



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA LA
GESTIÓN DEL ANÁLISIS DE PRECIOS UNITARIOS EN EL NEGOCIO
EMCO.**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero de Software

ÁREA: Desarrollo de software

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la información y Ciencia
de datos

AUTOR: Juan Esteban Espín Morales

TUTOR: MSc. Félix Fernández Peña, PhD.

Ambato - Ecuador

julio - 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN DEL ANÁLISIS DE PRECIOS UNITARIOS EN EL NEGOCIO EMCO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Juan Esteban Espín Morales, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2025.



MSc. Félix Fernández Peña, PhD.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN DEL ANÁLISIS DE PRECIOS UNITARIOS EN EL NEGOCIO EMCO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio 2025.



Juan Esteban Espín Morales

C.C. 1850062876

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio 2025.



Juan Esteban Espín Morales

C.C. 1850062876

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Juan Esteban Espín Morales, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN DEL ANÁLISIS DE PRECIOS UNITARIOS EN EL NEGOCIO EMCO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, julio 2025.



Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL



Ing. Gabriel León Paredes, Mg.
PROFESOR CALIFICADOR



Ing. Leonardo Torres Valverde Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedicado a mis padres, quienes han sido una piedra angular a lo largo de toda mi vida. No se puede medir el apoyo que me han brindado, ni calcular el impacto que sus enseñanzas han tenido en mí. El éxito de este trabajo no es solo mío si no también de ellos, quienes ven culminar 18 años de esfuerzo y sacrificio.

A mis hermanos, que hicieron de la monotonía del estudio una carga más llevadera.

A toda mi familia, que ha ayudado de forma oportuna.

AGRADECIMIENTO

Quiero agradecer a mis amigos, por su apoyo incondicional. No hubiese llegado tan lejos sin su presencia en mi vida. Con ustedes incluso las más amarga de las experiencias no es más que una anécdota risible.

A mis compañeros y amigos de curso, a quienes considero una segunda familia. Juntos hemos llegado al culmen del esfuerzo y sacrificio compartidos desde el primer momento.

A mi tutor, el inge Félix. Su guía fue fundamental para poder desarrollar un sistema a la altura. Es de incalculable valor el conocimiento compartido desinteresadamente a lo largo de la investigación.

Finalmente, quiero agradecer a las mentes brillantes detrás de los modelos de lenguaje. Es increíble lo mucho que estas herramientas elevan la eficiencia de un programador.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE ANEXOS	xviii
RESUMEN EJECUTIVO	xix
ABSTRACT	xx
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	11

1.3.1 Presupuesto	11
1.3.2 Rubro de obra.....	11
1.3.3 Cubicación	12
1.3.4 Análisis de Precios Unitarios	12
1.3.5 Costos Directos	12
1.3.6 Costos Indirectos.....	14
1.3.7 Especificaciones Técnicas.....	15
1.3.8 Plataforma de Aplicación.....	16
1.3.9 Arquitectura.....	16
1.3.10 Tecnologías de desarrollo de Backend.....	18
1.3.11 Tecnologías de Frontend	20
1.3.12 Sistema Gestor de Base de Datos (SGBD)	21
1.3.13 Metodología de Desarrollo.....	23
1.3.14 Validación de Usabilidad	26
1.3.15 Herramientas de Análisis de Precios Unitarios.....	27
1.4 Objetivos	28
1.4.1 Objetivo general.....	28
1.4.2 Objetivos específicos	28
CAPÍTULO II. METODOLOGÍA.....	29
2.1 Materiales.....	29
2.2 Métodos.....	29

2.2.1 Modalidad de la investigación	29
2.2.2 Población y muestra	30
2.2.3 Recolección de información.....	30
2.2.4 Procesamiento y análisis de datos	31
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	32
3.1 Planificación.....	32
3.1.1 Requerimientos funcionales	32
3.1.2 Requerimientos No Funcionales	33
3.1.3 Plan de Entrega	34
3.2 Diseño	34
3.2.1 Arquitectura.....	34
3.2.2 Base de datos.....	35
3.2.3 Diagramado	44
3.2.4 Prototipado	52
3.3 Desarrollo.....	63

3.3.1 Aplicación Funcional	64
3.3.2 Codificación	96
3.3.3 Consideraciones de Seguridad	111
3.3.4 Pruebas de funcionalidad	115
3.3.5 Evaluación de Resultados	118
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	121
4.1 Conclusiones	121
4.2 Recomendaciones.....	122
REFERENCIAS BIBLIOGRÁFICAS	123
ANEXOS	127

ÍNDICE DE TABLAS

Tabla 1. Cadenas de Búsqueda.....	3
Tabla 2. Comparativa de Arquitecturas	17
Tabla 3. Comparativa lenguajes backend.....	19
Tabla 4. Comparativa tecnologías frontend	21
Tabla 5. Comparativa de SGBD	23
Tabla 6. Comparativa de Metodologías	25
Tabla 7. Interpretación de SUS	26
Tabla 8. Análisis herramientas APU	27
Tabla 9. Requerimientos Funcionales	32
Tabla 10. Requerimientos No Funcionales	33
Tabla 11. Plan de Entrega	34
Tabla 12. Traducción nombres tablas	35
Tabla 13. Pruebas de funcionalidad	115
Tabla 14. Resultados test SUS	120

ÍNDICE DE FIGURAS

Figura 1. Arquitectura del Sistema.....	35
Figura 2. Diagrama de base de datos	36
Figura 3. Bases de equipo, materiales y mano de obra	37
Figura 4. Rubros del proyecto.....	39
Figura 5. Tablas de cubicación.....	40
Figura 6. Base de rubros	42
Figura 7. Tabla proyecto	43
Figura 8. Tabla unidades	44
Figura 9. Diagrama de Casos de Uso	44
Figura 10. Secuencia creación registro	45
Figura 11. Secuencia editar registro.....	46
Figura 12. Secuencia eliminar registro.....	46
Figura 13. Secuencia consulta registros	47
Figura 14. Secuencia filtrar registros	47
Figura 15. Secuencia visualizar especificaciones	48
Figura 16. Secuencia usar proyecto como base	49
Figura 17. Secuencia reordenamiento	50
Figura 18. Secuencia agregar componente desde base	51
Figura 19. Secuencia pegar imagen	52
Figura 20. Mapa de navegación	52

Figura 21. Prototipo inicio	53
Figura 22. Prototipo modal nuevo proyecto.....	54
Figura 23. Prototipo selección rubros de proyecto.....	54
Figura 24. Prototipo base de equipos	55
Figura 25. Prototipo base de mano de obra.....	55
Figura 26. Prototipo base de materiales	56
Figura 27. Prototipo agregar equipo a la base.....	56
Figura 28. Prototipo agregar mano de obra a la base.....	57
Figura 29. Prototipo agregar material a la base.....	57
Figura 30. Prototipo gestión categorías.....	58
Figura 31. Prototipo presupuesto	58
Figura 32. Prototipo modal indirectos.....	59
Figura 33. Prototipo buscar en base rubros.....	59
Figura 34. Prototipo añadir rubro a la base	60
Figura 35. Prototipo análisis de precios unitarios	61
Figura 36. Prototipo resumen APU	61
Figura 37. Prototipo cubicación	62
Figura 38. Prototipo especificaciones técnicas	62
Figura 39. Prototipo Exportación.....	63
Figura 40. Página de login	64
Figura 41. Página inicio	65

Figura 42. Opciones de usuario.....	65
Figura 43. Modal cambiar datos.....	66
Figura 44. Crear usuario.....	67
Figura 45. Sidebar	67
Figura 46. Modal base de equipos	68
Figura 47. Doble filtro	68
Figura 48. Modal agregar - editar equipo.....	69
Figura 49. Componente de filtrado y adición categoría.....	69
Figura 50. Modal confirmación eliminación.....	70
Figura 51. Modal de mano de obra	70
Figura 52. Modal agregar - editar mano de obra.....	70
Figura 53. Modal de materiales.....	71
Figura 54. Modal de agregar - editar material.....	71
Figura 55. Modal de categorías.....	72
Figura 56. Tabla de proyectos.....	72
Figura 57. Popup acciones proyecto	73
Figura 58. Modal de información adicional.....	73
Figura 59. Modal crear – editar proyecto.....	74
Figura 60. Modal de actualización	74
Figura 61. Modal usar como base	75
Figura 62. Página de presupuesto.....	76

Figura 63. Opciones adicionales sidebar.....	76
Figura 64. Modal unidades.....	77
Figura 65. Acciones del proyecto.....	77
Figura 66. Modal de indirectos	78
Figura 67. Exportación.....	78
Figura 68. Presupuesto.....	79
Figura 69. Modal agregar rubro de la base	79
Figura 70. Creación nuevo rubro	80
Figura 71. Popup opciones rubro	80
Figura 72. Modal edición rubro del proyecto.....	81
Figura 73. Modal agregar a la base de rubros	81
Figura 74. Modal rubro inconsistente	82
Figura 75. Modal posible rubro duplicado.....	82
Figura 76. Indicador ingreso manual	83
Figura 77. Botones componentes de rubro.....	83
Figura 78. Advertencia cubicación – análisis de precios unitarios	83
Figura 79. Página de cubicación	84
Figura 80. Modal editar ítem de cubicación.....	85
Figura 81. Ítem de cubicación incompleto.....	86
Figura 82. Indicador cubicación incompleta.....	86
Figura 83. Página de análisis de precios unitarios	86

Figura 84. Modal agregar componente de la base.....	87
Figura 85. Modal componente duplicado.....	88
Figura 86. Transporte	88
Figura 87. Resumen APU	89
Figura 88. Componente incompleto	89
Figura 89. Componente sin homónimo en la base	90
Figura 90. Indicador APU incompleto	90
Figura 91. Página de especificaciones	90
Figura 92. Modal agregar especificaciones de la base	91
Figura 93. Modal confirmar sobreescritura.....	91
Figura 94. Modal agregar a la base de especificaciones	92
Figura 95. Confirmar pérdida de información	92
Figura 96. Formato especificaciones.....	93
Figura 97. Base de rubros	93
Figura 98. Botones componentes del rubro.....	94
Figura 99. Análisis de precios unitarios	94
Figura 100. Especificaciones rubro.....	95
Figura 101. Modal base especificaciones.....	95
Figura 102. Página editar especificación	96
Figura 103. Página nueva especificación	96

ÍNDICE DE ANEXOS

Anexo A. Entrevista Técnica	127
Anexo B. Presupuesto exportado, formato SERCOP - PDF.....	130
Anexo C. Análisis de precios unitarios exportado, formato SERCOP – PDF	131
Anexo D. Cubicación exportada, formato EMCO – PDF.....	132
Anexo E. Especificaciones exportadas, formato EMCO	133
Anexo F. Presupuesto exportado, formato SERCOP – Excel.....	134
Anexo G. Análisis de precios unitarios exportado, formato SERCOP – Excel.....	135
Anexo H. Cubicación exportada, formato EMCO – Excel.....	136

RESUMEN EJECUTIVO

EMCO, un negocio dedicado a las actividades de ingeniería civil, enfrenta desafíos importantes al momento de generar presupuestos a través del Análisis de Precios Unitarios (APU). Las herramientas de software con las que cuenta actualmente, generan complicaciones debido a la fragmentación de la información, dificultad en la propagación de cambios, falta de integración entre los componentes del presupuesto y en general, una gestión engorrosa del mismo, todo ello sumado a una usabilidad limitada.

Para superar estas dificultades, se propuso un sistema web automatizado para la gestión del análisis de precios unitarios que integre todos los procesos subyacentes al análisis de precios: formulación del presupuesto, definición de rubros, gestión de materiales, equipos y mano de obra, cubicación y especificaciones técnicas.

Las tecnologías utilizadas para el desarrollo del sistema fueron MySQL como gestor de base de datos, Node.js para el backend y React para el frontend. En cuanto a la metodología de desarrollo, se utilizó un enfoque ágil con Rapid Application Development (RAD). Se evaluó la usabilidad del sistema utilizando un test SUS (System Usability Scale) aplicado a una muestra de 11 ingenieros civiles con años de experiencia en su área.

Luego de la aplicación del test se obtuvo un puntaje SUS de 80.91, de lo que se puede concluir que el sistema tiene una buena usabilidad, es intuitivo con una interfaz amigable y facilita la aceptación de su uso en la gestión del análisis de precios unitarios.

Palabras clave: Sistema web, presupuesto, obra civil, análisis de precios unitarios, cubicación, especificaciones técnicas, RAD, SUS

ABSTRACT

EMCO, a company engaged in civil engineering activities, faces significant challenges when generating budgets through Unit Price Analysis (UPA). The current used software tools create complications due to fragmented information, difficulties in propagating changes, lack of integration among budget components, and overall cumbersome management, all compounded by limited usability.

To overcome these difficulties, an automated web system was proposed for managing unit price analysis, integrating all underlying processes: budget formulation, definition of line items, management of materials, equipment, labor, cubage, and technical specifications.

The technologies used for the system development were MySQL as the database management system, Node.js for the backend, and React for the frontend. Regarding the development methodology, an agile approach was adopted using Rapid Application Development (RAD). The system's usability was evaluated using a System Usability Scale (SUS) test, applied to a sample of 11 civil engineers with years of experience in their field.

Following the test, a SUS score of 80.91 was obtained, leading to the conclusion that the system has good usability, is intuitive, has a user-friendly interface, and facilitates its adoption for unit price analysis management.

Keywords: Web system, budget, civil projects, unit price analysis, cubage, technical specifications, RAD, SUS

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN DEL ANÁLISIS DE PRECIOS UNITARIOS EN EL NEGOCIO EMCO.

1.1.1 Planteamiento del problema

Históricamente el sector de la construcción ha sido considerado como uno de los más importantes en Ecuador [1], debido a que existe una amplia cantidad de empresas dedicadas a la construcción o actividades relacionadas. De acuerdo a la información proporcionada por Vergara, solo en el año 2012 se registró que alrededor de 14.366 compañías se encontraban dentro de la industria de la construcción en el país [1]. Para el 2020, la construcción reportó un valor superior al 8% del PIB nacional [2]. Esto se traduce en una fuerte competencia, lo que motiva a los Ingenieros Civiles a realizar proyecciones de costos lo más pertinentes a la realidad en el momento en que se va a ejecutar el proyecto constructivo [3]. El método desarrollado para este fin es el Análisis de Precios Unitarios. Durante décadas se ha utilizado este método de presupuestación, para obtener de la forma más precisa posible los precios asociados a una obra civil [4].

EMCO se trata de un negocio dedicado a las actividades de ingeniería civil, donde se realizan Análisis de Precios Unitarios con frecuencia. Actualmente se utiliza la suite de Office, la cual ofrece un amplio abanico de funcionalidades para realizar análisis. Sin embargo, posee algunas limitantes que vuelven engorroso el proceso. Para empezar, todos los ficheros asociados al Análisis de Precios Unitarios (Cubicaciones, Análisis, Especificaciones y Presupuesto) se encuentran separados, debiendo establecer vínculos estáticos entre los archivos Excel y dejando las especificaciones aisladas. Por otro lado, usualmente un proyecto de obra civil tiene rubros en el orden de las decenas o incluso más de cien, lo que implica una amplia cantidad de información relacionada entre sí. Así pues, el análisis completo lo constituyen los vínculos a los archivos que contienen la información de los rubros, los vínculos al

archivo con la información de la cubicación y las especificaciones técnicas de cada rubro.

Toda esta estructura complica el proceso de cálculo y sobre todo la corrección de errores y propagación del cambio, teniendo que ser aplicado manualmente en los archivos aislados y verificado entre los archivos vinculados. De igual modo, esta forma de trabajo reduce la portabilidad de la información, debido al fuerte acoplamiento que existe entre los archivos por las rutas absolutas. Para mitigar estos problemas, EMCO ha intentado utilizar PUNIS, software dedicado al análisis de precios que especializa Excel a través del uso de Macros [3]. Si bien la creación de varios archivos Excel y manutención de vínculos desaparecen, la herramienta demuestra poseer sus propias limitantes. Se ha reportado un elevado consumo de recursos de hardware, lo que lleva a un desempeño pobre del software en computadores de gama media. Se ha podido observar que el tiempo de carga de los listados almacenados, el tiempo de carga de la siguiente pantalla, o incluso el tiempo que se tarda en modificar el rubro es excesivo, pudiendo cualquiera de estas operaciones tomar más de 5 segundos. Por otro lado, es de difícil uso debido a que carece de elementos de interfaz gráfica de usuario como barra de navegación, botones e íconos; la navegación y sus funcionalidades principales son accesibles únicamente a través de teclado. Esto aunado a los excesivos tiempos de carga, produce gran frustración al momento de cometer un error no solo de ingreso, también de navegación. Además, el sistema carece de elementos visuales de retroalimentación, volviendo difícil entender el estado del sistema y si la operación realizada fue exitosa. Finalmente, la cubicación, así como las especificaciones no son abordadas por el programa, por lo que permanecen igualmente aisladas en ficheros de herramientas externas como Excel y Word respectivamente.

1.2 Antecedentes investigativos

La revisión de literatura se realizó en varias bases de datos científicas, así como en el motor de búsqueda académica Google Scholar. Tras una búsqueda inicial en Taylor and Francis, Springer y Willey se encontró que fuera de Latinoamérica, el Análisis de Precios Unitarios no es el método utilizado para realizar presupuestos de obra civil. Por tanto, se utilizaron estas bases para recabar información sobre la usabilidad, interfaz de usuario y evaluación de la misma. Para información más orientada hacia el

análisis de precios, se utilizaron las bases Redyalic, Scielo y los repositorios de universidades (Google Scholar). Se utilizó cadenas de búsqueda con términos afines al propósito de investigación, como “Análisis de Precios Unitarios”, “Presupuesto”, “Obra Civil”, “Sistema”, “Base de Datos”, “Usability” o “Evaluation”. La Tabla 1 muestra las tres cadenas de búsqueda que se utilizaron, dos en español y una en inglés.

Tabla 1. Cadenas de Búsqueda

No.	Cadena
1	("sistema" OR "programa") AND ("web" OR "escritorio") AND "automatización" AND ("precios unitarios" OR "análisis de precios unitarios" OR "presupuesto de construcción") AND ("construcción civil" OR "civil" OR "ingeniería civil") AND "base de datos"
2	("éxito" OR "fracaso") AND "presupuesto" AND ("precios unitarios" OR "análisis de precios unitarios") AND "web"
3	("usability testing" OR "UI/UX" OR "testing usability") AND "heuristics" AND "web" AND "design"

Estas cadenas arrojaron un total de 577 posibles fuentes. Se descartaron tesis que se centraban en el análisis de precios en áreas fuera de la Ingeniería Civil, su producto no fuera un sistema (software) o el foco fuera la ejecución de una obra civil, más no el análisis de precios. En cuanto a los artículos, no se consideraron aquellos que no aplicaban una evaluación de usabilidad o no especifiquen el método empleado, así como aquellos que propusieran nuevos métodos de evaluación de usabilidad. A los artículos restantes se aplicaron nueve criterios de pertinencia:

1. Se analiza el impacto de realizar un APU en el presupuesto de obra.
2. Se propone una solución agilice el proceso de generar un APU.
3. Se busca mejorar la experiencia del usuario.
4. Se utiliza una metodología para el desarrollo del sistema.
5. El producto es un software web.
6. Presenta resultados obtenidos con el desarrollo del software.
7. Mide la usabilidad.
8. Evalúa los resultados de usabilidad.
9. Detalla las heurísticas utilizadas en la medición.

Estos criterios se analizaron en cada documento, asignando un punto por criterio cumplido y se escogieron las investigaciones que tuvieran 5 o más puntos, resultando así en un total de 7 artículos y 8 tesis. Así pues, 15 son los documentos revisados de los cuales se extrajo la información que se presenta a continuación.

Vásquez desarrolla un software de Análisis de Precios Unitarios con una base de datos para brindar información confiable y actualizada de los rubros utilizados en la construcción. El sistema fue desarrollado en Matlab con GUI, por tanto, se trata de una aplicación de escritorio. El SGBD usado es Microsoft Access. Se busca integrar los procesos implicados en el Análisis de Precios Unitarios como son el Análisis de Rubros, la carga de la información de cada componente a través de la base de datos, además de tener asociadas a cada rubro las especificaciones y por supuesto, un módulo para la generación de reportes Excel. Se menciona que la vinculación de un rubro a su especificación es una funcionalidad que no está disponible en ningún software a nivel nacional y que la base de datos agiliza el proceso al evitar un ingreso manual constante de los valores de componentes de rubro. En cuanto a resultados, se efectuó una encuesta a 40 individuos para certificar la pertinencia del software y la frecuencia con la que estos lo utilizarían [3].

Pinchao propone el desarrollo de un módulo de Análisis de Precios unitarios para mejorar la productividad de la empresa INVESERVICE FGL S.A.S, empresa dedicada a la construcción de invernaderos. Se desarrolló el programa para un entorno web utilizando Angular y Bootstrap para el front-end, NodeJS para el back-end y PostgreSQL como SGBD. El software se orienta enteramente a la parte del análisis de precios unitarios, es decir la sección de cálculo de precio a partir de Mano de Obra, Materiales, Transporte y Equipo, además de un módulo para la generación de reportes. Se busca la automatización del proceso y la integración de una base de datos con información de cada uno de los componentes del rubro. Adicionalmente, busca la usabilidad del sistema, garantizando facilidad de manipulación para usuarios con diferentes niveles de habilidades técnicas. Para validar el éxito del desarrollo, se aplicó una encuesta de 24 preguntas a 12 personas. Las preguntas estaban orientadas a evaluar la satisfacción del usuario, la calidad del sistema, la calidad de la información y los beneficios aportados. Se midió la homogeneidad de los ítems del cuestionario utilizando el Alfa de Cronbach, considerándose la encuesta como adecuada con un

valor de 0.878. Se obtuvo que el 95,75% de los encuestados encuentran utilidad en el sistema, además de estar dispuestos a usarlo más tiempo [4].

Montalván aborda el desarrollo de un programa para la determinación de análisis de precios unitarios y presupuesto en obras civiles con fines didácticos para la Facultad de Ingeniería Civil y Mecánica. Se trata de un sistema de escritorio el cual está desarrollado en Python con interfaz gráfica y SQLite como SGBD. El programa cubre procesos del Análisis de Precios Unitarios como: el análisis de precios de cada rubro con registros de cada componente (Equipos, Materiales, Mano de Obra) en una base de datos, el presupuesto final y un módulo para la generación de reportes en formato PDF. Todos los valores para los cálculos utilizan 5 decimales, y 2 decimales para los reportes. En general se trata de integrar estos procesos de cálculo en un solo sistema y que los reportes que se generen sean visualmente similares a los que se puede obtener por parte del SERCOP. En este caso, el sistema es evaluado en base a la precisión de los valores obtenidos con respecto a los valores presentados en un Análisis de Precios Unitarios disponible en la página del SERCOP. Se seleccionó un rubro, y se comparó cada uno de los valores de sus componentes. Tras la comparativa, se obtuvo valores con una diferencia de 0.3% que se atribuyen a la falta de decimales en el cálculo obtenido del SERCOP [5].

Albán y Vázquez desarrollan un sistema para el análisis de precios unitarios de obras electromecánicas. La propuesta se implementó usando VBA (Visual Basic for Applications), esto implica que el software es una solución dependiente de Excel. Incorpora una base de datos actualizable de Equipos, Materiales y Mano de Obra para ayudar en el cálculo del precio de cada rubro, un módulo para el seguimiento al avance de proyectos y un módulo para seguimiento de mantenimiento a Equipos. Pese a estar programado para Excel, este cuenta con todos los elementos de interfaz gráfica de usuario, como botones, input text, íconos, formularios y ventanas para cada formulario, lo que facilita el uso a personas no familiarizadas con el sistema. En este caso, se realiza una comparativa entre el desarrollo y la herramienta de pago InterPro a través de un proyecto que cuenta con 40 rubros. Se destaca el carácter open source de la solución, además de la especialización en obras electromecánicas como puntos a favor de la propuesta. De igual modo, la base de datos actualizable de los costos de mano de obra permitió obtener un presupuesto más preciso que el obtenido con InterPro [6].

Soto propone una aplicación de presupuestos para una vivienda de hasta dos pisos en hormigón para dispositivos móviles. Las tecnologías que usan son NextJS para el front-end, NodeJS para el Back-end y MongoDB como SGBD. El objetivo principal de la aplicación es entregar un presupuesto en función del área de la vivienda, materiales de la estructura (acero u hormigón) y materiales de los demás elementos constructivos. Para este fin se contó con una base de datos de 635 viviendas, sus rubros correspondientes, materiales constructivos y de estructura así como los costos asociados. A partir de esa información se promediaron los costos para obtener un valor aproximado generalizable. Tras evaluar los resultados obtenidos con el sistema al calcular precios de 298 viviendas, se obtuvo un error del 9.83% para las viviendas de acero, y 12.63 % para las viviendas de hormigón. Se busca garantizar usabilidad y velocidad de uso, ya que el usuario no debe ingresar ningún valor; simplemente se limita a escoger los materiales y el área como puntos neurálgicos, y el sistema arroja un valor estimado del costo [7].

Yagual desarrolla un Sistema Web para Elaboración de Análisis de Precios Unitarios para Construcción de Viviendas en el cantón La Libertad. Se trata de un sistema web que utiliza Ionic con Bootstrap para el front-end y el framework Laravel para el back-end, así como MySQL como SGBD. Se enfoca hacia la reducción del tiempo que lleva completar un presupuesto, así como la integración de una base de datos con la información a colocar en los componentes de cada rubro. Las funcionalidades que cubre son solamente el análisis de los rubros y la recopilación de estos en el presupuesto final, así como la generación de un reporte. Cabe acotar que, al estar especializado en la construcción de viviendas, la información que se debe ingresar es más limitada. Se evidencia un tiempo aproximado de 20 minutos en realizar un Análisis de Precios Unitarios, y una reducción del tiempo empleado en la generación de reportes [8].

Morillo y Narváez implementan un software como servicio para el análisis de precios unitarios en el sector de la construcción. Las tecnologías empleadas para este fin fueron Ionic con Bootstrap para el front-end, el framework Laravel para el back-end y MySQL como SGBD. La característica a destacar es el diseño de la interfaz minimalista e ideal para poder ser manipulada no solo desde computador, sino también a través de dispositivos móviles. Las funcionalidades que cubre se limitan al análisis

de cada rubro y al compendio de estos en el presupuesto. Integra la base de datos con los precios referenciales para los componentes de los rubros, así como usuarios y proyectos. Se persigue conseguir una interfaz intuitiva y fácil de utilizar. Para evaluar el éxito del sistema se aplicó una encuesta elaborada por los autores a una muestra de 60 individuos dónde se pretendía evaluar el impacto del sistema en el tiempo de generación de presupuestos [9].

Pachacuti propone un sistema web para el diseño y cálculo de presupuesto de obras de arte hidráulicas para mejorar la calidad de resultados obtenidos, al estar focalizado exclusivamente en las obras hidráulicas. Las herramientas utilizadas son JQuery con Bootstrap en el front-end, y el framework Codeigniter para el back-end junto con PostgreSQL como SGBD. El programa aborda la toma de información para los componentes del rubro desde la base de datos. Además, integra las especificaciones como parte del sistema, asociadas a los rubros. El acercamiento propuesto cuenta con plantillas predefinidas para generar obras, así como un módulo de reportes. La efectividad del sistema se mide en cuanto a la precisión de los resultados obtenidos respecto a los valores, que arrojan 3 herramientas similares como son SODA, HCanales y un programa de Hoja de Cálculo. Los resultados presentan una ínfima variación de $\pm 0.4\%$ [10].

Song, et al. presentan una revisión en profundidad de la Interfaz Humano-Computador (IHC) centrada en los métodos de evaluación existentes. Se define la IHC como el medio para comunicar las intenciones del manipulador al computador, así como el medio para mostrar retroalimentación sobre el estado o resultados producidos por el computador al manipulador. Este ciclo se ha de repetir hasta lograr el objetivo del manipulador. Se menciona que para evaluar la IHC se debe primero tener en claro los objetivos que persigue la evaluación, para determinar el aspecto de la IHC. Los aspectos principales mencionados son usabilidad, desempeño visual, estética apta para personas con capacidades especiales, y carga mental. En cuanto a usabilidad, se define como la eficiencia, efectividad y satisfacción de un producto. Entre las seis clasificaciones a los métodos de evaluación que propone, se destaca a continuación el de evaluación subjetiva, específicamente el de “investigación de usuario”. Este consiste en la obtención de información a través de encuestas, entrevistas u observación directa. Una de las limitantes del método es la cantidad de tiempo que se

puede invertir en comunicarse con el usuario, además de la falta de generalización que puede existir si el grupo no representa a un amplio segmento de la población [11].

Lindgaard y Dudek realizaron un estudio argumentando que, si bien la ISO define la usabilidad como la eficiencia, eficacia y satisfacción al usar un producto, no existe un método para evaluar la satisfacción. Menciona que los métodos de evaluación asumen que siempre que se presente una interfaz efectiva y eficiente produce satisfacción inmediata. Se llevó a cabo un test con 20 individuos a quienes se les pidió usar un sitio web con problemas de usabilidad, una vez solo para recorrer el sitio, y otra buscando completar 8 tareas. La satisfacción de los mismos se midió a través de 5 parámetros: gusto, emoción, estética, expectativas y usabilidad; estos valores fueron obtenidos de una entrevista realizada luego de cada vez que se usó el sistema. Para encontrar si existe alguna diferencia significativa entre la media de datos de los dos experimentos, se realizó una prueba T, encontrando que hay resultados son mayoritariamente positivos cuando el usuario solo navega a través del sitio sin objetivo aparente. Así pues, los autores concluyen que la satisfacción del usuario no está incluida en la usabilidad, si no al revés: la satisfacción del usuario depende de una buena usabilidad, además de la estética, expectativa inicial y el objetivo que persigue al usar el software [12].

Paz, et al. proponen una comparativa entre dos métodos de evaluación de usabilidad aplicados a un software móvil de e-commerce. El primer método es una evaluación heurística con los 10 parámetros definidos por Nielsen. El otro método es uno novedoso, propuesto por Inostroza y que se enfoca exclusivamente en evaluar usabilidad en dispositivos móviles. La prueba se llevó a cabo con 6 profesionales, 3 de los cuales evaluaron la herramienta con la heurística de Nielsen, y los otros 3 con el método de Inostroza. Los resultados arrojaron un número dispar de errores identificados. Los problemas más críticos son identificados con el método de Inostroza (31), mientras que quienes realizaron la heurística de Nielsen encontraron 25 problemas. Se identificó que un total de 15 problemas fueron comunes en ambas evaluaciones, 16 problemas adicionales y únicos fueron encontrados por el método de Inostroza y 10 por la heurística de Nielsen. Los autores atribuyen esta diferenciación al hecho de que los expertos estaban muy familiarizados con el método de Nielsen.

Esto podría sugerir una limitación al momento de identificar los problemas de usabilidad a través de evaluaciones heurísticas [13].

Hung Hsu y Hsiung Chen efectuaron un estudio de usabilidad en la interfaz de aplicaciones de transporte. El experimento se realizó con un grupo de 30 personas a las cuales se les solicitó realizar 5 tareas en 3 aplicaciones de transporte: Uber, Lyft y Gojek. En cuanto al tiempo de completitud de las tareas, se realizó un ANOVA para determinar si las diferencias encontradas son significativas. Se encontró que las tareas 1, 2, 3 y 5 tienen diferencias significativas en cuanto al tiempo de completitud entre aplicaciones. La tarea 4 no presenta diferencias significativas. Por otro lado, para la evaluación de la usabilidad, se utilizó SUS (System Usability Scale), un método que consiste en un cuestionario de 10 preguntas con 5 opciones de respuesta en la escala de Likert. Tiene 5 preguntas positivas y 5 preguntas negativas; las preguntas están predefinidas, y el resultado del procesamiento de los resultados se mide en una escala de 0 a 100. Tras la aplicación de SUS en el experimento, se obtuvieron valores de 66.75 para Uber, 60.25 para Lyft, y 62.75 para Gojek. Estos valores fueron analizados nuevamente con un ANOVA, y se encontró que no existe diferencia estadísticamente significativa entre los valores usabilidad percibida y que en general se encuentra en el rango marginal (de 50 a 68) [14].

Matsumoto y Yang desarrollan un estudio para conocer la influencia de la presentación de la interfaz en la escaneabilidad. Este término viene del inglés “scanability” y parece no tener una traducción al español, pero el estudio clarifica que esta palabra hace referencia a que tan entendible en poco tiempo es la información presentada. El objeto de análisis en este caso son las pantallas de “self-ordering” de restaurantes, y en como la disposición de los elementos en pantalla puede llevar a un mayor tiempo empleado en realizar la orden. El experimento contó con una muestra de 11 estudiantes los cuales usaron las pantallas y a través del método de pensar en voz alta y se registró que es lo que sentían al usar la interfaz. Luego se realizó un análisis comparativo de las interfaces de 8 pantallas ubicadas en locales de Japón y Corea del Sur para extraer algunas características destacables. Finalmente, se prueban el tamaño de la imagen y los márgenes en relación con el tiempo que se tarda en tomar una decisión [15].

Algunos de los resultados son pertinentes al objeto de estudio actual. Por ejemplo, se encontró que el tiempo de selección tiende a ser más largo cuando la escaneabilidad es pobre. Si los detalles de una imagen son el factor que permite decantarse por una opción u otra, entonces las imágenes deben ser grandes. La información relevante debe procurar ser presentada en una sola pantalla, sin necesidad de hacer scroll, para no incrementar el tiempo de selección. Por último, una pobre escaneabilidad conduce a un mayor número de errores [15]. Estos hallazgos ayudan como guía para mejorar la usabilidad en un sitio web. Nielsen et al. menciona que simplificar textos o cambiándolos a puntos clave puede mejorar un 124% la escaneabilidad del sitio, conduciendo a mejoras del tiempo en completar tareas, número de errores y tiempo requerido [16].

Maguire presenta una serie de guías para el diseño de e-commerce. Si bien la presente investigación no persigue generar un e-commerce, se han extraído los principios considerados como aplicables a un sistema de Análisis de Precios Unitarios, y que van de la mano con algunas de las heurísticas de Nielsen.

- La página principal o landing page es la primera impresión para el usuario. Es importante que este pueda encontrar fácilmente lo que busca.
- El contenido debe ser reducido y no producir contaminación visual que pueda confundir o distraer al usuario
- Se debe agrupar información relacionada con un encabezado para una rápida identificación
- La navegación no debe ser compleja, y el usuario debe llegar a donde quiere con el mínimo número de clicks. Por otro lado, el usuario debe saber dónde se encuentra, en todo momento.
- Seguir los patrones o estándares de las interfaces gráficas en cuanto a la disposición de los elementos. El exceso de innovación en estos aspectos puede causar un efecto adverso y alejar a los usuarios [17].

1.3 Fundamentación teórica

A continuación, se presenta los conceptos clave en los que se fundamenta el tema de investigación, para poder así entender mejor la terminología asociada al proyecto que se desarrolla. En virtud de ello, se abordarán los conceptos relacionados a la Ingeniería Civil; particularmente aquellos indispensables para entender el Análisis de Precios Unitarios. También se tratarán los conceptos intrínsecos a la Ingeniería de Software que se van a aplicar.

1.3.1 Presupuesto

El presupuesto de una obra se trata de un documento detallado que muestra todos los costos estimados para llevar a cabo un proyecto de construcción. Incluye costos de materiales, mano de obra, maquinaria, transporte y otros recursos. El presupuesto permite planificar y organizar el proyecto, además de trazar la ruta para el control del mismo. Es por ello que el profesional debe ser sumamente minucioso al momento de presupuestar, desglosando cada rubro en sus gastos asociados. Para obtener toda la información necesaria para el presupuesto, el ingeniero civil debe primero analizar en detalle los planos. A través de este análisis se determinan los rubros de proyecto, las unidades asociadas, los frentes constructivos y las medidas de los diferentes componentes de la obra civil [5]. Entonces, expresando la matemática subyacente en lenguaje común, se puede decir que el presupuesto es la suma del producto de las cantidades de obra por el precio unitario de cada rubro [3].

1.3.2 Rubro de obra

Al hablar de rubros se hace referencia a rubros de obra, los cuales son actividades integrales de un proyecto de obra civil y que al ser ejecutadas dan como resultado una parte tangible del proyecto. A nivel de presupuesto, un rubro tiene un nombre, una unidad de medida, cantidad, precio unitario y costo. Pese a conceptualmente ser actividades, sus nombres no siempre van asociados a un verbo; pueden ser también sustantivos que describan a un material o equipo. La cantidad es el resultado de la cubicación, mientras que el precio unitario es el resultado del Análisis de Precios

Unitarios. Finalmente, el costo es el resultado de multiplicar la cantidad y el precio unitario. A nivel de proyecto, además de lo previamente descrito un rubro debe contar con especificaciones [6].

1.3.3 Cubicación

También llamada cubicaje o cálculo de volúmenes, consiste en cuantificar las cantidades de obra involucradas en la ejecución de un rubro. Dichas cantidades pueden expresar no solo volumen, sino también peso, longitud, área, etc. Si se pone en palabras la matemática aplicada: es el producto de las dimensiones del cuerpo de análisis por el número de veces que este aparece en el rubro [5]. En la práctica, un rubro puede estar constituido por más de un cuerpo y estos pueden ser irregulares, lo que obliga a seccionar al cuerpo y sumar todas sus partes para obtener el total. Entonces, la cubicación en el cálculo de presupuestos es la suma de la cantidad de los cuerpos por el número de veces estos que aparecen, siendo que la cantidad de los mismos es el resultado de la suma de sus secciones [3].

1.3.4 Análisis de Precios Unitarios

El análisis de precios unitarios (APU) es un procedimiento que permite calcular el costo de cada actividad dentro de una obra civil por unidad de medida. Este análisis desglosa el rubro en sus componentes básicos y determina el costo de cada uno; a estos componentes se los conoce como costo directo. También se consideran los costos indirectos, entre los que se encuentra la utilidad. La suma de los costos directos e indirectos da como resultado el precio por cada unidad de trabajo, en función de los recursos necesarios [9].

1.3.5 Costos Directos

Son aquellos gastos directamente relacionados con la ejecución de los trabajos propios del proyecto; es decir, son los costos asociados a los recursos que se emplean de forma directa en la construcción. Para el análisis de precios unitarios, estos se dividen en cuatro categorías principales: equipos o maquinaria, mano de obra, materiales y transporte. Estas categorías son referidas como los componentes del análisis y son

fundamentales en el cálculo del presupuesto, ya que constituyen la base para determinar el costo unitario de un rubro [7]. A continuación, se detalla cada componente.

Equipos (M): Abarcan las herramientas o maquinaria necesaria para la ejecución del rubro. Para determinar el costo de cada uno se analiza:

- Cantidad (A)
- Tarifa (B)
- Costo Hora ($A * B = C$)
- Rendimiento R
- Costo ($R * C = D$)

Existe un caso especial, que es la Herramienta Menor, la cual representa el desgaste por uso de herramientas de mano. Para determinar su costo no se analiza ninguno de los puntos anteriores, si no que se asigna el cinco por ciento (5%) de la mano de obra del rubro.

Mano de Obra (N): Corresponde al personal que participa en la ejecución del rubro como albañiles, carpinteros, electricistas y operadores. Para determinar su costo se analiza:

- Cantidad (A)
- Tarifa/Hora (B)
- Costo Hora ($A * B = C$)
- Rendimiento (R)
- Costo ($C * R = D$)

Pese a tratarse de personas, las cantidades de mano de obra pueden ser decimales, ya que en este caso se mide la cantidad de incidencia del trabajador en el rubro; no siempre una persona dedica la totalidad de su incidencia a un solo rubro, si no que su actuación está distribuida entre varios de ellos. La Tarifa/Hora está dada por la CAMICON, y es un valor que considera el salario mensual, décimos y beneficios de ley para determinar cuánto se debe pagar por hora al trabajador.

Materiales (O): Incluyen todos los insumos necesarios para la construcción, como cemento, ladrillos, acero, etc. Para obtener su costo se analiza:

- Unidad
- Cantidad (A)
- Precio Unitario (B)
- Costo ($A * B = C$)

Pese a no participar de forma directa en el cálculo de su costo, la unidad es de vital importancia para entender sin ambigüedad la cantidad empleada, así como el costo asociado.

Transporte (P): Responde al traslado del material desde los puntos de explotación o compra, hasta la obra o proyecto en desarrollo. Su análisis es exactamente el mismo que los materiales, reemplazando el Precio Unitario por la Tarifa de Transporte.

- Unidad
- Cantidad (A)
- Tarifa de Transporte (B)
- Costo ($A * B = C$)

No solo coinciden los puntos de análisis, si no que existen tantos elementos de transporte como materiales tenga el rubro. Para la mayor parte de casos el total de este componente es cero; el costo de movilización de material suele estar incluido en el precio del material [5].

1.3.6 Costos Indirectos

Los costos indirectos de una obra son aquellos gastos que, aunque no están asociados directamente con un rubro específico de la obra, son necesarios para la correcta ejecución del mismo. Estos costos se dividen en tres categorías: Costos Fijos, Utilidad e Imprevistos.

Costos Fijos: Incluyen los gastos administrativos, gastos de oficina y otros gastos generales que apoyan la ejecución del proyecto. Subscripciones, afiliaciones, seguros también entran en esta categoría.

Imprevistos: Se debe tomar en cuenta a las eventualidades no planificadas en el análisis, ya que inversiones o costos adicionales pueden impactar negativamente en el proyecto, elevando el costo y representando a la larga pérdida para el contratista.

Utilidad: Se refiere a la ganancia que se espera percibir tras la ejecución de la obra. Uno de los factores más importantes para el contratista [7].

Los costos indirectos no suponen un valor fijo en el análisis, si no uno porcentual, es decir, los valores que pueden tomar cualquiera de las categorías anteriormente mencionadas aumenta o disminuye según el costo del rubro. Los indirectos son los mismos para todos los rubros de un proyecto y se aumentan a todos los análisis de precios unitarios [3].

1.3.7 Especificaciones Técnicas

Se trata de la descripción detallada del rubro que se analiza, y sirve para que las personas encargadas de la ejecución de la obra (constructor, dueño del proyecto y fiscalizador) tengan un punto de referencia común y sepan con exactitud cuáles son las condiciones que el equipo encargado del diseño del proyecto estableció para la ejecución del rubro. Algunos de los elementos de la especificación son:

- La definición del rubro.
- El detalle de las medidas, materiales, formas y dimensiones.
- Normas y estándares que deben cumplir los materiales.
- ¿Cómo se va a medir?
- Ensayos.
- Proceso constructivo.
- Referencias.
- Parámetros de fiscalización [3].

1.3.8 Plataforma de Aplicación

Al hablar de plataforma de aplicación se hace referencia al entorno en el que se ejecutará la aplicación una vez desarrollada. Por consenso general, las plataformas a las que se puede hacer referencia son web, escritorio, móvil y multiplataforma [8]. Tanto Vázquez y Montalván desarrollan un sistema de escritorio para el análisis de precios unitarios. Se menciona que la elección se dio para aprovechar al máximo el hardware del dispositivo, por cuanto el sistema requería de altos recursos de procesamiento. Además, se hace hincapié en la facilidad de desarrollo, ya que para ambos casos solo se requirió aprender un lenguaje de programación (Matlab y Python respectivamente), así como SQL para la persistencia de datos [3], [5].

Por otro lado, las tesis restantes optaron por un el desarrollo de una aplicación web, destacando principalmente el carácter multiplataforma de esta opción. Pinchao sobresalta la facilidad de actualización de aplicativos móviles, al tener que actualizar una sola vez y de forma centralizada, además de permitir a los usuarios acceder rápidamente a los cambios sin reinstalaciones [4]. Finalmente, Yagual indica que el uso de un sistema web aliviana las tareas de procesamiento en el cliente, dejando el procesamiento pesado para el servidor [8].

En base a lo expuesto, se ha decidido desarrollar un sistema web para dar solución a la problemática actual de EMCO. Principalmente, se tomó esta decisión por el carácter multiplataforma de una web y la facilidad e inmediatez de las actualizaciones.

1.3.9 Arquitectura

Una arquitectura de software se trata del diseño de la estructura de un sistema; es decir, cuáles son sus componentes, y cómo estos se relacionan entre sí. Dentro de las arquitecturas comunes en la nube están el modelo MVC, el cliente-servidor y SOA [8].

La arquitectura cliente servidor se trata de un modelo en el cuál las funciones de la aplicación se dividen en dos roles con funciones diferentes. Por un lado, el cliente es la parte encargada de la interacción con el usuario final, solicitando y recibiendo información del servidor. Por el otro lado el servidor, encargado de procesar

solicitudes y enviar las respuestas. Según la arquitectura física, este mismo será el encargado del almacenamiento, o en su defecto se comunicará con el servidor de base de datos. Pachacuti optó por el uso de esta arquitectura por su modelo centralizado, para facilitar la gestión e implantación de la aplicación. Se menciona como ventaja la escalabilidad del modelo, pudiendo agregar más servidores para atender un mayor número de solicitudes [10].

Por su parte, MVC (Modelo-Vista-Controlador) es un patrón de arquitectura de software que separa la aplicación en tres componentes, mejorando la organización del código. A continuación, se detalla la función de cada componente. El Modelo aísla toda la lógica del negocio y es el encargado de operar con la base de datos. La vista es la parte de la aplicación que interactúa con el usuario, mostrando información, capturando los eventos de interacción y comunicándolos al controlador. El Controlador actúa como intermediario entre los dos componentes anteriores, comunicando las solicitudes del usuario al modelo y actualizando la vista conforme este procese las mismas [6].

SOA (Service Oriented Architecture) se basa en la reutilización de servicios y la interoperabilidad, ya que busca integrar funcionalidades que están programadas en diferentes lenguajes. Cada servicio expone una interfaz y la comunicación se da a través de contratos bien definidos (REST o SOAP). Su uso es más común en el ámbito empresarial, donde se debe integrar arquitecturas y módulos diversos en tecnologías [4].

La Tabla 2 muestra las características más destacables de las arquitecturas más utilizadas en los documentos de la revisión bibliográfica.

Tabla 2. Comparativa de Arquitecturas

Criterio	MVC (Modelo-Vista-Controlador)	Cliente-Servidor	SOA (Arquitectura Orientada a Servicios)
Propósito principal	Separar responsabilidades dentro de una aplicación	Distribuir la carga de trabajo entre un cliente que solicita y un servidor que entrega.	Integrar funcionalidades distribuidas y reutilizables como servicios
Tipo de comunicación	Interna con funciones y eventos de la aplicación	Peticiones y respuestas por protocolos (HTTP/HTTPS, FTP, etc)	Basada en contratos (SOAP, REST)

Criterio	MVC (Modelo-Vista-Controlador)	Cliente-Servidor	SOA (Arquitectura Orientada a Servicios)
Acoplamiento	Bajo entre las capas	Medio: comparten protocolos	Muy bajo: servicios independientes
Escalabilidad	Limitada: se escala toda la aplicación	Buena: se puede escalar el servidor y los clientes	Alta: servicios distribuidos y escalables por separado
Tecnologías que la usan	Django, Ruby on Rails, ASP.NET MVC	Aplicaciones de escritorio/web con backend centralizado	Sistemas empresariales, ERP, integración de sistemas
Mantenibilidad	Alta (cada capa se modifica por separado)	Media (cliente y servidor deben estar sincronizados)	Alta (servicios se mantienen por separado)
Despliegue	Monolítico: En una sola aplicación o servidor web	En dos capas separadas	Distribuido (cada servicio en su propio entorno)

Se decidió utilizar una arquitectura cliente-servidor, debido a que EMCO no busca unificar sistemas diferentes, ni cuenta con múltiples servidores para poder desplegar una aplicación utilizando SOA. Por otro lado, el patrón MVC no se muestra como una opción atractiva ya que complicaría de manera innecesaria la aplicación, además de anular totalmente la centralización, siendo que cada cliente incorpora la lógica de negocio. Con eso claro, se escoge la arquitectura cliente-servidor, ampliando así el abanico de opciones tecnológicas que se pueden utilizar para el desarrollo, además de permitir realizar los procesos más pesados de la lógica de negocio fuera del cliente.

1.3.10 Tecnologías de desarrollo de Backend

En el contexto de una arquitectura cliente-servidor, al hablar del backend se hace referencia a la parte del sistema alojada en el servidor. Esta se encarga de gestionar las solicitudes de los usuarios y para ello ha de apoyarse en un lenguaje de programación o en un framework, el cual extiende las funcionalidades del lenguaje y proporciona una serie de librerías y utilidades para facilitar el desarrollo [4].

En los documentos presentados por [7] y [8], se presenta PHP como lenguaje de desarrollo, junto con el framework Laravel. Una de las razones que motivaron su elección fue su uso ampliamente extendido en aplicaciones empresariales, además del manejo de código embebido en HTML. El framework Laravel, incrementa las opciones disponibles de PHP, pero conduce a una estructura más orientada al uso de MVC (Modelo-Vista-Controlador). Como alternativa se presenta el framework CodeIgniter, con herramientas de seguridad integradas y funcionalidades

preprogramadas, lo que disminuye el tiempo de desarrollo y facilita la entrega del producto [10].

Los autores de [4] presentan a Node.js, como una elección efectiva para el desarrollo de aplicaciones web modernas, en línea con la tendencia actual. Se trata de un entorno de ejecución que utiliza JavaScript, lo que unifica el lenguaje de frontend y backend. Los autores destacan el amplio ecosistema de librerías que este presenta como su máximo punto a favor, facilitando un desarrollo rápido. Además, la libertad que ofrece JavaScript en cuanto al tipado, su baja curva de aprendizaje, su fuerte orientación hacia la programación funcional y su manejo de hilos facilita un manejo de eventos asíncronos con una sintaxis más natural. Para la comunicación ente el cliente y servidor se menciona el uso de Express, un framework minimalista para Node.js encargado de la gestión de solicitudes HTTP.

Los lenguajes y frameworks de programación utilizados para la parte del servidor son muy variados. La Tabla 3 muestra los criterios con los que se evaluaron las tecnologías para el backend.

Tabla 3. Comparativa lenguajes backend

Criterio	Java	Python	PHP	JavaScript
Velocidad de desarrollo	Lenta	Media	Rápida	Muy rápida
Curva de aprendizaje	Alta	Baja	Media	Baja
Ecosistema	Medio	Grande	Medio	Gigantesco (NPM)
Desempeño	Alto	Medio	Medio-Bajo	Alto
Flexibilidad	Rígida	Alta	Alta	Máxima
Tipado	Fuerte	Bajo	Bajo	Bajo
Multiplataforma	Sí	Sí	Web principalmente	Todas (Web, móvil, desktop, IoT)
Sintaxis	Verbosa	Limpia	Inconsistente	Limpia
Comunidad	Grande	Grande	Media	Grande y activa
Innovación	Lenta	Media	Lenta	Muy rápida
Escalabilidad	Excelente	Buena	Limitada	Excelente

Se optó por utilizar JavaScript, por ser una tecnología moderna, la amplia cantidad de librerías con las que cuenta y la facilidad que supone la instalación y uso de las mismas. De igual modo, la sintaxis de JavaScript es mucho más ligera y fácil de manejar, sin la necesidad de declaración de tipos, y con una fuerte aplicación de la programación funcional. Se descartó PHP por su bajo soporte en cuanto a librerías, además de su sintaxis poco amigable en cuanto a declaración de variables. Java se presentaba como

una buena opción, sin embargo, se descartó por su sintaxis verbosa y el tiempo que se invierte en la creación y modificación de clases.

1.3.11 Tecnologías de Frontend

En el contexto de una arquitectura cliente servidor el frontend es la parte con la que interactúan los usuarios, manipulada a través de un navegador. Esta capa se encarga de la estructura del sitio, el diseño visual y el comportamiento de la interfaz gráfica, apoyándose en tecnologías como HTML, CSS y JavaScript. Se encarga además de comunicar las peticiones del usuario al backend y mostrar las respuestas del servidor [4]. Existen frameworks y librerías que facilitan el desarrollo del frontend; su uso es lo más común por el amplio abanico de funcionalidades que ofrecen.

Albán y Vázquez optaron por el uso de Ionic debido a la orientación multiplataforma del desarrollo propuesto (Móvil y Web). Esto se debe a que Ionic permite crear aplicaciones con una sola base de código en tecnologías web estándar (HTML, CSS y JavaScript), y exportarlas a diferentes plataformas como iOS y Android. Además, cuenta con una librería integrada de componentes prefabricados y altamente personalizables, garantizando interfaces con un apartado visual estandarizado. Incluye además utilidades para acceder a funcionalidades móviles nativas, como cámara y GPS a través del uso de plugins. Su curva de aprendizaje puede ser elevada, ya que además de entender el flujo del framework, es necesario apoyarse de otros frameworks/librerías como Angular y React [6].

En [7] se hace uso de Next.js por ser un framework de desarrollo web basado en React, que permite construir aplicaciones tanto del lado del cliente como del servidor (renderizado híbrido). Next.js hereda el modelo de componentes de React, pero va más allá e incrementa las funcionalidades disponibles al ofrecer enrutamiento nativo, manejo de imágenes y generación de sitios estáticos optimizados.

Las tecnologías de frontend son principalmente frameworks o librerías, ya que para aplicaciones con complejidad media es inviable el uso de HTML, y JavaScript vanilla. La Tabla 4 muestra la comparativa de las principales tecnologías encontradas.

Tabla 4. Comparativa tecnologías frontend

Criterio	Angular	Ionic	React	Next.js
Tipo de tecnología	Framework completo	Framework híbrido	Librería UI	Framework sobre React
Curva de aprendizaje	Alta (arquitectura compleja)	Media	Media	Media
Popularidad	Alta, pero en descenso	Media	Alta	Alta (en crecimiento)
Flexibilidad	Baja, es bastante estructurado	Media, depende de con qué framework se utilice	Muy alta	Alta, con convenciones útiles
Enfoque principal	SPA empresariales	Aplicaciones móviles híbridas	UI reactivas y escalables	SSR, SSG, y aplicaciones web rápidas
Componentes reutilizables	Sí	Sí	Sí	Sí (hereda todo de React)
Ecosistema	Grande, pero cerrado	Integrado con Angular/React	Enorme, con muchas librerías compatibles	Muy rico, optimizado para producción
Comunidad	Grande	Mediana	Muy Grande	Grande y en crecimiento

Se decidió utilizar React, debido a su enfoque al desarrollo de componentes, la libertad que otorga en cuanto a la organización de los archivos y carpetas. Por otro lado, React propone un sistema de manejo de propagación de información entre componentes a través de props, o se puede utilizar contextos que inyectan la información. Ionic no es necesario ya que el proyecto no tiene miras a ser utilizado en dispositivos móviles. Por otro lado, se descartó Angular por ser un framework completo con una estructura más compleja cuando se trata de comunicación entre objetos, además de más rigidez en la organización de los archivos. Next.js puede verse como una alternativa viable; sin embargo, su orientación hacia un backend integrado con el frontend lo hace caer en el grupo de tecnologías de arquitectura MVC.

1.3.12 Sistema Gestor de Base de Datos (SGBD)

Las bases de datos son el conjunto de datos estructurados que son manipulados a través de un software gestor. Se diferencian de una hoja de cálculo por permitir almacenar cantidades masivas de información que puede ser accedida y manipulada por más de una persona [5]. El sistema gestor de base de datos es el encargado de permitir dicha manipulación (creación, lectura, modificación y eliminación de datos) y es muy común referirse al software gestor de bases como base de datos.

Las bases de datos más populares son las relacionales, las cuales requieren de un lenguaje estructurado de consulta (Structured Query Language) para su manipulación. Dentro de esta categoría y en el contexto de la revisión de literatura, se encontró a MySQL, PostgreSQL como los SGBD utilizados. También existen bases de datos no relacionales, las cuales tienen una orientación a documentos con formatos de almacenamiento flexibles [7]. En la revisión se encontró el uso de MongoDB para el desarrollo de un sistema de análisis de precios unitarios.

PostgreSQL es una base de datos relacional de código abierto escogida en [4] por su orientación a objetos y la capacidad nativa de manejar transacciones. Gracias a la facilidad con la que permite garantizar los principios ACID en transacciones, es ideal para aplicaciones que requieren procesos avanzados y manejo de datos estructurados y no estructurados. PostgreSQL es compatible con el estándar SQL, pero también permite el uso de JSON y XML, siendo esta la causa de su elección en [4], ya que en la propuesta se manipulan objetos XML.

MySQL es la base de datos relacional que destaca en la mayoría de tesis revisadas, debido a su popularidad, además de ser ampliamente utilizada para aplicaciones web. Es de código abierto y permite su uso de forma gratuita y legal a desarrolladores en fase de aprendizaje [6]. MySQL destaca por su buen rendimiento en consultas frecuentes, lo que la hace ideal para sitios de contenido dinámico y aplicaciones de comercio electrónico [8]. Además, existen librerías que integran los drivers necesarios para la gestión de la conexión de los distintos lenguajes de programación y frameworks. Su comunidad está activa y proporciona soporte constante, además de mejoras frecuentes [10].

MongoDB es una base de datos NoSQL de tipo documental que permite almacenar datos en formato JSON. Proporciona flexibilidad en el modelo de los datos (no hay una estructura fija) y es escalable horizontalmente; ideal para aplicaciones que aumentan su tamaño constantemente. Se describe su sistema de consultas como intuitivo y mucho más sencillo de aprender en relación con SQL [7]. Esta opción es ideal si lo que se busca es tener una estructura dinámica que puede variar no solo con el tiempo, si no entre registros y además esta puede crecer de forma exponencial.

Dada la naturaleza del problema que se abordó, fue necesario hacer uso de un Sistema Gestor de Base de Datos (SGBD). La Tabla 5 muestra las características relevantes de los sistemas gestores de bases de datos.

Tabla 5. Comparativa de SGBD

Criterio	MySQL	PostgreSQL	MongoDB
Tipo de base de datos	SQL	SQL	NoSQL
Estructura de datos	Tablas con filas y columnas	Tablas con filas y columnas	Documentos JSON-like (BSON)
Lenguaje de consultas	SQL	SQL con extensiones avanzadas	MongoDB Query Language (similar a JSON)
Soporte ACID	Sí (InnoDB)	Sí (nativo)	Parcial
Curva de aprendizaje	Baja (muy extendido y simple)	Moderada (más conceptos avanzados)	Baja si se conocen estructuras JSON
Licencia	Libre	Libre	Versión de pago o gratuita
Casos de uso ideales	Aplicaciones web, sitios CMS, e-commerce	Sistemas financieros, análisis de datos, GIS	Big data, IoT, analytics

La elección de la base de datos fue un proceso arduo ya que, en su mayoría las opciones encontradas en la literatura son buenas elecciones. Al final se optó por utilizar MySQL debido a su baja curva de aprendizaje y su orientación a aplicaciones web multipropósito. Se descartó PostgreSQL debido a que el tiempo que se debe invertir en práctica y aprendizaje es mayor, además de ofrecer soporte para funcionalidades mucho más complejas que las que se necesitaba. Por otro lado, MongoDB se descartó por considerar que el sistema debe mantener una estructura de datos bien definida y no tener miras a hacer analítica de datos.

1.3.13 Metodología de Desarrollo

Una metodología de desarrollo de software es un conjunto de prácticas y principios que guían el proceso de creación de software [6]. Se divide a las metodologías en tradicionales o clásicas y ágiles. Las tradicionales como el modelo en cascada se caracterizan por un enfoque lineal y secuencial, en el que cada fase del proyecto (análisis, diseño, desarrollo, pruebas, y despliegue) debe completarse antes de pasar a la siguiente [8]. Se recomienda estas metodologías solo si los requerimientos están claros desde el principio y no van a existir cambios sustanciales. En contraste, las metodologías ágiles son flexibles y se adaptan mejor a proyectos en los que los

requisitos pueden cambiar constantemente; promueven la entrega incremental de avances en ciclos cortos conocidos como iteraciones. La diferencia principal es que las metodologías ágiles priorizan la adaptación al cambio, mientras las tradicionales se enfocan en una planificación exhaustiva [10].

Se utilizó la metodología ágil Scrum, en [4] y [10] debido a su orientación hacia la organización del trabajo. Scrum permitió dividir el proyecto en tareas más pequeñas y manejables, lo cual facilitó el control y seguimiento del progreso. Al establecer sprints o ciclos de desarrollo cortos, los autores pudieron hacer entregas funcionales de forma regular. Uno de los puntos en contra del uso de esta metodología para el desarrollo de los dos sistemas propuestos, es la nula aparición de los roles propios de SCRUM, además de no cumplir con el mínimo de miembros en el equipo que sugiere la metodología, lo que a su vez conduce al incumplimiento de las ceremonias; con un solo individuo su ejecución no tiene sentido.

En [10] se utiliza la metodología ágil XP, por ser ideal para un solo desarrollador que busca maximizar la calidad del código y adaptarse rápidamente a cambios en los requisitos. Pese a que uno de sus principios es la programación en pares, no es obligatorio contar con dos desarrolladores para aplicar la metodología [7]. Se encuentra una severa desventaja en términos de tiempo por el fuerte enfoque hacia pruebas. Para asegurar la calidad del código, se desarrolla desde el principio con un enfoque en TDD (Test Driven Development), con prácticas que incluyen primero programar el test y luego el código en sí [4].

Por otro lado, se tiene RAD (*Rapid Application Development*) que es una metodología incremental que busca agilizar el ciclo de desarrollo mediante iteraciones rápidas y la participación activa de los usuarios. Esta metodología permite adaptación rápida a los cambios de requisitos y exige una mínima planificación. A diferencia de otros enfoques el usuario interviene lo más seguido posible validando prototipos los cuales se refinan hasta cumplir con sus expectativas. Entre sus principios se encuentra el uso de herramientas para generar código, herramientas de prototipado y la reutilización de componentes. En general, su objetivo es acelerar el desarrollo para conseguir un producto funcional que cumpla con los requerimientos del usuario en el menor tiempo posible [18].

La Tabla 6 resume los aspectos más relevantes de cada metodología.

Tabla 6. Comparativa de Metodologías

Criterio	SCRUM	XP (Extreme Programming)	RAD (Rapid Application Development)
Equipo	5–9 miembros	4–10 miembros	No está definido
Principales prácticas	- Sprints - Daily Scrum - Backlog - Reviews - Retrospectivas	- Programación en pareja - TDD - Refactorización - Integración continua	- Prototipado rápido - Reutilización - Desarrollo iterativo - Uso de herramientas CASE.
Documentación	Amplia, se documenta desde historias de usuario hasta retrasos en el desarrollo	Mínima, se prioriza el código funcional	Énfasis en prototipos
Pruebas	No son obligatorias	Pruebas automatizadas desde el inicio (TDD)	Pruebas de caja negra
Relación con el usuario/cliente	Indirecta: Product Owner representa al cliente	El cliente está presente todo el tiempo	Cliente presente desde los prototipos no funcionales
Roles	- Scrum Master - Product Owner - Equipo de Desarrollo	Incluye programadores, cliente, coach.	Sin roles predefinidos
Adecuado para	Proyectos complejos y cambiantes, con equipos autoorganizados	Proyectos con alta incertidumbre técnica y equipos muy colaborativos	Proyectos que requieren entrega rápida y prototipos funcionales
Desventajas	- Riesgo de mala implementación del marco - Requiere disciplina	- Se centra mucho en código de alta calidad - Difícil de escalar	- Puede sacrificar calidad por velocidad - Requiere compromiso alto del cliente
Herramientas clave	Jira, Trello, Azure DevOps	Jenkins, Git, JUnit, herramientas de CI/CD	Prototipadores visuales, CASE tools, RAD Studio

La metodología que se escogió fue RAD, debido su enfoque en entregas rápidas de valor, prototipos que facilitan la comunicación entre el cliente y el desarrollador, su libertad en cuanto a roles y la verificación de la funcionalidad del sistema a través de pruebas manuales. Presenta otras ventajas, como los principios de reutilización, el uso de herramientas CASE para generar código y la priorización de tareas relevantes y omisión de aquellas que comprometen la entrega en el plazo establecido. Todo ello va de la mano con el proyecto que se quiere implementar y el escenario donde se desarrollará, con el cliente en constante comunicación, retroalimentación inmediata y el uso de prototipos para alcanzar acuerdos de funcionalidad, sin mencionar la necesidad de un desarrollo rápido.

1.3.14 Validación de Usabilidad

Se ha encontrado que los métodos de evaluación predominantes son los de evaluación subjetiva, llevada a cabo a través de encuestas dirigidas directamente al usuario [11], [12], [14]. De estos, se destaca el System Usability Scale (SUS) debido a que presenta un cuestionario de 10 preguntas predefinidas con opciones de respuesta en la escala de Likert y dirigida a expertos. Esta encuesta puede ser aplicada a un mínimo de 5 individuos y los resultados obtenidos serán significativos al objeto de estudio [19]. Las investigaciones propuestas en [12] y [14], indican que para evaluar si los resultados en cuanto a usabilidad son significativamente mejores que los que se tenía respecto a mediciones previas, se utilizó ANNOVA y prueba T. Según se especifica, un ANNOVA se debe utilizar cuando se contrasta 3 o más grupos de análisis, mientras que una prueba T es ideal para 2 grupos.

En base a lo encontrado, se optó por escoger el test SUS como la herramienta para la evaluación de usabilidad del sistema propuesto. El método convierte el criterio subjetivo de los participantes en un valor numérico; por tanto, fue necesario realizar una interpretación de estos valores. La información para este fin se obtuvo del artículo propuesto por Sauro en [20]. En el expone cinco formas de interpretar los resultados obtenidos en base a percentiles, notas, adjetivos, rango, promotores y detractores. Sus hallazgos se resumen en la Tabla 7.

Tabla 7. Interpretación de SUS

Fuente: Traducido de [20]

Nota	SUS	Rango percentil	Adjetivo	Rango	NPS
A+	84.1-100	96-100	Mejor Imaginable	Aceptable	Promotor
A	80.8-84.0	90-95	Excelente	Aceptable	Promotor
A-	78.9-80.7	85-89		Aceptable	Promotor
B+	77.2-78.8	80-84		Aceptable	Pasivo
B	74.1 – 77.1	70 – 79		Aceptable	Pasivo
B-	72.6 – 74.0	65 – 69		Aceptable	Pasivo
C+	71.1 – 72.5	60 – 64	Bueno	Aceptable	Pasivo
C	65.0 – 71.0	41 – 59		Marginal	Pasivo
C-	62.7 – 64.9	35 – 40		Marginal	Pasivo
D	51.7 – 62.6	15 – 34	Regular	Marginal	Detractor
F	25.1 – 51.6	2– 14	Pobre	Inaceptable	Detractor
F	0-25	0-1.9	Peor Imaginable	Inaceptable	Detractor

1.3.15 Herramientas de Análisis de Precios Unitarios

La Tabla 8 muestra un análisis comparativo entre las herramientas probadas en EMCO, para poder obtener en claro las limitantes encontradas, fortalezas y áreas de mejora.

Tabla 8. Análisis herramientas APU

Característica	PUNIS v10	Microsoft Excel
Tipo de Herramienta	Aplicación especializada en el Análisis de Precios Unitarios desarrollada en Excel	Hoja de cálculo general
Funcionalidades	- Análisis de precios unitarios - Cálculo de presupuestos de obras - Cronogramas valorados - Gestión de insumos (materiales, equipo, mano de obra) - Fórmulas de reajuste y cuadrillas tipo - Copiar Presupuestos - Compatible con los formatos del SERCOP	- Cálculos matemáticos y financieros - Gráficos y tablas dinámicas - Análisis de datos - Macros y automatización - Amplia variedad de funciones para diversos campos
Facilidad de Uso	- Terminología de la interfaz diseñada para profesionales de la construcción - Métodos abreviados de teclado para mayor rapidez	- Interfaz generalista y de uso intuitivo - Requiere personalización para tareas específicas - El nivel de complejidad del programa dependerá de las funcionalidades que se quiere abarcar
Requisitos de Software	- Windows 8 o superior - Office 2010 o superior	- Disponible en múltiples plataformas (Windows, macOS) - Compatible con diversas versiones de Windows y Office
Requisitos de Hardware	Los requisitos no están especificados por el creador del sistema, pero se hicieron pruebas en una laptop con 16gb de RAM, y un procesador de 6 núcleos y 1.6 Ghz registrándose los tiempos de respuesta de hasta 5 segundos.	- Procesador de 2 núcleos y 1.6 Ghz. - 4 GB de memoria RAM, pero si se usa la versión para 32 bits solo 2 GB. - 4 GB de espacio en disco duro.
Soporte y Documentación	- Manual de usuario disponible en fuentes no oficiales y desactualizado - Comunidad de usuarios limitada - Cuenta con videos explicativos sobre su uso - Actualizaciones y reinstalaciones en forma permanente durante el periodo de garantía (personal o acceso remoto vía Internet)	- Amplia documentación oficial - Gran comunidad de usuarios y foros de soporte - Actualizaciones siempre disponibles - Variedad de cursos y videos sobre su uso
Licencia	De pago, adicional al pago de la licencia de Office para poder usar Excel	De pago, licencia comercial de Microsoft

Característica	PUNIS v10	Microsoft Excel
Ventajas	- Herramienta especializada para el sector de la construcción en Ecuador - Automatiza procesos específicos del análisis de precios unitarios - Ahorra tiempo en la elaboración de presupuestos y análisis	- Versatilidad para diversas tareas y sectores - Amplia gama de funciones y herramientas - Integración con otras aplicaciones de Microsoft Office.
Desventajas	- Dependencia de Excel y Windows - Carece casi en su totalidad de componentes gráficos - Elevado tiempo de respuesta en computadores de gama media - No integra las especificaciones ni cubicación	- Requiere personalización para tareas especializadas - No incluye funcionalidades específicas para el sector de la construcción - Requiere un nivel medio-alto en el uso de la herramienta para crear programas de análisis de precios unitarios automatizados

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema que integre los procesos subyacentes al Análisis de Precios Unitarios mejorando la usabilidad en la generación de presupuestos.

1.4.2 Objetivos específicos

- Identificar las tecnologías a utilizar para la implementación del software.
- Desarrollar un sistema automatizado para la gestión del Análisis de Precios Unitarios.
- Validar la usabilidad del sistema propuesto.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Se utilizaron diversos materiales para el desarrollo del proyecto, mismos que fueron esenciales en el análisis, codificación y redacción del documento. Entre los recursos físicos se incluye una laptop, lápiz y cuaderno. En cuanto a recursos tecnológicos está el internet, IDEs de desarrollo, librerías como soporte al desarrollo, herramientas CASE para las primeras etapas y finalmente almacenamiento en la nube tanto para las herramientas CASE como para el cuestionario, el cual se ejecutó a través de Google Forms.

2.2 Métodos

2.2.1 Modalidad de la investigación

Se detalla a continuación los dos tipos de investigación involucrados en el desarrollo de la propuesta.

Modalidad Bibliográfica

Se usó la modalidad bibliográfica de tipo analítica, ya que se centra en el análisis crítico y comparativo de información proveniente de diversas fuentes secundarias (artículos científicos, tesis y documentos técnicos) para fundamentar el desarrollo del sistema automatizado de Análisis de Precios Unitarios. A partir del análisis bibliográfico, se llegó a conclusiones prácticas que guiaron el diseño, implementación y evaluación del sistema.

Modalidad de Campo

Para el proyecto se utilizó la modalidad de investigación de campo de tipo descriptiva, ya que se busca recolectar y analizar información directamente de EMCO, negocio donde se optimizará el proceso de Análisis de Precios Unitarios. Se buscaba entender cuáles eran las herramientas de software utilizadas, las dificultades que estas

presentaban y los procesos que estaban conduciendo a retrasos en la generación de presupuestos. Gracias a este enfoque se pudo definir cómo se iba a validar la funcionalidad del sistema propuesto y delimitar el alcance de la solución para satisfacer las necesidades de EMCO.

2.2.2 Población y muestra

EMCO cuenta con 3 profesionales que trabajan en la ejecución de análisis de precios unitarios. Por ello se optó por ampliar la muestra con profesionales del GAD Municipalidad de Ambato, quienes evaluaron la usabilidad del sistema. Se contó con 4 ingenieros civiles del departamento de Planificación, en un rango de edades entre 32 y 50 años. Adicionalmente 4 ingenieros civiles del departamento de Obras Públicas entre los 27 y 35 años de edad. Finalmente, 2 ingenieros que trabajan en EMCO y el dueño. Los 11 profesionales que formaron parte de la muestra cuentan con entre 3 y 20 años de experiencia en su área y utilizan o han utilizado con anterioridad al menos un programa de análisis de precios unitarios.

2.2.3 Recolección de información

Se describen a continuación las técnicas que se utilizó en el proyecto

Entrevista Técnica

La entrevista técnica de carácter cualitativo se aplicó para entender las necesidades de EMCO, así como las expectativas respecto al sistema. Esta es la técnica de mayor importancia para el desarrollo del producto, ya que a partir de la información obtenida se levantaron los requerimientos.

Encuesta

La encuesta se trata de una técnica que permite obtener datos sobre opiniones, comportamientos o características específicas. Este instrumento es clave en la obtención de retroalimentación respecto a la efectividad del sistema desarrollado, debido al carácter cuantitativo de la evaluación. Esto quiere decir que, a diferencia de

la entrevista, la encuesta permite recolectar información para medir un parámetro. En este caso, la encuesta permitió medir la usabilidad del sistema.

Observación Directa

Se destaca la observación directa por su capacidad de proporcionar información de forma inmediata. En el caso de la investigación, se aplicó esta técnica a través de la observación de presupuestos modelo, formatos utilizados, y demás particularidades que pudieran pasarse por alto en la entrevista y que luego fueron esclarecidas en posteriores reuniones con el dueño de EMCO.

2.2.4 Procesamiento y análisis de datos

A través de la entrevista realizada al dueño de EMCO, se determinaron los requerimientos funcionales y no funcionales del sistema. El principal foco de automatización es la integración de cubicación y especificaciones como parte integral de un rubro. Además, se han de tener en cuenta la cantidad de decimales que se utiliza en los cálculos, funcionalidades de exportación, los formatos de exportación y finalmente, la posibilidad de incluir una base de Equipos, Mano de Obra, Materiales y una base de rubros. El conjunto de preguntas y respuestas se encuentra en el Anexo A.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

Como breve recordatorio de lo mencionado en el Capítulo I en la sección de Fundamentación teórica, la arquitectura a utilizar es el modelo cliente servidor. Como gestor de base de datos está MySQL, Node.js como herramienta para el desarrollo del backend y React para el frontend. La programación del sistema se hizo basándose en los principios de la metodología RAD y la evaluación de usabilidad se llevó a cabo con el test SUS.

3.1 Planificación

Se presenta los requerimientos obtenidos fruto de la entrevista y reuniones llevadas a cabo con el dueño de EMCO. Estos reflejan las necesidades identificadas, así como las expectativas para el proyecto.

3.1.1 Requerimientos funcionales

La Tabla 9 detalla los requerimientos funcionales, que son las especificaciones que describen lo que el sistema debe hacer.

Tabla 9. Requerimientos Funcionales

Código	Requerimiento	Descripción	Prioridad
RF01	Autenticación de Usuario	El sistema debe permitir a los usuarios autenticarse mediante credenciales válidas (correo y contraseña) para acceder al sistema.	Media
RF02	Registro de Usuario	El sistema debe permitir a los usuarios nuevos registrarse proporcionando el nombre, apellido, correo y contraseña.	Media
RF03	Gestión de Equipos	El sistema debe permitir registrar, consultar, modificar, eliminar, filtrar y categorizar información de equipos.	Alta
RF04	Gestión de Materiales	El sistema debe permitir registrar, consultar, modificar, eliminar, filtrar y categorizar información de materiales.	Alta
RF05	Gestión de Mano de Obra	El sistema debe permitir registrar, consultar, modificar, eliminar, filtrar y categorizar información de mano de obra.	Alta
RF06	Gestión de Rubros	El sistema debe permitir crear, consultar, modificar, eliminar, filtrar y categorizar rubros; esto implica la creación de equipos, materiales, mano de obra y especificaciones asociadas.	Alta

Código	Requerimiento	Descripción	Prioridad
RF07	Gestión de Especificaciones Técnicas	El sistema debe permitir crear, consultar, modificar, eliminar especificaciones técnicas. También debe permitir asociar dichas especificaciones a los rubros.	Alta
RF08	Gestión de Proyectos	El sistema debe permitir crear, consultar, modificar, duplicar y eliminar proyectos, así como reutilizar rubros de otros proyectos.	Alta
RF09	Gestión de Indirectos	El sistema debe permitir definir y modificar los valores de costos indirectos (utilidad y costos fijos) de un proyecto, expresados como porcentajes.	Alta
RF10	Generación de Presupuesto	El sistema debe mostrar el listado de rubros del presupuesto con su nombre, unidad, cantidad, precio unitario y total por rubro, así como el presupuesto total del proyecto.	Alta
RF11	Gestión de Rubros del Presupuesto	El sistema debe permitir agregar, consultar, modificar, eliminar, reordenar y categorizar los rubros dentro de un presupuesto, incluyendo la opción de cargar rubros desde la base o almacenarlos para uso futuro.	Alta
RF12	Gestión de Ítems de Cubicación	El sistema debe permitir agregar, consultar, modificar, eliminar, filtrar y categorizar ítems de cubicación.	Alta
RF13	Carga de Componentes en APU	El sistema debe permitir buscar y agregar equipos, materiales y mano de obra al análisis de precios unitarios, mostrando el total parcial de cada componente.	Alta
RF14	Análisis de Precios Unitarios (APU)	El sistema debe calcular el precio unitario de cada rubro, manejando hasta cinco decimales en los cálculos y dos decimales en el precio final.	Alta
RF15	Especificaciones Técnicas de Rubro	El sistema debe permitir asociar una especificación técnica a un rubro, o redactarla.	Alta
RF16	Exportación de Presupuesto	El sistema debe permitir exportar la información del presupuesto en formatos Excel y PDF.	Alta

3.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales describen el cómo se deben realizar las actividades del sistema. La Tabla 10 muestra los requerimientos no funcionales encontrados.

Tabla 10. Requerimientos No Funcionales

Código	Requerimiento	Descripción	Prioridad
RNF01	Usabilidad	La interfaz de usuario debe ser intuitiva y fácil de usar. El usuario debe poder entender donde está ubicado. Retroalimentación del sistema ya sea en éxito o fracaso. Solicitar confirmación previo acciones como eliminación o sobreescritura.	Alta
RNF02	Seguridad	Se debe garantizar que solo un usuario autenticado pueda usar el sistema	Media

3.1.3 Plan de Entrega

Una vez definidos los requerimientos, se procedió a elaborar un plan de entrega o *timeboxing* según las prácticas de RAD, mismo que se muestra en la Tabla 11. Cabe mencionar que, se ha omitido el requerimiento no funcional correspondiente a la usabilidad, dado que se busca desarrollar desde un principio pensando en el usuario, así que este es un requerimiento que se ejecuta en cada uno de los demás.

Tabla 11. Plan de Entrega

Iteración	Código	Requerimiento	Tiempo estimado (días)
Iteración 1	RF01	Autenticación de Usuario	2
	RF02	Registro de Usuario	1
	RF03	Gestión de Equipos	5
	RF04	Gestión de Materiales	5
	RF05	Gestión de Mano de Obra	5
Iteración 2	RF06	Gestión de Rubros	6
	RF07	Gestión de Especificaciones Técnicas	3
	RF08	Gestión de Proyectos	4
	RF09	Gestión de Indirectos	2
Iteración 3	RF10	Generación de Presupuesto	4
	RF11	Gestión de Rubros del Presupuesto	6
	RF12	Gestión de Ítems de Cubicación	5
Iteración 4	RF13	Carga de Componentes en APU	4
	RF14	Análisis de Precios Unitarios (APU)	5
	RF15	Especificaciones Técnicas de Rubro	3
	RF16	Exportación de Presupuesto	5
	RNF02	Seguridad	2
Total			67

3.2 Diseño

3.2.1 Arquitectura

Dado que se escogió la arquitectura Cliente-Servidor, la Figura 1 ilustra la arquitectura junto con las tecnologías utilizadas.

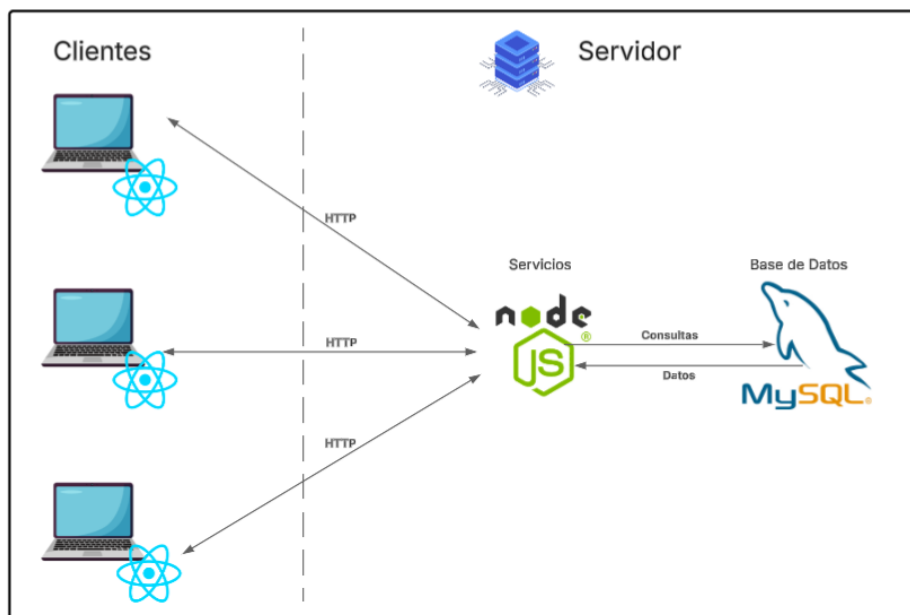


Figura 1. Arquitectura del Sistema

3.2.2 Base de datos

La Figura 2 muestra el diagrama desarrollado para representar la estructura lógica de la información. Los nombres tanto de las tablas como de los campos están dados en inglés; los términos utilizados corresponden en algunos casos a una adaptación en la traducción, cuando no existe palabra en inglés que signifique exactamente lo mismo. La Tabla 12 muestra los términos pertenecientes al análisis de precios unitarios en inglés y su traducción. A lo largo de esta sección se detallarán tablas o conjuntos de tablas que constituyen “bases de datos”. Esto no quiere decir que exista más de una base a la que se conecta el sistema si no que, en el argot de los análisis de precios unitarios, se hace referencia a “base de datos” para referirse a la información almacenada y disponible para consumo. Se usará esta variación del concepto de base de datos principalmente a lo largo de la sección de Prototipado y Desarrollo

Tabla 12. Traducción nombres tablas

Nombre Tabla	Traducción
Cubage	Cubicación
Equipment	Equipo
Labour	Mano de Obra
Line Item	Rubro
Material	Material
Specs	Especificaciones

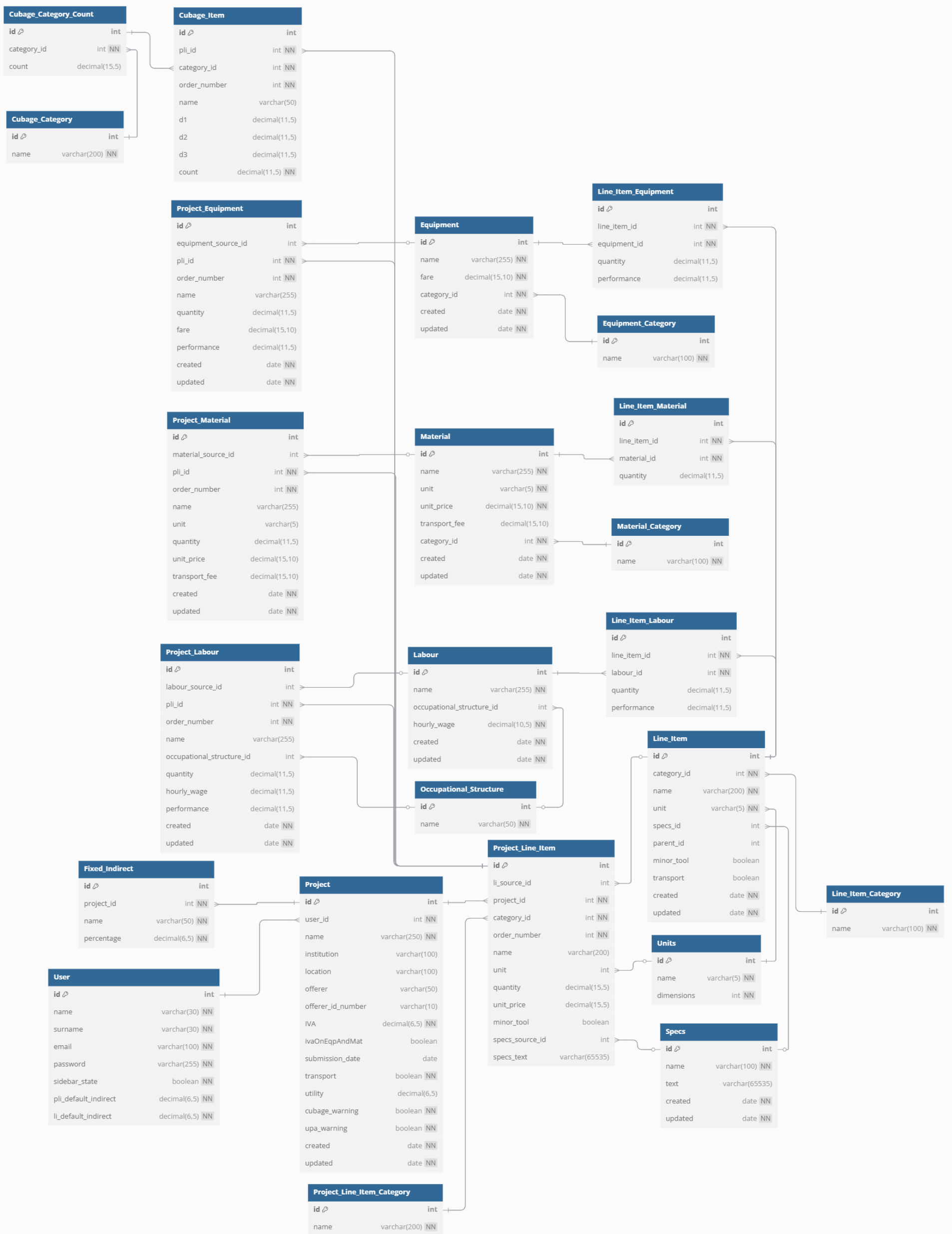


Figura 2. Diagrama de base de datos

La Figura 3 presenta de izquierda a derecha las tablas de Equipos (*Equipment*), Materiales (*Material*) y Mano de Obra (*Labour*) destinadas a almacenar la información que podrá ser cargada a los rubros del presupuesto. Todas ellas cuentan con un campo *created* y *updated*. El campo *created* se usó para control, mientras que el campo *updated* se utilizó para guardar la última fecha de modificación del registro, y posteriormente poder mostrar esta información al usuario. Con ello, este último puede decidir si los datos son aún válidos o requieren actualización. Al hablar de “bases” de equipos, materiales y mano de obra, se hace referencia a estas tablas. También se muestran las tablas de categoría/estructura ocupacional, y la relación uno a muchos que mantienen con las anteriores, respectivamente.

Los campos *name* de todas las tablas mostradas tienen la restricción *UNIQUE*, ya que no deben existir registros con el mismo nombre en ninguna de las bases ni en su categoría/estructura ocupacional.

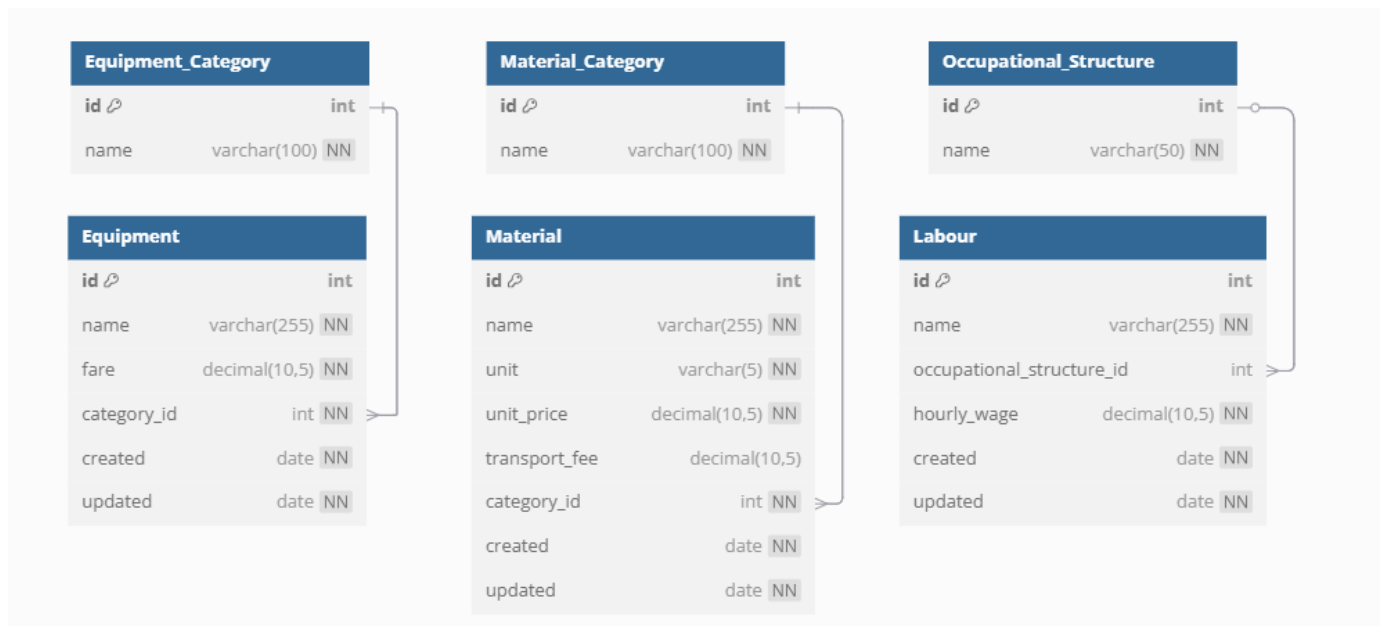


Figura 3. Bases de equipo, materiales y mano de obra

La estructura que corresponde al rubro del proyecto se muestra en la Figura 4. La tabla raíz de esta es *Project_Line_Item* que almacenará los rubros del proyecto, mismos que tendrán n materiales, equipos y mano de obra. Nótese que *Project_Material*, *Project_Labour*, y *Project_Equipment* contienen nuevamente los campos de las tablas *Equipment*, *Material* y *Labour*, en lugar de simplemente asumir estos

valores por referencia. Esto se pensó para evitar que una actualización de la información de estas afecte a los proyectos que hayan hecho uso de algún registro de las mismas. La información de estas tablas variará con el tiempo, pero los presupuestos deben conservar la información del momento en el que se realizó la presupuestación del proyecto.

Pese a esta duplicidad de información, las tablas *Project_Material*, *Project_Labour*, y *Project_Equipment* cuentan con un campo *table_source_id* que indica cuál es registro del que se tomó lo datos, esto para casos donde se quiera duplicar y actualizar el proyecto o usar rubros de un proyecto como base y actualizar su información. Lo mismo ocurre con las especificaciones, pero al existir una sola por rubro, constan como 2 campos en la tabla *Project_Line_Item*: *specs_source_id* para indicar el registro del que se tomaron las especificaciones y *specs_text* para almacenar la especificación.

En la tabla *Specs* el campo *text* y en la tabla *Project_Line_Item* el campo *specs_text* son de tipo de dato *MEDIUMTEXT*, el cual en MySQL tiene capacidad de almacenar hasta 16 MB de información, lo cual se consideró suficiente para almacenar las especificaciones.

Las tablas *Project_Line_Item*, *Project_Material*, *Project_Labour*, *Project_Equipment* y *Cubage_Item* cuentan con un campo *order_number* el cual almacena la posición en la que aparece un registro. Esto para permitir la reorganización de los rubros, equipos, materiales, mano de obra e ítems de cubicación y almacenar el nuevo orden.

La tabla *Project_Line_Item* cuenta con un campo *li_source_id*, el cual es una referencia hacia la tabla *Line_Item*. Esto se debe a que como se especificó en los requerimientos, el sistema debe permitir cargar rubros al presupuesto. Dada esta funcionalidad, es necesario poder validar que rubros ya se han utilizado y poder proceder adecuadamente en casos de almacenamiento duplicado. El campo *project_id* es una referencia hacia la tabla *Project*, para identificar a que proyecto corresponde un rubro dado. El campo *category_id*, es una referencia a la tabla de

categorías de rubros del proyecto *Project_Line_Item_Category*; el campo *name* de esta última tiene la restricción *UNIQUE*.

El campo *minor_tool* es un booleano el cual indica si el rubro tiene o no herramienta menor. Al ser la herramienta menor un valor calculado, no se almacena en la tabla *Project_Equipment* como un registro; pero se indica que un rubro debe realizar dicho cálculo como parte del análisis de precios unitarios. Los campos *quantity* y *unit_price* almacenan la cubicación y el APU respectivamente. Existen debido a que no siempre se va a calcular estos valores, si no que para algunos rubros se querrán ingresar manualmente.



Figura 4. Rubros del proyecto

En cuanto a los *Cubage_Item* y siguiendo con lo que se solicitó en los requerimientos, estos pueden agruparse en categorías, pero estas categorías tienen un conteo, o número de veces que aparecen. La Figura 5 muestra la estructura diseñada para cumplir con este requisito. Si se utilizaba una tabla *Cubage_Category* con el conteo incluido en ella, se terminaría con nombres de categorías repetidos hasta el infinito; ya que cada conteo sería diferente y por tanto, debería existir un nuevo registro para este. Para solucionar esta problemática, se pensó en una tabla intermedia *Cubage_Category_Count*. Esta tabla almacena una referencia hacia la tabla *Cubage_Category*, donde solo se guarda un *id* y *name*, evitando repetir n veces el nombre de la categoría. *Cubage_Category_Count* también cuenta con el conteo (*count*) y un *id*, mismo que será referenciado en *Cubage_Item*.

La tabla *Cubage_Item* también cuenta con un campo *count*, pero este es independiente del conteo de *Cubage_Category_Count*, y representa el número de veces que ese ítem aparece, que puede pertenecer a una categoría que aparece también n veces. Nótese que el campo *count* de ambas tablas es de tipo decimal, esto debido a que los valores no siempre serán enteros, según la información obtenida del dueño de EMCO. Finalmente, están los campos *d1*, *d2* y *d3* donde “d” es la contracción de *dimension*. Estos son los valores que tiene el ítem en cada una de sus dimensiones.

El campo *name* de la tabla *Cubage_Category* tiene la restricción *UNIQUE*.

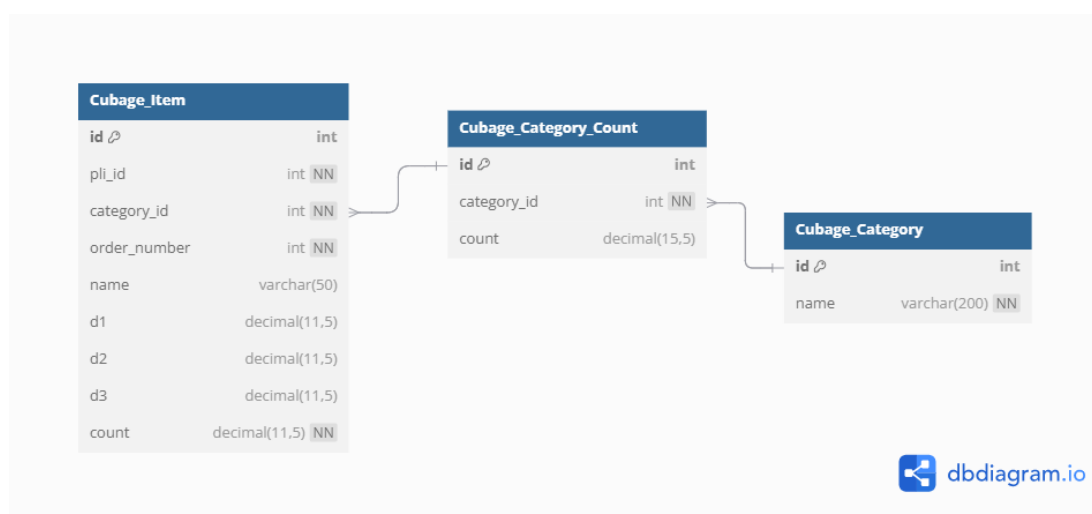


Figura 5. Tablas de cubicación

La estructura que almacena los rubros que se pueden cargar a un presupuesto se ilustra en la Figura 6. La raíz de esta es la tabla *Line_Item* donde se almacena la información básica de los rubros. Mantiene relación con las tablas *Line_Item_Equipment*, *Line_Item_Labour* y *Line_Item_Material*. En contraste con la estructura de los Rubros del Proyecto, estas si están normalizadas y simplemente tienen una referencia a la tabla *Equipment*, *Labour* y *Material*, respectivamente. Esto se hizo así, ya que la idea es que los equipos, mano de obra y materiales definidos en *Line_Item* estén siempre actualizados. Lo mismo ocurre con las especificaciones, donde solo almacena el campo *specs_id* que referencia a la tabla *Specs*. Al hablar de “base” de rubros se hace referencia a la estructura descrita.

La tabla *Line_Item* cuenta con el campo *category_id*, el cual es una referencia a la tabla *Line_Item_Category*, para agrupar los rubros por categorías. El campo *transport* de tipo booleano indica si el análisis de precios unitarios tiene o no transporte, ya que esto afecta al cálculo. El campo booleano *minor_tool* indica si el rubro tiene herramienta menor y, por tanto, si se debe aumentar esta al cálculo del APU. Los campos *created* y *updated* proporcionan información al usuario.

Finalmente, el campo *parent_id* es una referencia hacia cualquier rubro de la propia tabla. Este es un indicador para poder agrupar rubros relacionados. Los rubros pueden variar los componentes presentes (equipos, mano de obra o materiales) en un primer nivel, o variaciones en las cantidades o rendimiento de dichos componentes en un segundo nivel. Estas variaciones pueden suceder, pero el rubro que se analiza (ya en obra) no deja de ser el mismo. En respuesta a este comportamiento, el campo *parent_id* facilita la agrupación visual y lógica de rubros variantes. Por supuesto, en el sistema el nombre que se asigne al rubro variante debe ser diferente, para evitar ambigüedades y confusión para el usuario.

Los campos *name* de las tablas *Line_Item*, *Line_Item_Category* y *Specs* cuentan con la restricción *UNIQUE*.

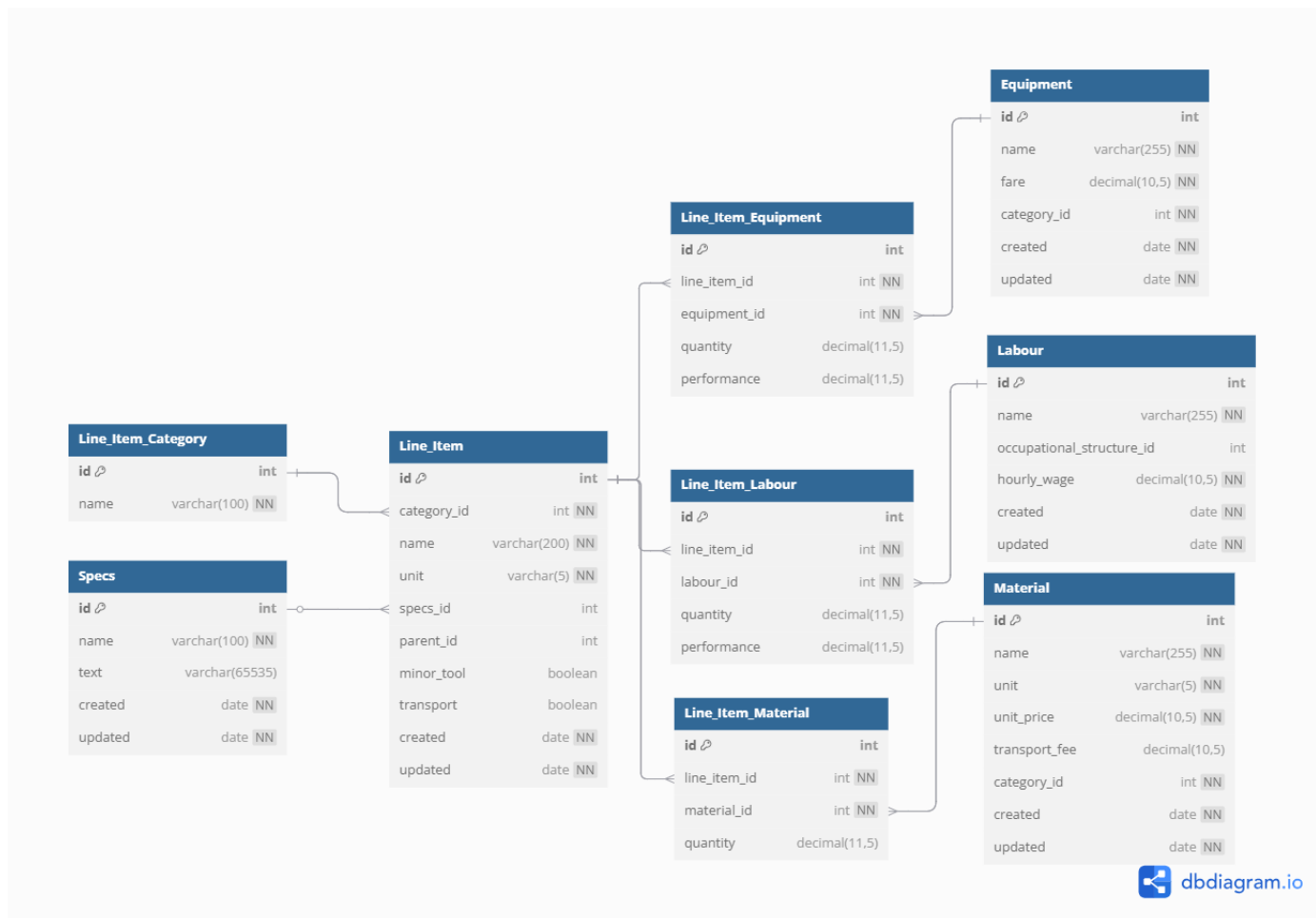


Figura 6. Base de rubros

La tabla *Project*, ilustrada en la Figura 7 cuenta con algunos campos a destacar. Dado que el proyecto puede o no tener IVA, se incluyó un campo de tipo decimal *IVA*. Se debe registrar este valor ya que, en Ecuador el IVA ha variado del 12% hasta diciembre del 2024 al 15% hasta 2025 y se espera que a finales de dicho año vuelva al 12%. Esto obliga a tener una forma de especificar qué porcentaje de IVA se está manejando al momento de presupuestar el proyecto. El campo booleano *ivaOnEqpAndMat* indica si el valor del IVA se aplica a los precios de materiales y equipos tanto del análisis de precios unitarios como de los seleccionados de sus respectivas bases. El campo booleano *transport* indica si se va a incluir el transporte al APU; esto afectará a todos los rubros.

Los campos booleanos *cubage_warning* y *upa_warning* identifican si en el proyecto se debe mostrar una notificación en caso de querer realizar el cálculo de la cubicación o del APU respectivamente, habiendo ingresado manualmente el campo de

cantidad o precio unitario de un rubro. En cada proyecto este valor es *TRUE* hasta que el usuario indique lo contrario. Finalmente, los campos *created* y *updated* son para visualización del usuario.

Project	
id 	int
user_id	int NN
name	varchar(100) NN
institution	varchar(100)
location	varchar(100)
offerer	varchar(50)
offerer_id_number	varchar(10)
IVA	decimal(6,5) NN
ivaOnEqpAndMat 	boolean
submission_date	date
transport	boolean NN
utility	decimal(6,5)
cubage_warning	boolean NN
upa_warning	boolean NN
created	date NN
updated	date NN

Figura 7. Tabla proyecto

La Figura 8 ilustra la tabla de unidades y sus relaciones. Esta se encuentra vinculada a la tabla *Project_Line_Item* y *Line_Item*. Su propósito es definir las dimensiones necesarias para determinar las columnas que deben mostrarse en la tabla de Cubicación, según la unidad del rubro correspondiente. Pese a que la tabla *Line_Item* no tiene relación directa con el cálculo de volúmenes, se conserva la relación con la unidad para garantizar coherencia cuando se agrega un rubro al presupuesto, el cual debe contar con una unidad válida.

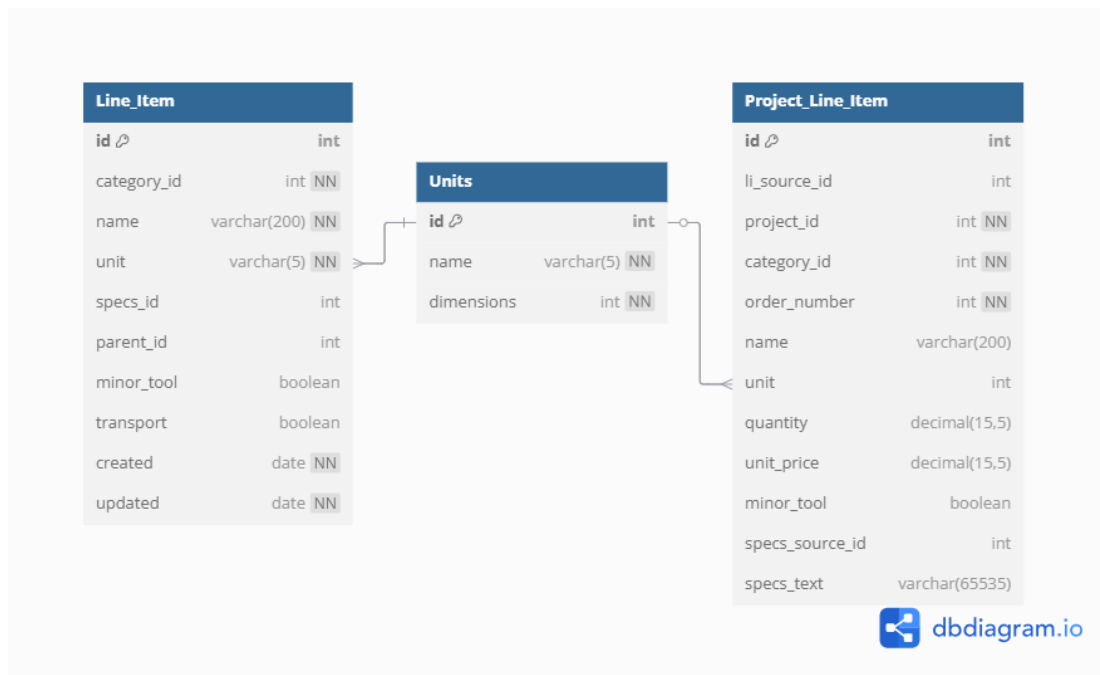


Figura 8. Tabla unidades

3.2.3 Diagramado

La Figura 9 muestra el diagrama de casos de uso propuesto en base a los requerimientos obtenidos. Hay que recordar que al hablar de “bases” no se habla de la base en informática, si no de una base en el contexto del análisis de precios unitarios

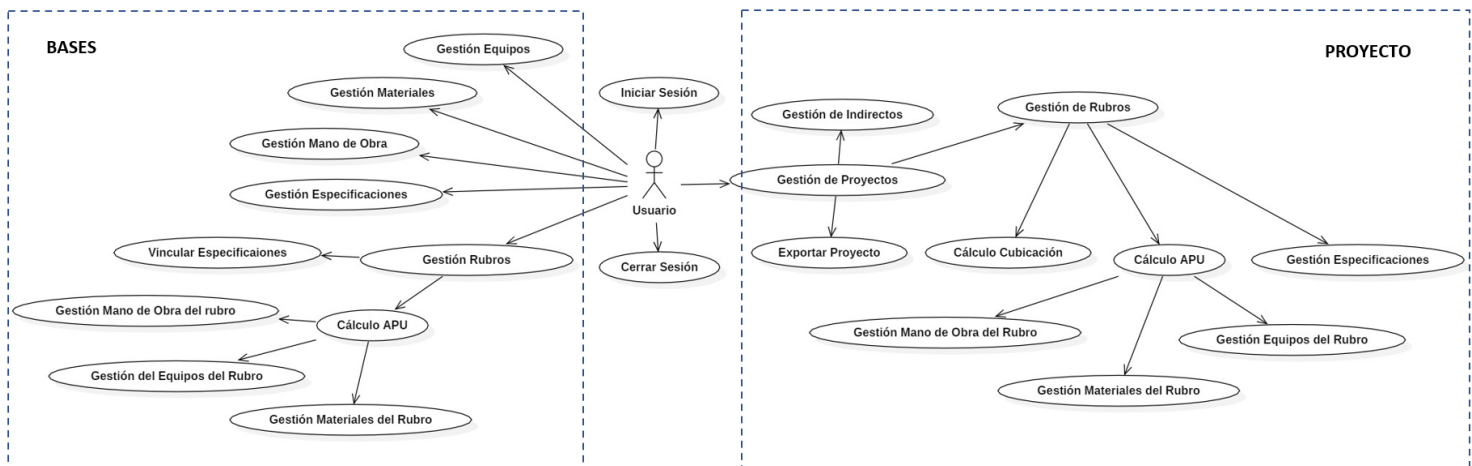


Figura 9. Diagrama de Casos de Uso

Para ilustrar algunas de las acciones que el usuario podrá realizar, se presentan a continuación diagramas de secuencia, que ilustran los actores y el orden en el que estos interactúan a lo largo del tiempo.

Para la gestión de cualquier componente se sigue un proceso similar, ya sea para equipos, mano de obra, materiales, rubros, rubros del proyecto, etc. Por ello se presentan diagramas que ilustran las interacciones y participantes en el CRUD básico.

Para la creación de un nuevo registro el usuario ingresa los datos requeridos los cuales son limpiados y formateados. El controlador recibe estos datos y los convierte a objeto JSON y envía al servicio correspondiente. Finalmente, el servicio lanza la consulta contra la base. La respuesta de la base se pasa hasta llegar al controlador quien ordena a la interfaz disparar retroalimentación visual acorde a la respuesta recibida. Este comportamiento se ilustra en la Figura 10. Se sigue un proceso similar para la edición de un registro, ilustrada en la Figura 11.

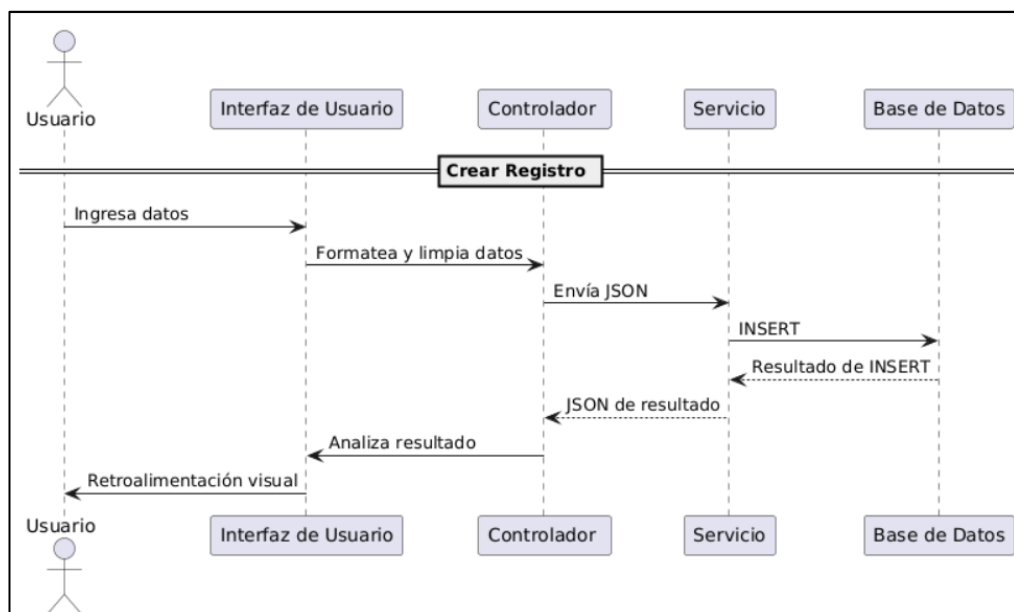


Figura 10. Secuencia creación registro

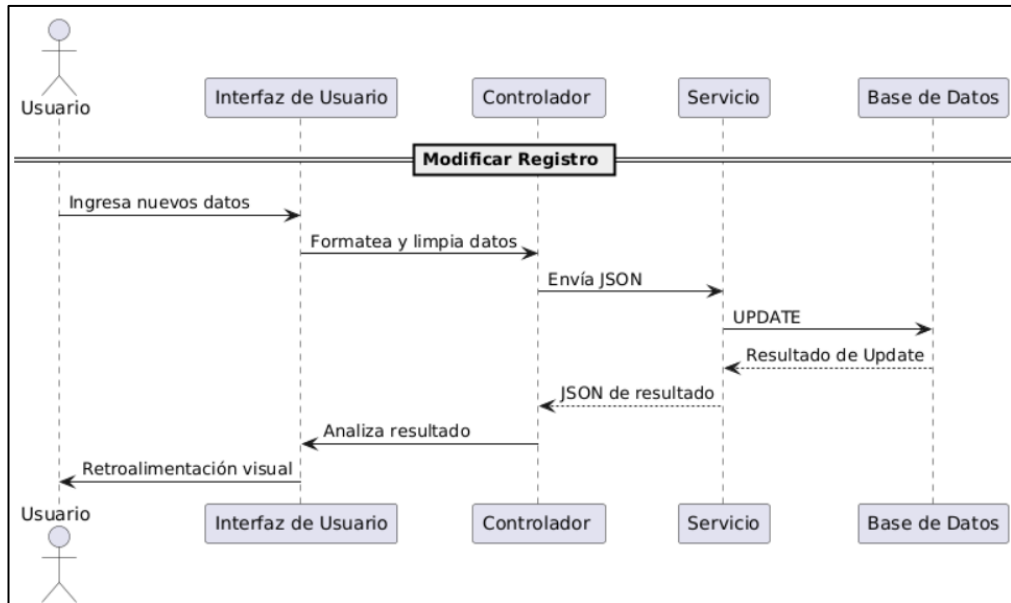


Figura 11. Secuencia editar registro

La eliminación sigue un proceso similar, pero la interfaz no limpia ni formatea datos ya que solo se envía el id del registro a eliminar. Así también ya no se envía un objeto JSON, dado que la solicitud con el verbo HTTP DELETE omite el cuerpo así sea enviado. Se ilustra el proceso en la Figura 12.

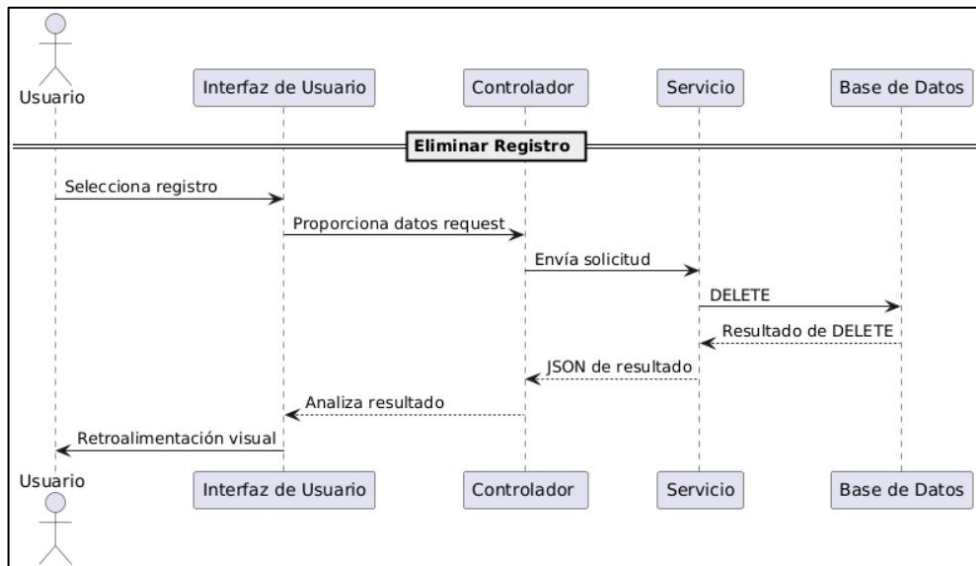


Figura 12. Secuencia eliminar registro

Para la consulta de registros, el usuario solicita el listado. Esto puede ser con un evento como la apertura de un modal, o el navegar a una página. La interfaz proporciona al

controlador los parámetros de selección y este envía la solicitud al servicio que consultará en la base. La base devuelve un array de objetos que el servicio convierte en JSON para ser enviados al controlador quien se los pasa a la interfaz, que finalmente muestra el listado. El proceso se ilustra en la Figura 13.

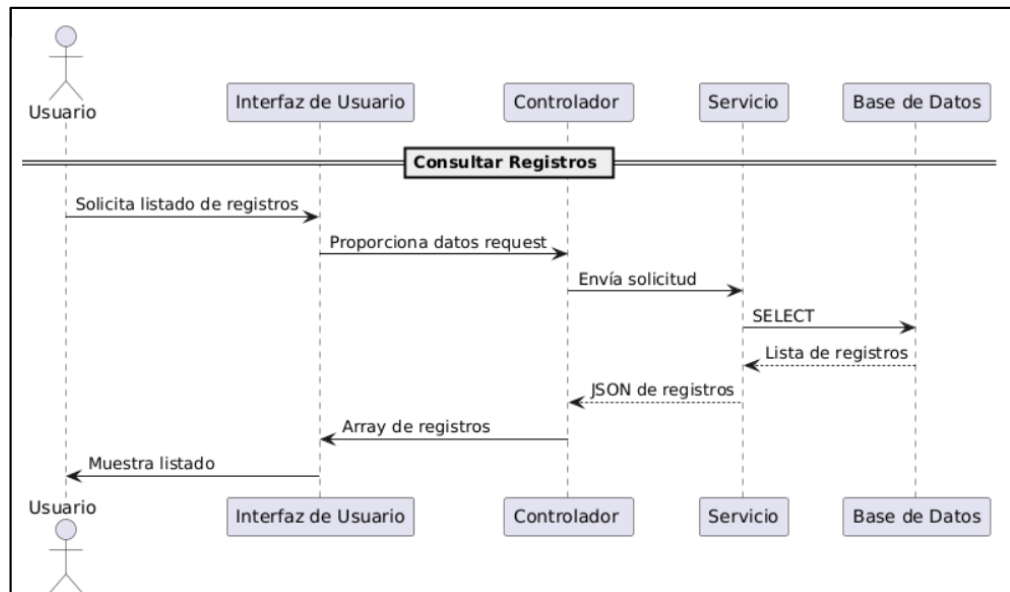


Figura 13. Secuencia consulta registros

La Figura 14 muestra la secuencia de filtrado. En este caso participa solo el usuario y la interfaz, ya que la información es filtrada en el frontend, sin necesidad de lanzar consultas contra la base.

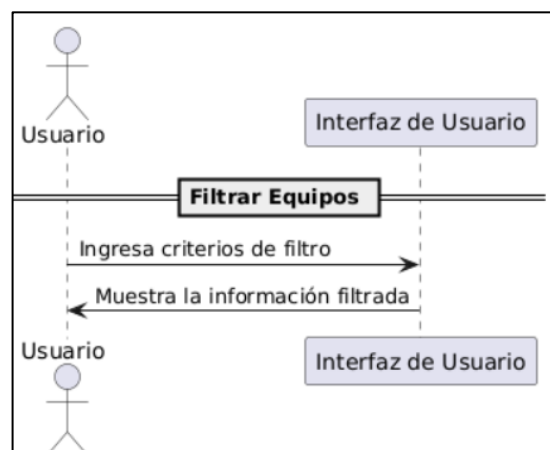


Figura 14. Secuencia filtrar registros

Para mostrar las especificaciones de la base, la secuencia inicia con el usuario que abre el modal. La interfaz llama al controlador adecuado, mismo que contacta con el servicio. Este último consulta en la base todos los nombres y fecha de creación de las especificaciones, mas no el contenido de la especificación. Esta información se muestra en el modal para el usuario, quien puede visualizar le contenido de alguna de estas especificaciones interactuando con la interfaz. En ese momento la interfaz envía el id de la especificación al controlador, que contacta con el servicio, mismo que consulta a la base. El texto de la especificación se pasa a la interfaz, la cual despliega la visualización del mismo. Este proceso se ilustra en la Figura 15.

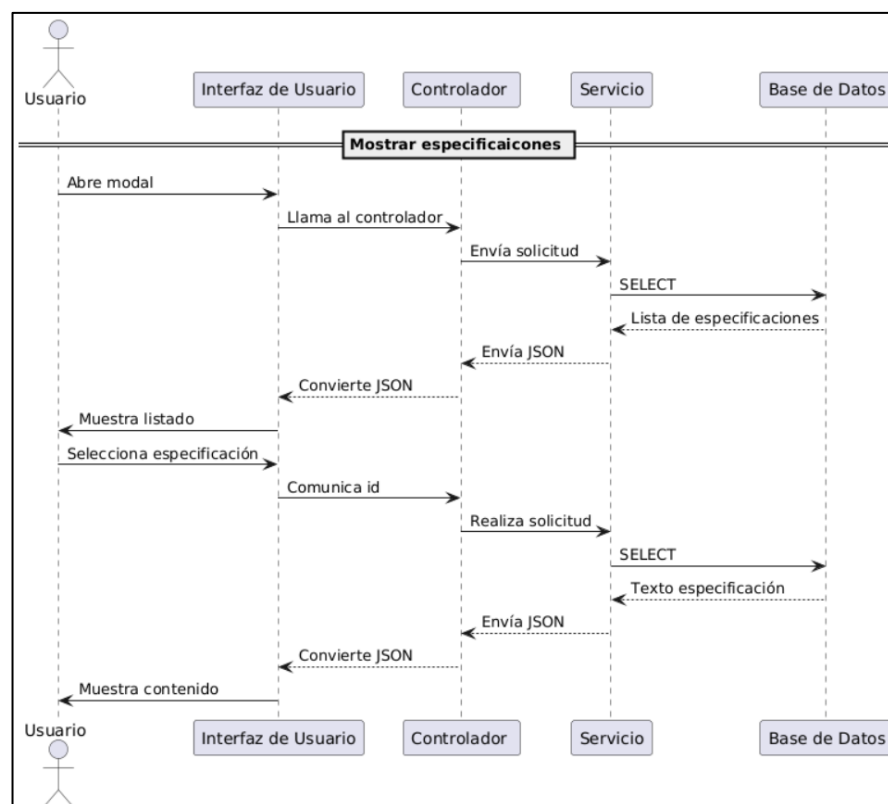


Figura 15. Secuencia visualizar especificaciones

Una de las funcionalidades que se solicitó fue poder tomar ciertos rubros de un proyecto y usarlos en uno nuevo. Este proceso se ha nombrado usar como base, y empieza cuando el usuario interactúa con el botón respectivo. Entonces la interfaz envía al controlador el id del proyecto correspondiente, quien a su vez realiza la solicitud al servicio y este se comunica con la base. La base devuelve el conjunto de rubros de ese proyecto, los cuales se visualizan en la interfaz. En este punto el usuario

selecciona los que desee, y continúa a la siguiente pantalla. La interfaz guarda los rubros seleccionados y los pasa a la siguiente pantalla. Aquí el usuario decide si actualizar la información del análisis de precios y especificaciones y pasa a la siguiente pantalla; la interfaz guarda su selección. Finalmente, la interfaz muestra el modal de inserción donde el proceso genérico de crear un registro se repite, solo que con todos los datos almacenados en pantallas previas. La Figura 16 ilustra lo previamente descrito.

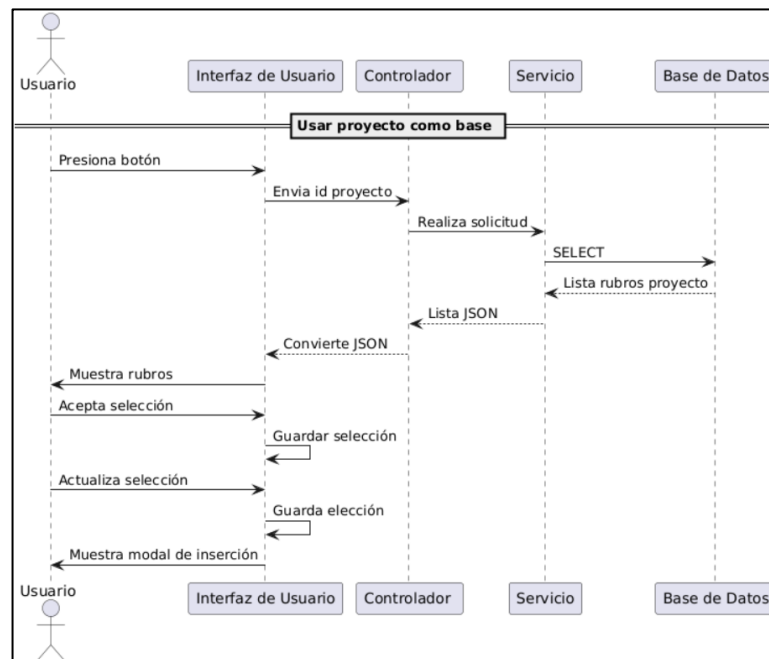


Figura 16. Secuencia usar proyecto como base

El reordenamiento de los componentes del análisis y de los ítems de cubicación requiere de un procesamiento recursivo entre el cliente y el servidor. Se inicia cuando el usuario termina la acción de arrastre para reordenamiento. El frontend envía al controlador un array de ids los cuales están reordenados según los cambios del usuario. El controlador convierte estos datos en objeto JSON y lo envía al servidor. Aquí el servidor recorre una a una las posiciones, realizando una consulta en la base por cada una. Cuando culmina el resultado de la última modificación se envía al cliente. El diagrama de la Figura 17 ilustra este comportamiento.

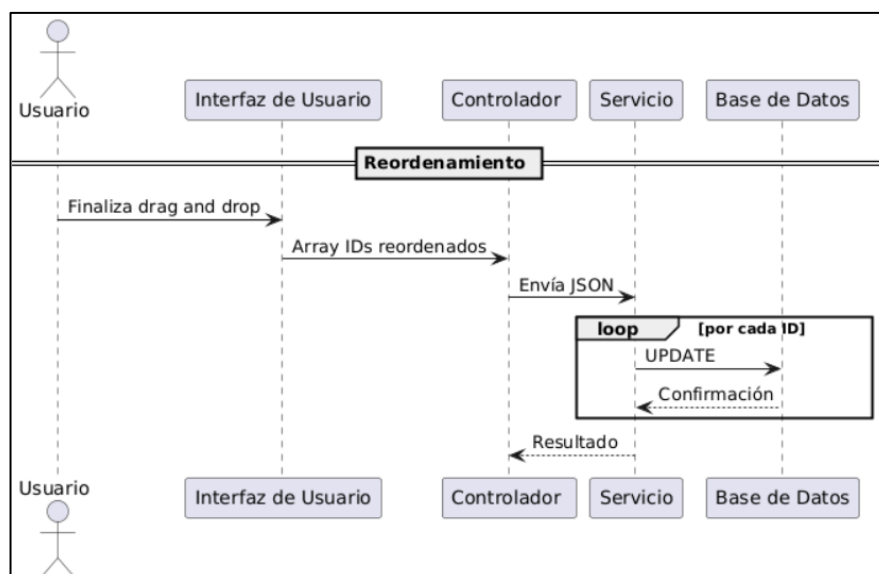


Figura 17. Secuencia reordenamiento

La adición de materiales, equipo o mano de obra desde la base al presupuesto inicia con el usuario aceptando la selección para ser agregada en el presupuesto. La interfaz envía la controlador un array de ids seleccionados, este a su vez se comunica con el servicio el cual usa el array para construir una sola consulta y enviarla a la base de datos. La respuesta vuelve a la interfaz, donde esta se encarga de actualizar el array de elementos agregables desde la base y actualiza la vista para mostrar los registros agregados. Finalmente muestra la notificación al usuario. La Figura 18 muestra este comportamiento

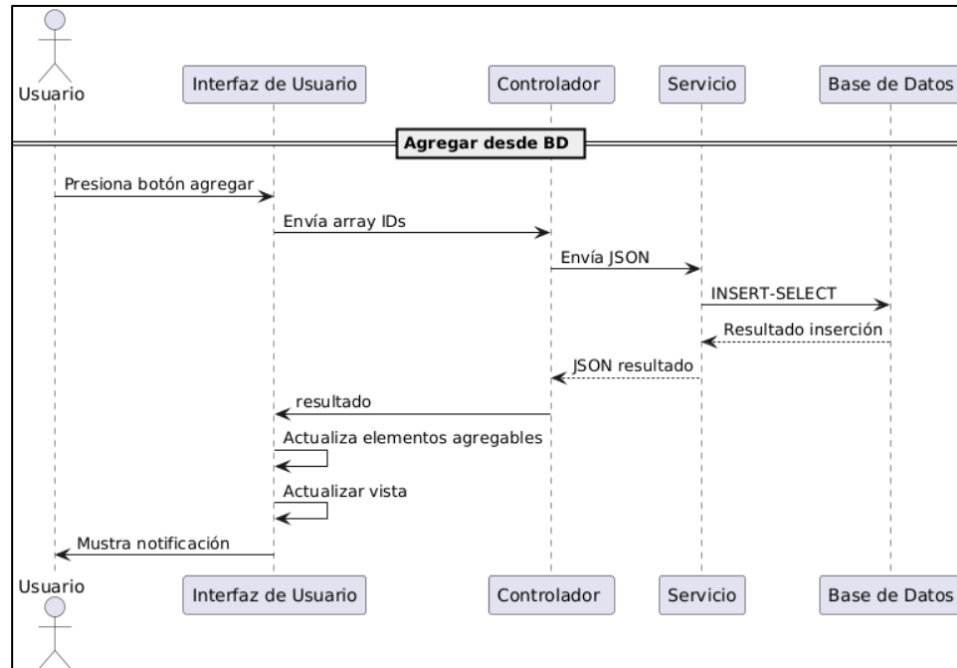


Figura 18. Secuencia agregar componente desde base

Las especificaciones pueden tener una imagen ilustrativa, para ello se debe asegurar que el editor de texto pueda manejar el almacenamiento de imágenes. El proceso inicia con el usuario que pega una imagen. La interfaz almacena esta imagen como un Blob en el navegador y muestra al usuario el éxito de su acción. A la par, el controlador envía el blob al servidor, donde se guarda la imagen asignando un nombre con el timestamp de cuando la imagen se envió seguido del timestamp de cuando llegó al servidor. Una vez guardada en el servidor, se forma una URL que apunta a la imagen y se envía el enlace de vuelta al cliente. El frontend entonces quita el blob de la imagen del navegador y la reemplaza por el recurso ahora disponible en el servidor, además de establecer la URL que identifica la imagen subida. El usuario no es consciente de este proceso. La Figura 19 ilustra lo previamente descrito.

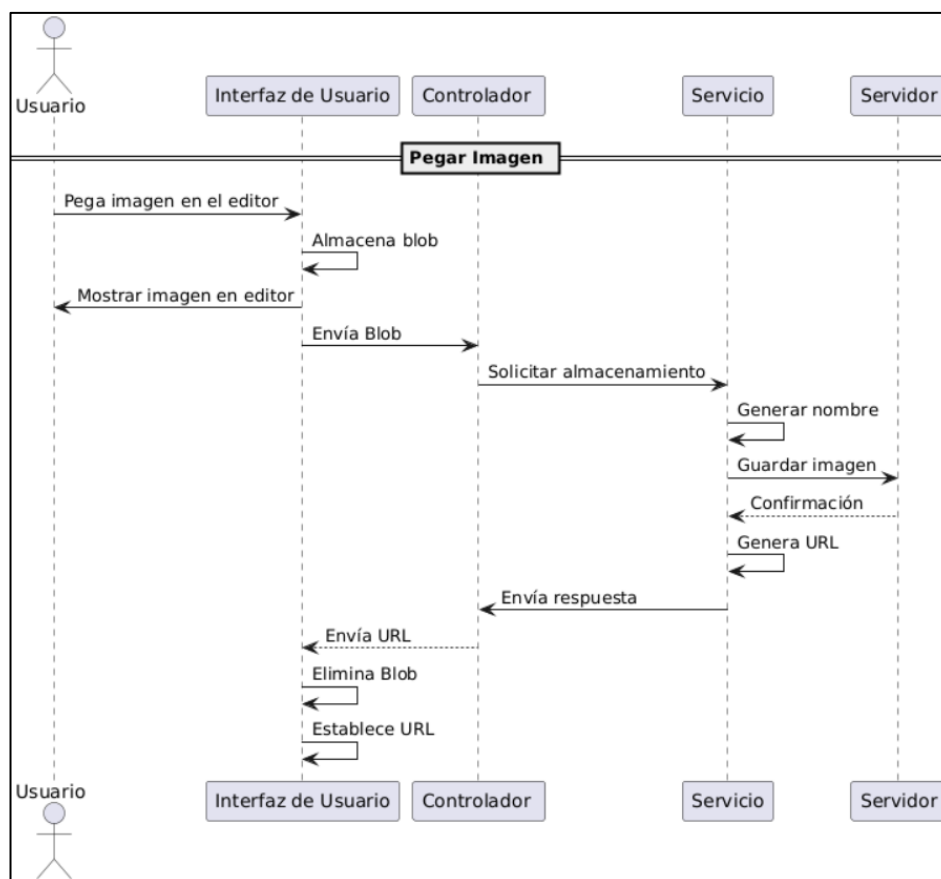


Figura 19. Secuencia pegar imagen

3.2.4 Prototipado

La Figura 20 ilustra el mapa de navegación propuesto para el sistema. Hay que mencionar que, la página de análisis de precios unitarios (APU) y especificaciones del presupuesto no son las mismas que las de la base de rubros.

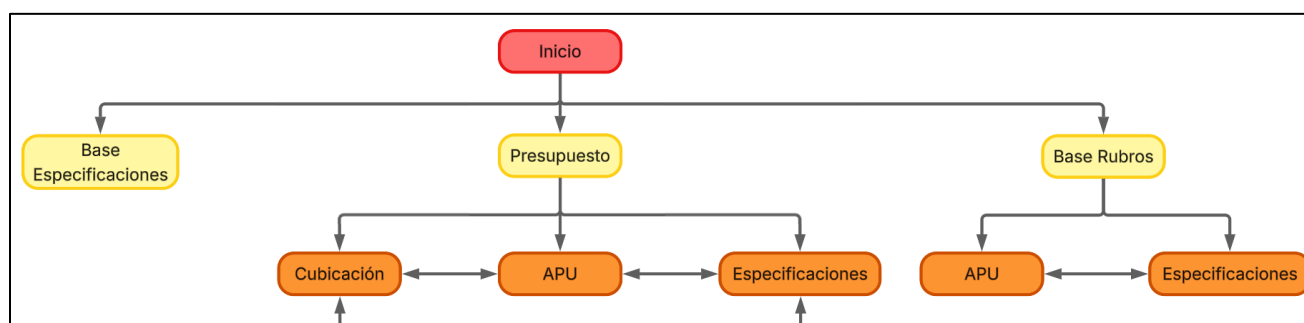


Figura 20. Mapa de navegación

La Figura 21 muestra la pantalla principal, accesible una vez se ha iniciado sesión. En el panel lateral izquierdo se encuentran las opciones para mostrar las bases de Equipos, Mano de obra, Materiales, Rubros y Especificaciones. En la parte central una tabla donde se mostrarán todos los proyectos y se podrá gestionarlos. Finalmente, una barra de búsqueda y un botón para cambiar el orden de la información.

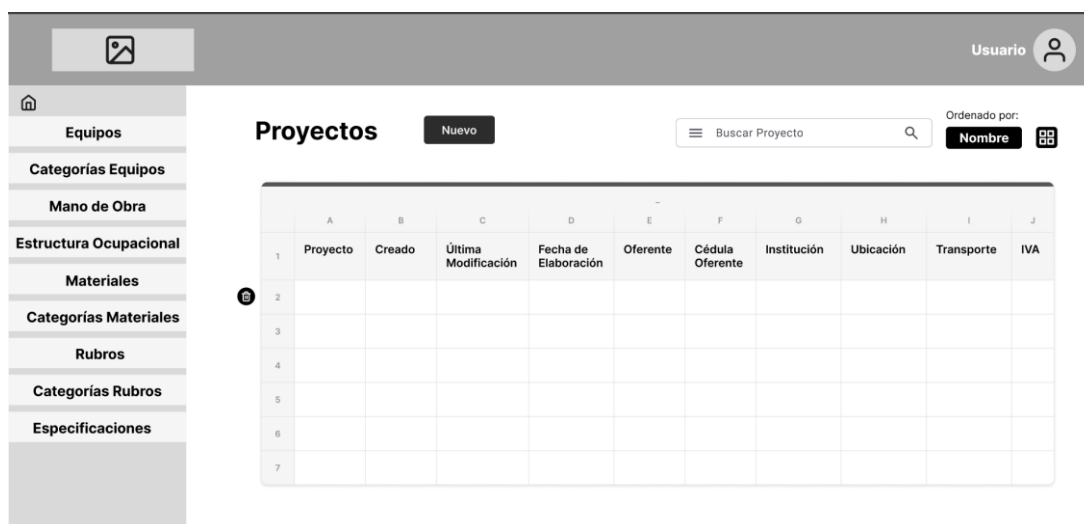


Figura 21. Prototipo inicio

La Figura 22 muestra el diseño propuesto para el modal de creación de un nuevo proyecto. El único campo obligatorio sin el cual no se puede crear un nuevo registro es el nombre; los demás campos son opcionales. De igual modo cuenta con el checkbox para indicar si se va a considerar transporte.

Nuevo Proyecto

Nombre* Institución

Ubicación

Oferente Cédula o RUC Oferente

Fecha de Elaboración

IVA (%)

☒ Transporte

Crear Proyecto

* Campos Obligatorios

Figura 22. Prototipo modal nuevo proyecto

La Figura 23 ilustra el prototipo del modal para la selección de rubros de un proyecto como base para crear un proyecto ya con datos. La idea es mostrar todos los rubros de un proyecto y que el usuario pueda seleccionar aquellos pertinentes al nuevo proyecto que esté realizando. Cuenta con una barra de búsqueda para optimizar tiempo.

Usar Proyecto como Base

	A	B	C
	Rubro	Unidad	Precio Unitario
☒	2		
☒	3		
☒	4		
☒	5		
☒	6		

Continuar

Figura 23. Prototipo selección rubros de proyecto

Las bases de equipos, materiales, y mano de obra tendrán una estructura similar a la mostrada en la Figura 24 cambiando solo las columnas acorde a los nombres de cada componente. Se trata de un modal el cuál muestra la tabla correspondiente con las respectivas operaciones de gestión, además de una barra para realizar búsquedas. El modal de la base de mano de obra se muestra en la Figura 25 y el de la base de materiales en la Figura 26.



Figura 24 es un prototipo de un modal de software titulado "Base de Equipos". El modal tiene un encabezado con el título y un botón de cierre (X). Debajo del título hay una barra de búsqueda con un icono de lupa y un botón "Nuevo". El cuerpo del modal contiene una tabla con tres columnas: "Equipo" (encabezada por "A"), "Tarifa" (encabezada por "B") y "Última Fecha de Modificación" (encabezada por "C"). La tabla tiene una fila de encabezados y seis filas de datos numeradas del 1 al 6. A la izquierda de la tabla hay un botón con un icono de documento.

	A	B	C
1	Equipo	Tarifa	Última Fecha de Modificación
2			
3			
4			
5			
6			

Figura 24. Prototipo base de equipos

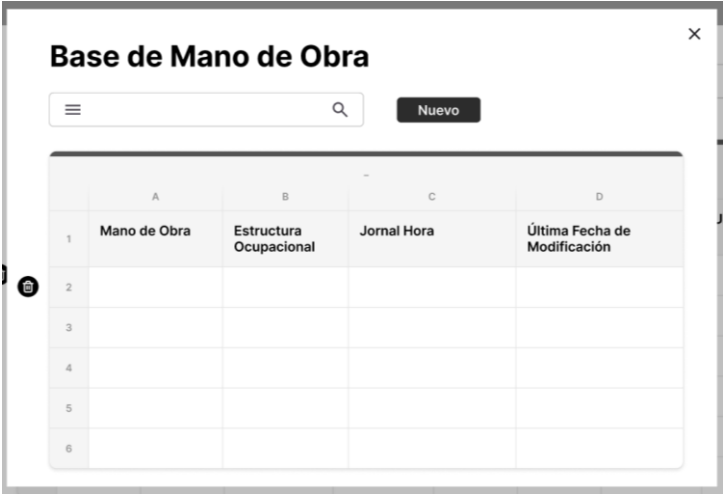


Figura 25 es un prototipo de un modal de software titulado "Base de Mano de Obra". El modal tiene un encabezado con el título y un botón de cierre (X). Debajo del título hay una barra de búsqueda con un icono de lupa y un botón "Nuevo". El cuerpo del modal contiene una tabla con cuatro columnas: "Mano de Obra" (encabezada por "A"), "Estructura Ocupacional" (encabezada por "B"), "Jornal Hora" (encabezada por "C") y "Última Fecha de Modificación" (encabezada por "D"). La tabla tiene una fila de encabezados y seis filas de datos numeradas del 1 al 6. A la izquierda de la tabla hay un botón con un icono de documento.

	A	B	C	D
1	Mano de Obra	Estructura Ocupacional	Jornal Hora	Última Fecha de Modificación
2				
3				
4				
5				
6				

Figura 25. Prototipo base de mano de obra

	A	B	C	D	E
	Material	Unidad	Precio Unitario	Tarifa Transporte	Última Fecha de Modificación
1					
2					
3					
4					
5					
6					

Figura 26. Prototipo base de materiales

La Figura 27 muestra el modal para la creación de un nuevo equipo en la base. Se presenta un campo adicional que es el IVA, esto debido a que los precios de Equipo pueden estar dados con IVA al momento de ingresar los datos. Se quiere almacenar en la base los precios sin el impuesto para posteriormente y de ser el caso, calcular el valor de la tarifa con IVA. El campo categoría permite seleccionar la categoría o de no existir, agregar una nueva. Esta funcionalidad es igual en todos los formularios que soliciten el ingreso de una categoría. El modal de agregar mano de obra se ilustra en la Figura 28, y el de agregar material en la Figura 29.

Nuevo Equipo

Equipo*

Nuevo Equipo

Tarifa* ? IVA (%)

0

Categoría*

☰

* Campos Obligatorios

Guardar

Figura 27. Prototipo agregar equipo a la base

Este es un prototipo de un modal de formulario titulado "Nueva Mano de Obra". El modal tiene un botón de cierre "X" en la esquina superior derecha. El formulario contiene tres campos de entrada: "Nombre*" con el valor "Nuevo Equipo", "Estructura Ocupacional" (vacío), y "Jornal hora*" (vacío). Los campos "Nombre*" y "Jornal hora*" tienen un asterisco que indica que son obligatorios. En la parte inferior derecha, hay un texto "* Campos Obligatorios" y un botón "Guardar".

Figura 28. Prototipo agregar mano de obra a la base

Este es un prototipo de un modal de formulario titulado "Nuevo Material". El modal tiene un botón de cierre "X" en la esquina superior derecha. El formulario contiene los siguientes campos: "Nombre*" (con el valor "Nuevo Equipo"), "Unidad*" (vacío), "Precio Unitario*" (vacío), "IVA (%)" (con el valor "0"), "Categoría*" (vacío) con un ícono de menú a la derecha, y "Tarifa de Transporte" (con el valor "0"). Los campos "Nombre*", "Unidad*", "Precio Unitario*" y "Categoría*" tienen un asterisco que indica que son obligatorios. En la parte inferior derecha, hay un texto "* Campos Obligatorios" y un botón "Guardar".

Figura 29. Prototipo agregar material a la base

El modal para la gestión de cualquier categoría es el mismo para todas las entidades que tengan una. La Figura 30 ilustra este modelo común, con la tabla donde se muestran los registros, la barra de búsqueda y el botón para agregar uno nuevo.

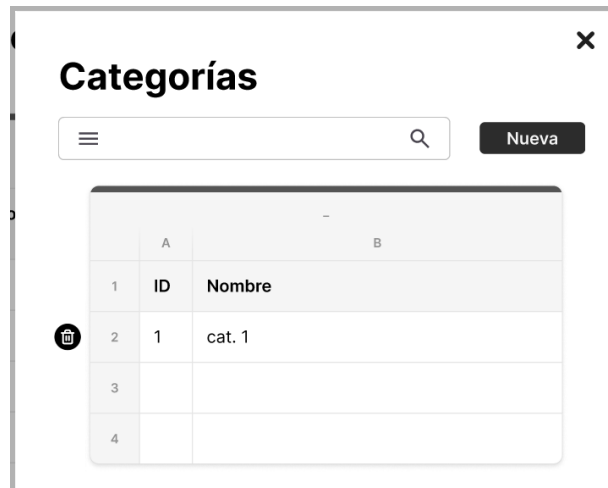


Figura 30. Prototipo gestión categorías

El presupuesto es el lugar donde se muestra el conjunto de rubros pertenecientes a un proyecto. En la interfaz se muestra en el panel izquierdo que hay dos opciones nuevas: presupuesto, que lleva a esta pantalla desde cualquier ventana del proyecto, y reportes, donde se puede exportar archivos Excel o PDF. En la zona principal, se encuentra la tabla que permite gestionar los rubros y la barra de búsqueda. A la derecha de la palabra presupuesto el botón para el modal de indirectos. Finalmente, a la derecha de la barra de búsqueda, el botón para buscar rubros en la base y cargarlos en el proyecto. La página de base de rubros presenta una vista similar, omitiendo la información que vincule los rubros a un proyecto como los indirectos, la cubicación y el agregar a la base.

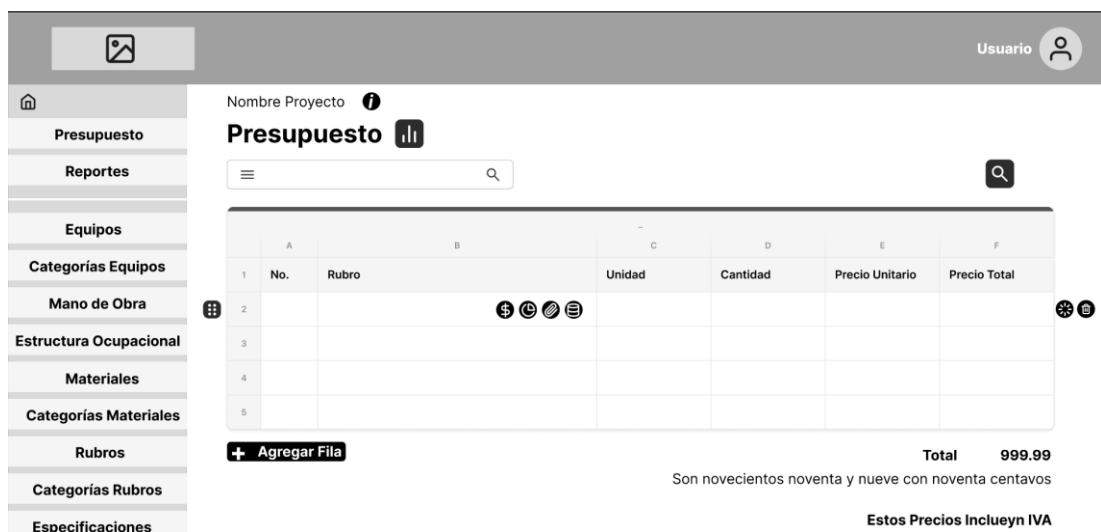


Figura 31. Prototipo presupuesto

El modal de indirectos se muestra en la Figura 32. Cuenta con una sección para editar la utilidad del proyecto, y otra con una tabla para gestionar los costos fijos. A la derecha del nombre del modal se encuentra un valor que indica el total porcentual de los indirectos, y es la suma de la utilidad y los costos fijos indirectos.

Indirectos[?]

25%

Utilidad %

Costos Fijos Indirectos

	A	B
1	Costo Fijo	Porcentaje
2	Indirecto 1	5%

+ Agregar Fila

Figura 32. Prototipo modal indirectos

Se muestra el modal para buscar rubros de la base correspondiente y cargarlos al proyecto en la Figura 33. Cuenta con un filtro por categoría, así como la barra de búsqueda. La tabla permite seleccionar varios rubros y agregarlos al presupuesto.

Base de Rubros[?]

Categoría

Cat. 1

	A	B	C
1	Rubro	Unidad	Precio Unitario
☒	2		
☒	3		
☒	4		
☒	5		
☒	6		

Agregar

Figura 33. Prototipo buscar en base rubros

Para finalizar con las acciones disponibles en el presupuesto, la Figura 34 ilustra el modelo propuesto para agregar un rubro a la base. Se pide la categoría, si el rubro es variante o no, y de serlo cuál es el rubro origen y el nuevo nombre que se le quiere dar.



El prototipo muestra una ventana modal con el título "Añadir Rubro a la Base" y un botón de cerrar (X) en la esquina superior derecha. El formulario contiene los siguientes campos:

- Categoría***: Un campo de texto vacío con un ícono de carpeta a la derecha.
- Variantes**: Una casilla de verificación marcada con el texto "Variante".
- Origen**: Un campo de texto con el valor "Rubro n".
- Nombre***: Un campo de texto con el valor "nombre variante".

Debajo de los campos, hay un texto explicativo: "Se guardará con el nombre: Nombre origen nombre variante al rubro como a las especificaciones en caso de que estas no existan". En la esquina inferior derecha hay un botón "Guardar".

Figura 34. Prototipo añadir rubro a la base

El análisis de precios unitarios y el aspecto que este tendrá se muestran en la Figura 35. La información que se presentará es el nombre del proyecto, el nombre del rubro y la unidad. Luego y hacia abajo se encuentran las tablas de equipos, materiales, mano de obra y transporte de ser el caso. Salvo la de transporte, todas tienen una apariencia similar a la mostrada en la figura: una lista desplegable para seleccionar la categoría a la que pertenecen, una barra de búsqueda por encima para mostrar los registros existentes en sus respectivas bases y un botón para agregar a la base de no existir lo que se busca. En la parte inferior derecha el subtotal y a la izquierda la opción para agregar más registros sin buscar en la base. La base de rubros muestra una pantalla idéntica para el APU, omitiendo la información que relaciona un rubro a un proyecto como el nombre de proyecto y la búsqueda en la base.

La página de cubicación consta de una tabla donde se pueden gestionar los ítems de cubicación y una barra de búsqueda como se ve en la Figura 37

Nombre Proyecto **Cubicación**

1 Rubro n Unidad u Cantidad 200

	A	B	C	D	E	F	G
	Descripción	a. Mayor	b. Menor	h. Altura	Volumen	Conteo	Subtotal
1							
2							
3							
4							
5							

+ Agregar Fila

Figura 37. Prototipo cubicación

Para la página de edición de especificaciones técnicas se propuso el prototipo mostrado en la Figura 38, con un área donde se puede editar a través de un editor de texto enriquecido (*Rich Text Editor*); aquí se carga el formato de especificaciones proporcionado por EMCO. Además, se puede buscar especificaciones de la base o enviar la especificación actual a la base. La página de edición de especificaciones para la base será idéntica.

Nombre Proyecto **Especificaciones Técnicas**

1. Rubro n u

a. Definición
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent vel libero fermentum, porta libero eu, consectetur nibh. Quisque pellentesque volutpat.

b. Especificaciones
c. Ensayos y Tolerancias
d. Referencias
e. Medición
f. Pago

Figura 38. Prototipo especificaciones técnicas

Finalmente, la página de exportación mostrada en la Figura 39. Este es accesible únicamente estando en un proyecto. Se permite al usuario seleccionar si quiere exportar todo, o solo ciertas partes y el formato de exportación.

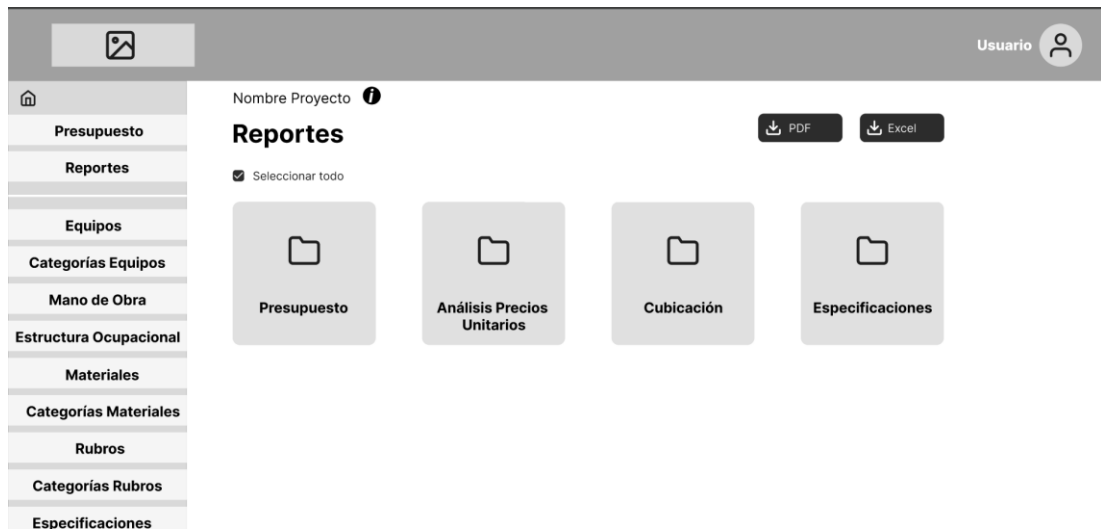


Figura 39. Prototipo Exportación

3.3 Desarrollo

Para el desarrollo del frontend se hizo uso de la librería de Material UI para utilizar componentes estilizados y con efectos visuales atractivos. Además, se usó Material React Table para las tablas por su amplio abanico de opciones y por usar Material UI, garantizando la coherencia visual. Para el manejo de rutas se usó React Router, para los formularios React Hook Form. Para los campos numéricos React Number Format, ya que permite validar los componentes TextField de React, y para los campos de fechas Dayjs, lo que permite tomar correctamente el timestamp proporcionado por los componentes date de Material. Para el editor de texto enriquecido se utilizó Froala, un editor de texto con versión community, lo que garantizó un uso legal y amplio, además de una documentación extensa y comprensible. Finalmente, para la comunicación se utilizó Axios, por su sintaxis simple y gestión de objetos JSON transparente al programador.

En cuanto al backend se utilizó la librería nodemon para tener una recarga automática del servidor en cambios del código, ahorrando así mucho tiempo en el desarrollo. Dotenv fue la librería utilizada para la gestión de variables de entorno que serían

accesibles desde cualquier punto del backend, y mysql2 la librería escogida para la gestión de la conexión con la base de datos. Se usó además el framework Express para poder exponer los endpoints de la API. En conjunto con este se usó la librería Cors, para evitar el error de Cross-Origin, además de gestionar más fácilmente las cabeceras de las peticiones, sin contar con que puede ser inyectado como middleware en Express. Se usó la librería jsonwebtoken para la creación de JSON Web Tokens en la seguridad de los endpoints y validación de rutas. Para la gestión de imágenes se usó Multer, una librería que permite gestionar archivos en el servidor. Finalmente, para la exportación se usó Archiver para la creación del archivo ZIP con más de un componente seleccionado, xlsx-populate para la creación de archivos Excel, PDF kit para la creación de los archivos que no son especificaciones y Puppeteer para las especificaciones, ya que este último permite generar PDF a partir de código HTML, que es el formato de información almacenado por Froala.

3.3.1 Aplicación Funcional

a. Inicio de sesión

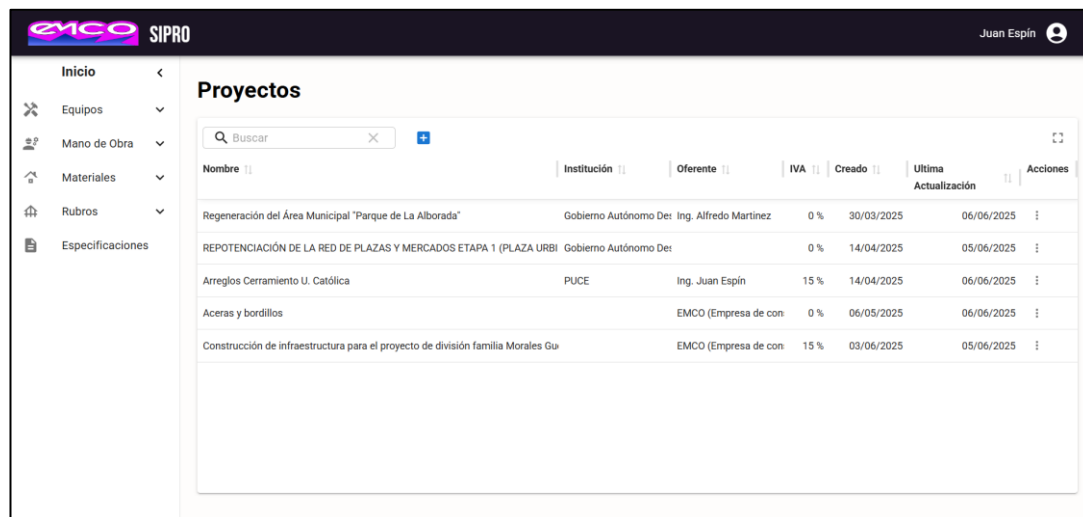
Si el usuario no se encuentra autenticado o no tiene cuenta, la única página a la que podrá acceder es al login. Se solicita la información básica que es el correo electrónico registrado, y la contraseña. La Figura 40 muestra cómo se ve el inicio de sesión.



Figura 40. Página de login

b. Inicio

Una vez autenticado el usuario, se accede a la página de inicio, desde donde se puede ir a todas las páginas del sistema. En la parte central se encuentra una tabla que muestra los proyectos asociados al usuario en cuestión. Esto se muestra en la Figura 41.



The screenshot shows the EMCO SIPRO system interface. On the left is a sidebar menu with options: Inicio, Equipos, Mano de Obra, Materiales, Rubros, and Especificaciones. The main area is titled 'Proyectos' and contains a table of projects. The table has columns for Nombre, Institución, Oferente, IVA, Creado, Última Actualización, and Acciones. The data rows are as follows:

Nombre	Institución	Oferente	IVA	Creado	Última Actualización	Acciones
Regeneración del Área Municipal "Parque de La Alborada"	Gobierno Autónomo De:	Ing. Alfredo Martinez	0 %	30/03/2025	06/06/2025	
REPOTENCIACIÓN DE LA RED DE PLAZAS Y MERCADOS ETAPA 1 (PLAZA URB)	Gobierno Autónomo De:		0 %	14/04/2025	05/06/2025	
Arreglos Cerramiento U. Católica	PUCE	Ing. Juan Espín	15 %	14/04/2025	06/06/2025	
Aceras y bordillos		EMCO (Empresa de con	0 %	06/05/2025	06/06/2025	
Construcción de infraestructura para el proyecto de división familia Morales Gu		EMCO (Empresa de con	15 %	03/06/2025	05/06/2025	

Figura 41. Página inicio

• Header

A la izquierda se encuentra el logo de EMCO y el nombre escogido para el sistema; al dar click en cualquiera de ellos se regresa a inicio. A la derecha se encuentra el nombre y apellido del usuario, así como el ícono de avatar. Al dar click en este se despliega un menú flotante con las acciones que puede realizar el usuario, ajenas al análisis de precios unitarios, como muestra la Figura 42.

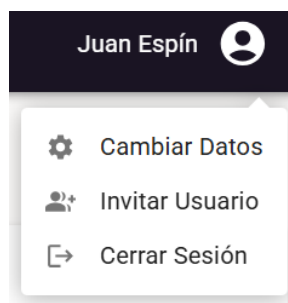
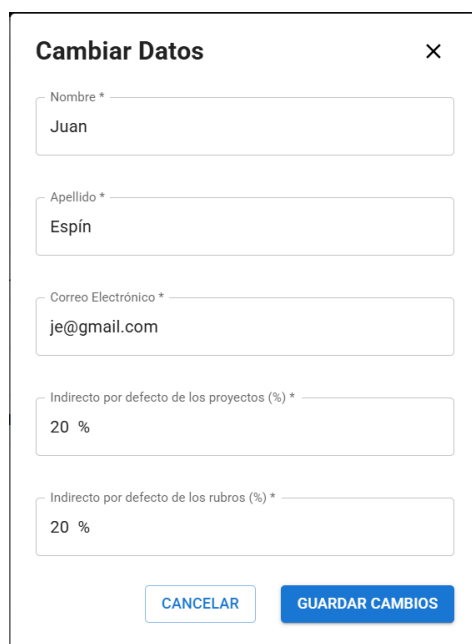


Figura 42. Opciones de usuario

Al seleccionar la opción “Cambiar Datos” se abre un modal para modificar la información del usuario. Además de los datos de identificación, se encuentra los porcentajes de utilidad por defecto al crear un proyecto y el porcentaje de indirectos a utilizar en la base de rubros; por defecto al crear un usuario ambos valores son de 20%. Esto se muestra en la Figura 42.



The image shows a modal window titled "Cambiar Datos" with a close button (X) in the top right corner. It contains five input fields with the following labels and values:

- Nombre *: Juan
- Apellido *: Espín
- Correo Electrónico *: je@gmail.com
- Indirecto por defecto de los proyectos (%) *: 20 %
- Indirecto por defecto de los rubros (%) *: 20 %

At the bottom of the modal, there are two buttons: "CANCELAR" (outlined) and "GUARDAR CAMBIOS" (solid blue).

Figura 43. Modal cambiar datos

Al dar click en la opción “Invitar Usuario” se direcciona a la página de creación de usuario. Con ello se logra que solo personas con acceso al sistema puedan “invitar” usuarios, y no cualquiera que acceda al login pueda crear una cuenta. Los campos que se solicitan son el nombre, apellido, correo y contraseña, como muestra la Figura 44.

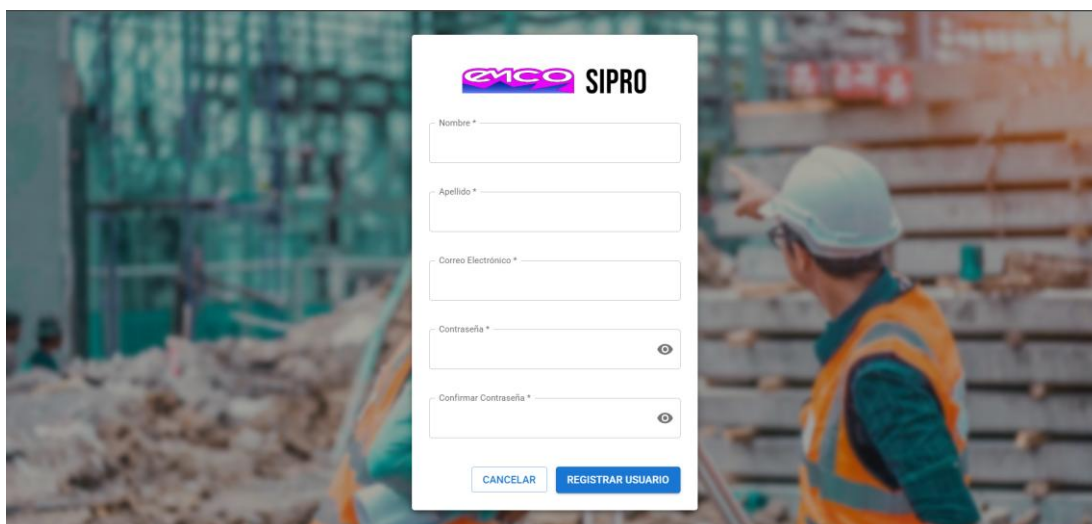


Figura 44. Crear usuario

- **Sidebar**

En la parte superior se coloca el nombre de la página en la que se encuentra actualmente. Debajo se encuentran los botones para acceder a cada una de las bases. Aquellas con el *chevron* hacia abajo contienen opciones anidadas, siendo estas las de Base de Datos y Categorías. A la derecha del nombre de la página se encuentra un *chevron* hacia la izquierda, al presionarlo se comprime el sidebar quedando visibles únicamente los íconos. Sin embargo, al hacer hover sobre estos se muestra un tooltip con el nombre correspondiente, como ilustra la Figura 45.

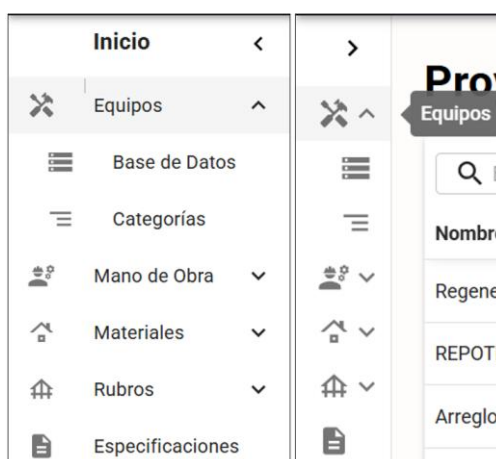


Figura 45. Sidebar

La base de equipos se muestra en la Figura 46. Se trata de un modal con una tabla para visualizar y editar los registros, siendo esta última funcionalidad accesible al dar click

en el campo deseado. Se puede observar que cada registro cuenta con las opciones de edición y eliminación, y están agrupados según su categoría (fila de color grisáceo). El botón de edición abre el modal de edición donde se puede cambiar la categoría a la que pertenece el registro.

Todas las tablas presentadas de aquí en adelante tendrán esta funcionalidad de visualización y edición de filas y el registro de tono más grisáceo que indica el inicio de un grupo de registros que pertenecen a una misma categoría.

Categoría	Equipo	Tarifa	Ultima Actualización	Acciones
Encofrados (1)	Encofrado metálico para bordillo tipo americano par 2.40m (alquiler día)	\$ 0.13043	04/06/2025	[Edit] [Delete]
Sin Categoría (7)	Amoladora	\$ 2.6087	04/06/2025	[Edit] [Delete]
	Compactador tipo plancha	\$ 4.34783	04/06/2025	[Edit] [Delete]
	Concretera 1 saco	\$ 4.34783	04/06/2025	[Edit] [Delete]
	Equipo topográfico (GPS de precisión-estación total)	\$ 17.3913	04/06/2025	[Edit] [Delete]
	Retroexcavadora	\$ 39.13043	03/06/2025	[Edit] [Delete]
	Vibrador - hormigón	\$ 2.6087	04/06/2025	[Edit] [Delete]
	Volqueta	\$ 26.08696	03/06/2025	[Edit] [Delete]

Figura 46. Modal base de equipos

Cabe mencionar que a partir de este momento todas las tablas que se muestren, salvo la de proyectos, contarán con las dos opciones de búsqueda de la Figura 47. Se trata de un doble filtrado, el primero por categoría y el segundo por coincidencia con lo que digite el usuario. El filtro de categorías es un input donde se puede escribir para filtrar las opciones o buscarla de una en una. Toda categoría tiene la opción “Sin Categoría”.

Categoría

Áridos
Clavos
Sin Categoría

Buscar

+

Material

Clavos 2" 2 1/2" 3" 3 1/2" 4"

Figura 47. Doble filtro

Los modales para crear/editar equipos se muestran en la Figura 48. Debajo de los campos de tarifa e IVA, se encuentra una nota que aclara que siempre se guardarán los valores sin IVA.

Nuevo Equipo

Nombre *

Tarifa *

IVA (%)

0 %

Se guardará la tarifa menos el porcentaje de IVA ingresado

Categoría

GUARDAR

Editar Equipo

Nombre *

Encofrado metálico para bordillo tipo americano

Tarifa *

\$ 0.13043

IVA (%)

0 %

Se guardará la tarifa menos el porcentaje de IVA ingresado

Categoría

Encofrados

GUARDAR CAMBIOS

Figura 48. Modal agregar - editar equipo

El input de categoría es un select que despliega las opciones, permite filtrar ingresando un nombre y si no hay coincidencias, permite crear una nueva categoría, como muestra la Figura 49. Se hará referencia a este como el componente de adición y filtrado.

Categoría

Áridos

Clavos

Sin Categoría

Categoría

cla

Clavos

Categoría

Metales

Añadir "Metales"

Figura 49. Componente de filtrado y adición categoría

En caso de seleccionar la opción de eliminación de registro, se mostrará un modal de confirmación de acción, como el de la Figura 50. Se solicitará confirmación para la eliminación en todo registro del sistema, y se mostrará un modal similar, cambiando el mensaje.

69

¿Desea eliminar el registro?

Se eliminará el equipo "Encofrado metálico para bordillo tipo americano par 2.40m (alquiler día)".
Esto afectará a todos los rubros que lo usen.

CANCELAR

ELIMINAR

Figura 50. Modal confirmación eliminación

La base de mano de obra se muestra en la Figura 51.

Mano de Obra				
Estructura Ocupacional Buscar				
Estructura Ocupacional	Mano de Obra	Tarifa por Hora	Ultima Actualización	Acciones
^ C1 (4)				
>	CHOFER: Volquetas (Estr. Oc. C1)	\$ 6.08	03/06/2025	✎ ✖
>	Maestro mayor en ejecución de obras civiles - C1	\$ 4.65	04/06/2025	✎ ✖
>	Operador de Retroexcavadora - C1	\$ 4.65	03/06/2025	✎ ✖
>	Topógrafo (En construcción - Estr. Oc. C1)	\$ 4.65	04/06/2025	✎ ✖
^ D2 (4)				
>	Albañil - D2	\$ 4.19	04/06/2025	✎ ✖
>	Ayudante de maquinaria - D2	\$ 4.26	04/06/2025	✎ ✖
>	Cadenero - D2	\$ 4.19	04/06/2025	✎ ✖
>	Operador de equipo liviano - D2	\$ 4.19	04/06/2025	✎ ✖
^ E2 (2)				
>	Ayudante albañil - E2	\$ 4.14	04/06/2025	✎ ✖
>	Peón - E2	\$ 4.14	03/06/2025	✎ ✖

Figura 51. Modal de mano de obra

El modal de crear/editar mano de obra se ilustra en la Figura 52. Este también cuenta con el componente para adición y filtrado de categorías, bajo el nombre de “Estructura ocupacional”.

Nuevo Mano de Obra

Nombre *

Tarifa Hora *

Estructura Ocupacional

GUARDAR

Editar Mano de Obra

Nombre *

Topógrafo (En construcción - Estr. Oc. C1)

Tarifa Hora *

\$ 4.65

Estructura Ocupacional

C1

GUARDAR CAMBIOS

Figura 52. Modal agregar - editar mano de obra

70

La Figura 53 muestra el modal de base de materiales.

Categoría	Material	Unidad	Precio Unitario	Tarifa de Transporte	Última Actualización	Acciones
Clavos (1)	Clavos 2", 2 1/2", 3", 3 1/2", 4"	Kg	\$ 1.04348	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
Sin Categoría (14)	Aditivo antisol	Kg	\$ 6.4	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Aditivo plastificante reductor de agua	Kg	\$ 6.4	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Agua (Potable)	m3	\$ 0.86957	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Arena	m3	\$ 3	\$ 0	01/06/2025	[Icono de editar] [Icono de borrar]
	Arena Lavada (INC. Transporte a Sitio)	m3	\$ 10.43478	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Base clase 2 (Incluye transporte a sitio)	m3	\$ 8.69565	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Cemento portland Tipo 1	Kg	\$ 0.13913	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Disco metálico para corte de pavimentos	u	\$ 78.26087	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Estacas de Madera	u	\$ 0.17391	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]
	Junta de poliuretano autonivelante (TUBO)	u	\$ 14.36522	\$ 0	04/06/2025	[Icono de editar] [Icono de borrar]

Figura 53. Modal de materiales

La Figura 54 muestra el modal de crear/editar materiales el cual muestra la aclaración del guardado sin IVA, además de tener el componente de adición y filtrado de categorías.

Nuevo Material

Nombre *

Unidad *

Precio Unitario *

IVA (%)

Se guardará la tarifa menos el porcentaje de IVA ingresado

Tarifa de Transporte *

Categoría

GUARDAR

Editar Material

Nombre *

Unidad *

Precio Unitario *

IVA (%)

Se guardará la tarifa menos el porcentaje de IVA ingresado

Tarifa de Transporte *

Categoría

GUARDAR CAMBIOS

Figura 54. Modal de agregar - editar material

Finalmente, la Figura 55 muestra el modal para la gestión de categorías de materiales. Este es similar para todos los componentes que tengan categorías, cambiando únicamente el título del recuadro.

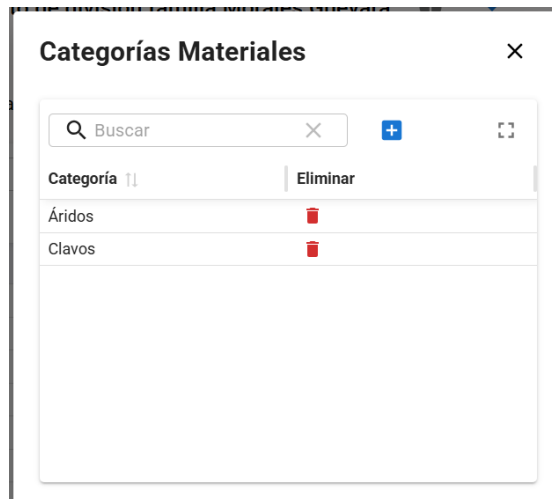


Figura 55. Modal de categorías

Las bases de rubros y especificaciones se abordarán más adelante, por presentar interfaces y funcionalidades similares a las de los rubros y especificaciones del proyecto, respectivamente.

- **Tabla de Proyectos**

La Figura 56 muestra que la tabla cuenta con un solo filtrado según lo que digite el usuario y al lado de la barra de búsqueda el botón para crear un nuevo proyecto. Al dar click en cualquiera de los registros se dirige hacia el presupuesto del proyecto. Cada uno cuenta con un botón en la parte derecha para mostrar más opciones del proyecto, como ilustra la Figura 57.

Q

Buscar

X

+

Nombre	Institución	Oferente	IVA	Creado	Ultima Actualización	Acciones
Regeneración del Área Municipal "Parque de La Alborada"	Gobierno Autónomo Desce	Ing. Alfredo Martinez	0 %	30/03/2025	06/06/2025	

Figura 56. Tabla de proyectos

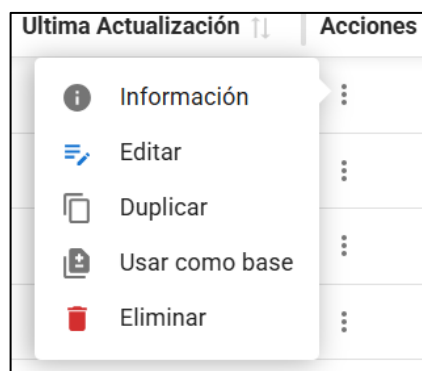


Figura 57. Popup acciones proyecto

Al seleccionar la opción “Información” se muestra un modal con los datos del proyecto. Esto se hizo ya que el espacio en la pantalla no fue suficiente para mostrar toda la información del registro en la tabla, y se dejaron solo los datos relevantes en ella y la posibilidad de mostrar los datos completos en el modal de la Figura 58.



Figura 58. Modal de información adicional

Al dar click en la opción “Editar”, se muestra el modal para modificar los datos del proyecto. La Figura 59 muestra el modal para crear/editar un proyecto; el único campo obligatorio para la edición/creación es el nombre, siendo todos los demás opcionales.

Nuevo Proyecto

Nombre *

Institución

Ubicación

Oferente

Cédula Oferente

Fecha de entrega

DD/MM/YYYY

IVA (%)

0 %

☐ Transporte

☐ IVA en Equipos y Materiales

GUARDAR

Editar Proyecto

Nombre *

Institución

Ubicación

Oferente

Cédula Oferente

Fecha de entrega

DD/MM/YYYY

IVA (%)

15 %

☐ Transporte

☒ IVA en Equipos y Materiales

GUARDAR CAMBIOS

Figura 59. Modal crear – editar proyecto

Al dar click en la opción “Duplicar” se muestra el modal de la Figura 60 para indicar si se desea actualizar los datos para el cálculo de precios unitarios, las especificaciones o ninguno. Al presionar el botón de continuar, se abre el modal de edición para ingresar los datos del nuevo proyecto. Esta opción se pensó para cuando el usuario necesite realizar un mismo proyecto constructivo o uno sumamente similar, o para cuando se pide presupuestar un proyecto previamente realizado, pero con los precios actualizados.

¿Desea Actualizar los datos?

¿Desea Actualizar la información del proyecto antes de duplicarlo?

No seleccione ninguna opción si no desea actualizar la información.

☐ Análisis de Precios Unitarios

☐ Especificaciones

CONTINUAR

Figura 60. Modal de actualización

Al dar click en la opción “Usar como Base”, se abre el modal de la Figura 61. Cuenta con dos tablas. La tabla de la izquierda muestra todos los rubros del proyecto que se usarán en el nuevo proyecto, la tabla derecha muestra los rubros que se han seleccionado. Cada registro en la tabla izquierda cuenta con un botón para pasarlo a

seleccionados, y cada registro de la tabla derecha tiene un botón para quitarlo de seleccionados. La tabla izquierda cuenta con el doble filtrado. Una vez se acepte la selección de rubros se presiona el botón de siguiente, el cual abre el modal de actualización. Esta opción se creó para cuando el usuario vaya a hacer un proyecto constructivo similar a uno previamente realizado, pero sin ocupar todos los rubros.

Seleccionar Rubros

Modal interface for selecting construction items (rubros). It features two side-by-side tables with filters and action buttons.

Left Table (Filtered):

Categoría	Nombre	Unidad	Cantidad	Precio	Acciones
Aceras y Bor					
>	Limpieza de terreno (a máquina)	m2	795.00	\$ 1.04	[Icono]
>	Excavación sin clasificar en tier	m3	295.80	\$ 3.77	[Icono]
>	Relleno compactado mecánico	m3	159.00	\$ 4.70	[Icono]
>	Relleno compactado mecánico	m3	159.00	\$ 16.70	[Icono]
>	Hormigón simple f'c=210 kg/cm²	m2	795.00	\$ 15.26	[Icono]
>	Hormigón ciclópeo, en Muros, 7	m3	9.80	\$ 113.85	[Icono]
Alcantarillad					
>	Limpieza de terreno (a máquina)	m2	250.00	\$ 1.04	[Icono]
>	Replanteo y nivelación para alci	Km	0.25	\$ 424.33	[Icono]
>	Excavación de zanja alcantarilla	m3	228.60	\$ 3.51	[Icono]

Right Table (Full List):

Categoría	Nombre	Unidad	Cantidad	Precio	Acciones
Aceras y Bor					
>	Hormigón simple f'c=210 kg/cm²	m3	51.76	\$ 137.06	[Icono]
>	Replanteo y nivelación de vías, 1	Km	0.25	\$ 605.56	[Icono]

Buttons: CANCELAR, SIGUIENTE

Figura 61. Modal usar como base

c. Presupuesto

Al dar click en cualquiera de los proyectos, se direccionará hacia la página de presupuesto. Esta cuenta en la parte superior con breadcrumbs para orientar al usuario. Debajo, se encuentran el nombre del proyecto y algunas opciones del mismo. Finalmente, la tabla con los rubros del presupuesto, como se ilustra en la Figura 62

Categoría	No.	Nombre	Unidad	Cantidad	Precio Unitario	Precio Total
Aceras y Bordil						\$ 25,837.62
	1	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	795.00	\$ 1.04	\$ 826.80
	2	Replanteo y nivelación de vías, equipo de precisión	Km	0.25	\$ 605.56	\$ 151.39
	3	Excavación sin clasificar en tierra a mano	m3	295.80	\$ 3.77	\$ 1,115.17
	4	Hormigón simple f'c=210 kg/cm2, concretera, en bordillo tipo americano, inc. E	m3	51.76	\$ 137.06	\$ 7,094.23
	5	Relleno compactado mecánico (mat. excavación, compactador mecánico planic	m3	159.00	\$ 4.70	\$ 747.30
	6	Relleno compactado mecánico (base clase 2, plancha)	m3	159.00	\$ 16.70	\$ 2,655.30
	7	Hormigón simple f'c=210 kg/cm2, en Acera, e=7cm, con concretera	m2	795.00	\$ 15.26	\$ 12,131.70
	8	Hormigón ciclópeo, en Muros, 70% H.S. f'c=210 kg/cm2, con concretera, 30% Pl	m3	9.80	\$ 113.85	\$ 1,115.73
Alcantarillado (3)						\$ 1,168.47
	9	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	250.00	\$ 1.04	\$ 260.00
	10	Replanteo y nivelación para alcantarillado, equipo de precisión	Km	0.25	\$ 424.33	\$ 106.08
	11	Excavación de zanja alcantarillado, máquina. H=0.00 - 2.75 m	m3	228.60	\$ 3.51	\$ 802.39

Figura 62. Página de presupuesto

- **Sidebar**

Aparecen 2 nuevas opciones como ilustra la Figura 63: “Ir al presupuesto”, la cual redirecciona al presupuesto desde cualquier punto, y la de “Proyecto” que contiene las opciones para el modal de categorías de rubros del proyecto y cubicación, así como la de Unidades. Estas opciones aparecen una vez se ingresa a un proyecto, más los registros que se mostrarán en cada una no son solo del proyecto, sino que son globales para todos los usuarios. Estas opciones no se muestran antes de ingresar a un proyecto, ya que se quería evitar confusiones entre las categorías de los rubros del proyecto y los rubros de la base.

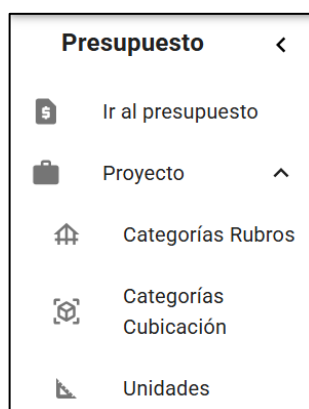


Figura 63. Opciones adicionales sidebar

Al seleccionar la opción de “Unidades” se abre el modal de la Figura 64 para la gestión de las mismas. Se debe proporcionar un nombre y un número de dimensiones. Esto se creó para poder asociar un número de dimensiones a las unidades de los rubros, funcionalidad que se usó en la cubicación.

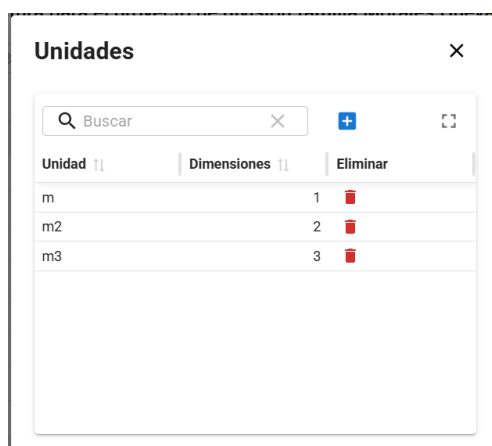


Figura 64. Modal unidades

- **Información del proyecto**

Como se ilustra en la Figura 65, a la derecha del nombre del proyecto se encuentran 3 botones. El primero de color gris muestra el modal con la información del proyecto. El segundo de color verde muestra el modal para gestión de indirectos ilustrado en la Figura 66. Finalmente, el botón azul que despliega las opciones de exportación del proyecto. Tanto los breadcrumbs como el nombre del proyecto y sus acciones estarán presentes en todas las páginas pertenecientes a un proyecto (Cubicación, APU y especificaciones.)

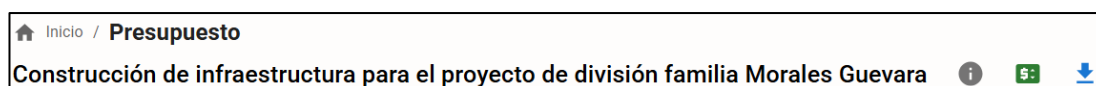


Figura 65. Acciones del proyecto

El modal de gestión de indirectos permite modificar la utilidad, así como el conjunto de valores de costos fijos. Todos los valores que se reciban aquí son porcentuales y en la parte superior, al lado del nombre del modal está una sumatoria del total de indirectos.

Indirectos 23.7 % [X]

Utilidad 20 %

Costos Fijos

Buscar [X] + [Fullscreen]

Costo Fijo	Porcentaje	Eliminar
Gastos administrativos	2.5 %	[X]
Movilización	1.2 %	[X]

Figura 66. Modal de indirectos

Al dar click en el botón de exportación se despliegan las partes del proyecto que se pueden exportar, así como los formatos (Excel y PDF); adicionalmente, se cambia el ícono de descarga por uno de cancelación. Si se escoge más de una opción, el archivo descargado será un ZIP con el nombre del proyecto. Si se escoge exportar las especificaciones, el botón de Excel se bloqueará, ya que estas solo pueden descargarse en formato PDF. Los Anexo B, Anexo C, Anexo D y Anexo E muestran los exportables PDF y los Anexo F, Anexo G y Anexo H los exportables Excel.

Construcción de infraestructura para el proyecto de división familia Morales Guevara [Info] [PDF] [Excel]

☒ Presupuesto ☒ Análisis de Precios Unitarios ☒ Cubicación ☐ Especificaciones

[PDF] [EXCEL]

Construcción de infraestructura para el proyecto de división familia Morales Guevara [Info] [PDF] [Excel]

☒ Presupuesto ☒ Análisis de Precios Unitarios ☒ Cubicación ☒ Especificaciones

[PDF] [EXCEL]

Figura 67. Exportación

- ***Tabla de rubros del presupuesto***

La tabla de la Figura 68 cuenta con el doble filtro, además de un botón de lupa a la derecha, este abre el modal para seleccionar rubros de la base. En cuanto a los rubros, estos se encuentran agrupados por categorías y cada una de estas muestra el subtotal que representan en el presupuesto. En la parte inferior derecha se puede ver el total del presupuesto, el IVA y cuanto es el total del proyecto. Cada registro cuenta con un botón en la parte izquierda, el cual permite arrastrar las filas para reordenarlas.

También se puede hacer este reordenamiento editando el campo “No.” O desde el modal de edición. Si se arrastra un registro a una categoría diferente se reordenan las categorías; es decir, todos los rubros bajo esta categoría pasan a estar en donde se encontraba la anterior.

[illegible]

Figura 68. Presupuesto

Al dar click en el botón de lupa se muestra el modal de la Figura 69. Este tiene la estructura del modal para usar un proyecto como base. Los rubros que aquí se muestran son aquellos de la base de rubros que estén completos (APU y especificaciones). Una vez se acepta la selección se agregan al proyecto los rubros, asignados a la última categoría encontrada.

Buscar Rubros

Categoría

Q Buscar

X

⌵ Categoría

Nombre

Uni... | Preci... | Acciones

⌵ Replanteo y n

>

Replanteo y nivelación de vías, equipo de precisió Km

\$ 557.83

⊕

>

Replanteo y nivelación para alcantarillado, equipo Km

\$ 397.13

⊕

⌵ Categoría

Nombre

Uni... | Preci... | Acciones

⌵ Sin Categoría

>

Limpieza de terreno (a máquina), eliminación cap m2

\$ 1.02

⊕

CANCELAR

AGREGAR SELECCIONADOS

Figura 69. Modal agregar rubro de la base

Para crear un nuevo rubro en el presupuesto se presiona el botón de “más” en la tabla, lo que crea una fila en blanco como muestra la Figura 70. Se carga automáticamente el número de rubro, pero este es editable, y el valor que se ingrese será la posición en donde se coloque el nuevo registro. El campo de cantidad corresponde al valor obtenido por la cubicación y el de Precio Unitario al APU. Sus valores pueden ser cargados manualmente o calculados. Si no se cambia el número, la categoría asignada será la misma que la del último rubro ingresado.

9	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desal	m2	250.00	\$ 1.04
10	Replanteo y nivelación para alcantarillado, equipo de precisión	Km	0.25	\$ 424.33
11	Excavación de zanja alcantarillado, máquina, H=0.00 - 2.75 m	m3	228.60	\$ 3.51
12	Nombre	Unidad	Cantidad	\$ Precio Unitar

Figura 70. Creación nuevo rubro

Cada registro de la tabla tiene un botón a la derecha para mostrar las opciones del rubro, como ilustra la Figura 71.

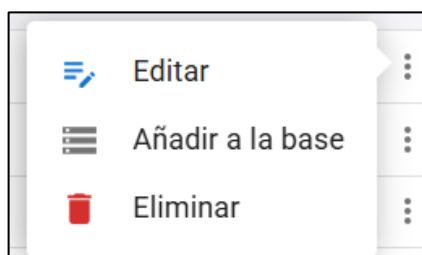


Figura 71. Popup opciones rubro

Al dar click en la opción “Editar” se abre el modal de edición representado en la Figura 72. En caso de que el campo Cantidad o Precio Unitario no sea ingresado manualmente si no calculado, este no será editable desde el modal. Desde aquí se puede modificar la categoría a la que pertenece el rubro y también el número; sin embargo, si la categoría no puede contener el número ingresado, entonces esta toma precedencia sobre el número, y se coloca al rubro al final de la categoría indicada, asignando el número correspondiente a esa posición. La categoría se selecciona a través del componente de filtrado/adición. Finalmente, el checkbox para indicar si el rubro tiene o no herramienta menor. Esta se trata de un registro en los equipos con valor calculado

(5% de la mano de obra). Al crear un rubro, se asume que tiene herramienta menor por defecto, al ser este el caso más común.

Editar Rubro ✕

No. * Unidad *

Nombre *

Cantidad Precio Unitario

Categoría

☒ Herramienta Menor

Figura 72. Modal edición rubro del proyecto

Al seleccionar la opción “Añadir a la Base” se muestra el modal de la Figura 73. Se precarga si tiene o no herramienta menor en base al rubro que se quiere añadir a la base. Dado que el transporte pertenece al proyecto, esto se deja a libre elección al momento de añadir a la base. La categoría que se muestra no es la misma que la del rubro de proyecto, y es seleccionada en el componente de filtrado/adición. Si se marca la casilla de “Variante”, se bloquea el ingreso de los campos excepto el nombre, ya que las demás características serán heredadas del rubro marcado como origen.

Guardar en la Base Rubro ✕

☐ Variante ☒ Herramienta Menor ☐ Transporte

Nombre * Unidad *

Categoría

Guardar en la Base Rubro ✕

☒ Variante

Rubro Origen *

Nombre *

Figura 73. Modal agregar a la base de rubros

Para poder añadir un rubro a la base, este debe cumplir con un conjunto de condiciones. En caso de que este no lo haga, se mostrará el modal de la Figura 74, además de una notificación de error con el punto concreto con el que no cumple el rubro.

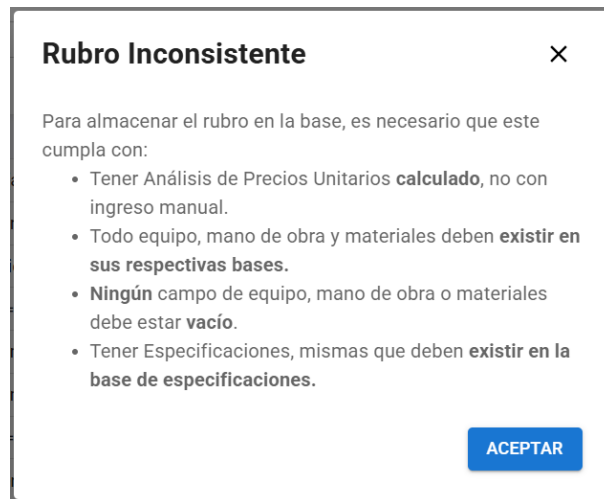


Figura 74. Modal rubro inconsistente

Hay que recordar que cada rubro en la base de datos tiene un campo *li_source_id*, el cual pasa de *NULL* a un valor cuando este fue tomado de la base de rubros. Puede darse el caso de querer añadir un rubro tomado de la base y que la información este duplicada, o que haya sido modificada. En caso de que el registro tenga el *li_source_id* entonces se muestra el modal de la Figura 75

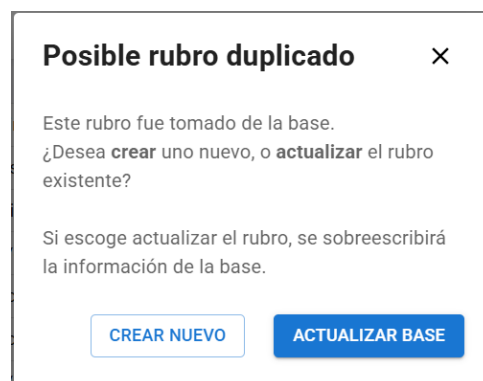


Figura 75. Modal posible rubro duplicado

Otra consideración con los registros de la tabla de presupuesto es que, si el campo Cantidad o el campo Precio Unitario son ingresados manualmente, entonces se mostrará un punto amarillo a su izquierda. Al hacer hover sobre este se indica que el valor fue ingresado manualmente, como muestra la Figura 76.

● 250.00	\$ 1.04
● 0.25	\$ 424.33
Ingresado manualmente	\$ 3.51
● 228.50	● \$ 5.24

Figura 76. Indicador ingreso manual

Finalmente, al hacer hover sobre el campo nombre del rubro se muestran tres botones, como muestra la Figura 77. El botón verde conduce a la cubicación del rubro, el amarillo al análisis de precios unitarios y el azul a las especificaciones.

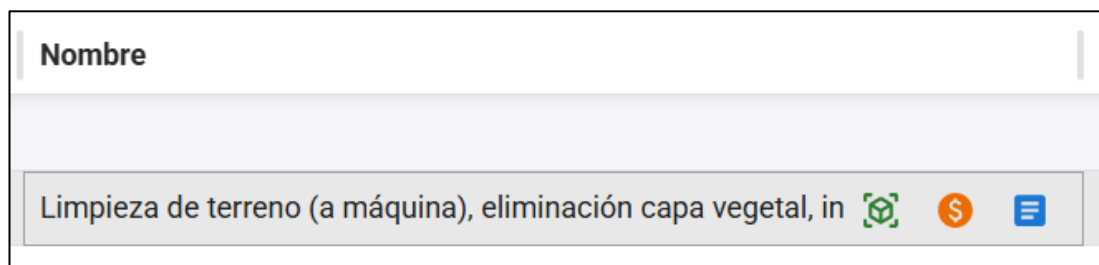


Figura 77. Botones componentes de rubro

En caso de seleccionar el botón de cubicación o de análisis de precios unitarios, lo primero que se mostrará es un modal indicando que, si el campo correspondiente ha sido ingresado manualmente, se eliminará y bloqueará al modificar la cubicación o análisis de precios, respectivamente. Si se marca la opción no volver a mostrar, esto evitará que el modal aparezca cada vez que se ingresa a la cubicación o APU por primera vez, solo en este proyecto.

Advertencia:

Se **bloqueará** el campo "Cantidad" del rubro y **eliminará** cualquier valor presente en el mismo.

[MOSTRAR MÁS](#) ▾

☐ No volver a mostrar

CONTINUAR

Advertencia:

Se **bloqueará** el campo "Precio Unitario" del rubro y **eliminará** cualquier valor presente en el mismo.

[MOSTRAR MÁS](#) ▾

☐ No volver a mostrar

CONTINUAR

Figura 78. Advertencia cubicación – análisis de precios unitarios

d. Cubicación

La página de cubicación se muestra en la Figura 79. En la parte superior cuenta con los breadcrumbs, el nombre del proyecto y los botones para visualizar la información del proyecto y exportación. Debajo del nombre del proyecto, se encuentra la información del rubro al que pertenece, como lo es el número, nombre y unidad. Al lado de la información del rubro se encuentran dos botones; el amarillo lleva al análisis de precios y el azul a las especificaciones. Este esquema estará presente también la página de análisis de precios unitarios y en la de las especificaciones, obviamente, uno de los botones (APU o especificaciones) será diferente según en cuál página se encuentre.

En la zona principal se encuentra la tabla con los ítems de cubicación; esta cuenta con el doble filtro. Los ítems pueden ser categorizados (registro grisáceo); pero sus categorías pueden también ocurrir n veces en el rubro, por lo que la columna “No. Elementos” es editable en la fila de categoría y el total por categoría es el resultado de sumar los totales de los ítems de cubicación y multiplicar este valor por el número de veces que ocurre la categoría. Cada registro cuenta con botones para editar y eliminar. Finalmente, en la zona inferior se muestra el total, que en este caso es el mismo de la categoría por solo existir una.

Categoría	Nombre	D1	D2	Producto	No. Elementos	Total (m2)	Acciones
^ Limpieza Aceras y Bordillos (9)					1	795.00	
	Calle A lado derecho e izquierdo	60.00	1.50	90.00	2.00	180.00	 
	Calle A curva	10.00	1.50	15.00	2.00	30.00	 
	Calle B lado derecho e izquierdo	40.00	1.50	60.00	2.00	120.00	 
	Calle B curva de retomo	50.00	1.50	75.00	1.00	75.00	 
	Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00	 
	Calle C curva de retomo	50.00	1.50	75.00	1.00	75.00	 
	Calle C curva	10.00	1.50	15.00	2.00	30.00	 
	Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00	 
	Calle C descuento tramo de calle "B"	10.00	1.50	15.00	-1.00	-15.00	 
						795.00	

Figura 79. Página de cubicación

El número de columnas de la tabla variará en función de cuantas dimensiones se hayan asociado a la unidad del rubro. Si no se encuentran coincidencias, se asume como una sola dimensión, y solo se muestra la columna “D1”. Si son dos dimensiones aparece también “D2” y si son tres, aparece “D3” también. Esto se refleja también en el modal de edición, como muestra la Figura 80. Según el número de dimensiones, serán los inputs disponibles. Además, el modal cuenta con el componente para filtrar/agregar categorías.

The figure displays three instances of the 'Actualizar Ítem' modal, each showing a different configuration of dimension inputs based on the number of dimensions associated with the unit.

- Modal 1 (Left):** Shows a single dimension input 'D1 *' with the value '75'. The 'No. Elementos *' is '1'. The category is 'Replanteo Aceras y Bordillos'.
- Modal 2 (Middle):** Shows two dimension inputs: 'D1 *' with '60' and 'D2 *' with '1.5'. The 'No. Elementos *' is '2'. The category is 'Limpieza Aceras y Bordillos'.
- Modal 3 (Right):** Shows three dimension inputs: 'D1 *' with '60', 'D2 *' with '0.35', and 'D3 *' with '0.5'. The 'No. Elementos *' is '2'. The category is 'Excavación Bordillos'.

Each modal includes a 'GUARDAR CAMBIOS' button at the bottom right.

Figura 80. Modal editar ítem de cubicación

Los campos de dimensiones no son obligatorios, y se pueden dejar en blanco como muestra la Figura 81, ya que pueden existir ítems que pese a tener más de una dimensión, el analista solo cuenta con el valor final (producto) por lo que se asumen los valores vacíos como 1. También al ser el número de columnas dinámico en base a las dimensiones asociadas a la unidad del rubro y esta poder cambiar en cualquier momento, pasando de 2 a 3 o 1 se encontró necesario notificar al usuario los ítems incompletos.

Se considera incompleto a un ítem cuando no todas sus dimensiones tienen un valor asignado y se asume el 1. Cada ítem incompleto muestra un punto rojo a la izquierda del nombre y al hacer hover sobre este se muestra un tooltip con el mensaje de ítem incompleto, como muestra la Figura 81. Además, en la tabla de presupuesto, en el campo Cantidad, se muestra el punto rojo, indicando que el cálculo está incompleto, como muestra la Figura 82.

Categoría	Buscar				
Categoría	Nombre	D1	D2	D3	Producto
Sin Asignar (1)					
	Item de cubicación 1		1.50		1.50
Item incompleto					

Figura 81. Ítem de cubicación incompleto

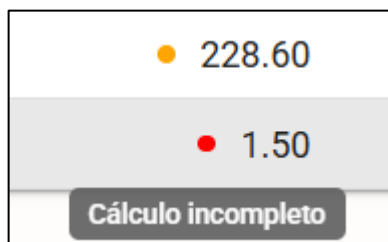


Figura 82. Indicador cubicación incompleta

e. Análisis de precios unitarios.

En el cuerpo de la página se puede observar las tablas para el ingreso de Equipos, Materiales, Mano de Obra y Transporte, según lo ilustrado en la Figura 83. A la derecha del nombre de la tabla hay dos botones, el primero con una lupa para buscar en la base correspondiente y cargar registros al APU, y el segundo para agregar al APU directamente el registro. En la parte inferior de cada tabla de componentes se muestra una sumatoria del costo de cada registro perteneciente al componente en cuestión.

SIPRO

Juan Espin

<

Figura 83. Página de análisis de precios unitarios

Al dar click en el botón con la lupa se abre el modal de selección con las dos tablas, como muestra la Figura 84. Este modal muestra los registros de la base correspondiente, con los precios con o sin IVA según la configuración del proyecto. Desde aquí se puede agregar nuevos registros o editar los existentes. Los registros agregados no tendrán la cantidad, ni rendimiento. Los Equipos, Materiales y Mano de Obra cuentan con la posibilidad de mostrar este modal.

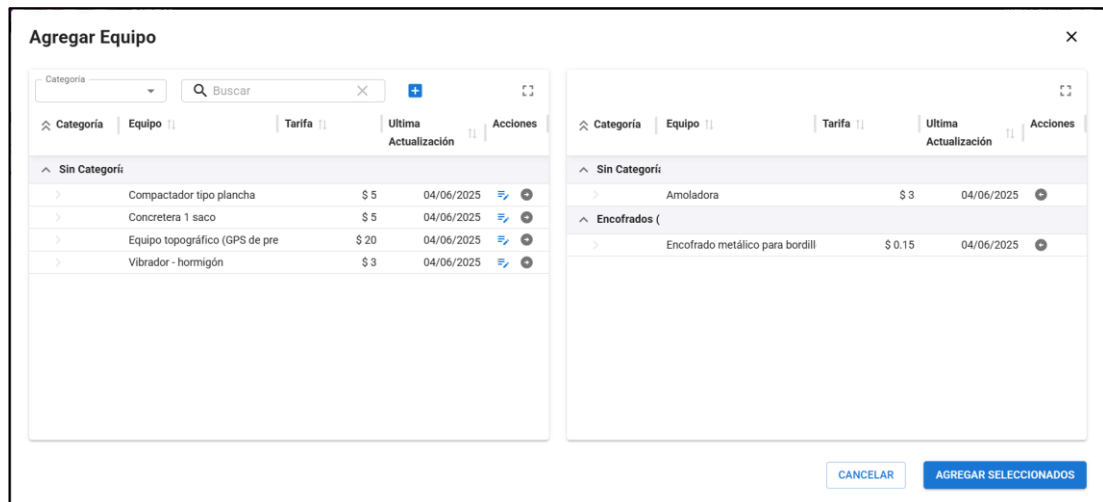


Figura 84. Modal agregar componente de la base

Al dar click en el botón de “más” se creará una fila en blanco para ingresar los datos. Al presionar el botón de guardado, pueden darse dos escenarios: se abrirá el modal para agregar el componente a su base respectiva, o se abrirá el modal de la Figura 85. Este modal indica que el equipo ya existe en la base. En la sección izquierda se muestra la información presente en la base, así como la última actualización; en la parte derecha los datos que se están ingresando. Para validar la existencia o no en la base se realiza una comparación con el nombre ingresado todo en minúsculas y sin espacios para evitar repeticiones por espacios intermedios. Se da a escoger al usuario si quiere utilizar los valores presentes en la base, o si desea actualizar la información de la base con la información ingresada.

Equipo Duplicado

×

El equipo que desea ingresar **ya existe** en la base de Equipo.
¿Desea **actualizar** la base o **utilizar** los datos existentes?

Datos Existentes

Nombre

Vibrador - hormigón

Tarifa

\$ 2.6087

IVA (%)

0 %

Categoría

Sin Categoría

Datos de Actualización

Nombre *

vibrador-hormigón

Tarifa *

\$ 1

IVA (%)

15 %

Categoría

Sin Categoría

Última Actualización: 04/06/2025

USAR EXISTENTES

ACTUALIZAR BASE

Figura 85. Modal componente duplicado

En caso de que se haya indicado que el proyecto tiene transporte, se mostrará esta tabla, pero no se puede agregar componentes. Esto se debe a que los registros de esta tabla son iguales a los presentes en la tabla de materiales, con la diferencia de que en lugar de precio unitario tienen tarifa de transporte, como se muestra en la Figura 86. El ejemplo de la imagen es solo ilustrativo, ya que los materiales utilizados no cuentan con tarifa de transporte y no afectan al cálculo.

Materiales  				
Nombre	Unidad	Cantidad	Precio Unitario	Costo
== Ripio Triturado (inc. Transporte a Sitio)	m3	0.80	10.00	
== Aditivo plastificante reductor de agua	Kg	0.50	7.36	
== Agua (Potable)	m3	0.20	1.00	
== Arena Lavada (INC. Transporte a Sitio)	m3	0.55	12.00	
== Cemento portland Tipo 1	Kg	350.00	0.16	
				Subtotal O
Transporte				
Nombre	Unidad	Cantidad	Tarifa Transporte	
Ripio Triturado (inc. Transporte a Sitio)	m3	0.80	0.00	
Aditivo plastificante reductor de agua	Kg	0.50	0.00	
Agua (Potable)	m3	0.20	0.00	
Arena Lavada (INC. Transporte a Sitio)	m3	0.55		
Cemento portland Tipo 1	Kg	350.00	0.00	
				Subtotal P

Figura 86. Transporte

En la parte inferior y debajo de la tabla de materiales o transporte según sea el caso, se muestra el total de costo, el porcentaje de indirectos, el precio del rubro que es la sumatoria de estos dos valores y el valor ofertado, que es el precio redondeado a dos decimales. Debajo de esta información, se encuentra una indicación de si los precios incluyen o no IVA, misma que aparece si se indicó que el proyecto tiene IVA en equipos y materiales. A la izquierda de la palabra indirectos se encuentra un botón que abre el modal de gestión de indirectos. El modificar estos valores afecta a los indirectos de todo el proyecto.

Subtotal O	\$ 0.00
Total Costo Directo (M+N+O+P):	\$ 0.86805
 Indirectos: 20 %	\$ 0.17361
Precio Total del Rubro:	\$ 1.04166
Valor Ofertado:	\$ 1.04
* Estos precios incluyen IVA	

Figura 87. Resumen APU

Se indica al usuario si alguno de los registros en el APU está incompleto. Se considera que un registro está incompleto si no tiene cantidad o rendimiento (ambos valores que no se llenan al agregar rubros de la base desde el modal). Al hacer hover sobre el punto rojo a la izquierda del nombre se indica que el registro está incompleto, como muestra la Figura 88. Si este punto es de color azul, eso indica que el registro no existe en la base. Este caso se da solo si un componente usado en el APU es eliminado de su base; la Figura 89 muestra el mensaje al hacer hover. De igual modo, si algún registro está incompleto se muestra en el campo de Precio Unitario de la tabla de presupuesto como muestra la Figura 90.

Nombre	Cantidad
 Herramienta Menor (5% M.O.)	
  Encofrado metálico para bordillo tipo americano par 2.40m (alquiler día)	
Registro incompleto	1.00

Figura 88. Componente incompleto

Nombre	Cantidad
Herramienta Menor (5% M.O.)	
● Encofrado metálico para bordillo tipo americano par 2.40m (alquiler día)	
● moladora	1.00
No existe en la base	

Figura 89. Componente sin homónimo en la base

\$ 424.33
\$ 3.51
● \$ 8.86
Cálculo incompleto

Figura 90. Indicador APU incompleto

f. Especificaciones

En esta página se encuentra el editor de texto enriquecido y en encima de este dos botones. El primero para buscar especificaciones en la base y el segundo para agregar la especificación actual a la base, como se muestra en la Figura 91. Nótese que en la esquina superior izquierda de la barra de acciones del editor de texto se encuentra un botón para guardar los cambios realizados, es decir no se cuenta con guardado automático.

The screenshot displays the 'Especificaciones' (Specifications) page within the Sipro application. The interface includes a top navigation bar with the Sipro logo and user information (Juan Espín). A left sidebar contains various icons for navigation. The main area features a text editor with a rich text toolbar. The editor content is organized into sections: 'a. DEFINICIÓN.' (Definition), 'ESPECIFICACIONES.' (Specifications), 'ENSAYOS Y TOLERANCIAS.' (Tests and Tolerances), 'REFERENCIAS.' (References), and 'MEDICIÓN.' (Measurement). The right sidebar provides project context, including the title 'Construcción de infraestructura para el proyecto de división familia Morales Guevara', the unit 'm3', and a status indicator.

Figura 91. Página de especificaciones

Al dar click en el botón con la lupa se muestra el modal de la Figura 92, para agregar una especificación al rubro sin redactarla. Este se divide en dos columnas. En la izquierda una tabla con los nombres de las especificaciones, fecha de creación y actualización. Al dar click sobre alguna de las filas se carga a la derecha el texto de la especificación para visualización. Una vez localizada la especificación se puede añadir presionando el botón de “más”. Antes de agregar una especificación se muestra el modal de la solicitando confirmación, ya que la acción sobre escribe la especificación previa en caso de existir, como se observa en la Figura 93.

Especificaciones

Buscar

Nombre	Actualizado	Creado
Excavación de zanja alcantarillado, m	06/06/2025	04/05/2025
Excavación sin clasificar en tierra a m	06/06/2025	06/06/2025
Hormigón simple f'c=210 kg/cm2, coi	06/06/2025	04/05/2025
Limpieza de terreno (a máquina), elim	04/06/2025	04/06/2025
Pintura Epóxica	10/05/2025	10/05/2025
Replanteo y nivelación de vías, equipc	04/06/2025	04/06/2025

a. DEFINICIÓN
Este trabajo consiste en la excavación y desalojo de todos los materiales de cualquier clase que sean encontrados en el transcurso de los trabajos, exceptuando aquellas excavaciones que son realizadas de acuerdo a otros rubros del contrato.

b. ESPECIFICACIONES
Los trabajos de excavación se realizarán en base a los alineamientos, pendientes, taludes y secciones transversales señalados en los planos o fijados por el Fiscalizador.
Incluye este rubro, la excavación, transporte y desalojo del material producto de estos trabajos.
El contratista deberá emplear en estos trabajos todo el equipo necesario para la ejecución eficiente y oportuna de los mismos; el equipo deberá contar con la aprobación del Fiscalizador y su disponibilidad en la obra dependerá de los procedimientos de trabajo que se empleen para la excavación.
El Fiscalizador procederá a verificar las pendientes, alineaciones y sección transversal, antes de manifestar su aprobación o reparos.
El Contratista estará obligado a conservar las referencias de niveles en mojones o estacas, hasta que la fiscalización lo creyese conveniente.

c. ENSAYOS Y TOLERANCIAS

Figura 92. Modal agregar especificaciones de la base

Advertencia

Se sobrescribirá la especificación del rubro en caso de existir.
¿Desea continuar?

CANCELAR CONTINUAR

Figura 93. Modal confirmar sobreescritura

Por el contrario, si se presiona el botón con el ícono de la base de datos (color gris), se muestra el modal para asignar un nombre a la especificación, como muestra la Figura 94.

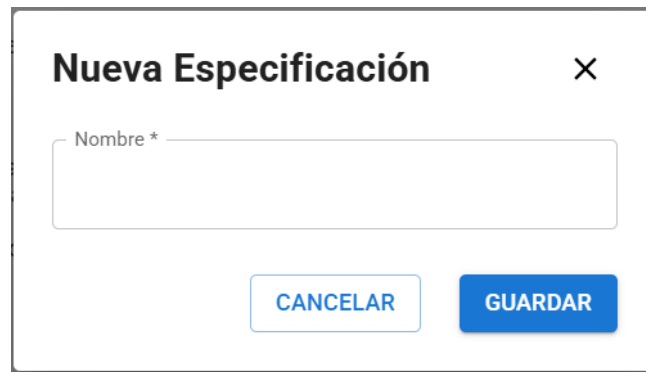
A modal window titled "Nueva Especificación" with a close button (X) in the top right corner. It contains a text input field with the label "Nombre *". Below the input field are two buttons: "CANCELAR" and "GUARDAR".

Figura 94. Modal agregar a la base de especificaciones

Dado que no se hace un guardado automático, se implementaron validaciones en caso de que se quiera abandonar la página de especificaciones sin guardar los cambios. La Figura 95 muestra los modales para este fin. Hay dos modelos, ya que al usar React Router, el sistema es básicamente un SPA, por lo que el modal de la izquierda es el que aparece en casos de navegación interna del sitio. Por otro lado, el modal de la derecha muestra el modal que aparece cuando se intenta recargar la página o cerrar la pestaña o el navegador sin haber guardado.

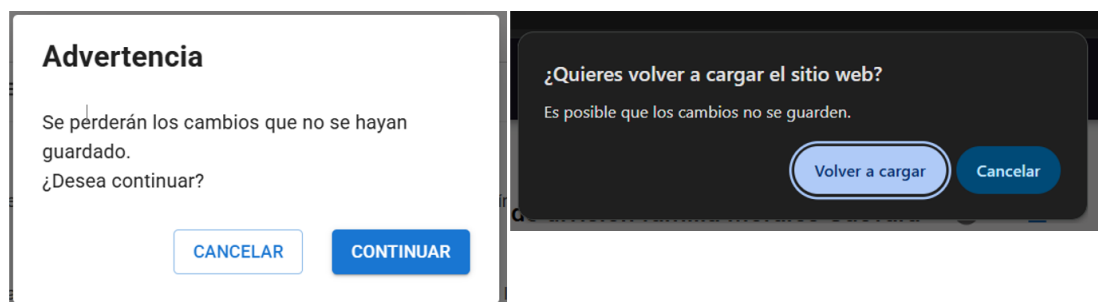
Two side-by-side warning modals. The left modal is titled "Advertencia" and contains the text "Se perderán los cambios que no se hayan guardado. ¿Desea continuar?" with "CANCELAR" and "CONTINUAR" buttons. The right modal is titled "¿Quieres volver a cargar el sitio web?" and contains the text "Es posible que los cambios no se guarden." with "Volver a cargar" and "Cancelar" buttons.

Figura 95. Confirmar pérdida de información

Finalmente, la Figura 96 muestra el formato de una especificación vacía. Este no se carga en la base de datos, sino que es el frontend quien lo muestra, siendo que en la base estas permanecen como *NULL*.



Figura 96. Formato especificaciones

g. Base de rubros

La página de la base de rubros es similar a la del presupuesto, con la diferencia de no contar con la información del proyecto. La tabla cuenta con el doble filtro, así como registros categorizados. Cada registro puede ser eliminado y editado y se puede agregar un rubro desde el botón “más”, como muestra la Figura 97. El modal de adición y edición del rubro es el mismo que se mostró en la Figura 73, con la diferencia de que, si se crea un nuevo rubro y se marca como variante, este precarga el nombre del rubro origen. Por otro lado, las opciones disponibles al hacer hover sobre el nombre del rubro son solo el análisis de precios y las especificaciones, dado que la cubicación es propia de cada proyecto y no se puede generalizar. Esto se muestra en la Figura 98.

Categoría	Nombre	Unidad	Precio Unitario	Acciones
Replanteo y nivelación (2)	Replanteo y nivelación de vías, equipo de precisión	Km	\$ 557.83	[Edit] [Delete]
Replanteo y nivelación para alcantarillado, equipo de precisión		Km	\$ 397.13	[Edit] [Delete]
Sin Categoría (1)	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	\$ 1.02	[Edit] [Delete]

Figura 97. Base de rubros

Nombre
(2)
Replanteo y nivelación de vías, equipo de precisión

- *Análisis de precios unitarios*

Esta página es idéntica a la del APU del proyecto, con excepción de que aquí solo se permite la carga de registros a cada componente a través del modal de búsqueda en la base, como indica la Figura 99.

emco
SIPRO

[Juan Espín](#)

>
Inicio / Base de Rubros / **Análisis de Precios Unitarios**

Rubro:

Unidad:

Replanteo y nivelación de vías, equipo de precisión

Km

Equipos

Nombre	Cantidad	Tarifa	Costo Hora	Rendimiento	Costo	Eliminar
Herramienta Menor (5% M.O.)					\$ 9.51111	
Equipo topográfico (GPS de precisión-estación total)	1.00	17.3913	\$ 17.39	11.11111	\$ 193.2367	
Subtotal M					\$ 202.74781	

Mano de Obra

Nombre	Cantidad	Tarifa hora	Costo Hora	Rendimiento	Costo	Eliminar
Cadenero - D2	1.00	4.19	\$ 4.19	11.11111	\$ 46.55555	
Peón - E2	2.00	4.14	\$ 8.28	11.11111	\$ 91.99999	
Topógrafo (En construcción - Estr. Oc. C1)	1.00	4.65	\$ 4.65	11.11111	\$ 51.66666	
Subtotal N					\$ 190.2222	

Materiales

Figura 99. Análisis de precios unitarios

- *Especificaciones*

Esta página solo permite visualizar la especificación asociada al rubro como se ve en la Figura 100. Se puede reemplazar la especificación con el botón de lupa ubicado debajo del nombre del rubro, el cual abre el modal mostrado en la Figura 92. También se puede editar la especificación, pero esto dirige a la edición en la base de especificaciones.

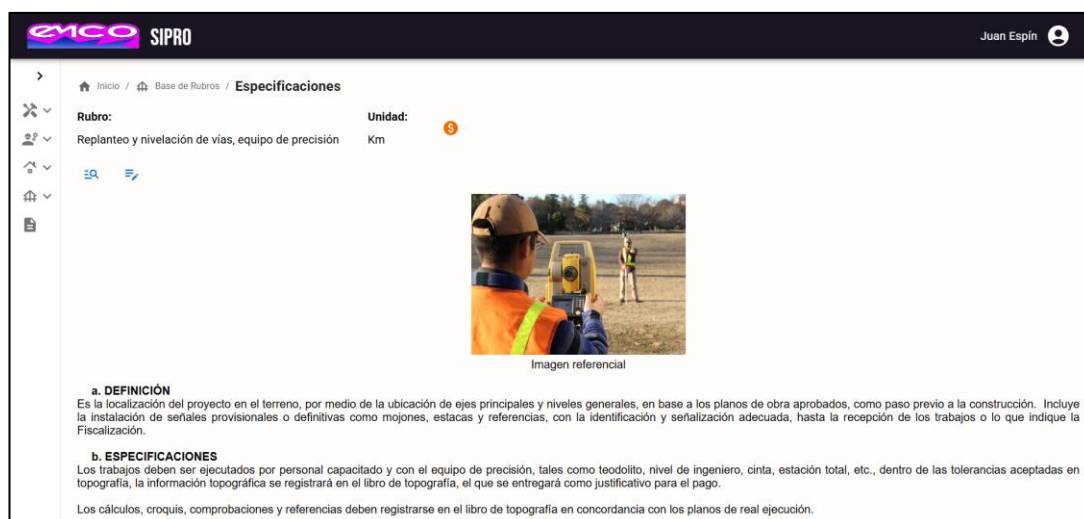


Figura 100. Especificaciones rubro

h. Base de especificaciones

Se trata de un modal el con estructura similar a la de los modales para añadir una especificación al proyecto o rubro de la base. La diferencia radica en que desde aquí se puede eliminar una especificación, o dirigir a la página de creación o edición de especificaciones, como muestra la Figura 101.



Figura 101. Modal base especificaciones

Al dar click en el botón de edición se muestra la página de la Figura 102, donde se observa el nombre de la especificación y a la derecha un botón que permite editarlo. Debajo del nombre hay tres botones, uno para eliminar la especificación, otro para ir

a la página de añadir especificación y otro como atajo para abrir la base de especificaciones y cambiar de especificación de ser el caso.

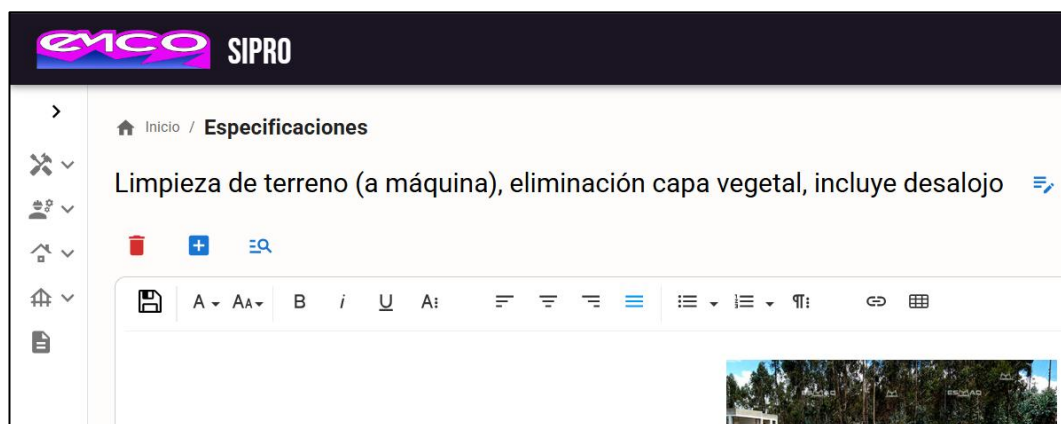


Figura 102. Página editar especificación

Al dar click en el botón de “más”, bien sea del modal de la Figura 101 o de la Figura 102, se dirige a la página de agregar especificación, en la cual se puede ingresar una especificación desde cero como muestra la Figura 103. Los botones en la parte superior del editor permiten cerrar el modo de creación, o abrir la base de especificaciones. Una vez que presione el botón de guardar se mostrará el modal solicitando el nombre de la especificación, y con el nombre proporcionado se redirige a la página de la Figura 102, para seguir editando la nueva especificación.

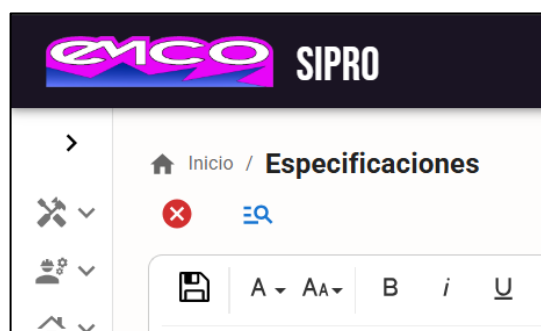


Figura 103. Página nueva especificación

3.3.2 Codificación

i. Únicos locales

Se encontró que se debía verificar que, por ejemplo, un proyecto nuevo o en edición no tome el mismo nombre de otro ya existente. Sin embargo, no se pudo utilizar la restricción *UNIQUE* de la base de datos, ya que otro usuario podía tener un proyecto con el mismo nombre y no debería existir conflicto. Este caso de un nombre único local se presentó no solo en los proyectos, sino también los rubros del proyecto, en los materiales del APU y otros sitios. Por ello, el Fragmento de Código 1 muestra la solución planteada, donde de forma dinámica se construye la consulta para verificar el único local. Se verifica el modo ya que la construcción de consulta debe evitar marcar como inválido el mismo nombre si es una actualización. Además, se debe proporcionar el nombre de la tabla, el nombre del campo que agrupa al elemento (*parentProp*), el valor del campo que agrupa (*parentId*) y el nombre que no debe existir (*name*). Con ellos se evita repetir las consultas y la lógica cada vez que se quiera verificar un único local.

```
exports.isNameAlreadyUsed = async (props) => {
  const { mode, targetTable, parentProp, parentId,
    itemId, name, response, mssg } = props;
  let data = [];
  if (mode == "insert") {
    [data] = await connection.query(
      `SELECT name
      FROM ${targetTable}
      WHERE ${parentProp} = ?
      AND name = ?`,
      [parentId, name]
    );
  } else {
    [data] = await connection.query(
      `SELECT name FROM ${targetTable}
      WHERE ${parentProp} = (
        SELECT ${parentProp}
        FROM ${targetTable}
        WHERE id = ?
      )
      AND name = ? AND id != ?`,
      [itemId, name, itemId]
    );
  }
  if (data.length > 0) {
    response.status(409).json({ error: mssg });
    return true;
  }
  return false;
};
```

Fragmento de Código 1. Constructor de consultas UNIQUE

j. Reordenamiento

Dado que la función de reordenamiento está presente en todos los componentes del APU e incluso en los ítems de cubicación, se decidió evitar la repetición de la lógica para cada uno y centralizarla en una función parametrizada para construir la consulta. Desafortunadamente, dada la naturaleza del reordenamiento, no se puede saber si fue hacia arriba o hacia abajo, por lo que el método espera recibir un array de ids, los cuales se encuentran en el nuevo orden. Recorriendo ese array envía a actualizar la base de datos buscando el id en la posición *idx*, y asignándole el valor *idx + 1*. Esto se muestra en el Fragmento de Código 2.

```
exports.updateOrderNumber = async (request, response, table) => {
  const change = request.body;
  let dbResponse;

  for(let idx = 0; idx < change.length; idx++){
    [dbResponse] = await connection.execute(
      `UPDATE ${table}
      SET order_number = ?
      WHERE id = ?`,
      [idx + 1, change[idx]]);
  }
  response.json(dbResponse);
}
```

Fragmento de Código 2. Función de reordenamiento

k. Cálculo de la cantidad.

Recordando que el cálculo del total de un ítem de cubicación viene dado en función del número de dimensiones asociadas a la unidad del rubro, se desarrolló una función para base de datos, la cual obtiene el total de la cantidad considerando las dimensiones del rubro.

El Fragmento de Código 3 muestra la función que se encarga de obtener el total de la cantidad. Se ejecuta una función de agregación *SUM* a un conjunto de valores que vienen de una tabla creada a partir de una consulta. Dicha consulta ejecuta a su vez una función de agregación *SUM*, pero esta vez al resultado de operar las dimensiones, el conteo del ítem y el conteo de categoría. Al existir n número de categorías, se agrupa por el id de categoría y como resultado se obtiene el conjunto de datos a sumar en la consulta padre.

Nótese que por cada registro se ejecuta la función *getDimensionValue* misma que se describe en el Fragmento de Código 4. Esta función obtiene el número de dimensiones asociadas a la unidad del rubro, y evalúa este número para decidir si acumula el valor de la siguiente unidad. En este caso se trata a una unidad nula como uno, para evitar volver cero el cálculo por un valor no proporcionado.

```
CREATE FUNCTION calculateQuantity(pliId INT) RETURNS DECIMAL(15,5)
BEGIN
    DECLARE quantity DECIMAL(15,5);
    SELECT SUM(totalPerCat) as quantity
    INTO quantity
    FROM ( SELECT
            SUM(getDimensionValue(pliId,
                                cubage.d1, cubage.d2, cubage.d3) * cubage.count) *
            categoryCount.count as totalPerCat
        FROM Cubage_Item as cubage, Cubage_Category_Count as categoryCount
        WHERE cubage.category_id = categoryCount.id AND cubage.pli_id=pliId
        GROUP BY categoryCount.id) as total;
    RETURN quantity;
END
```

Fragmento de Código 3. Función cálculo cantidad

```
CREATE FUNCTION getDimensionValue(
    pliId INT,
    d1 DECIMAL(11,5),
    d2 DECIMAL(11,5),
    d3 DECIMAL(11,5))
RETURNS DECIMAL(15,5)
BEGIN
    DECLARE unitDimensions INT;
    DECLARE product DECIMAL(15,5);
    SELECT dimensions
    INTO unitDimensions
    FROM Units
    WHERE id = ( SELECT unit
                FROM Project_Line_Item
                WHERE id = pliId );
    IF unitDimensions IS NULL OR unitDimensions >= 1 THEN
        SET product = IFNULL(d1, 1);
    END IF;
    IF unitDimensions >= 2 THEN
        SET product = product * IFNULL(d2, 1);
    END IF;
    IF unitDimensions >= 3 THEN
        SET product = product * IFNULL(d3, 1);
    END IF;
    RETURN product;
END
```

Fragmento de Código 4. Función para operar dimensiones

I. Notificaciones

Para garantizar la retroalimentación del usuario con notificaciones, se creó un contexto en React que se muestra en el Fragmento de Código 5.

```
export default function AlertDisplayContext(props) {
  const { children } = props;
  const [alerts, setAlerts] = useState([]);
  const timers = useRef({});
  const addAlert = (alertInfo) => {
    const id = Date.now();
    setAlerts((prev) => [...prev, { id, visible: true, ...alertInfo }]);
    startSelfClosing(id);
  };
  const startSelfClosing = (id) => {
    timers.current[id] = setTimeout(() => {
      setTimeout(() => closeAlert(id), 1000);
    }, 3000);
  };
  const stopSelfClosing = (id) => {
    clearTimeout(timers.current[id]);
    timers.current[id] = null;
  };
  const closeAlert = (id) => {
    clearTimeout(timers.current[id]);
    delete timers.current[id];
    setAlerts((prev) =>
      prev.map((alert) => {
        if(alert.id == id){
          alert.visible = false;
        }
        return alert;
      })
    );
  };
  return (
    <alertsContext.Provider value={[alerts, setAlerts]}>
      <timersContext.Provider value={timers}>
        <addAlertContext.Provider value={addAlert}>
          <startSelfClosingContext.Provider
            value={startSelfClosing}>
            <stopSelfClosingContext.Provider
              value={stopSelfClosing}>
              <closeAlertContext.Provider value={closeAlert}>
                {children}
              </closeAlertContext.Provider>
            </stopSelfClosingContext.Provider>
          </startSelfClosingContext.Provider>
        </addAlertContext.Provider>
      </timersContext.Provider>
    </alertsContext.Provider>
  );
}
```

Fragmento de Código 5. Contexto Notificaciones

Las alertas se cierran después de un intervalo de 3 segundos; los métodos *startSelfClosing* y *stopSelfClosing* manejan el auto cierre de la notificación o la cancelación del mismo respectivamente. De igual modo para un cierre manual de la notificación se creó el método *closeAlert*, que elimina el temporizador de cierre y también quita la notificación del array de notificaciones activas. Finalmente, el método más importante: *addAlert*, el cual agrega una alerta al array de alertas y la asocia a un temporizador. Todas las propiedades del contexto son inyectadas como *providers* en el HTML, y se toman los hijos de las props para que el contexto pueda ser un componente envoltorio y sus propiedades estén presentes en todos sus hijos. El Fragmento de Código 6 muestra el uso del contexto en la raíz del documento o el archivo *Main*.

```
createRoot(document.getElementById("root")).render(  
  <StrictMode>  
    <AlertDisplayerContext>  
      <RouterProvider router={router} />  
      <AlertDisplayer />  
    </AlertDisplayerContext>  
  </StrictMode>  
);
```

Fragmento de Código 6. Uso del contexto de notificaciones

Esta inyección del contexto permite que en cualquier punto de la aplicación se llame al método *getAddAlertCallback*, que devuelve el método *addAlert* del contexto y muestra la notificación. Sin embargo, esto solo crea la notificación lógica, mas no visible. De ello se encarga el componente *AlertDisplayer* mostrado en el Fragmento de Código 7, el cual hace uso de las demás propiedades del contexto para mostrar una alerta cada vez que se modifique el array de alertas, detiene el auto cierre de la alerta si el ratón se posa encima, lo retoma si se va, gestiona el click en el botón de cierre y muestra la información de la alerta.

```

export default function AlertDisplayer() {
  const [alerts, setAlerts] = getAlertsUseState();
  const timers = getTimers();
  const startSelfClosing = getSelfClosingCallback();
  const stopSelfClosing = getStopSelfClosingCallback();
  const closeAlert = getCloseAlertCallback();
  useEffect(() => {
    return () => {
      Object.values(timers.current).forEach(clearTimeout);
    };
  }, []);
  return (
    <Box>
      <Stack spacing={2} sx={{ mt: 2 }}>
        {alerts.map((alert, idx) => {
          return (
            <Slide
              in={alert.visible}
            >
              <Alert
                severity={alert.severity}
                onMouseEnter={() =>
                  stopSelfClosing(alert.id);
                }
                onMouseLeave={() => {
                  if (!timers.current[alert.id]) {
                    startSelfClosing(alert.id);
                  }
                }}
                onClose={() => {
                  closeAlert(alert.id);
                }}
              >
                {alert.message}
              </Alert>
            </Slide>
          );
        })}
      </Stack>
    </Box>
  );
}

```

Fragmento de Código 7. Componente AlertDisplayer

De igual modo, se definieron 4 niveles de alertas: información, advertencia, éxito y error. Dado que el proceso de creación de alertas es el mismo, pero que cambia el nivel de alerta, entonces se definieron funciones para obtener una alerta más fácilmente. Desafortunadamente, el llamar al contexto de añadir alerta no fue posible, dado que solo se podía utilizar dentro del cuerpo de un componente. Por tanto, el Fragmento de Código 8 muestra las funciones de generación de notificaciones, pero que reciben la función de *addAlert* desde fuera bajo el nombre *showAlert*.

```

export function infoAlert(mssg, showAlert) {
  showAlert({
    severity: "info",
    message: mssg,
  });
}
export function successAlert(mssg, showAlert) {
  showAlert({
    severity: "success",
    message: mssg,
  });
}
export function warningAlert(mssg, showAlert) {
  showAlert({
    severity: "warning",
    message: mssg,
  });
}
export function errorAlert(error, showAlert) {
  showAlert({
    severity: "error",
    message: error,
  });
}

```

Fragmento de Código 8. Obtener nivel de alerta

m. Generalización modales adición/creación

Dado que en el análisis de precios unitarios cada componente (equipos, materiales y mano de obra) cuenta con un modal de búsqueda, se creó un componente genérico para este fin. Sin embargo, cada uno debía a su vez permitir la inserción de nuevos registros a su respectiva base. Para ello se pensó en utilizar el mismo modal de creación edición, pero se debía garantizar que en el componente genérico se llamara a un componente genérico a su vez y que este escogiera que modal de creación concreto utilizar. El Fragmento de Código 9 muestra la función desarrollada para este fin. Para evitar el uso de condicionales y hacer más escalable la solución, se decidió utilizar un objeto *dialogs*, el cual tiene propiedades que coinciden con la prop *entity* de la función. Al final la función devuelve de este objeto la propiedad correspondiente. Esto es posible sin el uso de un array gracias a la *bracket notation* de JavaScript, que permite acceder a la propiedad de un objeto a través de su nombre como *String*.

```

export function getAddEditDialog(props) {
  const {
    entity,
    mode,
    openState,
    ...
  } = props;

  const dialogs = {
    equipment: (
      <AddEditEquipmentDialog
        mode={mode}
        openState={openState}
        ...
      />
    ),
    labour: (
      <AddEditLabourDialog
        mode={mode}
        openState={openState}
        ...
      />
    ),
    material: (
      <AddEditMaterialDialog
        mode={mode}
        openState={openState}
        ...
      />
    ),
  };
  return dialogs[entity];
}

```

Fragmento de Código 9. Selector de modales de creación

n. Filtrado de componentes ya usados

En el análisis de precios unitarios también se debe cargar la información de todos los equipos, mano de obra y materiales de sus respectivas bases. Es necesario conocer qué componentes ya están en uso en el análisis y aumentar o disminuir aquellos agregados o eliminados respectivamente. Para solucionar esto, se decidió filtrar en el frontend aquellos registros ya utilizados del total. El Fragmento de Código 10 muestra la implementación. Se recibe la totalidad de los registros y se llama a la función *filterUsedDbData*, que devuelve solo los componentes que ya existen en el análisis. Este array es pasado a una función setter de un *useState* de React, misma que después es utilizada dentro de los componentes que así lo requieran, pasando la prop a lo largo del árbol HMTL.

```

const fetchProjectLineItemEquipment = async (project, pli) => {
  let data = await getProjectLineItemEquipment(
    lineItemId,
    project.ivaOnEqpAndMat,
    showAlert
  );
  if (data) {
    setUsedDbEquipmentIds(filterUsedDbData(data, "equipmentSourceId"));
    if (pli.minorTool)
      data.unshift({
        id: null,
        name: "Herramienta Menor (5% M.O.)",
        cost: 0
      });
    setProjectLineItemEquipment(data);
  }
};

const fetchProjectLineItemMaterial = async (project) => {
  const data = await getProjectLineItemMaterial(
    lineItemId,
    project.ivaOnEqpAndMat,
    showAlert
  );
  if (data) {
    setUsedDbMaterialIds(filterUsedDbData(data, "materialSourceId"));
    setProjectLineItemMaterial(data);
  }
};

const fetchProjectLineItemLabour = async () => {
  const data = await getProjectLineItemLabour(lineItemId, showAlert);
  if (data) {
    setUsedDbLabourIds(filterUsedDbData(data, "labourSourceId"));
    setProjectLineItemLabour(data);
  }
};

```

Fragmento de Código 10. Exclusión de componentes usados

Dado que la lógica de filtrado es la misma para equipos, materiales y mano de obra, se aisló la funcionalidad en la función ya mencionada. El Fragmento de Código 11 muestra su implementación. Recibe como parámetros el array de registros de la base correspondiente y el nombre de la propiedad que se va a validar no sea nula. Se espera que este nombre sea siempre el del *sourceId*; si este no es nulo, entonces ya está en uso, y se evita comparar uno a uno con los registros del análisis optimizando recursos.

Para finalizar, hay que aclarar que no se crea un nuevo array donde no existan los elementos ya usados. Esto debido a que los registros usados en el APU pueden disminuir, en cuyo caso deben volver a ser seleccionables y a su vez, los valores que estos tengan pueden haber sido editados, por lo que adicionar los registros eliminados

a los seleccionables podría suponer inconsistencias con la información de la base de datos.

```
export function filterUsedDbData(data, sourceProp) {
  const usedDbData = [];
  data.forEach((item) => {
    if(item[sourceProp] != null){
      usedDbData.push(item[sourceProp]);
    }
  });
  return usedDbData;
}
```

Fragmento de Código 11. Función de filtrado

o. Jerarquización de rubros de la base

Para poder mostrar los rubros jerarquizados con sus variantes, el componente del frontend requiere que cada nodo padre tenga un array con los nodos hijos, mismos que pueden a su vez tener un array de nodos hijos. Sin embargo, obtener un objeto con estructura arbórea directamente con consultas SQL no se puede, por lo que el Fragmento de Código 12 muestra la solución desarrollada. Este procesamiento se realiza en el backend.

Esta consiste en obtener todos los rubros en un array plano, para posteriormente crear un *Map*, y llenarlo recorriendo los datos, asignar como *key* el id y aumentar la propiedad *subRows*. Posteriormente se vuelve a recorrer el array y se agregan los rubros que tengan *parentId* al array de *subRows* del padre correspondiente. Si no tiene un *parentId*, se agrega como un nodo padre de primer nivel en el array *hierarchy*. Al final se devuelve solo el array *hierarchy* que contiene el objeto estructurado. Se consiguió optimización ya que no se utiliza un ciclo anidado que busque rubro por rubro el padre correspondiente.

```

exports.buildLineItemHierarchy = (data) => {
  const liMap = new Map();
  const hierarchy = [];
  for (const item of data) {
    liMap.set(item.id, { ...item, subRows: [] });
  }
  for (const item of data) {
    const current = liMap.get(item.id);
    if (item.parentId !== null && liMap.has(item.parentId)) {
      liMap.get(item.parentId).subRows.push(current);
    } else {
      hierarchy.push(current);
    }
  }
  return hierarchy;
};

```

Fragmento de Código 12. Constructor de árbol

p. Evitar asignaciones inconsistentes de padres

Ya en el frontend, al momento de editar un rubro se permite cambiar el padre, pero se debe verificar que no se pueda asignar un rubro como hijo de uno de sus hijos, ya que se volvería un ciclo. Para obtener el array de rubros no asignables, se desarrolló la función del Fragmento de Código 13. Se trata de una función recursiva que recibe el rubro que se está editando (*lineItem*) y un array de ids excluidos (*bannedIds*).

```

const getBannedIds = (lineItem, bannedIds) => {
  if (lineItem.subRows.length === 0) return;
  for (const subRow of lineItem.subRows) {
    getBannedIds(subRow, bannedIds);
    bannedIds.push(subRow.id);
  }
};

```

Fragmento de Código 13. Función recursiva de exclusión

Esta función se utiliza en el Fragmento de Código 14, donde luego de agregados los id de los hijos del rubro, se agrega el rubro actual para no ser padre de sí mismo, y también el rubro padre, ya que no tendría sentido actualizar al mismo estado. Finalmente se asignan los posibles padres eliminando los id baneados, y se pasa este array a un setter de React.

```

useEffect(() => {
  let selectableParents = [];
  if (mode === "edit" && lineItemData) {
    const bannedIds = [];
    // Banea los rubros hijo
    getBannedIds(lineItemData, bannedIds);
    // Banea el rubro actual
    bannedIds.push(lineItemData?.id);
    // Banea el rubro padre
    lineItemData?.parentId && bannedIds.push(lineItemData?.parentId);
    // Excluye baneados
    selectableParents =
      possibleParents.filter((parent) =>
        !bannedIds.includes(parent.id)
      );
    // Agrega la opción ninguno para sacar de padre.
    selectableParents.unshift({ id: 0, name: "Ninguno" });
  } else {
    selectableParents = possibleParents;
  }
  setActualPossibleParents(selectableParents);
}, [openState[0], possibleParents, lineItemData]);

```

Fragmento de Código 14. Exclusión de padres completa

q. Exportación

Para la exportación en los diferentes formatos, se utilizó un patrón Factory. La clase *ExportManager* del Fragmento de Código 15 es la encargada de devolver una instancia de la clase concreta en base a si se quiere exportar más de un documento o si exporta solo uno, teniendo en cuenta el formato (PDF o Excel).

```

class ExportManager {
  static getExporter(moreThan1, format, requestBody, response) {
    if (moreThan1) return
      new ZipExporter(requestBody, response, format);
    switch (format) {
      case "pdf":
        return new PdfExporter(response);
      case "excel":
        return new XlsxExporter(response);
    }
  }
}

```

Fragmento de Código 15. Gestor de Exportadores

Al no tener forma de obligar a que se siga una plantilla, no se declara una clase abstracta o interfaz de la que hereden los constructores específicos, pero para seguir con el principio del patrón, todos deben cumplir con tener 2 métodos: *prepare* y *finalize*.

La clase *PdfExporter* del Fragmento de Código 16 en el método *prepare* modifica los encabezados de la respuesta, para que se envíe un archivo PDF. El método *finalize* está vacío ya que el stream de datos del PDF se debe escribir antes de generar el archivo, y esto es responsabilidad del método que genere el PDF.

```
class PdfExporter {
    constructor(response) {
        this.response = response;
    }
    prepare(file, fileName, specs = false) {
        this.response.
            setHeader("Content-Disposition",
                `attachment; filename="${fileName}.pdf"`);
        this.response.setHeader("Content-Type", "application/pdf");
        file.pipe(this.response);
    }
    finalize() {}
}
```

Fragmento de Código 16. Clase concreta PDF

La clase *XlsxExporter* en el método *prepare* del Fragmento de Código 17 establece los encabezados de la respuesta para enviar un archivo Excel, así como inicializar la instancia de *file* para posterior envío. Por otro lado, el método *finalize* se encarga de enviar la respuesta, dado que el archivo xlsx se procesa como un Buffer de datos, almacenable para envío luego de creado el archivo.

```
class XlsxExporter {
    constructor(response) {
        this.response = response;
    }
    prepare(file, fileName) {
        this.response.setHeader(
            "Content-Type",
            "application/vnd.openxmlformats-
            officedocument.spreadsheetml.sheet"
        );
        this.response.setHeader(
            "Content-Disposition",
            `attachment; filename="${fileName}.xlsx"`
        );
        this.file = file;
    }
    finalize() {
        this.response.send(this.file);
    }
}
```

Fragmento de Código 17. Clase concreta Excel

Las clases descritas previamente se utilizan cuando se quiere exportar solo un archivo ya sea en formato Excel o PDF. Si se quiere exportar más de uno, se agrupan estos archivos en un comprimido (.zip). La clase *ZipExporter* es la encargada de este proceso y se muestra en el Fragmento de Código 18.

En su constructor se establece la variable *zip*, la cual recibe el stream de datos donde se almacenarán los archivos a comprimir. Además, establece los encabezados para el envío de un archivo comprimido. En su método *prepare* verifica si se exporta Excel o PDF, dado que la forma de tratar los archivos creados es diferente. El método *finalize* envía el archivo comprimido.

```
class ZipExporter {
    constructor(requestBody, response, format) {
        this.response = response;
        this.response.setHeader("Content-Type", "application/zip");
        this.response.setHeader(
            "Content-Disposition",
            `attachment; filename="${requestBody.name} - ${requestBody.format.toUpperCase()}.zip"`
        );
        this.format = format;
        this.zip = archiver("zip", {
            zlib: { level: 9 },
        });
        this.zip.pipe(this.response);
    }
    prepare(file, fileName, specs = false) {
        if (this.format == "pdf") {
            if (specs) {
                console.log(file instanceof Buffer);
                this.zip.append(file, { name: `${fileName}.pdf` });
                return;
            }
            const stream = new PassThrough();
            file.pipe(stream);
            this.zip.append(stream, { name: `${fileName}.pdf` });
            return;
        }
        this.zip.append(file, { name: `${fileName}.xlsx` });
    }
    finalize() {
        this.zip.finalize();
    }
}
```

Fragmento de Código 18. Clase concreta ZIP

3.3.3 Consideraciones de Seguridad

Para la seguridad se decidió proteger las rutas en el frontend y para el backend se usó JWT para certificar que solo un usuario autenticado tenga acceso a los endpoints. A continuación, se muestran los detalles más relevantes de la implementación.

r. Backend

El token se debe generar una vez se ha verificado que el usuario existe en la base, para ello se utilizó una cadena como *SECRET*, la cual se toma de las variables de entorno. El Fragmento de Código 19 muestra la cadena usada como *SECRET*, así como la implementación para firmar el token. Al no pasarle opciones, por defecto el algoritmo de firma es HS256. Este token firmado es enviado al cliente quien deberá almacenarlo y enviarlo en la cabecera de todas las solicitudes para tener acceso a los endpoints.

```
SECRET='M50LEbK+Epa5tngiyp00sl6nwLfhdN2B0KQNgpHtTLqn4032mEQk7jpIPQUCnL/Up4qP
LGNYDn/jw0X+KJ6QAA'
exports.login = async (request, response) => {
  const { email, password } = request.body;
  const [data] = await connection.query(
    `SELECT id
     FROM USER WHERE EMAIL = ?
     AND PASSWORD = ?`,
    [email, password]
  );
  if (data.length == 0) {
    return response
      .status(404)
      .json({ error: "El usuario o contraseña son incorrectos" });
  }
  const authToken = jwt.sign(data[0], process.env.SECRET);
  response.json({ authToken }); // El cliente lo guarda
};
```

Fragmento de Código 19. Firma de JWT

El siguiente paso es en el archivo del servidor, antes de declarar el middleware de los endpoints que deben ser seguros se coloca el middleware de verificación de token. El Fragmento de Código 20 muestra la implementación. Se definió un array de rutas excluidas de la verificación: las rutas para la gestión de imágenes y el login, dado que este debe ser accesible siempre. Se excluyeron de la verificación las rutas para la carpeta de imágenes, debido a que Express se encarga de evitar que una carpeta para servir de forma estática liste su contenido si se accede directamente a la URL del

servidor. Por otro lado, las imágenes debían ser accedidas por las librerías de Froala para el editor de texto y por Puppeteer para la exportación de las especificaciones en PDF.

En la función *verifyToken* se verifica en primer lugar que la ruta no sea una de las excluidas de verificación, luego se certifica que la solicitud tiene la cabecera que contendrá el token. Si existe la cabecera, entonces se divide la cadena “*Bearer token*”, obteniendo solo el token. Finalmente se verifica el token con la librería y el *SECRET* definido. Si el token no es correcto se notifica al usuario y si lo es, se continúa con la solicitud. El aislar este proceso en un middleware permite definir la validación una sola vez, sin necesidad de implementarla en cada endpoint.

```
// En el archivo del servidor
app.use(verifyToken);
// En el archivo correspondiente
const WHITELIST = ["/ancalagon/image", "/ancalagon/userUploadedImages/",
"/ancalagon/login"];
const verifyToken = (request, response, next) => {
  if (WHITELIST.some((path) => request.path.startsWith(path))) {
    return next();
  }
  const authHeader = request.headers["authorization"];
  if (!authHeader) {
    return response.status(401).json({error:"Usuario no autorizado"});
  }
  const token = authHeader.split(" ")[1]; // obtiene solo el token
  if (!token) {
    return response.status(401).json({ error: "Token mal formado" });
  }
  jwt.verify(token, process.env.SECRET, (error, decoded) => {
    if (error) {
      return response.status(403).json({error:"Token inválido"});
    }
    next();
  });
};
```

Fragmento de Código 20. Agregar verificación JWT

s. *Frontend*

En la función *login* se recibe y almacena el token, así como el id del usuario. Adicionalmente, se indica a Axios que cada vez que se realice una solicitud se debe colocar la cabecera *Authorization*. Esto se logra a través del uso de *interceptors*. El Fragmento de Código 21 muestra la implementación. Pese a haber definido los

interceptors en el momento de iniciar sesión, al recargar la página estos se pierden, por lo que también deben ser colocados en el archivo *main.jsx* de la aplicación.

```
export async function login(data, navigate, showAlert) {
  const response =
    await axios
      .post(`${env.SERVER_MAIN_URL}/login`, data).catch((error) => {
        errorAlert(error, showAlert);
      });
  const authToken = response.data.authToken;
  localStorage.setItem("authToken", authToken);
  localStorage.setItem("userId", parseJwt(authToken).id);
  axios.interceptors.request.use(
    (config) => {
      config.headers.Authorization = `Bearer ${authToken}`;
      return config;
    },
    () => {}
  );
  navigate("/home");
}
export async function logout(navigate) {
  localStorage.removeItem("authToken");
  localStorage.removeItem("userId");
  navigate("/");
}
```

Fragmento de Código 21. Interceptors Axios

Con las cabeceras funcionando correctamente y el token almacenado, se debe proteger las rutas de la aplicación. Para ello se crea un componente de ruta protegida, el cual verifica si el token existe y de no ser el caso, redirecciona al inicio de sesión. El Fragmento de Código 22 muestra el componente.

```
const ProtectedRoute = ({ children }) => {
  const isAuth = !!localStorage.getItem("authToken");
  const location = useLocation();
  if (!isAuth) {
    return <Navigate to="/" state={{ from: location }} replace />;
  }
  return children;
};
```

Fragmento de Código 22. Protección de rutas

El Fragmento de Código 23 muestra un ejemplo de la implementación de la ruta protegida, así como la única ruta no protegida en el frontend: el inicio de sesión.

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <Login />,
  },
  {
    path: "/signup",
    element: (
      <ProtectedRoute>
        <Signup />
      </ProtectedRoute>
    ),
  },
],
```

Fragmento de Código 23. Implementación de ruta protegida

t. Trazabilidad temporal

El llevar un registro de la fecha de modificación de los proyectos, materiales, equipos, mano de obra y rubros permite tener una idea no solo para decidir si conviene usar un dato o debe actualizarse, si no también puede servir como registro de trazabilidad para indicar cambios inesperados o fuera de lugar.

u. Prevención de errores

El sistema cuenta con una elevada prevención de errores. Como se ha destacado en la sección de desarrollo, toda acción de eliminación muestra un modal solicitando confirmación y notificando las consecuencias del borrado. Por otro lado, también se muestran notificaciones para indicar eliminación de datos automáticamente, como es el caso de las notificaciones de eliminación y bloqueo de campo para las columnas precio unitario y cantidad en el presupuesto. También se incluye aquí la retroalimentación visual cuando el cálculo de la cantidad o del análisis de precios unitarios está incompleto o ha sido ingresado manualmente. Finalmente, se notifica sobre posibles pérdidas de información cuando el usuario no ha guardado los cambios en las especificaciones. Estas notificaciones aparecen cuando se intenta navegar en el sitio, o cuando se recarga la página, se cierra la pestaña o se cierra el navegador sin antes guardar las modificaciones en la página de especificaciones.

3.3.4 Pruebas de funcionalidad

En esta sección se detallan las pruebas de funcionalidad que se llevaron a cabo. Se han agrupado las funcionalidades según el requerimiento. La Tabla 13 muestra las pruebas llevadas a cabo.

Tabla 13. Pruebas de funcionalidad

Prueba	Requerimiento	Funcionalidad	Datos de entrada	Resultado	Estado
1	Autenticación de Usuario	Iniciar sesión	Correo y contraseña válidos	El usuario accede al sistema y visualiza su pantalla de inicio con los proyectos que tenga	Aprobado
2	Registro de Usuario	Crear Usuario	Nombre, apellido, correo electrónico y contraseña	El nuevo usuario es creado y puede acceder al sistema	Aprobado
3	Gestión de Equipos	Crear	Nombre, tarifa, IVA, categoría	El nuevo equipo es visible en la lista	Aprobado
4		Eliminar	Id (Dado por el sistema)	El equipo es eliminado y no aparece en la lista	Aprobado
5		Editar	Nombre, tarifa, IVA, categoría	El equipo actualizado se muestra con los nuevos datos	Aprobado
6		Filtrar	Cadena de búsqueda	Solo se muestran equipos que coincidan con la búsqueda	Aprobado
7	Gestión de Materiales	Crear	Nombre, unidad, precio unitario, IVA, tarifa de transporte y categoría.	El nuevo material es visible para el usuario	Aprobado
8		Modificar		Los cambios del material son reflejados en el sistema	Aprobado
9		Eliminar	Id (Dado por el sistema)	El material ya no aparece en la lista de materiales	Aprobado
10		Filtrar	Cadena de búsqueda	Solo se muestran los materiales que coincidan con la búsqueda	Aprobado
11	Gestión de Mano de Obra	Crear	Nombre, tarifa hora, estructura ocupacional.	La nueva mano de obra es visible para el usuario	Aprobado
12		Modificar		Los cambios en la mano de obra se reflejan correctamente	Aprobado
13		Eliminar	Id (Dado por el sistema)	La mano de obra ya no aparece en la lista	Aprobado
14		Filtrar	Cadena de búsqueda	Solo se muestra la mano de obra que coincide con la búsqueda	Aprobado
15	Gestión de Rubros	Crear	Nombre, unidad, herramienta	El nuevo rubro es visible	Aprobado

Prueba	Requerimiento	Funcionalidad	Datos de entrada	Resultado	Estado
16		Modificar	menor, transporte, rubro origen (opcional).	Los cambios realizados en el rubro se reflejan correctamente	Aprobado
17		Eliminar	Id (Dado por el sistema)	El rubro eliminado ya no aparece en el sistema	Aprobado
18		Filtrar	Cadena de búsqueda	Solo se muestran los rubros que coincidan con la búsqueda	Aprobado
19	Gestión de Especificaciones	Crear	Nombre, contenido	La nueva especificación técnica es visible en la lista	Aprobado
20		Modificar	Contenido	Los cambios en la especificación técnica se reflejan correctamente	Aprobado
21		Eliminar	Id (Dado por el sistema)	La especificación técnica ya no aparece en la lista	Aprobado
22		Asociar	Id (Dado por el sistema)	La especificación técnica queda vinculada al rubro seleccionado	Aprobado
23	Gestión de Proyectos	Crear	Nombre (Obligatorio), institución, localización, oferente, cédula	El nuevo proyecto es visible para el usuario	Aprobado
24		Modificar	oferente, fecha de elaboración (todos opcionales), transporte e IVA en equipos y materiales	Los cambios en el proyecto se reflejan correctamente	Aprobado
25		Duplicar	Proyecto base	Se crea un nuevo proyecto con los datos del proyecto original	Aprobado
26		Eliminar	Id (Dado por el sistema)	El proyecto eliminado ya no aparece en la lista	Aprobado
27		Usar como base	Conjunto de rubros a reutilizar	Los rubros seleccionados de otro proyecto se agregan al proyecto actual	Aprobado
28	Gestión de Indirectos	Crear	Nombre, valor porcentual.	Los valores de costos indirectos (utilidad y costos fijos) son guardados y reflejados correctamente en el proyecto.	Aprobado
29		Editar		Los valores de costos indirectos pueden ser modificados y los cambios se muestran actualizados en el proyecto.	Aprobado
30		Eliminar	Id (Dado por el sistema)	El indirecto deja de ser visible y se modifica el total asociado	Aprobado

Prueba	Requerimiento	Funcionalidad	Datos de entrada	Resultado	Estado
31	Generación de Presupuesto	Consultar	Id proyecto	El sistema muestra el listado completo de rubros del presupuesto con nombre, unidad, cantidad, precio unitario y total por rubro.	Aprobado
32		Calcular	Cantidad y precio unitario, ya sea manual o calculado	El sistema calcula correctamente el total por rubro y el presupuesto total del proyecto.	Aprobado
33	Gestión de rubros	Agregar	Número, nombre, unidad, cantidad (opcional), precio unitario (opcional), categoría y herramienta menor	El rubro añadido aparece correctamente en el presupuesto.	Aprobado
34		Modificar		Los cambios realizados en un rubro se guardan y se reflejan en el presupuesto.	Aprobado
35		Eliminar	Id (Dado por el sistema)	El rubro eliminado desaparece del presupuesto.	Aprobado
36		Reordenar	Nuevo número de orden	Los rubros pueden cambiar de orden y la nueva organización se guarda.	Aprobado
37		Cargar desde base	Conjunto de rubros a reutilizar	Los rubros importados desde la base de datos se agregan correctamente al presupuesto.	Aprobado
38		Almacenar para uso futuro	Id (Dado por el sistema)	El rubro aparece en la base de rubros	Aprobado
39	Gestión de Ítems de Cubicación	Agregar	Nombre, d1, d2, d3 y no. Elementos.	El nuevo ítem de cubicación se muestra en la lista.	Aprobado
40		Modificar		Las actualizaciones en un ítem se guardan correctamente y se reflejan en la interfaz.	Aprobado
41		Eliminar	Id (Dado por el sistema)	El ítem eliminado desaparece de la lista.	Aprobado
42		Filtrar	Cadena de búsqueda	El usuario puede ver solo los ítems que coincidan con la búsqueda.	Aprobado
43	Carga componentes en el APU	Buscar	Cadena de búsqueda	El sistema muestra una lista de equipos, materiales o mano de obra que coinciden con los criterios de búsqueda.	Aprobado
44		Agregar	Listado de equipos, mano de obra o materiales	Los componentes seleccionados se añaden al análisis de precios unitarios (APU) y aparecen en la lista correspondiente.	Aprobado

Prueba	Requerimiento	Funcionalidad	Datos de entrada	Resultado	Estado
45		Mostrar totales parciales	Conjunto de materiales, equipos o mano de obra	El sistema calcula y muestra correctamente el total parcial de cada componente agregado al APU.	Aprobado
46	Análisis de Precios Unitarios (APU)	Calcular	Conjunto de materiales, equipos y mano de obra	El sistema calcula el precio unitario del rubro con precisión de hasta cinco decimales en los cálculos internos.	Aprobado
47		Mostrar	Conjunto de materiales, equipos y mano de obra	El precio unitario final del rubro se muestra con dos decimales, reflejando correctamente el resultado del cálculo.	Aprobado
48	Exportación de Presupuesto	Exportar a Excel	Id proyecto	El sistema genera y descarga uno o varios archivos Excel. Los archivos pueden ser del presupuesto, análisis de precios o cubicación.	Aprobado
49		Exportar a PDF	Id proyecto	El sistema genera y descarga uno o varios archivos PDF. Los archivos pueden ser del presupuesto, cubicación, análisis de precios o especificaciones.	Aprobado

3.3.5 Evaluación de Resultados

Como se mencionó en la sección de Validación de Usabilidad del Capítulo I, de los métodos de evaluación subjetiva disponibles el más utilizado es el test SUS (System Usability Scale), y es el que se utilizó para evaluar el sistema propuesto. Este test consiste en una encuesta con diez afirmaciones en las que el evaluador debe escoger entre cinco opciones de respuesta, organizadas según la escala de Likert. Las afirmaciones en el test original se encuentran en inglés [21], por lo que se realizó una traducción que se presenta a continuación.

1. Creo que me gustaría usar este sistema con frecuencia
2. Encontré el sistema innecesariamente complejo
3. Me pareció que el sistema era fácil de usar
4. Creo que necesitaría el apoyo de un técnico para poder usar este sistema

5. Encontré que las diversas funciones del sistema estaban correctamente integradas
6. Me pareció que había muchas inconsistencias en este sistema
7. Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.
8. Encontré el sistema muy incómodo de usar
9. Sentía mucha confianza al usar el sistema
10. Necesité aprender muchas cosas antes de poder empezar a usar el sistema

De igual modo, se enumeran las posibles opciones de respuesta, así como el valor numérico asignado a cada una.

- Totalmente en desacuerdo = 1
- En desacuerdo = 2
- Indiferente = 3
- De acuerdo = 4
- Totalmente de acuerdo = 5

Los resultados del test se analizan a través de un proceso matemático, donde:

- A los valores de las respuestas impares (1,3,5,7,9) se les resta 1.
- Se resta el valor de las respuestas pares (2,4,6,8,10) de 5,
- Se suman los valores obtenidos de ambos pasos anteriores.
- El resultado anterior se multiplica por 2.5

La Ecuación 1 ilustra estos pasos de forma matemática, donde A_n es el número de la afirmación

$$x = [(A_1 - 1) + (A_3 - 1) + (A_5 - 1) + (A_7 - 1) + (A_9 - 1) + (5 - A_2) + (5 - A_4) + (5 - A_6) + (5 - A_8) + (5 - A_{10})] * 2.5 \quad (1)$$

Estos pasos se ejecutan por cada evaluador, promediando al final los puntajes obtenidos para así tener el puntaje de usabilidad del sistema.

Una vez definidas las reglas de evaluación, se procedió a aplicar el test a los 11 individuos que componen la muestra como se especifica en la sección Población y muestra. Para este fin se organizaron reuniones donde se presentó el sistema a los profesionales. El test se ejecutó inmediatamente después de la presentación a través de la herramienta Google Forms. La Tabla 14 muestra los resultados obtenidos. Como columnas y hacia la derecha, con números ordinales se encuentran las afirmaciones en el orden descrito anteriormente. Hacia abajo, como filas se numera a los evaluadores del test. Se muestra además el resultado individual y el promedio. Adicionalmente se ha colocado una nota y el rango en el que se encuentran los resultados, para una mejor interpretación, basado en la Tabla 7.

Tabla 14. Resultados test SUS

Afir. Enc.	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°	Puntaje SUS	Nota SUS	Rango
1	4	2	4	4	5	2	4	2	4	3	70,00	C	Marginal
2	4	4	2	3	5	3	2	2	4	2	57,50	D	Marginal
3	5	2	4	3	5	1	4	2	4	2	80,00	A-	Aceptable
4	5	1	5	3	5	1	5	1	5	2	92,50	A+	Aceptable
5	4	2	4	4	5	2	4	1	4	2	75,00	B	Aceptable
6	5	1	5	1	4	1	5	1	5	1	97,50	A+	Aceptable
7	5	3	5	2	5	2	5	1	5	1	90,00	A+	Aceptable
8	5	2	5	3	4	1	4	2	4	1	82,50	A	Aceptable
9	5	3	5	5	5	3	5	5	5	3	65,00	C	Marginal
10	5	1	5	2	5	1	5	1	4	3	90,00	A+	Aceptable
11	5	2	5	2	4	1	5	1	5	2	90,00	A+	Aceptable
TOTAL											80,91	A	Aceptable

Dado que el valor obtenido tras la ejecución del SUS es de 80.91 y considerando la interpretación de resultados propuesta, se puede afirmar que el sistema cumple con tener una buena usabilidad. Basado en las afirmaciones, los usuarios encuentran el sistema intuitivo y fácil de usar, además de presentar disposición a usarlo para la gestión de análisis de precios unitarios.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se identificaron tanto las tecnologías necesarias para el desarrollo del sistema como la plataforma de la aplicación, arquitectura, metodología de desarrollo y método de evaluación de usabilidad del sistema. Las decisiones tomadas se sustentaron a través del uso de tablas comparativas donde se contrastaba la información sobre herramientas usadas por otros autores en el desarrollo de sistemas para análisis de precios unitarios. Al final se optó por el uso de tecnologías modernas con ecosistemas amplios, para agilizar al máximo el proceso de desarrollo. Estas tecnologías incluyen React, Node.js y MySQL. En cuanto a la metodología se optó por RAD, un enfoque no encontrado en las investigaciones revisadas, pero con prácticas que guiaban a tener un mínimo de documentación y optimizar el tiempo.
- Se desarrolló el sistema, encontrando que la participación activa del cliente fue fundamental, siendo este el punto de apoyo para validarlo. Dada la orientación del sistema a cubrir una necesidad en el campo de la ingeniería civil, fue importante su asesoría en los conceptos más específicos de esta rama, así como su guía en el proceso de prototipado. Sin embargo, se descubrió que los prototipos de interfaz no guían al cliente si este no está familiarizado con el desarrollo del software. El cliente deseaba poder probar la aplicación, manipular las funcionalidades y en general, realizar acciones con resultados visibles. La atención se perdía cuando se empezaba a hablar de lo que se “hará” sin poder demostrar dichas propuestas. Esto podría evidenciar una enorme limitante en los prototipos visuales, al menos cuando se trata de comunicar la estructura y funcionamiento del software a aquellas personas sin conocimientos técnicos sobre prototipado.
- Se evaluó el sistema a través del test SUS, encontrando que es ideal para poder cuantificar la usabilidad percibida por los usuarios, siendo que sus resultados indican si el usuario podría y querría usar el sistema con gusto. Además, los

estudios revisados lo señalan como una herramienta ideal cuando se quiere resultados significativos y se cuenta con una muestra pequeña. Sin embargo, se encontró enriquecedora y como un complemento ideal la retroalimentación proporcionada por los usuarios durante la exposición del sistema. Se dio libertad para que estos pudieran opinar sobre las diferentes funcionalidades mientras estas se mostraban, pudiendo evaluar su reacción y qué es lo que pensaban al ver el sistema. Si bien esta información no fue evaluada, se considera de incalculable valor para el autor, siendo un punto de referencia para mejorar futuros desarrollos.

4.2 Recomendaciones

- Se recomienda el uso de tecnologías que cuenten con una comunidad activa, guías de uso y con una amplia documentación. Esto garantiza que se pueda solventar problemas y se encuentren librerías o recursos externos que apoyen el desarrollo. En cuanto a metodologías, es ideal el uso de aquellas que prioricen resultados en poco tiempo y no pierdan tiempo en documentación innecesaria.
- Se sugiere la implementación de interfaces responsivas, para aumentar la gama de dispositivos desde los cuales se puede acceder al sistema.
- Se recomienda llevar por escrito y de ser posible firmados los acuerdos a los que se llega en cuanto a funcionalidad del sistema, para evitar malos entendidos y retrasos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Vergara, “Industria de la Construcción en el Ecuador”, *Tesis de Pregrado*, Colegio de Administración y Economía, UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ, Quito, Ecuador, 2017. [En línea]. Disponible: <https://core.ac.uk/download/pdf/147370346.pdf>
- [2] M. Tenorio, D. Heras, P. Vásquez, “*Búsqueda de Oportunidades de la Base de Precios Unitarios en el Sector Público – Caso GAD Cuenca*”, Polo del Conocimiento, vol. 8, no. 1, pp. 1582-1620, enero 2023. [En línea]. Disponible: [10.23857/pc.v8i1](https://doi.org/10.23857/pc.v8i1)
- [3] V. Vásquez, “DESARROLLO DE UN SOFTWARE PROGRAMADO EN MATLAB CON UNA BASE DE DATOS, QUE PERMITA GENERAR PRECIOS UNITARIOS, PRESUPUESTOS, REAJUSTE, CRONOGRAMA VALORADO DE TRABAJO Y LAS ESPECIFICACIONES TÉCNICAS DE CADA RUBRO”, *Tesis de Pregrado*, FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA, UNIVERSIDAD TÉCNICA DE AMBATO, Ambato, Ecuador, 2021. [En línea]. Disponible: <https://repositorio.uta.edu.ec:8443/bitstream/123456789/32497/1/Tesis%20I.%20C.%201461%20-%20Vasquez%20Herrera%20Vicente%20Andr%c3%a9s.pdf>
- [4] J. Pinchao, “DESARROLLO DE UN MÓDULO DE ANÁLISIS DE PRECIOS UNITARIOS PARA MEJORAR LA PRODUCTIVIDAD DE LA EMPRESA INVESERVICE FGL S.A.S UTILIZANDO MICROSERVICIOS REST.”, *Tesis de Pregrado*, Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Ibarra, Ecuador, 2024. [En línea]. Disponible: <https://repositorio.utn.edu.ec/bitstream/123456789/15513/2/04%20SOF%20030%20TRABAJO%20GRADO.pdf>
- [5] J. Montalván, “PROGRAMACIÓN Y PROCESAMIENTO DE DATOS EN LENGUAJE DE PYTHON PARA LA DETERMINACIÓN DE ANÁLISIS DE PRECIOS UNITARIOS Y PRESUPUESTO, PARA CONSTRUCCIÓN DE OBRAS CIVILES.”, *Tesis de Pregrado*, Facultad de Ingeniería Civil y Mecánica,

Universidad Técnica de Ambato, Ambato, Ecuador, 2022. [En línea]. Disponible: <https://repositorio.uta.edu.ec:8443/bitstream/123456789/36295/1/Tesis%20I.C.%201631%20-%20Montalv%c3%a1n%20Castillo%20Janina%20de%20los%20Angeles%20y%20Bonilla%20Freire%20Lenin%20Fernando.pdf>

- [6] L. Albán, L. Vásquez, “DESARROLLO DE UN SISTEMA PARA EL ANÁLISIS DE PRECIOS UNITARIOS DE OBRAS ELECTROMECAÓNICAS”, *Tesis de Pregrado*, FACULTAD DE LAS CIENCIAS DE LA INGENIERÍA Y APLICADAS, Universidad Técnica de Cotopaxi, Latacunga, Ecuador, 2022. [En línea]. Disponible: <https://repositorio.utc.edu.ec/items/6a1ca9ef-52e4-4f06-a484-c203cb003d2d>
- [7] A. Soto, “Presupuestos para una vivienda de hasta dos pisos en hormigón o acero a través de una aplicación”, *Tesis de Pregrado*, Unidad Académica de Ingeniería, Industria y Construcción, Universidad Católica de Cuenca, Cuenca, Ecuador, 2023. [En línea]. Disponible: <https://dspace.ucacue.edu.ec/handle/ucacue/15408>
- [8] F. Yagual, “Sistema Web para Elaboración de Análisis de Precios Unitarios para Construcción de Viviendas en el Cantón La Libertad”, *Tesis de Pregrado*, FACUTAD DE SISTEMAS Y TELECOMUNICACIONES, UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA, La Libertad, Ecuador, 2021. [En línea]. Disponible: <https://repositorio.upse.edu.ec/bitstream/46000/5978/1/UPSE-TTI-2021-0027.pdf>
- [9] D. Morillo, B. Narváez, “Software como servicio para el análisis de precios unitarios en el sector de la construcción.”, *Tesis de Pregrado*, Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables, Universidad Nacional de Loja, Loja, Ecuador, 2017. [En línea]. Disponible: <http://dspace.unl.edu.ec/jspui/handle/123456789/19401>
- [10] M. Pachacuti, “Sistema Web para el diseño y cálculo de presupuesto de obras de arte hidráulicas Caso: Consultora Argos S.R.L.”, *Tesis de Pregrado*, Facultad de Ciencias Puras y Naturales, Universidad Mayor de San Andrés, La Paz, Bolivia,

2015. [En línea]. Disponible: <http://repositorio.umsa.bo/xmlui/handle/123456789/8710>

- [11] X. Song, M. Liu, L. Gong, Y. Gu, M. Shidujaman, "A Review of Human-Computer Interface Evaluation Research Based on Evaluation Process Elements", *Human-Computer Interaction. HCII 2023*, M. Kurosu and A. Hashizume, Eds., Lecture Notes in Computer Science, vol. 14011, Cham: Springer, 2023, pp. 262–289. [En línea]. Disponible: https://doi-org.uta.lookproxy.com/10.1007/978-3-031-35596-7_17
- [12] G. Lindgaard, C. Dudek, "User satisfaction, aesthetics and usability: Beyond reductionism", *Usability: Gaining a Competitive Edge, IFIP World Computer Congress 2002*, J. Hammond, T. Gross, and J. Wesson, Eds., Kluwer Academic Publishers, 2002, pp 231–246. [En línea]. Disponible: https://doi-org.uta.lookproxy.com/10.1007/978-0-387-35610-5_16
- [13] F. Paz, A. Huaracha, F. A. Paz, A. Moquillaza, A. Lecaros, F. Falconi, J. Aguirre, A. Tapia, "Comparing two sets of usability heuristics in a user experience evaluation: A case study in an e-commerce mobile application", *Design, User Experience, and Usability. HCII 2023*, A. Marcus, E. Rosenzweig, and M. M. Soares, Eds., Lecture Notes in Computer Science, vol. 14032, Cham: Springer, 2023, pp 69-81. [En línea]. Disponible: https://link-springer-com.uta.lookproxy.com/chapter/10.1007/978-3-031-35702-2_5#citeas
- [14] Y. Hung Hsu and C. Hsiung Chen, "Usability study on the user interface design of ride-hailing applications", *Design, User Experience, and Usability. HCII 2023*, A. Marcus, E. Rosenzweig, and M. M. Soares, Eds., vol. 14032, Lecture Notes in Computer Science, Cham: Springer, 2023, pp 208–216. [En línea]. Disponible: https://link-springer-com.uta.lookproxy.com/chapter/10.1007/978-3-031-35702-2_15#citeas
- [15] K. Matsumoto and W. Yang, "Research about influence of information presentation in UI on scanability," *Design, User Experience, and Usability. HCII 2023*, A. Marcus, E. Rosenzweig, and M. M. Soares, Eds., vol. 14032, Lecture Notes in Computer Science, Cham: Springer, 2023, pp 227–238. [En línea].

Disponible: https://link-springer-com.uta.lookproxy.com/chapter/10.1007/978-3-031-35702-2_17

- [16] J. Nielsen, "How users read on the web," *Nielsen Norman Group*, Sep. 30, 1997. [En línea]. Disponible: <https://www.nngroup.com/articles/how-users-read-on-the-web/>.
- [17] M. Maguire, "A review of usability guidelines for e-commerce website design," in *Design, User Experience, and Usability. HCII 2023*, A. Marcus, E. Rosenzweig, and M. M. Soares, Eds., vol. 14032, Lecture Notes in Computer Science, Cham: Springer, pp. 30 – 43. [En línea]. Disponible: https://link-springer-com.uta.lookproxy.com/chapter/10.1007/978-3-031-35702-2_3
- [18] A. J. Jácome Perrazo, "Aplicación web para la gestión de historias clínicas odontológicas con odontograma geométrico digital en la Clínica Saúde Medical Group del cantón Pelileo", *Trabajo de titulación*, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador, 2024. [En línea]. Disponible: <https://repositorio.uta.edu.ec/items/7d2aa054-1e62-4244-b701-666ce499e311>
- [19] J. Sauro, "10 Things to Know About the System Usability Scale (SUS)," *MeasuringU*, 18-Jun-2013. [En línea]. Disponible: <https://measuringu.com/10-things-sus/>
- [20] J. Sauro, "5 Ways to Interpret a SUS Score", *MeasuringU*, Sep. 19, 2018. [En línea]. Disponible: <https://measuringu.com/interpret-sus-score/>
- [21] J. Wierusz, "System Usability Scale (SUS)", *SurveyLab Blog*, Mar. 12, 2024. [En línea]. Disponible: <https://www.surveylab.com/blog/system-usability-scale-sus/>

ANEXOS

Anexo A. Entrevista Técnica

1. ¿Cuáles son las principales dificultades que enfrenta actualmente en el proceso de cálculo del Análisis de Precios Unitarios?

Más que nada, la dificultad de tener que manejar Excel; si bien es una herramienta completa, da problemas al momento de querer corregir el nombre de un rubro, porque esto significa que hay que cambiar todos los lugares donde este nombre aparezca. También es complicado mantener vinculados los análisis de precios, la cubicación y las especificaciones a los rubros que aparecen en el presupuesto. Estábamos intentando con PUNIS, pero el programa se tarda demasiado en cargar de una ventana a otra y no soluciona el inconveniente de mantener la misma información entre archivos si se cambia algún dato.

2. ¿Qué funcionalidades considera imprescindibles en una herramienta de Análisis de Precios Unitarios para su flujo de trabajo?

Primero que nada, debe tener el presupuesto, que es donde se agrupan los rubros. También debe permitir calcular el APU, y dejar libertades en cuanto a si se llena o no la tabla de Transporte, por ejemplo, la cual no siempre tiene datos.

Tiene que haber una base de materiales, equipos y mano de obra para no tener que estar cargando una y otra vez los mismos datos en diferentes proyectos. La idea es poder mantener los precios actualizados.

Hasta ahí son las funcionalidades que tienen otros sistemas, pero queremos también poder asociar el cálculo de volúmenes (cubicación) al rubro, y también las especificaciones.

También hay rubros que son afines entre proyectos, y de los cuales queremos tener todo el análisis de precios unitarios y las especificaciones, para solo copiar y pegar, sin tener que calcular nuevamente. Debe haber opciones de búsqueda, para filtrar rápido rápido.

3. ¿Cómo se comparte actualmente la información de las bases de materiales, mano de obra y maquinaria?

Nos pasamos los archivos Excel con las bases si están actualizadas, si no simplemente buscamos los precios más actualizados y lo ponemos en el cálculo. A veces los archivos de otros proyectos recientes son los que sirven como base, por comodidad.

4. ¿Cuál es el proceso que se sigue para realizar un análisis de precios unitarios?

Se define un rubro y se realiza el análisis de precios unitario correspondiente, y así con todos los rubros que se hayan definido para el proyecto. Con los análisis de precios ya bien hechos y corregidos, se pasa el nombre y unidad y precio unitario al presupuesto. Finalmente, se realiza el cálculo de volúmenes para ese rubro, lo que da como resultado la cantidad que se ocupa de este rubro.

Se multiplica la cantidad por el precio unitario por cada rubro y al final se suma. Si el proyecto es para la parte privada, entonces debe tener IVA y se agrega a esta sumatoria. Con todo el presupuesto listo, o a veces antes, se redactan las especificaciones. La verdad no importa en que punto de hagan, pero estas deben existir, ya que son la guía o lenguaje común entre el contratista, fiscalizador y el cliente.

El problema viene cuando hay que cambiar algo, o se vinculan mal los archivos...algo que suele pasar, ya que algunos proyectos para el sector público pueden tener hasta cientos de rubros.

5. Respecto a los indirectos, ¿Tiene un valor asignado o puede variar entre presupuestos?

Sí, los indirectos son muy dependientes del presupuesto que se esté realizando y del cliente que lo solicite. Por eso debe ser editable el porcentaje asignado, por que claro...los indirectos y la utilidad son porcentajes del precio del rubro, más no valores fijos.

6. ¿Los precios almacenados en las diferentes bases van directamente al análisis de precios unitarios, o reciben algún tratamiento, modificación o proceso previo?

No, la información que esté en la base es la que se coloca directamente en el presupuesto, esa es la idea de tener las bases. Lo que si, deben existir 2 bases, una para

precios de materiales y equipos con IVA y otra sin IVA. Así es como manejamos ahorita. Si el presupuesto es para la parte privada, obviamente se paga IVA en las compras, pero si el presupuesto es para el sector público, por ley no se debe incluir IVA en equipos ni materiales. La mano de obra no importa, porque se toman los valores de las tablas de la CAMICON, qué es la que pone los salarios de los maestros.

7. ¿Prefiere que la aplicación sea accesible desde dispositivos móviles?

Podría ser una funcionalidad interesante el poder ver análisis desde el teléfono, pero no es necesario. La mayor parte del tiempo trabajamos en la computadora, ya sean las de escritorio o laptop, por lo que pasamos dibujamos en el AutoCAD, o trabajamos con Excel.

8. ¿Desea sacar la información del sistema, es decir exportar los presupuestos?

Si, es importante poder sacar los datos. Sería ideal poder sacar un PDF y un Excel. El PDF sería para presentar el resumen de los cálculos realizados, y el Excel para compartir un archivo fácilmente editable. La mayoría de ingenieros conocen más el uso de Excel y se debe poder compartir con ellos los datos.

9. ¿Qué nivel de integración considera necesario con otras herramientas que utiliza, como Excel?

Como dije anteriormente, con una exportación a Excel sería suficiente, ya que se supone que el sistema ya me va a dejar hacer las cosas que hacía en Excel.

10. ¿Qué medidas de seguridad considera importantes para proteger la información de los presupuestos y los datos de la obra?

Bueno, entiendo que como la aplicación va a verse en un navegador y todos vamos a acceder al mismo lugar, debe verificar quien está accediendo a través de un usuario y contraseña. No quisiera que luego los archivos estén hechos un rebulicio, mezclado lo de una y otra persona, o que por ejemplo se modifiquen sin querer trabajos ajenos.

Anexo B. Presupuesto exportado, formato SERCOP - PDF

INSTITUCIÓN:

PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara

UBICACIÓN: Ambato

OFERENTE: EMCO (Empresa de construcciones)

ELABORADO: 17/05/2025

TABLA DE DESCRIPCIÓN DE RUBROS, UNIDADES, CANTIDADES Y PRECIOS

N°	<u>Rubro / Descripción</u>	<u>Unidad</u>	<u>Cantidad</u>	<u>Precio Unitario</u>	<u>Precio Total</u>
ACERAS Y BORDILLOS					
1	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	795.00	1.04	826.80
2	Replanteo y nivelación de vías, equipo de precisión	Km	0.25	605.56	151.39
3	Excavación sin clasificar en tierra a mano	m3	295.80	3.77	1,115.17
4	Hormigón simple f'c=210 kg/cm2, concretera, en bordillo tipo americano, inc. Encofrado y desencofrado	m3	51.76	137.06	7,094.23
5	Relleno compactado mecánico (mat. excavación, compactador mecánico plancha)	m3	159.00	4.70	747.30
6	Relleno compactado mecánico (base clase 2, plancha)	m3	159.00	16.70	2,655.30
7	Hormigón simple f'c=210 kg/cm2, en Acera, e=7cm, con concretera	m2	795.00	15.26	12,131.70
8	Hormigón ciclópeo, en Muros, 70% H.S. f'c=210 kg/cm2, con concretera, 30% Piedra	m3	9.80	113.85	1,115.73
ALCANTARILLADO					
9	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	250.00	1.04	260.00
10	Replanteo y nivelación para alcantarillado, equipo de precisión	Km	0.25	424.33	106.08
11	Excavación de zanja alcantarillado, máquina, H=0.00 - 2.75 m	m3	228.60	3.51	802.39
12	Rubro con vacíos	m3	1.50	8.86	13.29
TOTAL:					27,019.38
IVA: 15.00%					4,052.91
TOTAL PROYECTO:					31,072.29

ELABORADO: 17/05/2025

Anexo C. Análisis de precios unitarios exportado, formato SERCOP – PDF

PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara
ELABORADO: 17/05/2025

ANÁLISIS DE PRECIOS UNITARIOS

N° 1

RUBRO: Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo

UNIDAD: m²

EQUIPO DESCRIPCIÓN	CANTIDAD A	TARIFA B	COSTO HORA C=A*B	RENDIMIENTO R	COSTO D=C*R
Herramienta Menor (5% M.O.)					0.03419
Retroexcavadora	0.0125	45.00	0.56	0.16	0.09
Volqueta	0.0125	30.00	0.38	0.16	0.06
SUBTOTAL M					0.18419

MANO DE OBRA DESCRIPCIÓN	CANTIDAD A	JORNAL/HR B	COSTO HORA C=A*B	RENDIMIENTO R	COSTO D=C*R
CHOFER: Volquetas (Estr. Oc. C1)	0.0125	6.08	0.08	0.16	0.01216
Peón - E2	1.00	4.14	4.14	0.16	0.6624
Operador de Retroexcavadora - C1	0.0125	4.65	0.06	0.16	0.0093
SUBTOTAL N					0.68386

MATERIALES DESCRIPCIÓN	UNIDAD	CANTIDAD A	PRECIO UNITARIO B	COSTO C=A*B
SUBTOTAL O				0.00

TRANSPORTE DESCRIPCIÓN	UNIDAD	CANTIDAD A	TARIFA B	COSTO C=A*B
SUBTOTAL P				0.00

TOTAL COSTO DIRECTO (M+N+O+P)	0.86805
INDIRECTOS (%)	0.00%
UTILIDAD (%)	20.00%
PRECIO TOTAL DEL RUBRO	1.04166
VALOR UNITARIO	1.04

ESTOS PRECIOS INCLUYEN IVA

Anexo D. Cubicación exportada, formato EMCO – PDF

PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara
ELABORADO: 17/05/2025

RESUMEN DE CÁLCULO DE VOLÚMENES

N° 1

RUBRO: Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo

UNIDAD: m2

ÍTEM	D1	D2	PRODUCTO	NO. ELEMENTOS	TOTAL
Calle A lado derecho e izquierdo	60.00	1.50	90.00	2.00	180.00
Calle A curva	10.00	1.50	15.00	2.00	30.00
Calle B lado derecho e izquierdo	40.00	1.50	60.00	2.00	120.00
Calle B curva de retorno	50.00	1.50	75.00	1.00	75.00
Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00
Calle C curva de retorno	50.00	1.50	75.00	1.00	75.00
Calle C curva	10.00	1.50	15.00	2.00	30.00
Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00
Calle C descuento tramo de calle "B"	10.00	1.50	15.00	-1.00	-15.00
				TOTAL:	795.00

ELABORADO: 17/05/2025

ESPECIFICACIONES TÉCNICAS

PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara

ELABORADO: 17/05/2025

1. LIMPIEZA DE TERRENO (A MÁQUINA), ELIMINACIÓN CAPA VEGETAL, INCLUYE DESALOJO

UNIDAD:
m2



Imagen referencial

a. DEFINICIÓN.

Consiste en la remoción y desalojo de vegetación existente, de capa vegetal, escombros y otros materiales extraños en un espesor mínimo de 30 cm; que afecten única y exclusivamente el área de construcción del proyecto.

b. ESPECIFICACIONES.

Se procederá a la limpieza del terreno, remoción y eliminación de capa vegetal del área definida por la extensión total del proyecto o la autorizada por Fiscalización, incluye la nivelación del terreno de manera que permita la ejecución de los trabajos de construcción. Esta actividad se realizará con las herramientas y equipo adecuados.

Todo material resultado de la limpieza, el contratista deberá desalojar hacia los lugares autorizados legalmente como botaderos de escombros; lo que estará sujeto a la calificación por parte de Fiscalización.

c. ENSAYOS Y TOLERANCIAS.

Anexo F. Presupuesto exportado, formato SERCOP – Excel

A1					
	A	B	C	D	E
1					
2		INSTITUCIÓN:			
3		PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara			
4		UBICACIÓN: Ambato			
5		OFERENTE: EMCO (Empresa de construcciones)			
6		ELABORADO: 17/05/2025			
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					

TABLA DE DESCRIPCIÓN DE RUBROS, UNIDADES, CANTIDADES Y PRECIOS					
N°	Rubro / Descripción	Unidad	Cantidad	Precio Unitario	Precio Total
ACERAS Y BORDILLOS					
1	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	795.00	1.04	826.80
2	Replanteo y nivelación de vías, equipo de precisión	Km	0.25	605.56	151.39
3	Excavación sin clasificar en tierra a mano	m3	295.80	3.77	1,115.17
4	Hormigón simple f'c=210 kg/cm2, concretera, en bordillo tipo americano, inc. Encofrado y desencofrado	m3	51.76	137.06	7,094.23
5	Relleno compactado mecánico (mat. excavación, compactador mecánico plancha)	m3	159.00	4.70	747.30
6	Relleno compactado mecánico (base clase 2, plancha)	m3	159.00	16.70	2,655.30
7	Hormigón simple f'c=210 kg/cm2, en Acera, e=7cm, con concretera	m2	795.00	15.26	12,131.70
8	Hormigón ciclópeo, en Muros, 70% H.S. f'c=210 kg/cm2, con concretera, 30% Piedra	m3	9.80	113.85	1,115.73
ALCANTARILLADO					
9	Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo	m2	250.00	1.04	260.00
10	Replanteo y nivelación para alcantarillado, equipo de precisión	Km	0.25	424.33	106.08
11	Excavación de zanja alcantarillado, máquina, H=0.00 - 2.75 m	m3	228.60	3.51	802.39
12	Rubro con vacíos	m3	1.50	8.86	13.29
				TOTAL:	27,019.38
				IVA: 15.00%	4,052.91
				TOTAL PROYECTO:	31,072.29

Anexo G. Análisis de precios unitarios exportado, formato SERCOP – Excel

A1							
	A	B	C	D	E	F	G
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							

ANÁLISIS DE PRECIOS UNITARIOS

N° 1

Rubro: Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo

Unidad: m2

EQUIPO DESCRIPCIÓN	CANTIDAD A	TARIFA B	COSTO HORA C=A*B	RENDIMIENTO R	COSTO D=C*R
Herramienta Menor (5% M.O.)					0,034193
Retroexcavadora	0.0125	45.00	0.56	0.16	0.09
Volqueta	0.0125	30.00	0.38	0.16	0.06
SUBTOTAL M					0.18419

MANO DE OBRA DESCRIPCIÓN	CANTIDAD A	JORNAL/HR B	COSTO HORA C=A*B	RENDIMIENTO R	COSTO D=C*R
CHOFER: Volquetas (Estr. Oc. C1)	0.0125	6.08	0.08	0.16	0.01216
Peón - E2	1.00	4.14	4.14	0.16	0.6624
Operador de Retroexcavadora - C1	0.0125	4.65	0.06	0.16	0.0093
SUBTOTAL N					0.68386

MATERIALES DESCRIPCIÓN	UNIDAD	CANTIDAD A	PRECIO UNITARIO B	COSTO C=A*B
SUBTOTAL O				0.00

TRANSPORTE DESCRIPCIÓN	UNIDAD	CANTIDAD A	TARIFA B	COSTO C=A*B
SUBTOTAL P				0.00

TOTAL COSTO DIRECTO (M+N+O+P)	0.86805
INDIRECTOS (%)	0.00%
UTILIDAD (%)	20.00%
PRECIO TOTAL DEL RUBRO	1.04166
VALOR UNITARIO	1.04

ESTOS PRECIOS INCLUYEN IVA

Anexo H. Cubicación exportada, formato EMCO – Excel

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

PROYECTO: Construcción de infraestructura para el proyecto de división familia Morales Guevara

ELABORADO: 17/05/2025

RESUMEN DE CÁLCULO DE VOLÚMENES

N° 1

Rubro: Limpieza de terreno (a máquina), eliminación capa vegetal, incluye desalojo

Unidad: m2

ÍTEM	D1	D2	PRODUCTO	NO. ELEMENTOS	TOTAL
Calle A lado derecho e izquierdo	60.00	1.50	90.00	2.00	180.00
Calle A curva	10.00	1.50	15.00	2.00	30.00
Calle B lado derecho e izquierdo	40.00	1.50	60.00	2.00	120.00
Calle B curva de retorno	50.00	1.50	75.00	1.00	75.00
Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00
Calle C curva de retorno	50.00	1.50	75.00	1.00	75.00
Calle C curva	10.00	1.50	15.00	2.00	30.00
Calle C lado derecho e izquierdo	50.00	1.50	75.00	2.00	150.00
Calle C descuento tramo de calle "B"	10.00	1.50	15.00	-1.00	-15.00
				TOTAL:	795.00

ELABORADO: 17/05/2025