161     global total_label
162     total_label = ttk.Label(frame_superior, text=f"Total: {total_formateado}",
163                           font=("Century Gothic", 13, "italic"), foreground="red")
164     total_label.pack(side="right", padx=10)
165
166     # Definir las columnas de la tabla
167     columnas = ("CODIGO", "DESCRIPCION", "UNIDAD", "CANTIDAD", "PRECIO UNITARIO", "TOTAL")
168
169     # Crear la tabla con las columnas definidas
170     tabla = ttk.Treeview(tab2, columns=columnas, show="tree headings", style="Treeview")
171
172     # Definir el ancho de las columnas
173     tabla.column("#0", width=30, minwidth=80, stretch=False)
174     tabla.column("CODIGO", width=80, minwidth=70, stretch=False)
175     tabla.column("DESCRIPCION", width=800, minwidth=80, stretch=False)
176     tabla.column("UNIDAD", width=80, minwidth=80, stretch=False)
177     tabla.column("CANTIDAD", width=100, minwidth=80, stretch=False)
178     tabla.column("PRECIO UNITARIO", width=130, minwidth=100, stretch=False)
179     tabla.column("TOTAL", width=130, minwidth=80, stretch=False)
180
181     # Configurar los encabezados de las columnas y su alineación
182     for col in columnas:
183         tabla.heading(col, text=col, anchor="w") # Encabezados centrados
184         tabla.column(col, anchor="w", stretch=True) # Contenido alineado a la izquierda
185
186     # Insertar elementos desde el CSV con jerarquía visual
187     nodos = {}
188     for _, row in df.iterrows():
189         codigo = str(row["CODIGO"]) # Mantener el código sin cambios
190         descripcion = row["DESCRIPCION"]
191         unidad = row["UNIDAD"] if pd.notna(row["UNIDAD"]) else ""
192         cantidad = row["CANTIDAD"] if pd.notna(row["CANTIDAD"]) else ""
193         precio_unitario = row["PRECIO UNITARIO"] if pd.notna(row["PRECIO UNITARIO"]) else ""
194
195         total = row["TOTAL"] if pd.notna(row["TOTAL"]) else ""
196
197         # Determinar el nivel de jerarquía
198         partes = codigo.split(".")
199         nivel = len([p for p in partes if p != "0"])
200
201         # Aplicar formato visual a la jerarquía
202         espaciado = "|_" * (nivel - 1) # Representación visual de niveles
203         codigo_mostrar = f"{espaciado}{codigo}"
204
205         # Determinar el nodo padre
206         if nivel == 1:
207             nodo_padre = ""
208             tag = "Nivel1"

```



```

208         elif nivel == 2:
209             nodo_padre = nodos.get(".".join(partes[:1]) + ".0", "")
210             tag = "Nivel2"
211         elif nivel == 3:
212             nodo_padre = nodos.get(".".join(partes[:2]) + ".0", "")
213             tag = "Nivel3"
214         else:
215             nodo_padre = ""
216             tag = ""
217
218         # Insertar en la tabla y asignar etiquetas
219         nodos[codigo] = tabla.insert(
220             nodo_padre,
221             "end",
222             text="",
223             values=(codigo_mostrar, descripcion, unidad, cantidad, precio_unitario,
total),
224             tags=(tag,),
225             open=True # open=(nivel == 1) , para que se contraiga la lista
226         )
227
228
229         # Aplicar los estilos a los niveles
230         tabla.tag_configure("Nivel1", background="#00796B", foreground="white", font=("Century
Gothic", 10)) # Verde turquesa con texto blanco
231         tabla.tag_configure("Nivel2", background="#E0F2F1", font=("Century Gothic", 10)) #
Nivel 2 amarillo
232         tabla.tag_configure("Nivel3", background="white", foreground="#0000FF") # Nivel 3
fondo blanco, texto azul
233
234
235         # Permitir la edición de celdas mediante doble clic
236         tabla.bind("<Double-1>", lambda event: editar_celda(event, tabla, tab2))
237
238         # Agregar la tabla a la pestaña y expandirla para ocupar todo el espacio disponible
239         tabla.pack(expand=True, fill="both", padx=10, pady=10)
240
241         # Crear un Frame para el botón y colocarlo debajo de la tabla
242         frame_boton = ttk.Frame(tab2)
243         frame_boton.pack(fill="x", padx=10, pady=10)
244
245         boton_actualizar_guardar = ttk.Button(
246             frame_boton,
247             text="Actualizar y Guardar",
248             cursor="hand2",
249             command=lambda: actualizar_dataframe_y_guardar(tabla, "BBDD_monRP.csv")
250         )
251         boton_actualizar_guardar.pack(side="left", padx=5, pady=5)
252
253         boton_recargar_csv = ttk.Button(
254             frame_boton,
255             text="Actualizar CSV",
256             cursor="hand2",
257             command=lambda: recargar_csv_en_tabla(tabla),

```

11/12/25, 12:10 p.m.

B2_mon_rp.py

```
258     bootstyle="info"  
259     )  
260     boton_recargar_csv.pack(side="left", padx=5, pady=5)  
261  
262     return tab2 # Retornar la pestaña creada
```

localhost:63616/578ffffb-6aaf-4806-b807-9031131aad6d/

6/6

ANEXO D

COTIZACIONES Y PRECIOS REFERENCIALES

DETALLE DEL PRODUCTO / INSUMO	UND.	PRECIO (S/)
1. POSTES Y ACCESORIOS DE CONCRETO		
Poste de Concreto Armado (C.A.C.) 8/200/140/260	u	441.00
Poste de Concreto Armado (C.A.C.) 8/300/150/270	u	526.00
Caja de Concreto P.T. (0.30x0.40x0.40 m) con tapa	u	50.00
2. AISLADORES		
Aislador de Porcelana, Clase ANSI 54-2 (Para retenidas)	u	11.00
3. CONDUCTORES ELÉCTRICOS		
Conductor Cobre Desnudo 25mm ² , 7 Hilos (Puesta a tierra)	m	9.28
Cable Autoportante Aluminio CAAI 2x25+16/N25 mm ²	m	8.65
Cable Autoportante Aluminio CAAI 2x25/N25 mm ²	m	6.14
4. FERRETERÍA DE MONTAJE		
Grapa de Anclaje Cónica Tipo Cocodrilo	Und	18.00
Hebilla para Fleje de Acero Inoxidable 19 mm	Und	2.15
Fleje de Acero Inoxidable 19mm	Und	6.50
Plancha Gancho de Suspensión (Carga= 15 Kn)	Und	22.00
Grapa de Suspensión Aluminio Plastificada (SO 14.1)	Und	18.00
Perno con Gancho de Suspensión D=16mm (SOT 15.8)	Und	15.00
Arandela Cuadrada Curva 57x57x5mm	Und	4.00
Perno con Ojal D=16 mm, L=228 mm	Und	18.00
Tuerca Ojo, D=16 mm	Und	9.20
Portalínea Unipolar p/ Aislador Carrete ANSI	Und	10.00
Perno Maquinado L= 229 mm, D= 13 mm	Und	6.08
5. ACCESORIOS PARA RETENIDAS		
Cable de Acero Galvanizado D=10.0mm	m	4.00
Alambre Galvanizado N°14 AWG	Und	0.80
Varilla de Anclaje con Ojal L=2400mm D=16mm	Und	56.00
Canaleta Guardacable L=2100mm	m	35.00
Abrazadera para Soporte de Retenida D=130mm	Und	38.00
Grapa Paralela de A °G ° de Tres Pernos D=10mm	Und	12.00
Plancha Cuadrada Fierro Galv. 300x300x6.4 mm	Und	40.00
Guardacabo de Acero Galvanizado D=10mm	Und	2.00

DETALLE DEL PRODUCTO / INSUMO	UND.	PRECIO (\$)
Templador Fe Galv. para Retenida 255x16mm	Und	35.00
Contrapunta L=1.00m D=50mm con abrazadera	Und	65.00
6. SISTEMA DE PUESTA A TIERRA		
Plancha Antirrobo Bronce P.A.T. 200x200mm	Und	45.00
Varilla de cobre 19mm x 2.40m con conector	Und	213.54
Conector de Cobre Tipo AB P/Varilla 19 mm	Und	8.00
Cemento Conductivo x 25Kg	Bls	81.30
Tierra Negra Cernida - Arcilla	m3	35.00
7. ALUMBRADO PÚBLICO		
Luminaria LED 50 W, 120-277 V, IP66	u	491.90
Pastoral de Acero Galv. AH=1000mm AV=850mm	u	35.00
Abrazadera Partida Simple Para Pastoral	u	12.50
Conector de derivación dos vías Al-Cu	u	8.00

* Precios referenciales sujetos a variación del mercado local.

ANEXO D

MANUAL DE USO DEL SOFTWARE PARA GESTIÓN DE COSTOS Y PRESUPUESTOS EN SISTEMAS DE DISTRIBUCIÓN

INTRODUCCIÓN

Este software desarrollado en Python permite gestionar costos y presupuestos para proyectos de electrificación, específicamente aplicado al proyecto Challabamba Alta. La aplicación cuenta con una interfaz gráfica moderna dividida en secciones organizadas que facilitan la administración de suministros, montajes, partidas y análisis de costos.

Requisitos del Sistema

Python 3.8+

Bibliotecas: ttkbootstrap, pandas, tkinter, matplotlib, openpyxl, csv

AutoCAD (opcional, para funcionalidades de vinculación)

Sistema operativo: Windows 10/11 recomendado

INTERFAZ PRINCIPAL

Estructura de la Ventana Principal

La ventana principal está dividida en tres áreas principales:

The screenshot displays the 'Gestión de Costos y Presupuestos en Distribución' application. The interface is divided into three main sections:

- BARRA SUPERIOR DE BOTONES:** A top navigation bar containing icons for saving, exporting PDF, adding items, adding equipment, modifying items, viewing costs, linking to AutoCAD, information, and linking to AutoCAD.
- NOTEBOOK 1:** A large table displaying cost data. The table has columns for 'DESCRIPCION', 'REDES PRIMARIAS', 'REDES SECUNDARIAS', and 'TOTAL'. It lists various electrical components and their associated costs.
- NOTEBOOK 2:** A detailed view of a specific item, showing its 'Codigo', 'Descripcion', 'Cant', 'Unid', 'Cant', 'F.UNIT', 'SUBTO', and 'TOTAL'. It also includes a 'Rendimiento' field and a 'Unidad' dropdown.
- NOTEBOOK 3:** A section titled 'Análisis de Costos del Proyecto' featuring a pie chart and a table showing the distribution of costs across different categories: MATERIALES (11.8%), MANO DE OBRA (40.1%), EQUIPOS (58.4%), and TRABAJOS COMPLEMENTARIOS (2.4%).

Barra Superior de Botones

The 'Barra Superior de Botones' contains the following icons and labels:

- Icono guardar
- Exportar pdf
- Ingresar Insumo
- Agregar Armado
- Modificar Partida
- Ver costos
- Vincular Diredcad
- Info
- Vincular Autocad

Contiene íconos con funciones específicas:

Botón	Función	Descripción
Guardar	guardar_excel()	Exporta los datos del proyecto a un archivo Excel.
Exportar PDF	guardar_pdf()	Genera un reporte del proyecto en formato PDF.
Ingresar Insumo	ingresar_insumo()	Permite añadir nuevos insumos a la base de datos del sistema.
Agregar Armado	open_add_armado_window()	Incorpora armados predefinidos al análisis de partidas.
Modificar Partida	run_main_gui()	Permite editar partidas existentes del proyecto.
Vincular DiredCAD	Actualizar_metrado_diredcad()	Importa metrados desde el software DiredCAD.
Información	info()	Muestra información general del sistema.
Vincular AutoCAD	Actualizar_metrado_autocad()	Importa metrados desde archivos de AutoCAD.
Ejecutar	mostrar_carga()	Actualiza y recalcula los costos generales del proyecto.

SECCIONES PRINCIPALES

Notebook 1 (Panel Izquierdo)

Resumen General	Gastos Generales	Sum RP	mon RP	Sum RS	mon RS	Partidas RP	Partidas RS
Recargar Datos	Exportar a PDF						
Tabla de Resumen							
DESCRIPCION	REDES PRIMARIAS	REDES SECUNDARIAS	TOTAL				
SUMINISTRO DE MATERIALES	-	-	-	-	-	-	-
AISLADORES	\$/ 830.00						
ALUMBRADO PUBLICO	-						
CONDUCTORES	\$/ 950.08						
FERRETERIA	\$/ 2,318.60						
POSTES Y ACCESORIOS DE CONCRETO ARMADO	\$/ 2,150.00						
PUESTA A TIERRA	\$/ 1,957.80						
RETENIDAS	\$/ 717.60						
SECCIONADORES, PARARRAYOS, TERMINACIONES Y ACCESORIOS	\$/ 3,014.10	-					
TRANSFORMADORES	\$/ 6,413.16	-					
TOTAL SUMINISTRO DE MATERIALES	\$/ 18,351.34	\$/ 40,492.85					
MONTAJE ELECTROMECHANICO	-	-					
AISLADORES	\$/ 430.22	\$/ 813.15					
CONDUCTORES Y CABLES	\$/ 353.31	\$/ 1,312.18					
FERRETERIA	\$/ 1,199.60	\$/ 7,123.69					
MONTAJE ELECTROMECHANICO ALUMBRADO PÚBLICO	-	\$/ 4,051.89					
POSTES Y ACCESORIOS DE CONCRETO ARMADO	\$/ 1,765.10	\$/ 12,892.36					
PRUEBAS Y PUESTA EN SERVICIO	\$/ 1,765.10	\$/ 1,550.60					
PUESTA A TIERRA	\$/ 1,195.29	\$/ 1,210.72					
RETENIDAS	\$/ 561.19	\$/ 11,407.33					
SECCIONADORES, PARARRAYOS Y TERMINACIONES	\$/ 763.36	-					
TRABAJOS PRELIMINARES	\$/ 2,175.05	\$/ 722.19					
TRANSFORMADORES	\$/ 494.73	-					
TOTAL MONTAJE ELECTROMECHANICO	\$/ 10,702.95	\$/ 41,084.11					
A. Suministro de Materiales	\$/ 18,351.34	\$/ 40,492.85					
B. Montaje Electromecánico	\$/ 10,702.95	\$/ 41,084.11					
C. Transporte	\$/ 1,651.62	\$/ 3,644.36					
TOTAL COSTO DIRECTO	\$/ 30,705.91	\$/ 85,221.32					
D. Gastos Generales y Utilidades	\$/ 9,211.77	\$/ 25,566.39					

Contiene 8 pestañas con diferentes funcionalidades:

Módulo	Descripción / Funcionalidades principales
Resumen	Presenta el resumen general de costos del proyecto, incluyendo subtotales por categorías y el total general con IGV.
Gastos Generales	Permite la configuración de porcentajes de gastos generales, definición de parámetros de costos indirectos y el cálculo automático de utilidades.
Suministro RP (Red Primaria)	Gestiona los materiales correspondientes a la red primaria, incorporando códigos jerárquicos (Nivel 1, 2 y 3) y el cálculo automático de cantidades y costos totales.
Montaje RP (Red Primaria)	Registra los costos de instalación y montaje de la red primaria, considerando mano de obra, equipos y cálculos basados en rendimientos.
Suministro RS (Red Secundaria)	Administra los materiales específicos de la red secundaria, manteniendo una estructura similar al módulo de suministro de red primaria.
Montaje RS (Red Secundaria)	Registra los costos de instalación de la red secundaria, incluyendo luminarias y acometidas.
Partidas Montaje RP	Presenta el detalle de las partidas correspondientes al montaje de la red primaria, incorporando el análisis de precios unitarios.
Partidas Montaje RS	Presenta el detalle de las partidas correspondientes al montaje de la red secundaria, con cálculo de costos unitarios.

Notebook 2 (Panel Derecho Superior) - "Descripción Generales"

Descripción generales

MonRP
1.1.1
Modificar

Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)

CODIGO	DESCRIPCION	CUAI	UND	CAN1	P.UNIT	SUBTO'	TOTAL
1.1.1	MATERIALES						100.95
1.1.1	Piedra Mediana de Canter		M3	0.37	50.0	18.5	
1.1.1	Cemento Portland Tipo I		Blsa	2.05	23.0	47.15	
1.1.1	Hormigón		M3	0.49	70.0	34.3	
1.1.1	Agua Transportada		M3	0.2	5.0	1.0	
1.1.1	MANO DE OBRA						82.2
1.1.1	Operario	0.1	HH	0.1	15.26	1.53	
1.1.1	Oficial	1	HH	1.0	12.57	12.57	

RENDIMIENTO: 8
UNIDAD: Pza/Día

Funcionalidades y características del módulo de partidas

Aspecto	Descripción
Selección de archivo	Permite seleccionar, mediante un combobox, el archivo correspondiente a MonRP o MonRS.
Selección de código	Muestra una lista de códigos disponibles según el archivo seleccionado.
Visualización de descripción	Presenta la descripción general de la partida asociada al código seleccionado.
Tabla de datos	Muestra una tabla detallada que incluye: Código, Descripción, Cantidad, Unidad (UND), Precio Unitario, Subtotal y Total.
Sección de rendimiento	Incluye una sección editable para la modificación de rendimientos asociados a la partida.
Botón "Modificar"	Permite editar las descripciones generales o los valores de rendimiento de la partida seleccionada.
Resaltado automático	Resalta de forma automática categorías relevantes como Materiales, Mano de Obra, entre otras.
Modificación en tiempo real	Permite realizar cambios dinámicos en descripciones y rendimientos durante la operación del sistema.
Actualización automática	Recalcula automáticamente los subtotales y totales al modificar los rendimientos.

Notebook 3 (Panel Derecho Inferior) - "Análisis de Costos"



Funcionalidad	Descripción
Gráfico circular (Pie Chart)	Representa la distribución porcentual de los costos del proyecto por categoría.
Visualización por categoría	Muestra cada categoría con colores diferenciados para facilitar la identificación.
Valores y porcentajes	Presenta los porcentajes y valores monetarios expresados en soles (S/).
Categorías analizadas	Incluye las categorías: Materiales, Mano de Obra, Equipos, Equipos de Seguridad y Herramientas, y Trabajos Complementarios.
Título del gráfico	Incorpora el título "Análisis de Costos del Proyecto".
Botón de actualización	Permite refrescar los datos del gráfico de forma manual.
Leyenda interactiva	Muestra información detallada de cada categoría, incluyendo valores y porcentajes.

FUNCIONALIDADES ESPECÍFICAS

Ingreso de Insumos

Sistema de Gestión de Niveles

Formulario de Datos

Nivel: 1

Código: 1 . .

Descripción: Postes 12/200

Unidad:

Cantidad:

Precio Unitario:

Mostrar Limpiar Guardar y Ajustar

Registros

NIVEL	CODIGO	DESCRIPCION
1	2.0.0	Postes 12/300

Ventana emergente para añadir nuevos ítems:

Campos del Formulario:

1. **Nivel:** Selección entre 1, 2 o 3 (jerarquía)
2. **Código:** Generado automáticamente con spinboxes
3. **Descripción:** Campo de texto amplio (obligatorio)
4. **Unidad:** Solo para nivel 3
5. **Cantidad:** Solo para nivel 3
6. **Precio Unitario:** Solo para nivel 3

Botones:

- **Mostrar:** Añade el ítem a la tabla temporal
- **Limpiar:** Reinicia los campos manteniendo estructura
- **Guardar y Ajustar:** Guarda en CSV y ajusta numeración automáticamente

La función `ingresar_insumo()` abre una ventana emergente especializada que permite agregar nuevos ítems o insumos al sistema de costos del proyecto. Esta herramienta es fundamental para expandir y personalizar la base de datos de materiales, mano de obra y equipos según las necesidades específicas del proyecto Challabamba Alta.

La ventana opera como un formulario estructurado que garantiza que todos los nuevos elementos sigan la jerarquía de codificación establecida en el sistema. Esto es crucial para mantener la coherencia en los cálculos posteriores, especialmente en la generación automática de presupuestos y la distribución de costos.

Agregar Armados (open_add_armado_window())

Selector de Armado

Selección de Armado

Código: 4 2

Selección de Armado

ARMADO AB1

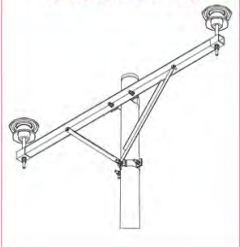
Actualizar Inserir Armado

Componentes del Armado

CODIGO	DESCRIPCION	MULTIPLICADOR	UNIDAD	CANTIDAD	PRECIO UNITARIO	TOTAL
4.2.1	Abrazadera Tipo CAS Simple de 64mm, E=6.4mm, D=160mm, C/3P/3T/3CT/6A/3AI	1.0	Und	16	48	768
4.2.1	GRAPA DE SUSPENSIÓN DE ALUMINIO CUBIERTA DE PINTURA PLASTIFICADA (ISO	1.0	Und	101	5/ 18.00	5/ 51.00
4.2.2	Abrazadera de Fijación Tipo U de 64mm, E=6.4mm, D=150mm, C/2P/2T/2CT/4A/2	1.0	Und	16	45	720
4.2.2	PERNO CON GANCHO DE SUSPENSIÓN D=16 mm, L=228 mm, C/T/CT/2A/AP (SOT	1.0	Und	101	5/ 15.00	5/ 45.00
4.2.3	Cruceta de Perfil Angular de P"6" de 64x64x2,500mm, E=6.4mm, con 3 dados de	1.0	Und	16		
4.2.3	ARANDELA CUADRADA CURVA DE 57x57x5MM, D=17.5MM	2.0	Und	202		
4.2.4	Perno Maquinado L=50mm, D=13mm, C/T/CT/2A/AP	2.0	Und	32		
4.2.5	Arandela Cuadrada Curva de A"6" de 57x57x5 mm, agujero de 17.5mmØ	2.0	Und	32		
4.2.6	Riostra de Perfil Angular de P"6" de 38x38x1,000mm, E=5mm	2.0	Und	32		

Imagen del Armado

ARMADO MONOFASICO (BIFILAR) DE ALINEAMIENTO AB1



Componentes:

1. Selector de Armado: Combobox con armados disponibles

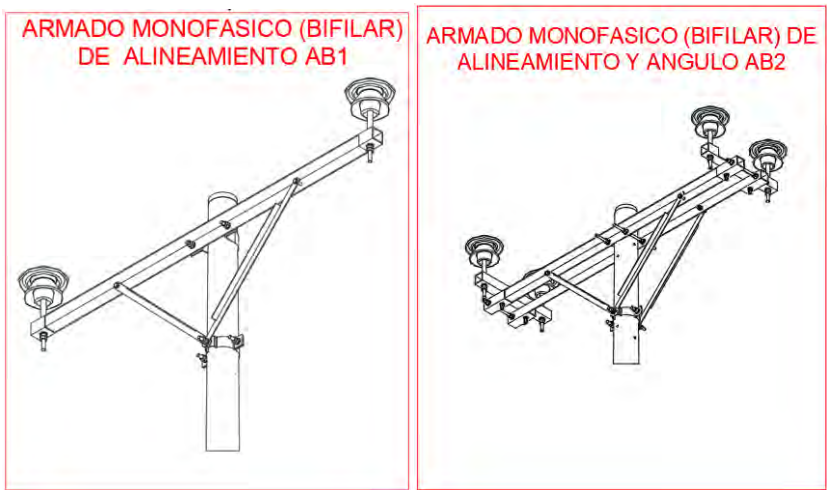
2. Código Personalizable: Spinboxes para definir código base

3. Tabla de Componentes: Muestra todos los elementos del armado

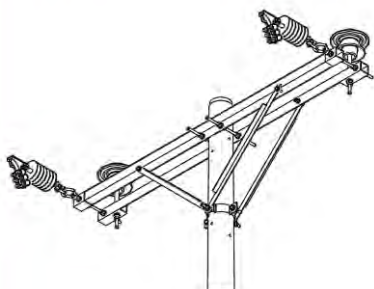
4. Visualización de Imagen: Muestra imagen del armado seleccionado

La función open_add_armado_window() despliega una ventana especializada diseñada para seleccionar y personalizar conjuntos preconfigurados de componentes eléctricos, conocidos en el ámbito de distribución eléctrica como "armados". Un armado representa un ensamblaje completo de elementos necesarios para una configuración específica en la red eléctrica, como un cruce de líneas, una derivación, un apoyo especial, o cualquier combinación estandarizada de postes, aisladores, conductores, herrajes y accesorios.

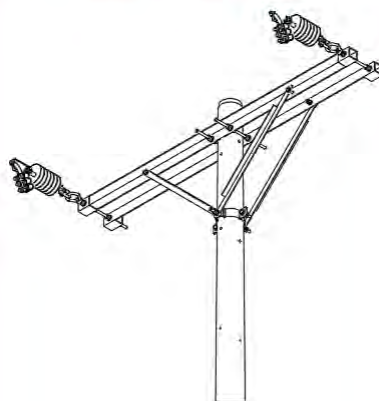
Esta funcionalidad permite visualizar diferentes tipos de armados:



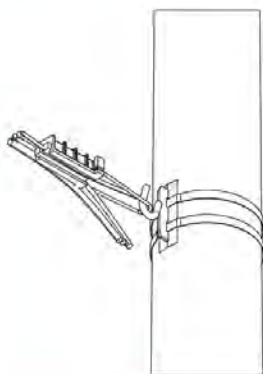
**ARMADO MONOFASICO (BIFILAR) DE
ALINIAMIENTO Y ANCLAJE AB4**



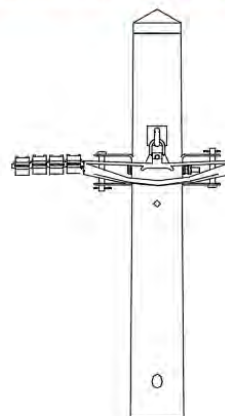
**ARMADO MONOFASICO (BIFILAR) DE
FIN DE LINEA AB5**



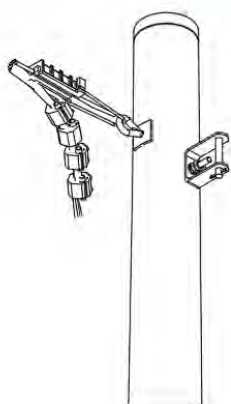
**ARMADO DE INICIO EN SUBESTACIÓN PARA
CONDUCTOR AUTOPORTANTE E0**



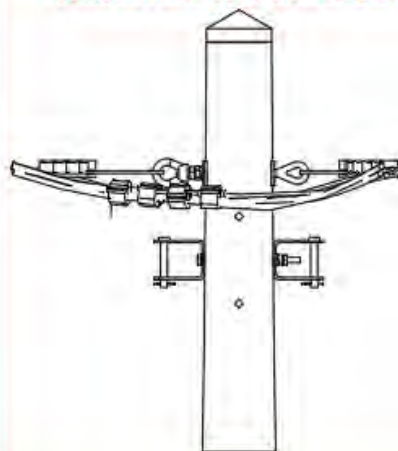
**ARMADO DE ALINEAMIENTO PARA
CONDUCTOR AUTOPORTANTE E1**



**ARMADO DE FIN DE LINEA PARA
CONDUCTOR AUTOPORTANTE E3**



**ARMADO DE CAMBIO DE DIRECCION PARA
CONDUCTOR AUTOPORTANTE E4**



Esta herramienta es crucial para optimizar el proceso de presupuestación en proyectos de electrificación como Challabamba Alta, donde ciertas configuraciones se repiten múltiples veces a lo

largo de la red. En lugar de ingresar manualmente docenas de componentes individuales para cada ocurrencia de un armado específico, el usuario puede seleccionar un armado predefinido y el sistema automáticamente calcula y despliega todos sus componentes con sus cantidades, multiplicadores y costos asociados.

Modificar Partidas

Gestión de Datos de Partidas

Descripción generales

MonRP 1.1.1 Modificar

Izaje Poste de C.A.C. de 12/200 KG. (Hoyo de 1.80x0.80x0.80m) (en terreno normal)

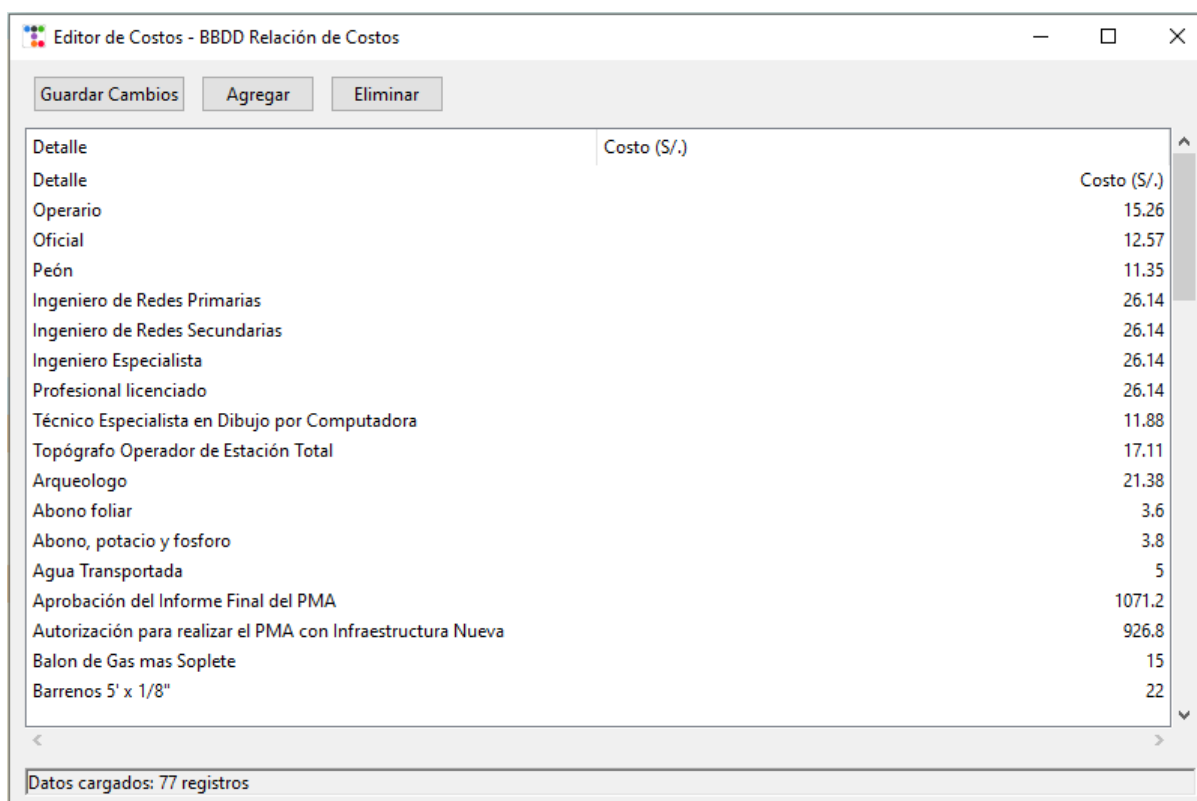
CODIGO	DESCRIPCION	CUANTIA	UND	CANTIDAD	P.UNIT	SUBTOTAL	TOTAL
1.1.1	MATERIALES						100.95
1.1.1	Piedra Mediana de Cantera		M3	0.37	50.0	18.5	
1.1.1	Cemento Portland Tipo I		Blsa	2.05	23.0	47.15	
1.1.1	Hormigón		M3	0.49	70.0	34.3	
1.1.1	Agua Transportada		M3	0.2	5.0	1.0	
1.1.1	MANO DE OBRA						82.2
1.1.1	Operario	0.1	HH	0.1	15.26	1.53	
1.1.1	Oficial	1	HH	1.0	12.57	12.57	
1.1.1	Peón	6	HH	6.0	11.35	68.1	
1.1.1	EQUIPOS						59.6
1.1.1	Camion Grua 6 Tn.	0.5	HM	0.5	104.01	52.01	
1.1.1	Camioneta Rural 4 x 4 130 HP 5 Pasa	0.15	HM	0.15	50.59	7.59	
1.1.1	EQUIPOS DE SEGURIDAD Y HERRAMIE						8.22
1.1.1	Equipos de Seguridad	5	% MO			4.11	

RENDIMIENTO: 8 UNIDAD: Pza/Día

La función `run_main_gui()` activa un editor visual especializado para partidas detalladas que opera como puente entre la flexibilidad de edición de Microsoft Excel y la estructura de datos controlada del sistema de presupuestos. Esta herramienta está diseñada específicamente para el proceso de análisis de precios unitarios (APU), donde cada partida requiere un desglose minucioso de recursos, cantidades, rendimientos y costos.

La necesidad de esta funcionalidad surge de la complejidad inherente a las partidas de construcción eléctrica, donde una sola partida (como "Instalación de poste de concreto 12/200") involucra múltiples recursos agrupados en categorías: materiales (cemento, arena, acero), mano de obra (operarios, oficiales), equipos (mezcladora, grúa), herramientas especializadas, y trabajos complementarios. Cada uno de estos recursos tiene su propia cuantificación, unidad de medida y precio, que deben calcularse con precisión para determinar el costo unitario final de la partida.

Relación de Costos



Detalle	Costo (\$/.)
Detalle	Costo (\$/.)
Operario	15.26
Oficial	12.57
Peón	11.35
Ingeniero de Redes Primarias	26.14
Ingeniero de Redes Secundarias	26.14
Ingeniero Especialista	26.14
Profesional licenciado	26.14
Técnico Especialista en Dibujo por Computadora	11.88
Topógrafo Operador de Estación Total	17.11
Arqueologo	21.38
Abono foliar	3.6
Abono, potasio y fosforo	3.8
Agua Transportada	5
Aprobación del Informe Final del PMA	1071.2
Autorización para realizar el PMA con Infraestructura Nueva	926.8
Balon de Gas mas Soplete	15
Barrenos 5' x 1/8"	22

Datos cargados: 77 registros

La función `abrir_relacion_costos()` despliega un editor especializado para la base de datos central de precios del sistema, representada en el archivo `BBDD_Relacion_de_Costos.csv`. Esta base de datos constituye el corazón del sistema de costos, ya que contiene los precios unitarios de referencia para todos los recursos, materiales, mano de obra y equipos utilizados en el proyecto de electrificación.

La importancia crítica de esta funcionalidad radica en que todos los cálculos de costos del sistema dependen directamente de los valores almacenados en esta relación. Cuando se modifican precios aquí, el cambio se propaga automáticamente a través de:

- Partidas detalladas (APU)

- Suministros de materiales

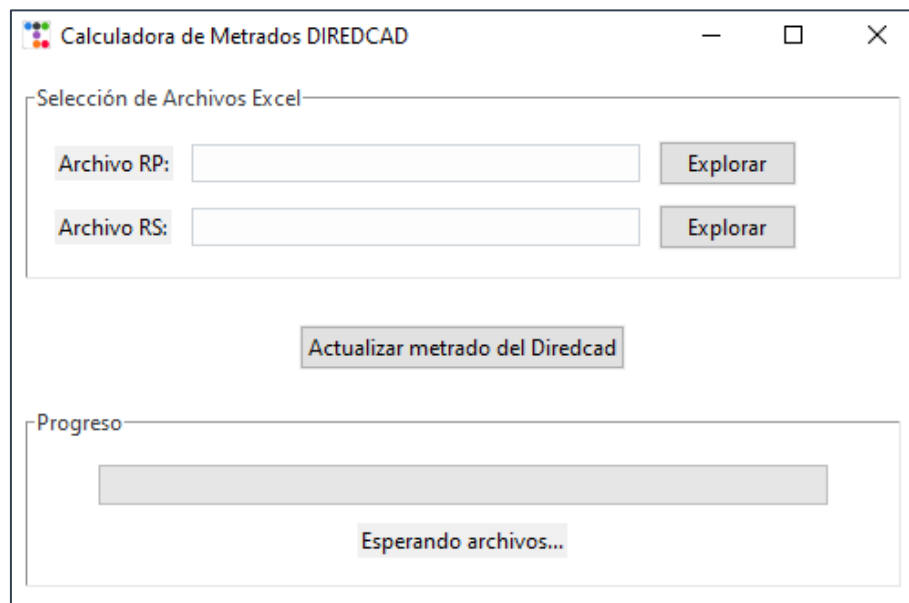
- Costos de montaje

- Análisis de costos generales

- Resúmenes presupuestarios

Este editor proporciona un control granular sobre la estructura de costos sin necesidad de manipular manualmente archivos CSV, reduciendo el riesgo de errores de formato o corrupción de datos.

Vinculación con DiredCAD



La vinculación con AutoCAD permite extraer metrados directamente de los planos CAD del proyecto Challabamba Alta, automatizando el proceso de cuantificación de elementos de red secundaria. A diferencia de DiredCAD que procesa archivos Excel, esta funcionalidad se conecta en tiempo real con una instancia activa de AutoCAD, permitiendo seleccionar objetos gráficos específicos y contar/medir automáticamente los elementos del plano.

Esta herramienta es especialmente valiosa cuando:

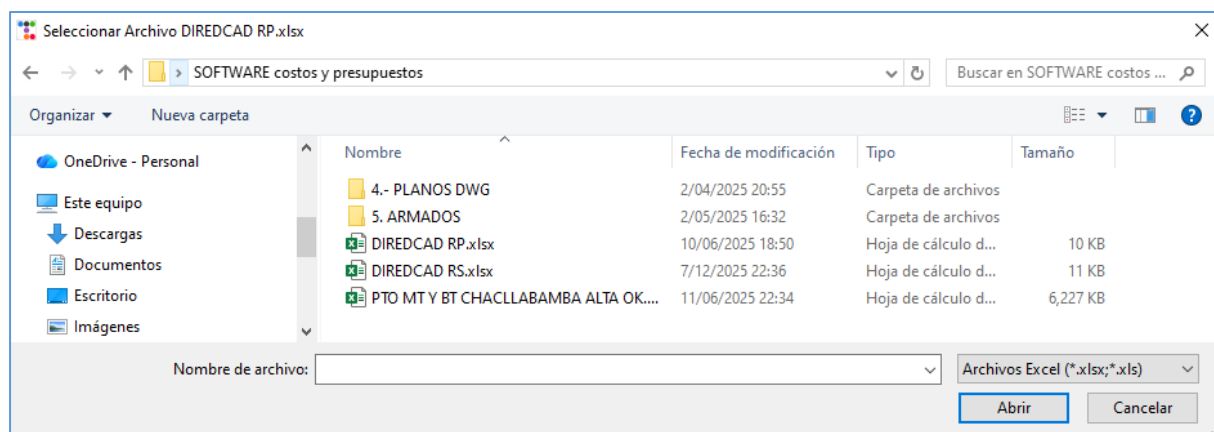
Los planos han sido modificados directamente en AutoCAD

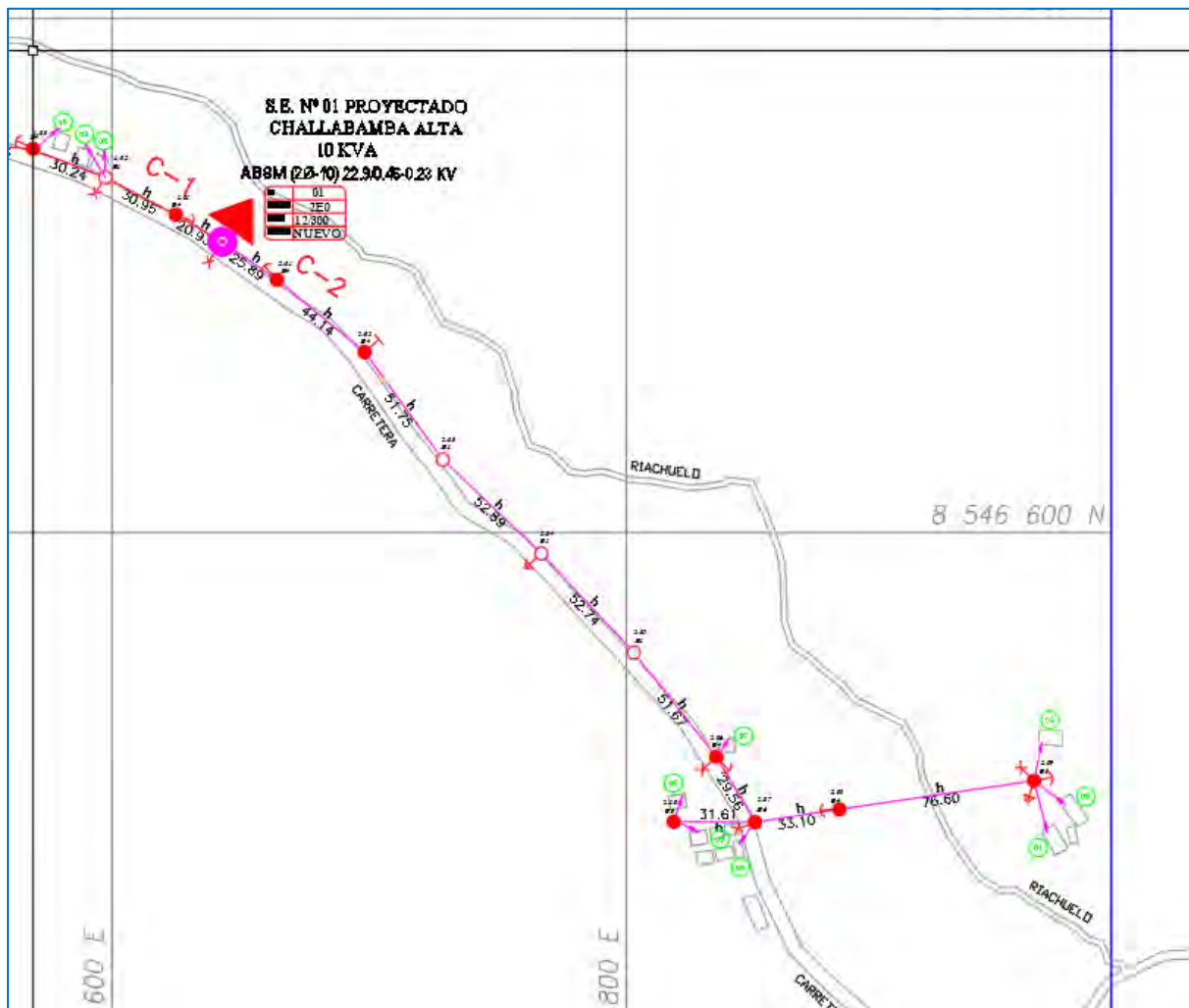
No se dispone de reportes DiredCAD actualizados

Se requiere verificación visual directa de los elementos

Se trabaja con planos existentes no generados en DiredCAD.

Ubicación de archivos





CONSEJOS Y MEJORES PRÁCTICAS

Gestión de Códigos

Mantener la estructura jerárquica (Nivel 1 > Nivel 2 > Nivel 3)

Usar "Ingresar Insumo" para mantener consistencia

No eliminar códigos base sin revisar dependencias

Actualización de Datos

Siempre hacer backup antes de importar grandes cambios

Verificar que las imágenes de armados existan (.png)

Mantener actualizada la relación de costos

Rendimiento

Para proyectos grandes, trabajar por secciones

Usar la función "Actualizar" frecuentemente

Cerrar Excel/AutoCAD cuando no se usen

Solución de Problemas Comunes

Error de encoding: Verificar que archivos CSV usen UTF-8-SIG

Imágenes no encontradas: Verificar nombres y rutas

AutoCAD no responde: Asegurar que esté abierto antes de vincular

Información del programa:

