



UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INGENIERÍA,  
INDUSTRIA Y CONSTRUCCIÓN.**

**CARRERA DE INGENIERÍA CIVIL**

**“PRESUPUESTOS PARA UNA VIVIENDA DE HASTA  
DOS PISOS EN HORMIGÓN O ACERO A TRAVÉS DE  
UNA APLICACIÓN”**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO CIVIL**

**AUTOR: ANABEL SOLANGE SOTO MERINO**

**DIRECTOR: CARLOS JULIO CORDERO CABRERA**

**CUENCA – ECUADOR**

**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INGENIERÍA,  
INDUSTRIA Y CONSTRUCCIÓN**

**CARRERA DE INGENIERÍA CIVIL**

“PRESUPUESTOS PARA UNA VIVIENDA DE HASTA DOS PISOS EN  
HORMIGÓN O ACERO A TRAVÉS DE UNA APLICACIÓN”

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO CIVIL**

**AUTOR: ANABEL SOLANGE SOTO MERINO**

**DIRECTOR: CARLOS JULIO CORDERO CABRERA**

**CUENCA - ECUADOR**

**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**

## DECLARATORIA DE AUTORIDAD Y RESPONSABILIDAD

Anabel Solange Soto Merino, portadora de la cédula de ciudadanía N° 1104069669. Declaro ser autora de la obra: “Presupuestos para una vivienda de hasta dos pisos en hormigón o acero a través de una aplicación”, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 06 de septiembre de 2023



F:.....

Anabel Solange Soto Merino

C.I. 1104069669

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Anabel Solange Soto Merino, bajo mi supervisión.



---

**ING. MSC. CARLOS JULIO CORDERO**

**DIRECTOR DE TRABAJO DE TITULACIÓN**

## **DEDICATORIA**

Dedico con todo mi corazón y esfuerzo esta tesis a mi hijo Ian, mi mayor inspiración y felicidad en mi vida, quien me acompañó desde sus 21 días de nacido en el largo camino para cumplir mi sueño. Con esto quiero demostrarte que no importan las piedras en el camino, si le pones dedicación, esfuerzo y amor a todo lo que te propones, lo lograrás. Quiero que sepas que has sido una pieza fundamental en este camino hacia el logro de mi meta. Te doy mi trabajo en agradecimiento a toda tu paciencia y comprensión a tu corta edad durante estos años acompañando a mamá a estudiar.

## **AGRADECIMIENTOS**

Agradezco a mis padres por ser un gran apoyo a lo largo de mi vida, quienes con su amor y todo su esfuerzo posible me permitieron llegar hasta aquí, un logro más alcanzado, gracias por inculcar en mí un claro ejemplo de valentía y dedicación, por no temer a aquellas adversidades que se presenten porque Dios estará conmigo siempre guiándome junto a vosotros. A mi madre Ligia Merino por enseñarme a ser una mujer fuerte e independiente, y a mi padre Jose Soto por su fuerte apoyo para mi crecimiento intelectual.

A mi tía Libia Soto por ser una segunda madre, quien estuvo para mí desde mi niñez y en especial a lo largo de mi carrera universitaria de forma incondicional.

Y a mi hermana Nayeli Soto, quien, a pesar de la diferencia de edad, siempre estuvo ahí para motivarme, animarme y alentarme en cada paso que daba hacia mi objetivo.

## RESUMEN

Esta tesis consiste en el desarrollo de una aplicación para generar precios estimados para una casa de hasta dos pisos elaborada con hormigón armado o acero, con el fin de resolver la problemática que como ingenieros civiles se experimenta al enfrentarse con los clientes, ya que los mismos, solicitan el presupuesto estimado de sus futuros domicilios de forma inmediata.

Los datos para la aplicación se obtienen de fuentes públicas y privadas, como informes de mercado y datos de la industria. En el estudio se analizan los factores que influyen en el precio de una casa, como son: el material de la estructura y el área de construcción. Utilizando estos factores, se creará una base de datos de 635 viviendas que comprenden áreas desde 36 m<sup>2</sup> como área mínima hasta 120 m<sup>2</sup> como área máxima, que permita mediante un análisis estadístico establecer las cantidades de los rubros que intervienen en la construcción de la vivienda.

Los precios unitarios son creados con base al mercado local y salarios vigentes de mano de obra. La aplicación se desarrolla con la ayuda de la creación de un algoritmo que es el encargado de calcular la cantidad de obra para cada rubro perteneciente al tipo de vivienda y así se permita generar estimaciones de precios para las casas. El objetivo final del proyecto es ofrecer a los usuarios una forma rápida de calcular el precio estimado de sus propiedades con un rango de error del costo real que no supere en  $\pm 15\%$  el estimado.

*Palabras clave:* vivienda, presupuesto, aplicación, hormigón armado, acero.

## ABSTRACT

This thesis focuses on developing an application to generate estimated prices for housing of up to two floors constructed with reinforced concrete or steel to solve a common issue faced by civil engineers when dealing with clients who require immediate estimated budgets for their prospective houses.

Application data is obtained from public and private sources, including market reports and industry information. The study analyzes the key factors influencing housing prices, such as constructions materials and building area. To facilitate this analysis, a database encompassing 635 houses is compiled, ranging in size from a minimum area of 36 m<sup>2</sup> to a maximum area of 120 m<sup>2</sup>. This dataset enables statistical analysis to determine the quantities of materials and labor involved in housing construction.

Unit prices are established based on the local market rates and prevailing labor wages. The application is developed with the help of creating an algorithm responsible for calculating the work amount for each item belonging to the type of housing, thus making it possible to generate price estimates for them. The project's ultimate aim is to provide users with a quick way to calculate the estimated price of their properties with an error range of the actual cost not exceeding  $\pm 15\%$  of the estimated.

*Keywords:* housing, budget, application, reinforced concrete, steel

## TABLA DE CONTENIDO

Resumen .....	5
Palabras clave .....	5
CAPÍTULO I.....	14
1.1 Introducción.....	14
1.2 Delimitación del problema .....	15
1.3 Objetivos.....	16
<i>1.3.1 Objetivo general:</i> .....	16
<i>1.3.2 Objetivos generales:</i> .....	16
1.4 Justificación .....	16
CAPÍTULO II.....	18
2.1 Marco teórico.....	18
<i>2.1.1 Definición de elementos estructurales.</i> .....	18
<i>2.1.2 Conceptos de costos de construcción.</i> .....	21
<i>2.1.3 Conceptos que comprenden la aplicación.</i> .....	22
<i>2.1.4 Herramientas para la creación de la aplicación.</i> .....	22
<i>2.1.4 Normativa empleada para las viviendas.</i> .....	23
2.2. Estado del arte. ....	25
CAPÍTULO III .....	28
3.1 Materiales y métodos.....	28
<i>3.1.1 Detalle de la base de datos</i> .....	28
<i>3.1.2 Fórmula para el cálculo de la muestra de la población.</i> .....	30

3.1.3 Desarrollo de la aplicación.....	31
3.1.4 Estructura de la aplicación.....	31
3.1.5 Diagrama de flujo.....	34
3.1.6 Cálculo de la cantidad de obra de cada rubro.....	35
3.1.7 Cálculo del presupuesto.....	37
3.1.8 Cálculo del porcentaje de error de la aplicación.....	38
CAPÍTULO IV.....	39
4.1 Resultados.....	39
4.1.1 Resultados obtenidos estadísticamente de la base de datos.....	39
4.1.2 Resultados obtenidos de la aplicación de presupuestos para vivienda.....	45
4.1.3 Algoritmo para la creación del presupuesto.....	49
4.1.4 Resultados del cálculo del porcentaje de error de la aplicación.....	50
4.1.5 Resultado de la muestra de la población.....	53
4.1.6 Corrida de la aplicación.....	53
4.1.6.1 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión de pago en acero:.....	53
4.1.6.2 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión gratuita en acero:.....	54
4.1.6.3 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión gratuita en hormigón:.....	55
4.1.6.4 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión de pago en hormigón:.....	55
CAPÍTULO V.....	56

5.1 Conclusiones.....	56
5.2 Recomendaciones .....	57
BIBLIOGRAFÍA .....	58
GLOSARIO .....	61
ANEXOS .....	62
Código de la aplicación web:.....	62
Código del servidor de la base de datos.....	69

## Tabla de contenido de ilustraciones

Ilustración 1. Base de datos de la cantidad de obra de las viviendas tipo. Fuente: Elaboración propia.....	28
Ilustración 2. Base de datos de precios unitarios de las viviendas tipo. Fuente: Elaboración propia .....	29
Ilustración 3. Base de datos de dimensiones de las áreas principales de una vivienda. Fuente: Elaboración propia.....	30
Ilustración 4. Estructura de la aplicación. Fuente: Elaboración propia. ....	33
Ilustración 5. Diagrama de flujo de la aplicación. Fuente: Elaboración propia .....	34
Ilustración 6. Código para hacer el cálculo del presupuesto. Fuente: Elaboración propia .....	37
Ilustración 7. Ventana de rubros de la aplicación MyHousingCosts. Fuente: Elaboración propia.....	46
Ilustración 8. Ventana de casas de la aplicación MyHousingCosts. Fuente: Elaboración propia .....	46
Ilustración 9. Ventana de presupuestos de la aplicación MyHousingCosts. Fuente: Elaboración propia.....	47
Ilustración 10. Ejemplo de la versión gratuita de la aplicación MyHousingCosts. Fuente: Elaboración propia.....	48
Ilustración 11. Ejemplo de la versión de pago de la aplicación MyHousingCosts. Fuente: Elaboración propia.....	48
Ilustración 12. Ejemplo de cómo eliminar los rubros que no se requieran en la ventana de presupuestos. Fuente: Elaboración propia .....	48
Ilustración 13. Presupuesto en versión de pago de vivienda en acero- Fuente: Elaboración propia.....	54

Ilustración 14. . Presupuesto en versión gratuita de vivienda en acero- Fuente: Elaboración propia.....	54
Ilustración 15. Presupuesto en versión gratuita de vivienda en hormigón- Fuente: Elaboración propia.....	55
Ilustración 16. Presupuesto en versión de pago de vivienda en hormigón- Fuente: Elaboración propia.....	55

### Tabla de contenido de tablas:

Tabla 1: Tabla para crear las gráficas de las cuales se obtienen las ecuaciones para el cálculo de la cantidad de obra. Fuente: Elaboración propia.....	36
Tabla 2. Resumen de precio unitario de los siete rubros más importantes. Fuente: Elaboración propia.....	45
Tabla 3: Tabla del cálculo del porcentaje de error de las viviendas en hormigón. Fuente: Elaboración propia.....	51
Tabla 4: Tabla del cálculo del porcentaje de error para las viviendas en acero Fuente: Elaboración propia.....	52

## Tabla de contenido de gráficas:

Gráfica 1: Porcentaje de viviendas según el área de cada vivienda. Fuente: Elaboración propia .....	40
Gráfica 2: Porcentaje de viviendas según el material. Fuente: Elaboración propia .....	40
Gráfica 3. Gráfica estadística de cantidad de obra y precio unitario para hormigón simple $f'c= 210 \text{ kg/cm}^2$ . Fuente: Elaboración propia.....	42
Gráfica 4: Gráfica estadística de cantidad de obra y precio unitario para acero de refuerzo. Fuente: Elaboración propia.....	42
Gráfica 5: Gráfica estadística de cantidad de obra y precio unitario para cubierta de fibrocemento con 2 caídas. Fuente: Elaboración propia.....	42
Gráfica 6: Gráfica estadística de cantidad de obra y precio unitario para pintura para interiores, 2 manos. Fuente: Elaboración propia .....	43
Gráfica 7: Gráfica estadística de cantidad de obra y precio unitario para pintura para exteriores, 2 manos. Fuente: Elaboración propia .....	43
Gráfica 8: Gráfica estadística de cantidad de obra y precio unitario para mampostería de bloque 15 x 20 x 40 cm. Fuente: Elaboración propia.....	44
Gráfica 9: Gráfica estadística de cantidad de obra y precio unitario para enlucido con mortero 1:3. Fuente: Elaboración propia.....	44
Gráfica 10: Porcentaje de error para las viviendas en hormigón. Fuente: Elaboración propia .....	52
Gráfica 11: Porcentaje de error para las viviendas en acero. Fuente: Elaboración propia .....	53

# CAPÍTULO I

## 1.1 INTRODUCCIÓN

El costo de una vivienda es un factor importante a tener en cuenta a la hora de realizar cualquier inversión. Puede ser difícil evaluar con precisión el valor de una propiedad, especialmente si no está familiarizado con el mercado y carece de conocimientos especializados. Para facilitar este proceso, se ha desarrollado una aplicación que ayuda a los usuarios a calcular rápidamente el precio estimado de las casas de hasta dos pisos construidas con hormigón armado o acero desde sus teléfonos celulares.

Esta aplicación tiene en cuenta muchos factores que afectan los precios, como los materiales utilizados en la construcción, la calidad de los acabados, los elementos de diseño y las áreas de construcción, entre otros; lo que permite, estimaciones de costos más precisas que las que podrían proporcionar los métodos tradicionales por sí solos. Esta información servirá como una guía para los clientes que buscan invertir en sus futuras casas para que puedan tomar decisiones informadas sobre qué tipo de casa deben comprar sin tener que preocuparse por gastar en exceso debido a estimaciones inexactas o confiar únicamente en conjeturas.

En general, el objetivo de la creación de esta aplicación es simple: crear una aplicación móvil fácil de usar que brinde a las personas acceso a información confiable sobre los costos de vivienda para que puedan avanzar con confianza en las decisiones de compra sabiendo con precisión cuánto dinero deben reservar para renovaciones u otros proyectos relacionados con su inversión en sus nuevas viviendas.

## **1.2 DELIMITACIÓN DEL PROBLEMA**

La investigación se extendería hasta el punto en que se logrará crear una aplicación que proporcione precios estimados y confiables para el costo de una vivienda unifamiliar, con acabados de gama media, y con una estructura ya sea de hormigón armado o acero. Esto implicaría recopilar datos relevantes sobre materiales, áreas, diseño, etc. que permitan proporcionar una valoración del costo. Además, esta investigación implicará 1) la creación de un algoritmo para el cálculo del costo, 2) la creación de una base de datos para almacenar los datos recopilados y 3) la creación de una interfaz de usuario que permita ingresar los datos relevantes. Esta información servirá como una guía para determinar un precio estimado de la obra.

La aplicación se delimitará en arrojar un presupuesto estimado únicamente para la construcción de casas de hasta dos pisos en hormigón armado o acero, utilizando zapatas aisladas como cimentación. Sin embargo, se espera que en el futuro se incluyan otros tipos de estructuras, como las de madera. Por lo tanto, este proyecto establecerá la cantidad y tipo de materiales necesarios para su construcción, incluyendo la cantidad de obra para los diferentes rubros que se propongan. Una vez se tenga la información, se deberá calcular el presupuesto estimado, para viviendas únicamente destinadas para la ciudad de Cuenca.

La normativa que se regirá para la construcción de las viviendas será la Normativa Ecuatoriana de la Construcción (NEC), específicamente la Normativa Ecuatoriana de la Construcción en Estructuras de Hormigón Armado (NEC-SE-HM), la Normativa Ecuatoriana de la Construcción en Estructuras de acero (NEC-SE-AC) y la Normativa Ecuatoriana de la Construcción en Geotecnia y Cimentaciones (NEC-SE-GC), puesto que en Ecuador es la normativa que generalmente, es empleada. En esta normativa se detalla el conjunto de regulaciones y normas técnicas que establecen los requisitos mínimos para

la construcción de edificaciones en Ecuador garantizando la seguridad, la calidad y la eficiencia de las mismas.

### **1.3 OBJETIVOS**

#### ***1.3.1 Objetivo general:***

- Desarrollar una aplicación que permita a los usuarios calcular de manera rápida el precio estimado de sus casas de dos pisos construidas con hormigón armado o acero desde la comodidad de sus teléfonos celulares.

#### ***1.3.2 Objetivos generales:***

- Recopilar datos relevantes sobre materiales, mano de obra, diseño, tamaños, etc. que permitan proporcionar una estimación precisa del costo, lo que les permitirá tomar decisiones informadas en cuanto al costo que tendrán sus futuras viviendas.
- Crear un algoritmo para el cálculo del costo, junto a una base de datos, para almacenar los datos recopilados; y una interfaz de usuario, que permita ingresar los datos relevantes; que servirá como una guía para determinar un precio estimado de la obra, lo que permitirá tomar decisiones informadas en cuanto al costo que tendrán sus futuras viviendas.
- Ofrecer una solución simple y fácil de usar que ayuda a los clientes a comprender los muchos factores que afectan los precios de las casas

### **1.4 JUSTIFICACIÓN**

La principal razón para investigar la creación de una aplicación para dar precios estimados del costo de una casa es ayudar a los usuarios a realizar estimaciones de costos más precisas y rápidas. Esto es particularmente importante en proyectos de construcción donde los tiempos de entrega son muy limitados y los errores en los presupuestos pueden tener consecuencias graves. Una aplicación con un algoritmo que tome en cuenta todos los factores relevantes podría ayudar a los usuarios a realizar evaluaciones de los costos

de las futuras viviendas en un tiempo razonable. Actualmente el sistema en el que se basan la mayoría de revistas especializadas en el tema, es desarrollar el diseño de una vivienda tipo y a través de esto proponer un costo promedio por metro cuadrado. Lo que causa una problemática debido a que este parámetro así obtenido no representa un dato estadístico representativo del costo real.

Además, esta aplicación ayudará en la planificación con base al capital disponible, si el proyecto es viable económicamente o no, dando una idea clara de la prefactibilidad del proyecto a través de una aplicación de teléfono.

## CAPÍTULO II

### 2.1 MARCO TEÓRICO

#### 2.1.1 Definición de elementos estructurales.

##### ▪ Tipos de estructuras:

Las estructuras constituyen uno de los más notables símbolos de la civilización. El objetivo del profesional en diseño estructural radica en la consecución de una estructura que sea económicamente viable, segura y capaz de satisfacer tanto los requisitos funcionales como los estéticos establecidos. (*Estructuras de acero. Introducción al diseño*, 2006)

- a) Estructuras de acero: una estructura de acero se define como una estructura construida de acero laminado, para uso como estructura de una edificación, cuyos componentes son en su mayoría, columnas y vigas de acero. (9.- *NEC-SE-AC-Estructuras-de-Acero.pdf*, 2014)
- b) Estructuras de hormigón: una estructura de hormigón se define como una estructura construida con hormigón premezclado o prefabricado para uso como estructura de una edificación. (8.- *NEC-SE-HM-Hormigón-Armado.pdf*, 2014)

##### ▪ Cimientos:

“La principal función de la cimentación es la transferencia adecuada al subsuelo de las cargas vivas y muertas de las edificaciones, las cargas sísmicas sostenidas por el edificio y las cargas sísmicas impartidas del propio suelo.” (7.- *NEC - SE - GC-Geotécnia-y-Cimentaciones.pdf*, 2014)

“Esta superestructura debe ubicarse sobre materiales que dispongan de características geomecánicas suficientes para garantizar la resistencia al corte, un idóneo desempeño

de la estructura para los asentamientos generados en el subsuelo y que la resistencia se mantenga por debajo de los estados límite y de servicio en los diseños de cimentaciones.”(7.- *NEC - SE - GC-Geotécnia-y-Cimentaciones.pdf*, 2014) De esta manera, se reduce el riesgo de desplazamiento, distorsión o grietas en paredes, losas, vigas y columnas.

- a) “Zapata aislada: se trata de una cimentación puntual que recibe un solo sistema de carga como son los pilares, se emplea en terreno firme y competente con una tensión de media a alta con asentamientos pequeños a moderados, además es la cimentación más económica sobre roca o terrenos con tensiones admisibles mayores a 0.15 N/mm<sup>2</sup>.” (Yepes Piqueras, 2019)

▪ **Acero de refuerzo:**

Es el refuerzo que se emplea en vigas, columnas, vigas y columnas confinantes, elementos colocados en el interior del muro y/o en el exterior del mismo, estará constituido por barras corrugadas según NTE INEN 2167, por malla de acero según NTE INEN 2209, por alambres corrugados laminados en frío según NTE INEN 1511 o por armaduras electro-soldadas por resistencia eléctrica de alambre de acero según NTE INEN 2209. Se admitirá el uso de barras lisas únicamente en estribos, mallas de alambre soldado o en conectores. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

▪ **Bloque de hormigón:**

Elemento de mampostería que a pesar de no cumplir con la especificación INEN 640, se usa regularmente para edificaciones de interés social y de bajo costo, cuya resistencia bruta no debe ser menor que  $f'_m = 3 \text{ MPa}$  (30 kg/cm<sup>2</sup>). (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Columna:**

Elemento estructural vertical reforzado con barras, armaduras electro-soldadas, alambres corrugados o lisos (estribos) de acero, cuya función es transmitir las cargas de la estructura al cimiento. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Losa de entrepiso o de cubierta:**

Elemento estructural horizontal o aproximadamente horizontal, maciza o con nervaduras, que trabaja en una o dos direcciones, de espesor pequeño en relación con sus otras dos dimensiones. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Mortero de pega:**

Mezcla plástica de materiales cementantes, agregados finos (arena) y agua utilizado para unir las piezas de mampostería. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Muro de mampostería:**

Elemento de colocación manual, de características pétreas y estabilidad dimensional que, unido con mortero configura la pared de mampostería. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Viga:**

Elemento estructural horizontal reforzado con barras, armadura electro-soldada, alambres corrugados o lisos (estribos) de acero, cuya función es soportar y transmitir las cargas transversales hacia los elementos de apoyo. (12.- *NEC-SE-VIVIENDA-parte-1.pdf*, 2014)

- **Cubierta:**

“Elemento estructural horizontal o aproximadamente horizontal, maciza o con nervaduras, con estructura metálica que trabaja en una o dos direcciones, de espesor

pequeño en relación con sus otras dos dimensiones elaborada con fibrocemento.”(Paúl & Acosta, 2008)

### ***2.1.2 Conceptos de costos de construcción.***

- **Rubros:**

Consiste en las actividades que se realizan en un proyecto de construcción orientada a la elaboración de bienes inmuebles, estando presentes en la prestación de servicios de construcción.(Soliz Baldovinos, 2013)

- **Unidad:**

En costos de construcción la unidad hace referencia a la medida con la que se contabiliza la cantidad de un elemento o material dentro de un rubro. (Soliz Baldovinos, 2013)

- **Cantidad de obra:**

El concepto de cantidad de obra alude a la cantidad de unidades de un rubro para llevar a cabo una obra especificada. (Soliz Baldovinos, 2013)

- **Precio unitario:**

Es el importe de remuneración individual que debe pagarse al contratista por cada unidad de trabajo finalizada y realizada de acuerdo con el proyecto, las especificaciones de construcción y estándares de calidad.(Soliz Baldovinos, 2013)

- **Presupuesto:**

El presupuesto de obra es el elemento primordial en la planeación y ejecución de una obra y se procede a la elaboración del mismo en el momento en que se tome la decisión de construir. Su objetivo es determinar los costos requeridos para completar un proyecto de acuerdo a unos planos y unas especificaciones técnicas brindadas por

el especialista. Puede definirse como la presentación ordenada de los resultados previstos de un plan o un proyecto.(Cáceres Tume, 2013)

### ***2.1.3 Conceptos que comprenden la aplicación.***

- Algoritmo: los algoritmos proporcionan una secuencia lógica para realizar tareas específicas como hacer un cálculo o hallar la solución de un tipo de problemas. (Sarría, 2017)
- Base de datos: sistema que permite el acceso a grandes cantidades organizadas e integradas de información de modo que sean accesibles individualmente por medios electrónicos o de otra forma. (Sarría, 2017)
- Interfaz de usuario: punto clave entre el usuario y el sistema que permite una comunicación fluida sin errores. (Sarría, 2017)

### ***2.1.4 Herramientas para la creación de la aplicación.***

- JavaScript es un lenguaje de programación utilizado en la creación de páginas web dinámicas, las mismas tienen como principal función la incorporación de interactividad, permitiendo la activación y desactivación de acciones al interactuar con elementos como botones y ventanas, así como la generación de mensajes de aviso dirigidos al usuario.(Eguíluz Pérez, 2009)
- Node.js es una plataforma de desarrollo de aplicaciones de servidor y redes de código abierto, basada en el lenguaje JavaScript. Su objetivo principal es proporcionar un entorno unificado para la creación de aplicaciones web completas, abarcando desde la capa de presentación y experiencia de usuario hasta la capacidad de conectarse con bases de datos. Esta plataforma permite a los desarrolladores aprovechar un único lenguaje de programación para construir aplicaciones web altamente funcionales y escalables.(JS, 2020)

- React es una biblioteca de JavaScript de código abierto desarrollada por Facebook, Inc. para construir interfaces de usuario. React proporciona una forma fácil de crear aplicaciones interactivas con interfaces ricas, utilizando componentes reutilizables. (Gackenheimer, 2015)
- Next.js es un marco de JavaScript de código abierto para la creación de aplicaciones web de servidor. Está basado en React, ofrece una solución única para la construcción de aplicaciones web y sitios web dinámicos escalables. Proporciona herramientas y características que permiten a los desarrolladores crear aplicaciones web de servidor de forma rápida y eficiente. (Sobre Next.js, 2020).
- MongoDB es un sistema de base de datos de código abierto, orientado a documentos, que ofrece una amplia variedad de herramientas para su administración. Asimismo, almacena datos en documentos estructurados en forma de matriz, que se organizan en colecciones. Esta base de datos permite almacenar y procesar grandes cantidades de datos de una manera eficiente, sin tener que preocuparse por el diseño de la base de datos. Además, MongoDB admite diferentes lenguajes de programación, como Java, JavaScript, Python, PHP, etc. (MongoDB Inc., 2020)

#### ***2.1.4 Normativa empleada para las viviendas.***

- **NEC-SE-GC:** establece las directrices y requisitos para el diseño, construcción y mantenimiento de estructuras y cimentaciones a fin de alcanzar una adecuada seguridad estructural. Asimismo, se ha diseñado para garantizar que se cumplan los requisitos de resistencia del suelo, estabilidad de taludes, estabilidad de laderas y resistencia a la corrosión. Esta norma incluye los requisitos para el diseño de estructuras de cimentación, la determinación de la resistencia del suelo, la

selección y diseño de materiales de cimentación, la aplicación de métodos de construcción y la aplicación de los procedimientos de control de calidad. También incluye los requisitos para la inspección y pruebas de cimentaciones, así como los criterios para las pruebas de campo. (*Capítulos de la NEC (Norma Ecuatoriana de la Construcción) – MIDUVI – Ministerio de Desarrollo Urbano y Vivienda, 2014*)

- **NEC-SE-HM:** establece los requisitos generales para el diseño y construcción de estructuras de hormigón armado en el Ecuador. Esta norma se aplica a edificios, estructuras de acero, puentes, túneles, muros, pozos, tanques, presas y otras estructuras. Establece requisitos de diseño, dimensionamiento, selección y verificación de materiales, formas de construcción, fabricación y montaje, así como los requisitos de control de calidad para el hormigón armado. También establece directrices para la inspección, ensayo, reparación y mantenimiento de las estructuras de hormigón armado. (*Capítulos de la NEC (Norma Ecuatoriana de la Construcción) – MIDUVI – Ministerio de Desarrollo Urbano y Vivienda, 2014*)
- **NEC-SE-AC:** establece los requisitos para la construcción de estructuras de acero. Esta norma se aplica a estructuras de acero utilizadas en edificios, puentes, torres y otras estructuras. Implanta los requisitos para la selección, diseño, fabricación, ensamblaje, instalación, mantenimiento y operación de estructuras de acero, además de los requisitos para la inspección, ensayos, certificación y aprobación de materiales para estructuras de acero, los requisitos para el control de calidad de los materiales de acero y de la construcción. (*Capítulos de la NEC (Norma Ecuatoriana de la Construcción) – MIDUVI – Ministerio de Desarrollo Urbano y Vivienda, 2014*)

## 2.2. ESTADO DEL ARTE.

En el campo de la ingeniería y la construcción, el uso de software especializado se ha vuelto esencial para garantizar la eficiencia y la precisión en el diseño, cálculo y presupuestación de proyectos. En este contexto, algunas herramientas destacadas son 1) CYPECAD, 2) TCQ, el 3) Generador de precios de CYPE, 4) InterPro, 5) Pro-excel.

Los tres primeros softwares mencionados son desarrollados por la reconocida empresa CYPE ofrecen soluciones integrales y avanzadas para los profesionales del sector, facilitando la realización de tareas clave en el proceso constructivo. (*Presupuestos y condiciones técnicas - ITeC - Instituto de Tecnología de la Construcción, 2015*)

CYPECAD es una potente aplicación de diseño y cálculo de estructuras que se ha consolidado como una referencia en el ámbito de la ingeniería civil y la arquitectura. Esta herramienta permite realizar el diseño estructural de edificios de manera eficiente, considerando las normativas y regulaciones vigentes. CYPECAD ofrece una amplia variedad de opciones para el análisis y diseño de estructuras, incluyendo elementos como pilares, vigas, losas y muros, así como la posibilidad de trabajar con distintos materiales y tipologías estructurales. Además, permite la realización de presupuestos proporcionando herramientas para la gestión y control de costos y para la creación de informes y documentación técnica. (*Manual del ingeniero de edificación. Guía para el cálculo de estructuras con CYPECAD., 2012*)

Por otro lado, TCQ (Trabajo, Costes y Calidad) es un software especializado en la gestión y control de costes en la construcción. Esta herramienta facilita la elaboración de presupuestos detallados y precisos, teniendo en cuenta tanto los materiales necesarios como la mano de obra y otros gastos asociados al proyecto. TCQ permite realizar mediciones, valoraciones y comparativas de precios de forma eficiente, brindando a los profesionales del sector una herramienta imprescindible para el control económico de las

obras y la toma de decisiones informadas. Gracias a sus diferentes módulos de aplicación, TCQ es capaz de gestionar conjuntamente y de forma integrada los datos técnicos, económicos y temporales que intervienen en el ciclo completo de la obra. (*Presupuestos y condiciones técnicas - ITeC - Instituto de Tecnología de la Construcción*, 2015)

En cuanto al Generador de precios de CYPE, es una herramienta que simplifica y agiliza el proceso de elaboración de presupuestos. Este software permite generar presupuestos de obras de manera rápida y precisa, basándose en una amplia base de datos que incluye precios actualizados de materiales, mano de obra y equipos, de diferentes países. El Generador de precios de CYPE es una valiosa herramienta para los profesionales del sector, ya que les permite optimizar el tiempo invertido en la elaboración de presupuestos y garantizar la coherencia y la exactitud de los mismos. (*Generador de precios de la construcción. Ecuador. CYPE Ingenieros, S.A.*, 2022)

El software InterPro provee servicios para gestionar de una forma fácil. Rápida y confiable proyectos de construcción. Despliega varios módulos de software tanto de cliente como de servidor. Su funcionalidad considera, en primer lugar, la realización de análisis de precios unitarios y presupuestos. Permite, además, elaborar una completa planificación de un proyecto tanto alcance, tiempo y costo. Asimismo, a partir del análisis de variación de línea base, ayuda a realizar el monitoreo y control de la obra. (InterPro, 2023)

ProExcel es un sistema para el análisis de costos de un proyecto, el cual tiene una interfaz muy parecida a la del Excel lo cual hace su uso mucho más sencillo, así sin tener una capacitación muy profunda sobre sus aplicaciones se puede llegar a entender la gran mayoría de sus funciones y realizar un análisis de precios unitarios y de presupuestos muy avanzado. Este programa tiene una ventaja muy amplia que se pueden exportar e importar

archivos desde el Excel, modificarlos sin problemas y así poder presentar las ofertas sin inconvenientes. (Felipe, 2018)

## CAPÍTULO III

### 3.1 MATERIALES Y MÉTODOS

#### 3.1.1 Detalle de la base de datos

La base de datos es fundamental para el desarrollo de la aplicación de generación de presupuestos destinada a la construcción de viviendas de acero u hormigón armado, por lo que se ha estructurado en tres tipos. Para su creación, se empleó el software Excel, que permitió registrar y organizar con eficiencia la información necesaria. Comprende un total de 635 viviendas, 16 capítulos y 140 rubros.

- El primer tipo de base de datos está conformada por 55 columnas y 160 filas. En las filas 3, 4, 6, 7, 8 de la columna 1 se contempla los rubros necesarios para llevar a cabo la construcción de las viviendas, especificando la cantidad de obra asignada a cada rubro que se presenta en las columnas 3, 4, 5; la cual es obtenida de presupuestos de viviendas tipo que han sido adquiridos de empresas públicas y privadas (ilustración 1).

Descripción	Unidad	CANTIDAD DE OBRA		
		Casa 1 A=121,20 m2	Casa 2 A= 60,60 m2	Casa 3 A= 100 m2
<b>AREA PROMEDIO</b>				
<b>OBRAS PRELIMINARES</b>				
Desbroce y limpieza del terreno	m2	120,00	60,00	96,00
Replanteo y nivelación	m2	60,00	30,00	96,00
<b>MOVIMIENTO DE TIERRAS</b>				
Cargada de material a máquina	m3	16,00	8,00	10,45
Cargada de material a mano	m3	3,00	1,50	1,39
Excavación a mano en suelo sin clasificar, profundidad entre 0 y 2m	m3	2,00	1,00	1,39

*Ilustración 1. Base de datos de la cantidad de obra de las viviendas tipo. Fuente: Elaboración propia.*

Así como su precio unitario (ilustración 2). Asimismo, se detalla el área de construcción que comprende las columnas 4,5,6 de la fila 1; correspondiente a cada vivienda tipo mencionada anteriormente, que ha sido empleada en la creación de la base de datos. Finalmente, en la columna 2, se muestra la unidad de medida de cada rubro.

	Descripción	Unidad	P.Unitario		
<b>RUBROS</b>			asa 1 A=1	asa 2 A= 100	asa 3 A=100
	<b>AREA PROMEDIO</b>				
	<b>OBRAS PRELIMINARES</b>				
	Desbroce y limpieza del terreno	m2	2,20	3,45	3,45
	Replanteo y nivelación	m2	1,34	1,79	1,79
<b>CAPÍTULO</b>	<b>MOVIMIENTO DE TIERRAS</b>				
	Cargada de material a máquina	m3	1,69	2,06	2,06
	Cargada de material a mano	m3	15,05	12,00	7,16
	Excavación a mano en suelo sin clasificar, profundidad entre 0 y 2m	m3	14,79	12,00	12,00

Ilustración 2. Base de datos de precios unitarios de las viviendas tipo. Fuente: Elaboración propia

- La segunda base de datos tiene como objetivo detallar aspectos específicos de las viviendas tipo de acero u hormigón armado, tales como las columnas, vigas, mampostería, losas, cubiertas, puertas y ventanas, así como los suministros correspondientes para las instalaciones eléctricas e hidrosanitarias. Estos datos han sido obtenidos a partir de los planos correspondientes a los prototipos de viviendas empleados como ejemplo. De esta manera, se establece una estructura

de datos detallada y bien organizada que garantiza la precisión en la elaboración de presupuestos y la eficiencia en la planificación de las obras.

- Por último, la tercera base de datos (ilustración 3) está compuesta por detalles específicos de las viviendas, como las dimensiones de las luces y las áreas asignadas a salones, baños, dormitorios y cocinas en cada uno de los ejemplos de vivienda utilizados. Esta información ha sido cuidadosamente registrada y organizada en una estructura de datos bien definida, que garantiza la precisión en los presupuestos.

	CASA A=100 m2		CASA A=36m2		CASA A=40,16 m2		CASA A= 72 m2	
Luz x	3,40	2,60	3,07	2,93	2,92	3,43	2,90	2,90
Luz y	5,00	8,00	2,93	2,93	2,93		2,93	2,93
A. Sala	31,68		8,06		10,05		7,98	
A. Cocina	10,35		6,33		10,05		4,93	
A. Baño	3,54		5,65		3,01		5,08	
A. Dormitorio	9,24		9,00		8,56		8,50	

*Ilustración 3. Base de datos de dimensiones de las áreas principales de una vivienda. Fuente: Elaboración propia.*

### **3.1.2 Fórmula para el cálculo de la muestra de la población.**

Es importante realizar este cálculo para comprobar cuál es la cantidad de viviendas que van a ser representativas para conseguir un resultado del presupuesto estimado que sea lo certero posible.

$$n = \frac{Z^2 * P * (1 - P)}{d^2}$$

### ***3.1.3 Desarrollo de la aplicación***

Para el desarrollo de la aplicación se emplea la metodología SCRUM (Sistema Razonablemente Unificado Método) en donde se permite trabajar para formular soluciones útiles en tecnología e informática moderna con herramientas avanzadas tales como:

- Para el lenguaje de programación se emplea JavaScript.
- Para el servidor se utiliza Node.js.
- Para el framework se usa React y Next.js.
- Finalmente, para la base de datos, se emplea MongoDB.

### ***3.1.4 Estructura de la aplicación***

En esta sección se pretende mostrar la representación gráfica que permite esquematizar un algoritmo, mostrando de manera visual los pasos o procesos requeridos para alcanzar una solución a un problema. A continuación, se presenta la estructura de la aplicación (ilustración 4).

La estructura de la aplicación se divide en dos partes fundamentales, por una parte, está el Frontend, que comprende a la interfaz de usuario, donde el usuario o cliente es quien va a interactuar con esta parte desde diferentes dispositivos, ya sean tabletas, teléfonos móviles u ordenadores, siempre y cuando se tenga acceso a internet. Esta interfaz de usuario será creada con la herramienta mencionada en el apartado anterior denominada, React.js, que es donde se va a mostrar opciones para poder ingresar o seleccionar parámetros determinados por cada vivienda.

Por otra parte, está el Backend, el mismo que comprende al servidor, es decir donde se van a encontrar alojados todos los servicios que van a hacer que funcione la interfaz de usuario para que la aplicación sea interactiva. Asimismo, se podrá guardar información para que se comuniquen con la base de datos que, por su parte, será donde se guardará la

información para que sea persistente y en un futuro se proceda a crear un registro de los presupuestos generados. Esta base de datos será creada con MongoDB y el servidor en Node.js.

La manera en la que estas partes mencionadas se van a comunicar será a través de un protocolo conocido como Protocolo HTTPS, este sirve para poder intercambiar información tanto de Frontend y Backend.

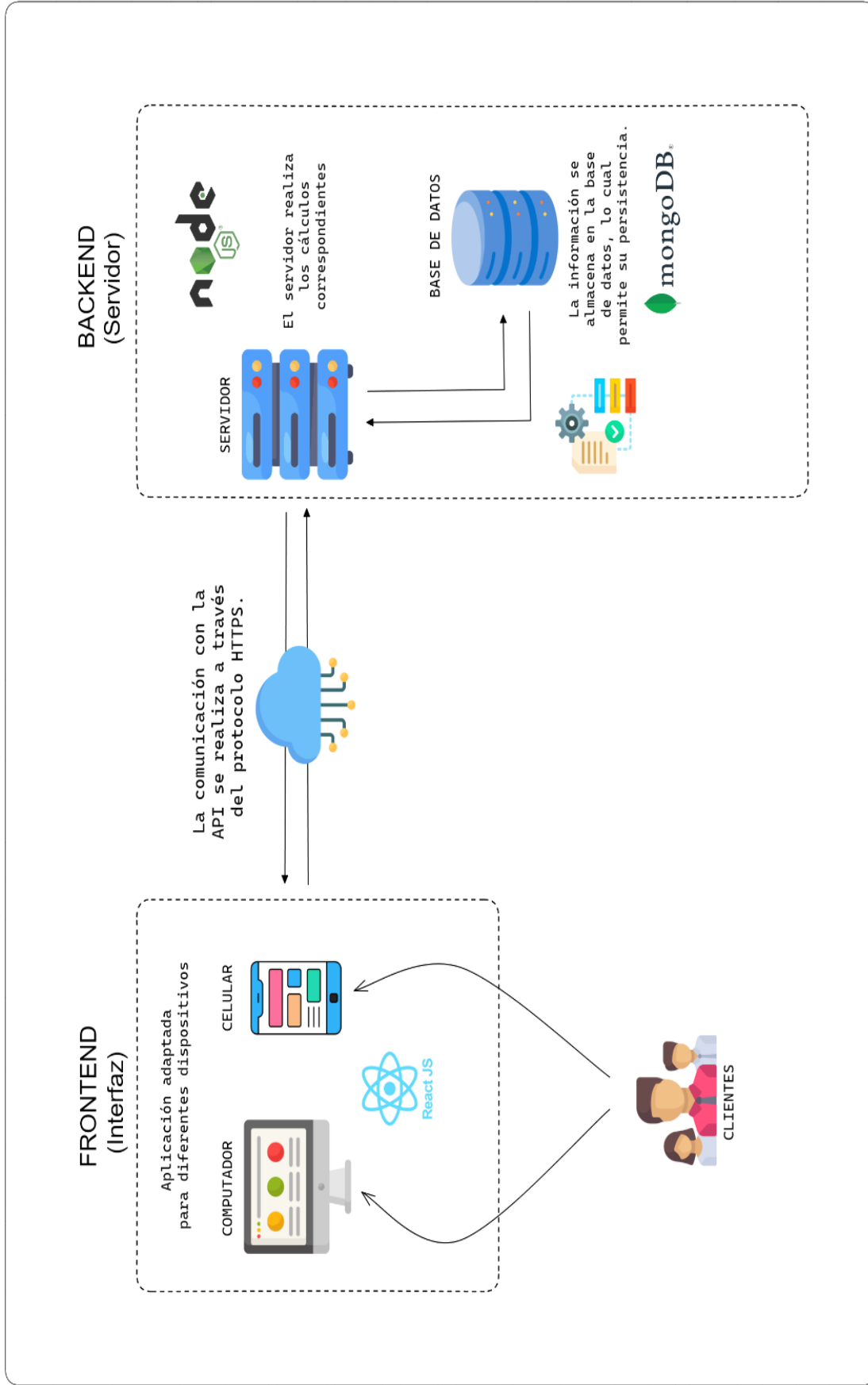


Ilustración 4. Estructura de la aplicación. Fuente: Elaboración propia.

### 3.1.5 Diagrama de flujo

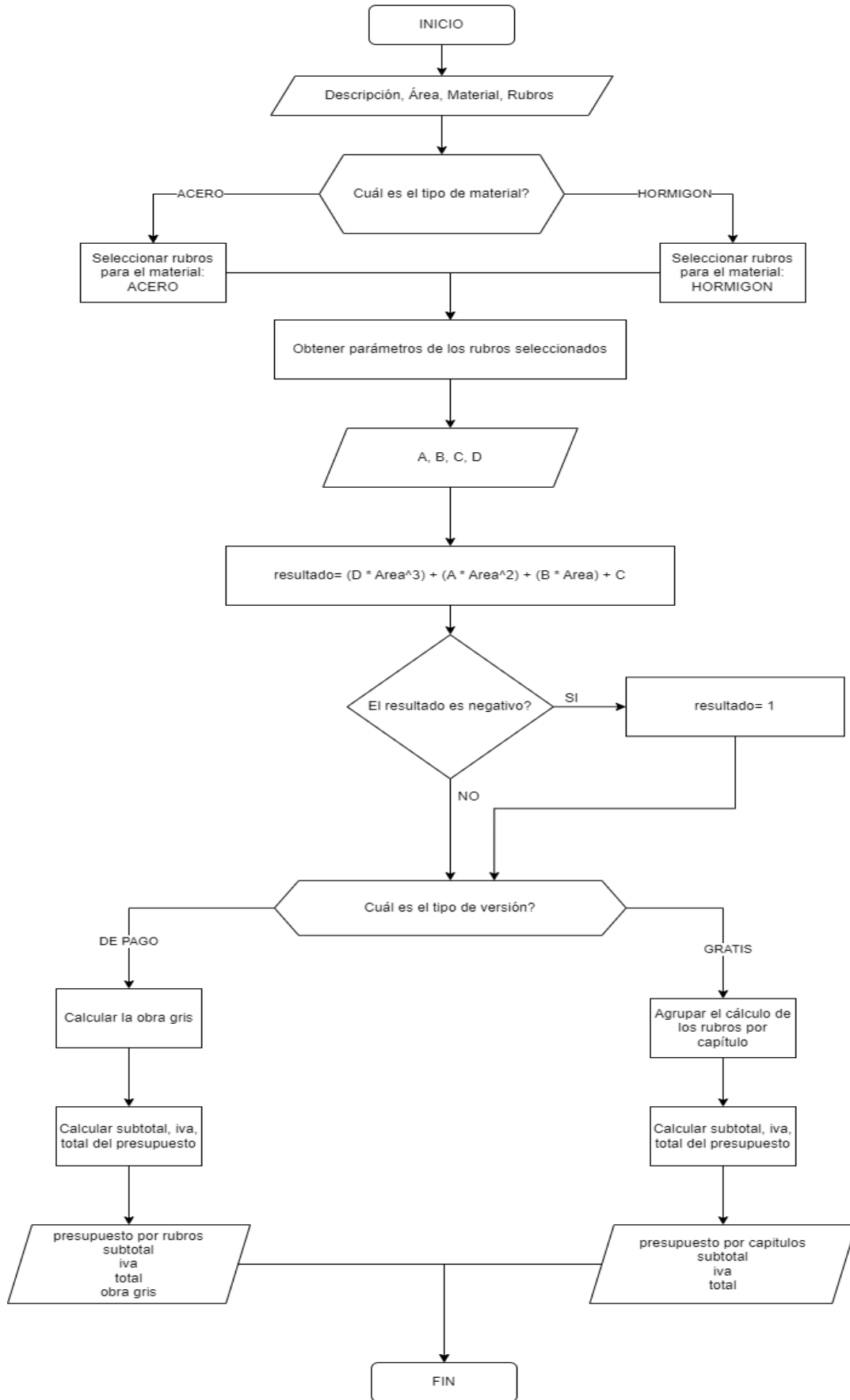


Ilustración 5. Diagrama de flujo de la aplicación. Fuente: Elaboración propia

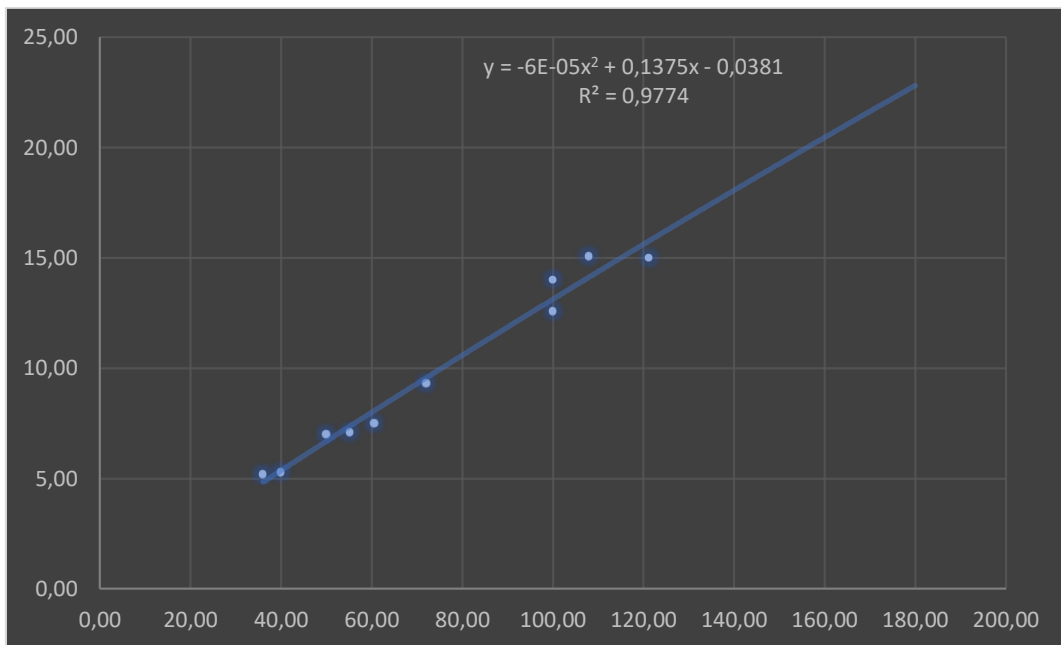
En el diagrama de flujo, (Ilustración 5) muestra el funcionamiento de la aplicación. Se empieza introduciendo los parámetros descripción, área y material, seguidamente, se muestra que, a la hora de seleccionar el material, se eligen rubros correspondientes a las estructuras de las viviendas, ya sean acero u hormigón. A continuación, se introducen los parámetros A, B, C y D correspondientes a las ecuaciones obtenidas para calcular la cantidad de obra, en el caso, donde el resultado de dichas ecuaciones sea negativo o menor a 0, el algoritmo dará como resultado 1 y continuará con el proceso, caso contrario se tomará el resultado obtenido de la ecuación. Finalmente, se escoge el tipo de versión que se requiera, ya sea de pago, en la cual se obtendrá un resultado del subtotal, IVA, presupuesto total, y obra gris, o gratuita, en la cual se agruparán los resultados por capítulos para la cantidad de obra, y se tendrá el subtotal, IVA y presupuesto total.

### ***3.1.6 Cálculo de la cantidad de obra de cada rubro***

El cálculo de la cantidad de obra para cada rubro es un proceso importante en el presupuesto de construcción. Para llevar a cabo este procedimiento, se realizan gráficas estadísticas (gráfica 11) con los datos recopilados de la base de datos sobre las viviendas tipo. Estas gráficas permiten determinar la línea polinómica tendencia que genera una ecuación para calcular el requerimiento exactamente necesario. Una vez obtenido este resultado, se ingresa al programa la ecuación y dentro del apartado "presupuestos" aparece automáticamente el resultado del cálculo en los diferentes rubros involucrados en dicha presupuestación. Esto facilita a los contratistas que buscan obtener rápidamente estimaciones precisas sobre cuánto les pueden costar sus proyectos sin tener un amplio conocimiento sobre los costos unitarios de cada rubro y la cantidad de obra de los mismos para cada vivienda.

ÁREA		121,20	60,60	100,00	100,00	36,00	108,00
Relleno compactado con material de mejoramiento (plancha vibratoria)	m3	15,00	7,50	5,08	12,58	5,02	15,06

Tabla 1: Tabla para crear las gráficas de las cuales se obtienen las ecuaciones para el cálculo de la cantidad de obra. Fuente: Elaboración propia



Gráfica 11. Gráfica ejemplo de cómo se obtuvo las ecuaciones para calcular la cantidad de obra. Fuente: Elaboración propia.

### 3.1.7 Cálculo del presupuesto

```
1 private calculateBudget({ area, entriesStats, isPaid }: { area: number; entriesStats: any[]; isPaid: boolean }) {
2   const statsOfPaid = entriesStats.map(({ parameterD, parameterA, parameterB, parameterC, measure, ...rest }) => {
3     // Resolve equation parameterD(x^3) + parameterA(x^2) + parameterB(x) + parameterC when x = area
4     const equation = parameterD * Math.pow(area, 3) + parameterA * Math.pow(area, 2) + parameterB * area + parameterC;
5
6     // If result of equation is negative, then amountWork is 1
7     const amountWork = equation < 0 ? 1 : equation;
8     // console.log({ equation, amountWork });
9     return {
10      ...rest,
11      measure,
12      // If measure is U or P measure then amountWork is the ceiling of amountWork
13      amountWork: measure === 'U' || measure === 'P' ? Math.ceil(amountWork) : amountWork,
14      total: amountWork * rest.unitPrice,
15    };
16  });
17  // console.log(statsOfPaid);
18
19  // If grayWork is true then sum amountWork and total
20  const grayWork = statsOfPaid.reduce(
21    (acc, stat) => {
22      if (stat.grayWork) {
23        acc.amountWork += stat.amountWork;
24        acc.total += stat.total;
25      }
26      return acc;
27    },
28    { amountWork: 0, total: 0 },
29  );
30
31  // Group by chapter and sum amountWork and total and return an array of objects
32  const statsOfFree = statsOfPaid.reduce<any[]>((acc, stat) => {
33    const chapter = stat.chapter;
34
35    const chapterEqual = acc.find((statAcc: any) => statAcc.chapter === chapter);
36
37    if (chapterEqual) {
38      chapterEqual.amountWork += stat.amountWork;
39      chapterEqual.total += stat.total;
40    } else {
41      acc.push({
42        chapter,
43        amountWork: stat.amountWork,
44        total: stat.total,
45      });
46    }
47
48    return acc;
49  }, []);
50
51  const subtotal = statsOfPaid.reduce((acc, stat) => acc + stat.total, 0);
52  // IVA 12%
53  const iva = subtotal * 0.12;
54
55  // Total with IVA
56  const totalWithIva = subtotal + iva;
57
58  // Total
59  const total = totalWithIva;
60
61  return {
62    approximateBudget: isPaid ? statsOfPaid : statsOfFree,
63    subtotal,
64    iva,
65    total,
66    grayWork: isPaid ? grayWork : null,
67  };
68 }
```

Ilustración 6. Código para hacer el cálculo del presupuesto. Fuente: Elaboración propia

El código anterior (ilustración 6) es una función de TypeScript llamada calcular presupuesto que toma un objeto con tres propiedades como parámetro: área, estadísticas de entrada. La función calcula el presupuesto para un área determinada y devuelve un objeto con el presupuesto aproximado, el subtotal, el IVA, el total y obra gris.

La función primero mapea sobre la matriz y calcula la cantidad de trabajo requerida para cada entrada según el área dada. Si el resultado de la ecuación es negativo, entonces la cantidad de trabajo se establece en 1. Si la medida del rubro es Unidad o Punto, entonces la cantidad de trabajo se redondea al número entero más cercano. Luego, el costo total de cada entrada se calcula multiplicando la cantidad de trabajo por el precio unitario.

Seguidamente, la función verifica si es verdadero y suma la cantidad de trabajo y el costo total para cada entrada que tiene establecido en verdadero. A continuación, agrupa las entradas por capítulo y suma la cantidad de trabajo y el costo total de cada capítulo.

El subtotal se calcula sumando el costo total de cada entrada. El IVA se calcula multiplicando el subtotal por 0,12. El total con IVA se calcula sumando el subtotal y el IVA. Finalmente, el total se establece en el total con IVA y la función devuelve un objeto con el presupuesto aproximado, el subtotal, el IVA, y el total.

### ***3.1.8 Cálculo del porcentaje de error de la aplicación***

Para corroborar el rango de error se empleó la siguiente fórmula. Dicha fórmula se empleará por separado para las viviendas de acero y para las viviendas de hormigón, comparando el resultado que arroja la aplicación con el presupuesto de las viviendas tipo que se usaron para la creación de la base de datos.

$$\%e = \frac{|VALOR APROXIMADO - VALOR REAL|}{VALOR REAL} * 100$$

## CAPÍTULO IV

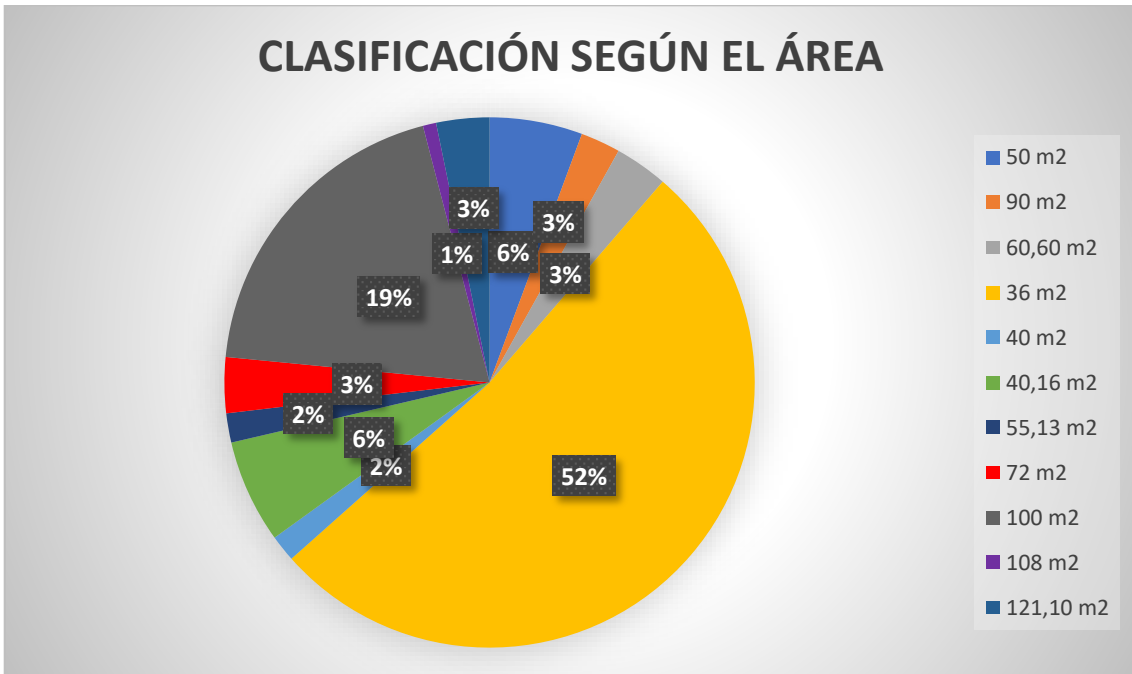
### 4.1 RESULTADOS

#### *4.1.1 Resultados obtenidos estadísticamente de la base de datos*

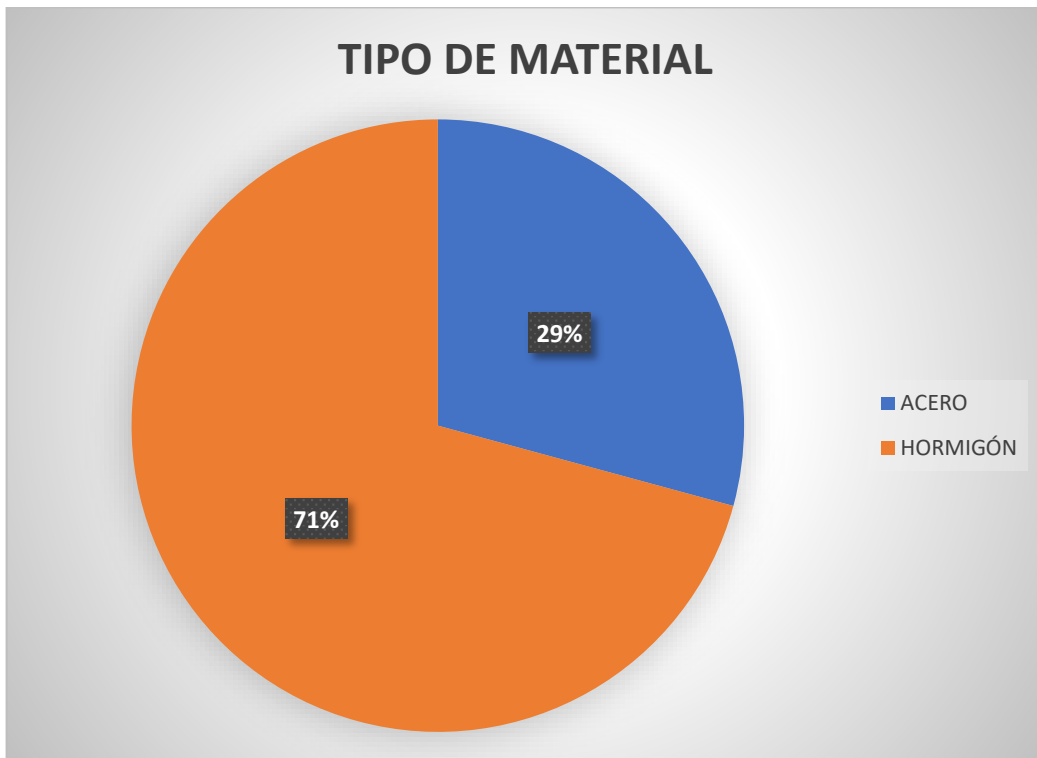
La base de datos MongoDB, tiene un costo mensual de \$7.00. Está conformada por un total de 635 viviendas. La base de datos está alojada en el Servicio MongoDB Atlas, el cual es un conjunto integrado de base de datos en la nube y servicios de datos para acelerar y simplificar la forma de construir con datos. En anexos se encuentra el link del servidor donde está alojada.

Las gráficas estadísticas presentadas a continuación, muestran las dos variables que se utilizan para calcular la cantidad de obra y presupuesto necesarios para construir futuras viviendas las cuales son el resultado obtenido tras realizar la base de datos, de donde se obtuvo el área, material y número de pisos de cada vivienda que sirvió de ejemplo.

La primera característica es el área de construcción (gráfica 1), lo cual determina el tamaño del proyecto. Como se observa se emplearon viviendas tipo de diferentes áreas en las cuales se observa que las áreas de construcción de las viviendas comprenden entre los 36m<sup>2</sup> de área mínima, y de 121.10 m<sup>2</sup> como área máxima. La segunda característica es el tipo de material a utilizar en la estructura de la vivienda (gráfica 2), ya sea acero o hormigón. Estos factores juntos permiten realizar un análisis preciso sobre la cantidad de obra exacta que requiere cada proyecto.



Gráfica 1: Porcentaje de viviendas según el área de cada vivienda. Fuente: Elaboración propia

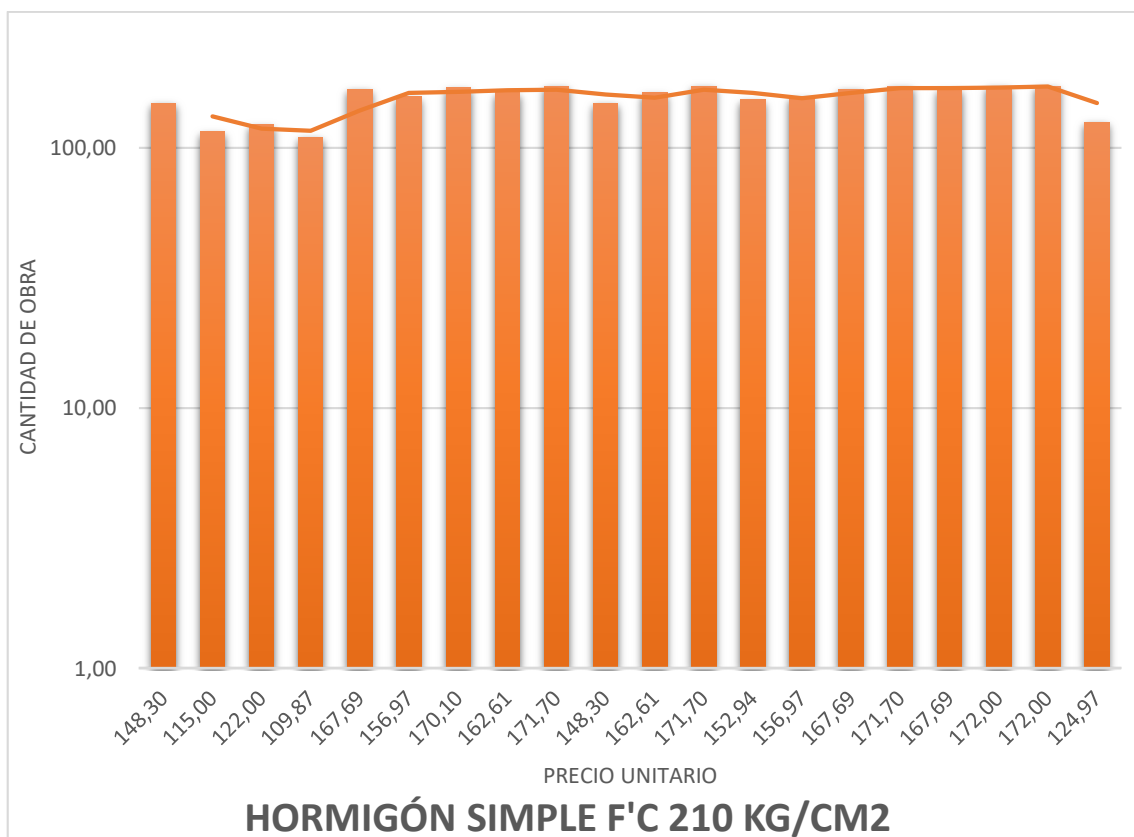


Gráfica 2: Porcentaje de viviendas según el material. Fuente: Elaboración propia

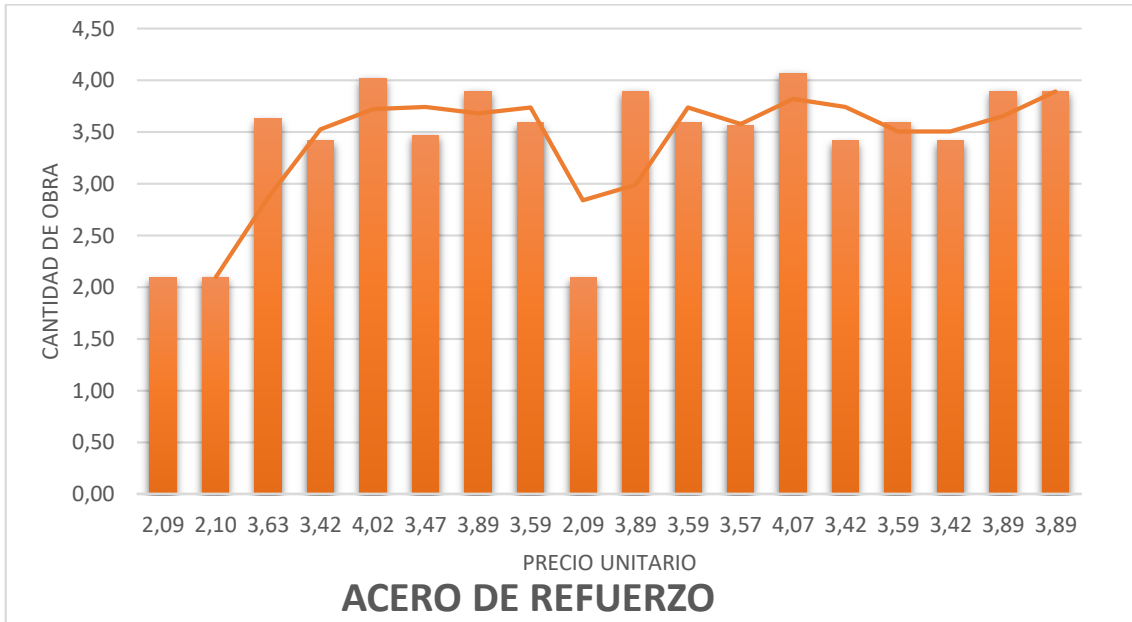
El análisis de los datos estadísticos presentados a continuación (gráficas 4 – 10), muestran que el costo unitario para la construcción de viviendas es una parte importante del presupuesto general. Los gráficos demuestran que hay siete rubros principales como son: hormigón simple f'c 210 kg/cm<sup>2</sup>, acero de refuerzo, cubierta, pintura para interiores y exteriores, mampostería, y enlucido con mortero 1:3; en los cuales se incurren gastos: materiales, mano de obra y equipamiento.

Para la obtención de estos gráficos se elaboró la base de datos (ilustración 1 y 2) en donde se recopila la información adquirida de presupuestos de viviendas tipo, y una vez introducida toda la información se procede a calcular el promedio de todos los datos de los precios unitarios por rubro.

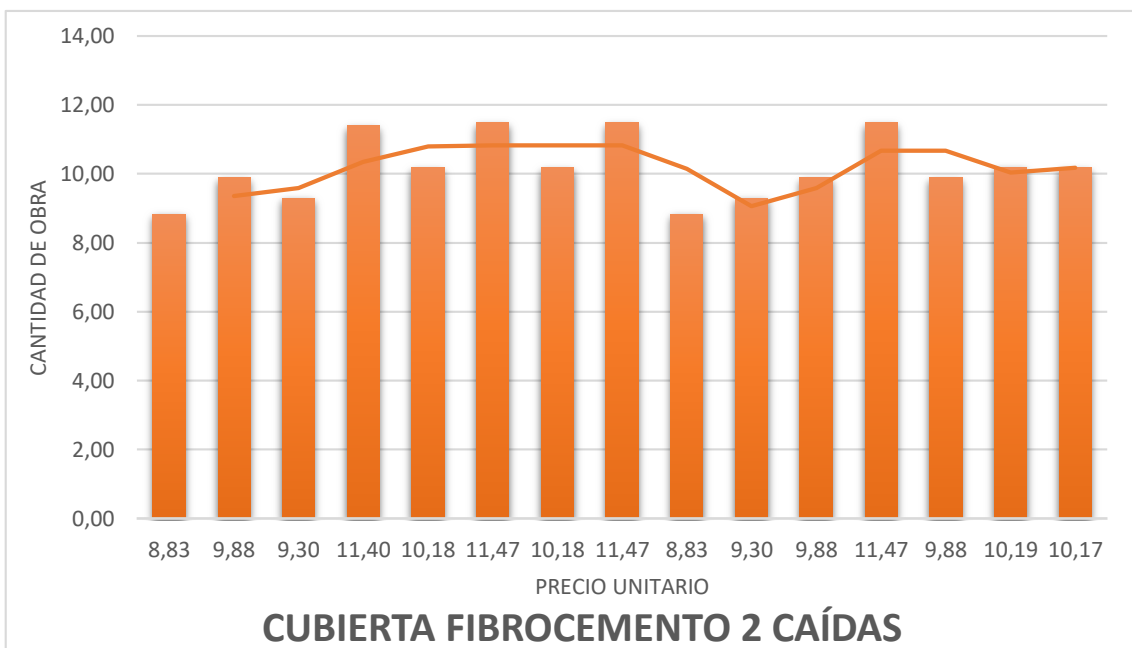
Este análisis ha ayudado a identificar claramente distintas categorías sobre las cuales se incurren gastos en la construcción de viviendas y el costo promedio de cada una de ellas.



Gráfica 3. Gráfica estadística de cantidad de obra y precio unitario para hormigón simple  $f'c= 210 \text{ kg/cm}^2$ . Fuente: Elaboración propia.



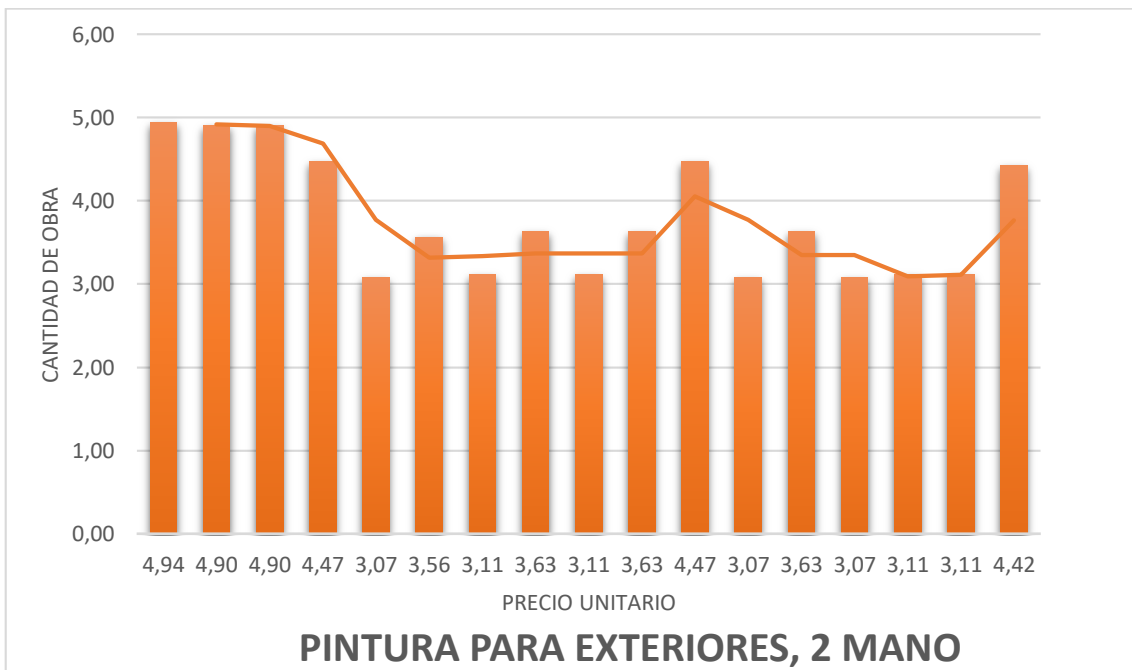
Gráfica 4: Gráfica estadística de cantidad de obra y precio unitario para acero de refuerzo. Fuente: Elaboración propia



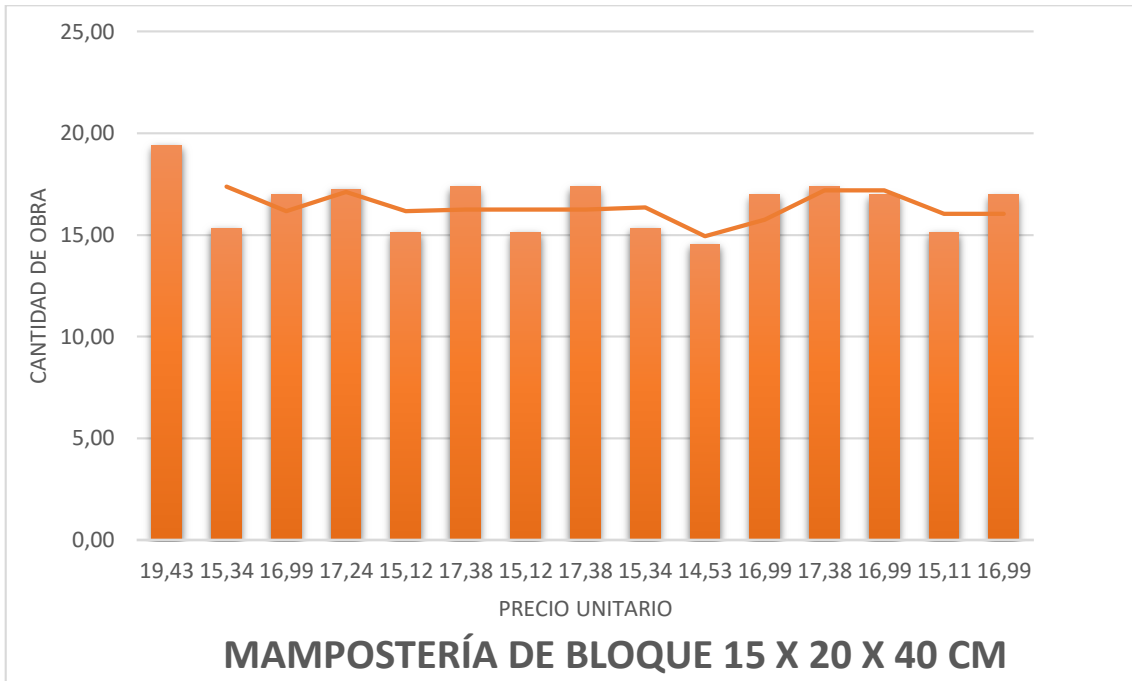
Gráfica 5: Gráfica estadística de cantidad de obra y precio unitario para cubierta de fibrocemento con 2 caídas. Fuente: Elaboración propia



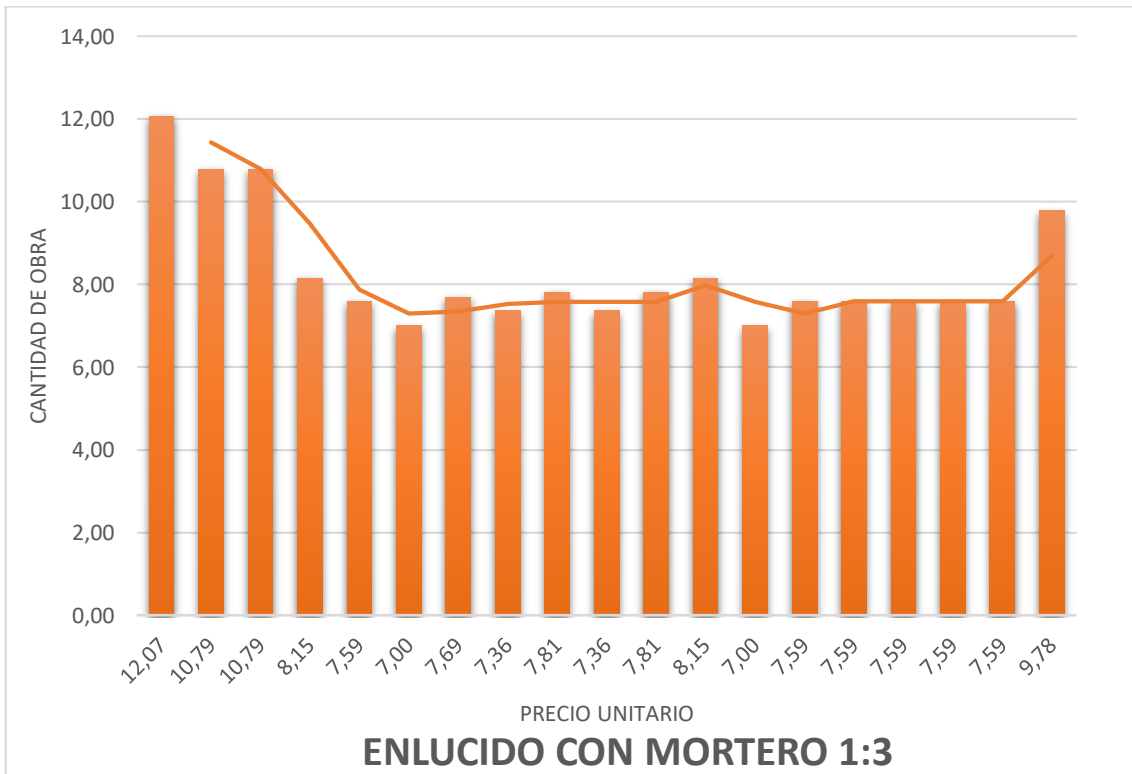
Gráfica 6: Gráfica estadística de cantidad de obra y precio unitario para pintura para interiores, 2 manos. Fuente: Elaboración propia



Gráfica 7: Gráfica estadística de cantidad de obra y precio unitario para pintura para exteriores, 2 manos. Fuente: Elaboración propia



Gráfica 8: Gráfica estadística de cantidad de obra y precio unitario para mampostería de bloque 15 x 20 x 40 cm. Fuente: Elaboración propia



Gráfica 9: Gráfica estadística de cantidad de obra y precio unitario para enlucido con mortero 1:3. Fuente: Elaboración propia

Tras analizar las gráficas se obtuvieron los siguientes promedios para los siete rubros más importantes.

<b>Hormigón simple <math>f'c=210</math> kg/cm<sup>2</sup> para zapatas</b>	
Promedio	\$154,64
<b>Acero de refuerzo</b>	
Promedio	\$3,42
<b>Cubierta de fibrocemento dos caídas</b>	
Promedio	\$10,16
<b>Pintura para interiores, 2 manos</b>	
Promedio	\$3,76
<b>Pintura para exteriores, 2 manos</b>	
Promedio	\$3,78
<b>Mampostería de bloque 15 x 20 x 40 cm</b>	
Promedio	\$16,49
<b>Enlucido con mortero 1:3</b>	
Promedio	\$7,70

*Tabla 2. Resumen de precio unitario de los siete rubros más importantes. Fuente: Elaboración propia*

#### **4.1.2 Resultados obtenidos de la aplicación de presupuestos para vivienda**

Una vez finalizada la creación de la aplicación, se procederá a detallar su funcionamiento, incluyendo la forma de introducir la información necesaria para la presupuestación de cada vivienda. Para ello, se presentan imágenes explicativas que permitan comprender de manera clara y precisa el proceso.

La aplicación, MyHousingCosts, nombre de la aplicación asignado por la tesista, se divide en tres ventanas, cada una adaptada a un propósito específico. La primera ventana (ilustración 7), detalla todos los rubros que componen la construcción de una vivienda y proporciona información de costo por unidad para cada componente. Estos rubros se pueden eliminar, o incrementar según se requiera necesario, sin embargo, la aplicación tiene rubros limitados para que el estudio de los mismos fuera más conciso.

## RUBROS

Nuevo rubro

DESCRIPCIÓN	MEDIDA	CATEGORIA	PRECIO UNITARIO	OBRA GRIS	ACCIÓN
Desbroce y limpieza del terreno	M2	Obras preliminares	2.67	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>
Replanteo y nivelación	M2	Obras preliminares	1.84	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>
Cargada de material a máquina	M3	Movimiento de tierras	1.96	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>
Cargada de material a mano	M3	Movimiento de tierras	9.21	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>
Excavación a mano en suelo sin clasificar, profundidad entre 0 y 2m	M3	Movimiento de tierras	12.24	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>
Excavación a máquina con retroexcavadora	M3	Movimiento de tierras	9.12	SI	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ilustración 7. Ventana de rubros de la aplicación MyHousingCosts. Fuente: Elaboración propia.

La segunda ventana (ilustración 8), contiene una lista de tipos de vivienda tipo utilizados como puntos de referencia al crear presupuestos en MyHousingCosts, con costos, número de pisos y material detallados asociados con cada tipo de vivienda incluida en la base de datos.

## CASAS

Nueva casa

DESCRIPCIÓN	AREA	PISOS	MATERIAL	ACCIÓN
Casa 1	121.21	2	Acero	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 2	100	2	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 3	100	2	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 4	36	1	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 5	108	2	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 6	40.16	1	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>
Casa 7	72	1	Concreto	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ilustración 8. Ventana de casas de la aplicación MyHousingCosts. Fuente: Elaboración propia

Finalmente, la tercera y última ventana (ilustración 9), permite a los usuarios crear presupuestos personalizados basados en sus propias necesidades y preferencias individuales ingresando variables relevantes como el área, número de pisos y material de la vivienda. Asimismo, presenta dos versiones: gratuita o de pago; ambos brindan resultados útiles, pero difieren ligeramente en términos de detalles proporcionados; mientras que la versión gratuita (ilustración 10), le brindará cifras generales divididas en capítulos, la versión premium (ilustración 11), ofrecerá desgloses más granulares, incluidos los costos de obra gris.

Los rubros están seleccionados de forma automática tanto para viviendas en hormigón como en acero, sin embargo, en caso de que se requieran eliminar rubros, existe la opción eliminar en esta misma ventana, haciendo click en la x. (Ilustración 12)

The screenshot shows the 'MyHousingCosts' application interface for creating a budget. On the left is a navigation menu with three items: 'Presupuestos' (Budgets), 'Rubros' (Items), and 'Casas' (Houses). The main content area contains several input fields and a list of items:

- Descripción:** Ejemplo 1
- Area de construcción:** 166
- Numero de pisos:** 2
- Material:** Acero
- Capítulo:** Todos los rubros
- Rubro:** Selecciona un rubro

Below these fields is a scrollable list titled 'Rubros agregados (135)'. The visible items are:

- Instalaciones eléctricas
- Cargada de material a máquina
- Instalaciones eléctricas
- Replanteo y nivelación
- Instalaciones eléctricas
- Desbroce y limpieza del terreno

At the bottom right, there are two buttons: 'GRATIS' and 'Calcular presupuesto'.

*Ilustración 9. Ventana de presupuestos de la aplicación MyHousingCosts. Fuente: Elaboración propia*

**MyHousingCosts** GRATIS

DESCRIPCIÓN	CANTIDAD DE OBRA	P. TOTAL
Obras preliminares	330.75	\$745.32
Movimiento de tierras	138.74	\$1,581.74
Cimentación	84.32	\$3,329.39
Estructura	2,729.19	\$24,090.26
Cubierta	1,687.98	\$15,213.37

Los acabados son de categoría media

Subtotal:	\$191,678.33
IVA (12%):	\$23,001.40
<b>Total:</b>	<b>\$214,679.73</b>

*Ilustración 10. Ejemplo de la versión gratuita de la aplicación MyHousingCosts.  
Fuente: Elaboración propia*

**MyHousingCosts** DE PAGO

Cimentación Replanteo de hormigón pobre f'c=140 kg/cm2	M3	2.48	\$110.75	\$274.22
Cimentación Hormigón ciclopeo (60% H.S y 40% piedra) f'c=180 kg/cm2	M3	13.90	\$88.11	\$1,225.16
Cimentación Encofrado de madera recto zapatas (2 usos)	M2	23.94	\$9.76	\$233.70
Cimentación Hormigón simple f'c=210 kg/cm2 para zapatas	M3	8.36	\$154.16	\$1,288.05

Los acabados son de categoría media

Subtotal:	\$191,678.33
IVA (12%):	\$23,001.40
<b>Total:</b>	<b>\$214,679.73</b>

Obra gris	
Cantidad de Obra:	8,747.94
Precio total	\$57,182.18

*Ilustración 11. Ejemplo de la versión de pago de la aplicación MyHousingCosts.  
Fuente: Elaboración propia*

Descripción: 135.32      Area de construcción: 135.32

Material:

Capítulo: Todos los rubros      Rubro: Selecciona un rubro

Rubros agregados (100)

Obras preliminares	<input type="button" value="X"/>
Desbroce y limpieza del terreno	

*Ilustración 12. Ejemplo de cómo eliminar los rubros que no se requieran en la ventana de presupuestos. Fuente: Elaboración propia*

#### ***4.4.3 Algoritmo para la creación del presupuesto***

- Paso 1: Ingreso de Información

El usuario comienza ingresando los datos necesarios para el cálculo del presupuesto de construcción: descripción, área de construcción y material (Concreto u Hormigón).

- Paso 2: Selección Automática de Capítulos y Rubros

Dependiendo del material seleccionado (Concreto u Hormigón), el sistema seleccionará automáticamente los capítulos y rubros correspondientes.

- Paso 3: Cálculo del Presupuesto

Una vez seleccionados los capítulos y rubros, el sistema procederá a calcular el presupuesto utilizando una ecuación.

- Paso 4: Elección de Versión: GRATUITA o PAGO

Después de calcular el presupuesto, el usuario tiene la opción de elegir entre la versión GRATUITA y la versión de PAGO para ver los detalles adicionales.

- Paso 5: Versión PAGO: Detalles de Rubros

En la versión de PAGO, se mostrarán detalladamente la cantidad de obra, el precio unitario y el precio total de cada rubro seleccionado.

- Paso 6: Cálculo de OBRA GRIS

Además, en la versión de PAGO, se calculará la cantidad de OBRA GRIS y se mostrará su total.

- Paso 7: Versión GRATUITA: Agrupación por Capítulo

En la versión GRATUITA, solo se mostrará la cantidad de obra agrupada por capítulo y su precio total.

- Paso 8: Cálculo de Subtotal

En ambas versiones, se calculará el subtotal de los precios totales de los rubros.

- Paso 9: Cálculo del IVA (12%)

Se calculará el IVA correspondiente al subtotal utilizando una tasa del 12%.

- Paso 10: Cálculo del Total

Finalmente, se calculará el total sumando el subtotal y el IVA.

- Paso 11: Resultado Final

El sistema mostrará el resultado final del presupuesto, que incluirá el Subtotal, el IVA y el Total, según la versión elegida por el usuario.

#### ***4.1.4 Resultados del cálculo del porcentaje de error de la aplicación***

Para corroborar el rango de error se empleó la fórmula presentada en el capítulo 3, con la cual se obtuvo un rango de error del 9.83% para las viviendas de acero, y 12.63 % para las viviendas de hormigón.

A continuación, se muestra la forma en la que se obtuvieron los porcentajes de error, para ellos se realizaron las siguientes tablas (tabla 3, tabla 4):

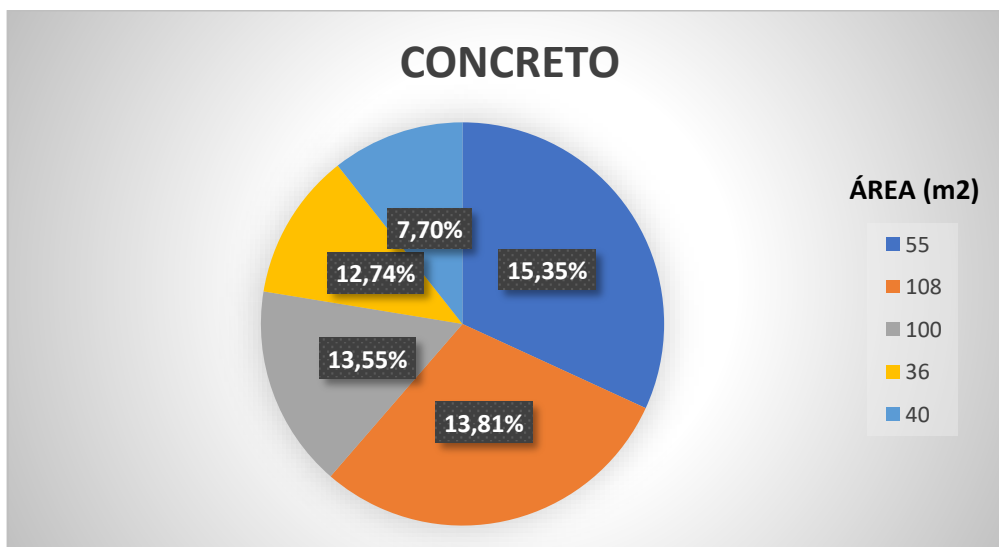
<b>CONCRETO</b>	
108m2 concreto	
Valor exacto	\$54319,66
Valor aproximado	\$61823,32
% de Error	13,81%
100m2 concreto	
Valor exacto	\$50295,06
Valor aproximado	\$57111,88
% de Error	13,55%
55m2 concreto	
Valor exacto	\$25765,15
Valor aproximado	\$29719,62
% de Error	15,35%
40m2 concreto	
Valor exacto	\$20787,19
Valor aproximado	\$22386,99
% de Error	7,70%
36m2 concreto	
Valor exacto	\$18787,47
Valor aproximado	\$21180,89
% de Error	12,74%
<b>PROMEDIO</b>	<b>12,63%</b>

*Tabla 3: Tabla del cálculo del porcentaje de error de las viviendas en hormigón.  
Fuente: Elaboración propia.*

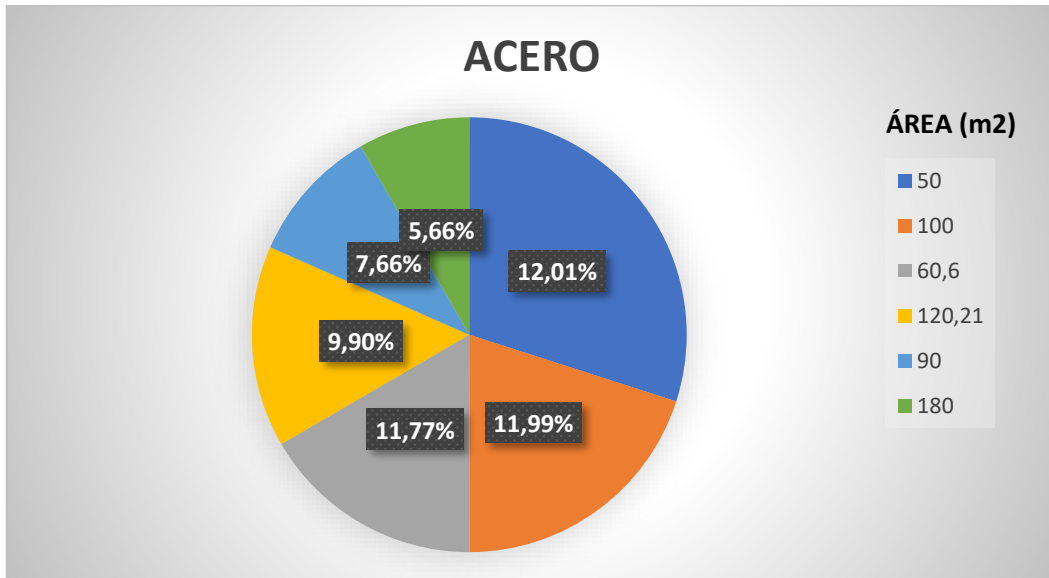
<b>ACERO</b>	
120,21m2 acero	
Valor exacto	\$72917,53
Valor aproximado	\$80136,35
% de Error	9,90%
100m2 acero	
Valor exacto	\$60658,45
Valor aproximado	\$67933,07
% de Error	11,99%
90m2 acero	
Valor exacto	\$57927,45
Valor aproximado	\$61203,53
% de Error	7,66%
60,60m2 acero	
Valor exacto	\$36458,77
Valor aproximado	\$40750,87
% de Error	11,77%
50m2 acero	
Valor exacto	\$30329,23
Valor aproximado	\$33972,35
% de Error	12,01%
<b>PROMEDIO</b>	<b>9,83%</b>

Tabla 4: Tabla del cálculo del porcentaje de error para las viviendas en acero Fuente: Elaboración propia.

Con las siguientes gráficas (gráfica 11 y 12) se puede observar de forma clara como se encuentran distribuidos los porcentajes de error tanto para viviendas de acero como de hormigón.



Gráfica 10: Porcentaje de error para las viviendas en hormigón. Fuente: Elaboración propia



Gráfica 11: Porcentaje de error para las viviendas en acero. Fuente: Elaboración propia

#### 4.1.5 Resultado de la muestra de la población

Empleando la siguiente ecuación con una población de 1300 permisos de construcción que se hicieron en el año 2022 (Cuenca, 2022), se obtiene un tamaño muestra de la población de 298.

$$n = \frac{Z^2 * P * (1 - P)}{d^2}$$

#### 4.1.6 Corrida de la aplicación

4.1.6.1 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión de pago en acero:

DE PAGO | ▾

Calcular presupuesto

DESCRIPCIÓN	UNIDAD	CANTIDAD DE OBRA	P. UNITARIO	P. TOTAL
Obras preliminares Desbroce y limpieza del terreno	M2	49.95	\$2.67	\$133.38
Obras preliminares Replanteo y nivelación	M2	50.00	\$1.79	\$89.50
Movimiento de tierras Cargada de material a máquina	M3	5.96	\$1.96	\$11.67
Movimiento de tierras Cargada de material a mano	M3	0.75	\$12.00	\$9.04

Los acabados son de categoría media

Obra gris	
Cantidad de Obra:	2,819.85
Precio total	\$14,200.69

Subtotal:	\$30,261.08
IVA (12%):	\$3,631.33
Total:	\$33,892.41

*Ilustración 13. Presupuesto en versión de pago de vivienda en acero- Fuente: Elaboración propia.*

4.1.6.2 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión gratuita en acero:

GRATIS | ▾

Calcular presupuesto

DESCRIPCIÓN	CANTIDAD DE OBRA	P. TOTAL
Obras preliminares	99.95	\$222.88
Movimiento de tierras	31.42	\$190.64
Cimentación	20.41	\$1,103.40
Estructura	1,901.61	\$6,669.66
Cubierta	243.77	\$1,690.19

Los acabados son de categoría media

Subtotal:	\$30,261.08
IVA (12%):	\$3,631.33
Total:	\$33,892.41

*Ilustración 14. . Presupuesto en versión gratuita de vivienda en acero- Fuente: Elaboración propia.*

4.1.6.3 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión gratuita en hormigón:

GRATIS |

DESCRIPCIÓN	CANTIDAD DE OBRA	P. TOTAL
Obras preliminares	99.95	\$222.88
Movimiento de tierras	31.42	\$190.64
Cimentación	20.41	\$1,103.40
Estructura	364.00	\$2,058.79
Cubierta	251.00	\$1,757.53

Los acabados son de categoría media

Subtotal:	\$23,826.25
IVA (12%):	\$2,859.15
Total:	\$26,685.40

*Ilustración 15. Presupuesto en versión gratuita de vivienda en hormigón- Fuente: Elaboración propia.*

4.1.6.4 Presupuesto del valor aproximado de una vivienda obtenido de la aplicación MyHousingCost versión de pago en hormigón:

DE PAGO |

DESCRIPCIÓN	UNIDAD	CANTIDAD DE OBRA	P. UNITARIO	P. TOTAL
Obras preliminares Desbroce y limpieza del terreno	M2	49.95	\$2.67	\$133.38
Obras preliminares Replanteo y nivelación	M2	50.00	\$1.79	\$89.50
Movimiento de tierras Cargada de material a máquina	M3	5.96	\$1.96	\$11.67
Movimiento de tierras Cargada de material a mano	M3	0.75	\$12.00	\$9.04

Los acabados son de categoría media

Obra gris	
Cantidad de Obra:	1,023.86
Precio total	\$8,789.49

Subtotal:	\$23,826.25
IVA (12%):	\$2,859.15
Total:	\$26,685.40

*Ilustración 16. Presupuesto en versión de pago de vivienda en hormigón- Fuente: Elaboración propia.*

## CAPÍTULO V

### 5.1 CONCLUSIONES

En este trabajo de titulación de ingeniería civil, se ha abordado el diseño y desarrollo de una aplicación que permite la presupuestación de viviendas con áreas comprendidas entre 36 m<sup>2</sup> hasta 120 m<sup>2</sup>, ya sean con estructura de acero u hormigón armado en la ciudad de Cuenca, Ecuador. Esta aplicación se presenta en dos versiones: una gratuita y otra de pago. En la versión gratuita, los usuarios pueden obtener un presupuesto y la cantidad de obra desglosada por capítulos. Por otro lado, la versión de pago ofrece un presupuesto más detallado que incluye la cantidad de obra por rubros, así como el costo y la cantidad de la obra gris.

La recopilación de información previa y la planificación rigurosa son fundamentales para garantizar la precisión en la obtención de un presupuesto. Por tanto, la base de datos conformada por más de 635 viviendas se ha convertido en un recurso valioso para este propósito, debido que al hacer el cálculo estadístico para comprobar el tamaño de la muestra necesaria para que la aplicación arroje un resultado certero es de 298 viviendas, por lo que la cantidad de viviendas empleadas para la creación de la aplicación está en un rango superior al necesario.

Se ha determinado un margen de error del 9.83% para las viviendas de acero, y 12.63 % para las viviendas de hormigón en la aplicación MyHousingCosts, tomando en consideración que algunos de los presupuestos de las viviendas tipo utilizados en la creación de la base de datos no abarcaron todos los aspectos fundamentales asociados a la construcción de una vivienda, siendo empleadas un total de 635 viviendas para la creación y un total de 597 viviendas empleadas para la verificación del rango de error que arroja la aplicación.

En general, MyHousingCosts ha sido diseñado específicamente para aquellos que trabajan en la industria de la construcción firmemente en su núcleo; dándoles acceso no solo a estimaciones rápidas sino también a desgloses detallados para que puedan estar seguros de que han tenido en cuenta con precisión todos los aspectos involucrados al elaborar el presupuesto de su proyecto.

## **5.2 RECOMENDACIONES**

- Realizar una actualización anual de la base de datos es una práctica recomendable para mantener los costos unitarios por rubro al día. Esto ayuda a evitar problemas financieros en el futuro, ya que, si los costes no se actualizan regularmente, podrían surgir errores o desajustes cuando llegue el momento de realizar presupuestaciones y cotizaciones. Además, hacer actualizaciones de los rubros con base a la implementación de nuevas tecnologías constructivas de forma periódica.
- Retroalimentar la base de datos añadiendo más viviendas, puesto que la aplicación da precios estimados únicamente para viviendas desde 36 m<sup>2</sup>, hasta 120 m<sup>2</sup>, puesto que esas son las áreas de las viviendas tipo que se investigaron. En caso de intentar calcular para viviendas con áreas mayores a 120 m<sup>2</sup>, el resultado no es confiable.
- Se recomienda que, a futuro, el programa calcule por sí solo la ecuación para el cálculo de la cantidad de obra de cada rubro, teniendo en cuenta el número de pisos de la vivienda, puesto que actualmente este paso se realiza de forma manual.
- Mantener al día el coste mensual fijo asociado con el uso general del sistema informático para administrar su base de datos; este coste es normalmente \$7 USD/mes, para así evitar que por alguna razón la base de datos sea eliminada.

## BIBLIOGRAFÍA

- Cruz, D., López de León, F., Pascual, L., & Battaglia, M. (2010). *Guía Técnica de producción de hongos comestibles de la especie de Hongos Ostra*.
- Cuenca, C. d. (2022). Cifras de la construcción. *Edificaciones en Altura*, 54.
- Felipe, D. J. (2018). *Implementación del programa Proexcel 2016 en la Empresa Panecons s.a., para su uso eficiente en la elaboración de presupuestos de obra dentro de la empresa*. Quito: PUCE.
- *InterPro*. (15 de Agosto de 2023). Obtenido de <https://www.interpro.ec/>
- *MongoDB Inc.* (2020). Obtenido de <https://www.mongodb.com/es/what-is-mongodb>
- *Sobre Next.js*. (2020). Obtenido de <https://nextjs.org/>
- 7.- *NEC - SE - GC-Geotécnia-y-Cimentaciones.pdf*. (2014, diciembre). <https://www.habitatyvivienda.gob.ec/wp-content/uploads/2023/03/7.-NEC-SE-GC-Geotecnia-y-Cimentaciones.pdf>
- 8.- *NEC-SE-HM-Hormigón-Armado.pdf*. (2014, diciembre). <https://www.habitatyvivienda.gob.ec/wp-content/uploads/2023/03/8.-NEC-SE-HM-Hormigon-Armado.pdf>
- 9.- *NEC-SE-AC-Estructuras-de-Acero.pdf*. (2014, diciembre). <https://www.habitatyvivienda.gob.ec/wp-content/uploads/2023/03/9.-NEC-SE-AC-Estructuras-de-Acero.pdf>
- 12.- *NEC-SE-VIVIENDA-parte-1.pdf*. (2014, diciembre). <https://www.habitatyvivienda.gob.ec/wp-content/uploads/2023/03/12.-NEC-SE-VIVIENDA-parte-1.pdf>

- Cáceres Tume, K. V. (2013). *Estimación de Costos de Proyectos de Infraestructura Municipal*.  
[https://pirhua.udep.edu.pe/bitstream/handle/11042/1430/ICI\\_142.pdf?sequence=3](https://pirhua.udep.edu.pe/bitstream/handle/11042/1430/ICI_142.pdf?sequence=3)
- Cairó, O. (2005). *Metodología de la programación*. Alpha Editorial.
- *Capítulos de la NEC (Norma Ecuatoriana de la Construcción) – MIDUVI – Ministerio de Desarrollo Urbano y Vivienda*. (2014). Recuperado 18 de mayo de 2023, de <https://www.habitatyvivienda.gob.ec/documentos-normativos-nec-norma-ecuatoriana-de-la-construccion/>
- Eguíluz Pérez, J. (2009). *Introducción a JavaScript*.
- *Estructuras de acero. Introducción al diseño*. (2006). Univ. Nacional de Colombia.
- Gackenheimer, C. (2015). *Introduction to React*. Apress.
- *Generador de precios de la construcción. Ecuador. CYPE Ingenieros, S.A.* (2022). <http://www.ecuador.generadordeprecios.info/>
- JS, N. (2020). *NODE.JS*. <https://nodejs.org/es>.
- *Manual del ingeniero de edificación. Guía para el cálculo de estructuras con CYPECAD*. (2012).
- Paúl, S., & Acosta, E. (2008). *Diseño estructural de la cubierta metálica para dos canchas de ecuavoley* [BachelorThesis, QUITO/ EPN/ 2008]. <http://bibdigital.epn.edu.ec/handle/15000/651>

- *Presupuestos y condiciones técnicas—ITeC - Instituto de Tecnología de la Construcción.* (2015). <https://itec.es/programas/tcqi/presupuesto-condiciones-tecnicas/>
- Sarría, F. A. (2017). *Sistemas de Información Geográfica.*
- Soliz Baldovinos, I. (2013). *Costos Indirectos de la Construcción.*  
<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/3163/Costos%20Indirectos%20en%20la%20Construccion.pdf?sequence=1>
- Yepes Piqueras, V. (2019). *Zapatas aisladas.*  
<https://riunet.upv.es/handle/10251/119673>

## GLOSARIO

- Vivienda: Lugar cerrado y cubierto construido para ser habitado por personas.
- Aplicación: Programa informático diseñado como herramienta para permitir a los usuarios realizar uno o diversos tipos de trabajo.
- Presupuesto: Cálculo detallado y anticipado del precio de construcción de una obra.
- IVA: Sigla de impuesto sobre el valor añadido o de impuesto sobre el valor agregado, impuesto que grava el valor añadido o agregado de un producto en las distintas fases de su producción.
- Algoritmo: Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.
- Base de datos: Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.
- Interfaz de usuario: Medio por el cual una persona controla una aplicación de software o dispositivo de hardware.
- TypeScript: Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft.

## ANEXOS

### Código de la aplicación web:

```
1 import { SidebarPro } from '@components/ui/SidebarPro'
2 import { TabBar } from '@components/ui/TabBar'
3 import QueryProvider from '@providers/QueryProvider'
4 import { Poppins } from 'next/font/google'
5 import NextTopLoader from 'nextjs-toploader'
6 import './globals.css'
7
8 const poppins = Poppins({
9   weight: ['300', '400', '500', '600', '700', '800'],
10  subsets: ['latin'],
11 })
12
13 export const metadata = {
14   title: 'MyHousingCosts',
15   description: 'Aplicación para crear presupuestos de manera rápida y sencilla',
16 }
17
18 export default function RootLayout({ children }: { children: React.ReactNode }) {
19   return (
20     <html lang="es">
21       <body className={poppins.className}>
22         <NextTopLoader color="#fb7185" height={4} showSpinner />
23         <QueryProvider>
24           <main className="flex h-full w-full">
25             <SidebarPro />
26
27             <div className="w-full flex-shrink flex-grow overflow-auto pb-14" id="main-area">
28               {children}
29             </div>
30             <TabBar />
31           </main>
32         </QueryProvider>
33       </body>
34     </html>
35   )
36 }
```

```
1 import { Props } from '@types/queries/queries'
2 import { Props } from '@types/queries/queries'
3 import { Props } from '@types/queries/queries'
4 import { Props } from '@types/queries/queries'
5 import { Props } from '@types/queries/queries'
6
7 export const propTypes = {
8   // ...
9 }
10
11 export const defaultProps = {
12   // ...
13 }
14
15 export const styles = {
16   // ...
17 }
18
19 export const components = {
20   // ...
21 }
22
23 export const hooks = {
24   // ...
25 }
26
27 export const utils = {
28   // ...
29 }
30
31 export const constants = {
32   // ...
33 }
34
35 export const enums = {
36   // ...
37 }
38
39 export const interfaces = {
40   // ...
41 }
42
43 export const types = {
44   // ...
45 }
46
47 export const declarations = {
48   // ...
49 }
50
51 export const namespace = {
52   // ...
53 }
54
55 export const module = {
56   // ...
57 }
58
59 export const namespace = {
60   // ...
61 }
62
63 export const module = {
64   // ...
65 }
66
67 export const namespace = {
68   // ...
69 }
70
71 export const module = {
72   // ...
73 }
74
75 export const namespace = {
76   // ...
77 }
78
79 export const module = {
80   // ...
81 }
82
83 export const namespace = {
84   // ...
85 }
86
87 export const module = {
88   // ...
89 }
90
91 export const namespace = {
92   // ...
93 }
94
95 export const module = {
96   // ...
97 }
98
99 export const namespace = {
100  // ...
101 }
102
103 export const module = {
104  // ...
105 }
106
107 export const namespace = {
108  // ...
109 }
110
111 export const module = {
112  // ...
113 }
114
115 export const namespace = {
116  // ...
117 }
118
119 export const module = {
120  // ...
121 }
122
123 export const namespace = {
124  // ...
125 }
126
127 export const module = {
128  // ...
129 }
130
131 export const namespace = {
132  // ...
133 }
134
135 export const module = {
136  // ...
137 }
138
139 export const namespace = {
140  // ...
141 }
142
143 export const module = {
144  // ...
145 }
146
147 export const namespace = {
148  // ...
149 }
150
151 export const module = {
152  // ...
153 }
154
155 export const namespace = {
156  // ...
157 }
158
159 export const module = {
160  // ...
161 }
162
163 export const namespace = {
164  // ...
165 }
166
167 export const module = {
168  // ...
169 }
170
171 export const namespace = {
172  // ...
173 }
174
175 export const module = {
176  // ...
177 }
178
179 export const namespace = {
180  // ...
181 }
182
183 export const module = {
184  // ...
185 }
186
187 export const namespace = {
188  // ...
189 }
190
191 export const module = {
192  // ...
193 }
194
195 export const namespace = {
196  // ...
197 }
198
199 export const module = {
200  // ...
201 }
202
203 export const namespace = {
204  // ...
205 }
206
207 export const module = {
208  // ...
209 }
210
211 export const namespace = {
212  // ...
213 }
214
215 export const module = {
216  // ...
217 }
218
219 export const namespace = {
220  // ...
221 }
222
223 export const module = {
224  // ...
225 }
226
227 export const namespace = {
228  // ...
229 }
230
231 export const module = {
232  // ...
233 }
234
235 export const namespace = {
236  // ...
237 }
238
239 export const module = {
240  // ...
241 }
242
243 export const namespace = {
244  // ...
245 }
246
247 export const module = {
248  // ...
249 }
250
251 export const namespace = {
252  // ...
253 }
254
255 export const module = {
256  // ...
257 }
258
259 export const namespace = {
260  // ...
261 }
262
263 export const module = {
264  // ...
265 }
266
267 export const namespace = {
268  // ...
269 }
270
271 export const module = {
272  // ...
273 }
274
275 export const namespace = {
276  // ...
277 }
278
279 export const module = {
280  // ...
281 }
282
283 export const namespace = {
284  // ...
285 }
286
287 export const module = {
288  // ...
289 }
290
291 export const namespace = {
292  // ...
293 }
294
295 export const module = {
296  // ...
297 }
298
299 export const namespace = {
300  // ...
301 }
302
303 export const module = {
304  // ...
305 }
306
307 export const namespace = {
308  // ...
309 }
310
311 export const module = {
312  // ...
313 }
314
315 export const namespace = {
316  // ...
317 }
318
319 export const module = {
320  // ...
321 }
322
323 export const namespace = {
324  // ...
325 }
326
327 export const module = {
328  // ...
329 }
330
331 export const namespace = {
332  // ...
333 }
334
335 export const module = {
336  // ...
337 }
338
339 export const namespace = {
340  // ...
341 }
342
343 export const module = {
344  // ...
345 }
346
347 export const namespace = {
348  // ...
349 }
350
351 export const module = {
352  // ...
353 }
354
355 export const namespace = {
356  // ...
357 }
358
359 export const module = {
360  // ...
361 }
362
363 export const namespace = {
364  // ...
365 }
366
367 export const module = {
368  // ...
369 }
370
371 export const namespace = {
372  // ...
373 }
374
375 export const module = {
376  // ...
377 }
378
379 export const namespace = {
380  // ...
381 }
382
383 export const module = {
384  // ...
385 }
386
387 export const namespace = {
388  // ...
389 }
390
391 export const module = {
392  // ...
393 }
394
395 export const namespace = {
396  // ...
397 }
398
399 export const module = {
400  // ...
401 }
402
403 export const namespace = {
404  // ...
405 }
406
407 export const module = {
408  // ...
409 }
410
411 export const namespace = {
412  // ...
413 }
414
415 export const module = {
416  // ...
417 }
418
419 export const namespace = {
420  // ...
421 }
422
423 export const module = {
424  // ...
425 }
426
427 export const namespace = {
428  // ...
429 }
430
431 export const module = {
432  // ...
433 }
434
435 export const namespace = {
436  // ...
437 }
438
439 export const module = {
440  // ...
441 }
442
443 export const namespace = {
444  // ...
445 }
446
447 export const module = {
448  // ...
449 }
450
451 export const namespace = {
452  // ...
453 }
454
455 export const module = {
456  // ...
457 }
458
459 export const namespace = {
460  // ...
461 }
462
463 export const module = {
464  // ...
465 }
466
467 export const namespace = {
468  // ...
469 }
470
471 export const module = {
472  // ...
473 }
474
475 export const namespace = {
476  // ...
477 }
478
479 export const module = {
480  // ...
481 }
482
483 export const namespace = {
484  // ...
485 }
486
487 export const module = {
488  // ...
489 }
490
491 export const namespace = {
492  // ...
493 }
494
495 export const module = {
496  // ...
497 }
498
499 export const namespace = {
500  // ...
501 }
502
503 export const module = {
504  // ...
505 }
506
507 export const namespace = {
508  // ...
509 }
510
511 export const module = {
512  // ...
513 }
514
515 export const namespace = {
516  // ...
517 }
518
519 export const module = {
520  // ...
521 }
522
523 export const namespace = {
524  // ...
525 }
526
527 export const module = {
528  // ...
529 }
530
531 export const namespace = {
532  // ...
533 }
534
535 export const module = {
536  // ...
537 }
538
539 export const namespace = {
540  // ...
541 }
542
543 export const module = {
544  // ...
545 }
546
547 export const namespace = {
548  // ...
549 }
550
551 export const module = {
552  // ...
553 }
554
555 export const namespace = {
556  // ...
557 }
558
559 export const module = {
560  // ...
561 }
562
563 export const namespace = {
564  // ...
565 }
566
567 export const module = {
568  // ...
569 }
570
571 export const namespace = {
572  // ...
573 }
574
575 export const module = {
576  // ...
577 }
578
579 export const namespace = {
580  // ...
581 }
582
583 export const module = {
584  // ...
585 }
586
587 export const namespace = {
588  // ...
589 }
590
591 export const module = {
592  // ...
593 }
594
595 export const namespace = {
596  // ...
597 }
598
599 export const module = {
600  // ...
601 }
602
603 export const namespace = {
604  // ...
605 }
606
607 export const module = {
608  // ...
609 }
610
611 export const namespace = {
612  // ...
613 }
614
615 export const module = {
616  // ...
617 }
618
619 export const namespace = {
620  // ...
621 }
622
623 export const module = {
624  // ...
625 }
626
627 export const namespace = {
628  // ...
629 }
630
631 export const module = {
632  // ...
633 }
634
635 export const namespace = {
636  // ...
637 }
638
639 export const module = {
640  // ...
641 }
642
643 export const namespace = {
644  // ...
645 }
646
647 export const module = {
648  // ...
649 }
650
651 export const namespace = {
652  // ...
653 }
654
655 export const module = {
656  // ...
657 }
658
659 export const namespace = {
660  // ...
661 }
662
663 export const module = {
664  // ...
665 }
666
667 export const namespace = {
668  // ...
669 }
670
671 export const module = {
672  // ...
673 }
674
675 export const namespace = {
676  // ...
677 }
678
679 export const module = {
680  // ...
681 }
682
683 export const namespace = {
684  // ...
685 }
686
687 export const module = {
688  // ...
689 }
690
691 export const namespace = {
692  // ...
693 }
694
695 export const module = {
696  // ...
697 }
698
699 export const namespace = {
700  // ...
701 }
702
703 export const module = {
704  // ...
705 }
706
707 export const namespace = {
708  // ...
709 }
710
711 export const module = {
712  // ...
713 }
714
715 export const namespace = {
716  // ...
717 }
718
719 export const module = {
720  // ...
721 }
722
723 export const namespace = {
724  // ...
725 }
726
727 export const module = {
728  // ...
729 }
730
731 export const namespace = {
732  // ...
733 }
734
735 export const module = {
736  // ...
737 }
738
739 export const namespace = {
740  // ...
741 }
742
743 export const module = {
744  // ...
745 }
746
747 export const namespace = {
748  // ...
749 }
750
751 export const module = {
752  // ...
753 }
754
755 export const namespace = {
756  // ...
757 }
758
759 export const module = {
760  // ...
761 }
762
763 export const namespace = {
764  // ...
765 }
766
767 export const module = {
768  // ...
769 }
770
771 export const namespace = {
772  // ...
773 }
774
775 export const module = {
776  // ...
777 }
778
779 export const namespace = {
780  // ...
781 }
782
783 export const module = {
784  // ...
785 }
786
787 export const namespace = {
788  // ...
789 }
790
791 export const module = {
792  // ...
793 }
794
795 export const namespace = {
796  // ...
797 }
798
799 export const module = {
800  // ...
801 }
802
803 export const namespace = {
804  // ...
805 }
806
807 export const module = {
808  // ...
809 }
810
811 export const namespace = {
812  // ...
813 }
814
815 export const module = {
816  // ...
817 }
818
819 export const namespace = {
820  // ...
821 }
822
823 export const module = {
824  // ...
825 }
826
827 export const namespace = {
828  // ...
829 }
830
831 export const module = {
832  // ...
833 }
834
835 export const namespace = {
836  // ...
837 }
838
839 export const module = {
840  // ...
841 }
842
843 export const namespace = {
844  // ...
845 }
846
847 export const module = {
848  // ...
849 }
850
851 export const namespace = {
852  // ...
853 }
854
855 export const module = {
856  // ...
857 }
858
859 export const namespace = {
860  // ...
861 }
862
863 export const module = {
864  // ...
865 }
866
867 export const namespace = {
868  // ...
869 }
870
871 export const module = {
872  // ...
873 }
874
875 export const namespace = {
876  // ...
877 }
878
879 export const module = {
880  // ...
881 }
882
883 export const namespace = {
884  // ...
885 }
886
887 export const module = {
888  // ...
889 }
890
891 export const namespace = {
892  // ...
893 }
894
895 export const module = {
896  // ...
897 }
898
899 export const namespace = {
900  // ...
901 }
902
903 export const module = {
904  // ...
905 }
906
907 export const namespace = {
908  // ...
909 }
910
911 export const module = {
912  // ...
913 }
914
915 export const namespace = {
916  // ...
917 }
918
919 export const module = {
920  // ...
921 }
922
923 export const namespace = {
924  // ...
925 }
926
927 export const module = {
928  // ...
929 }
930
931 export const namespace = {
932  // ...
933 }
934
935 export const module = {
936  // ...
937 }
938
939 export const namespace = {
940  // ...
941 }
942
943 export const module = {
944  // ...
945 }
946
947 export const namespace = {
948  // ...
949 }
950
951 export const module = {
952  // ...
953 }
954
955 export const namespace = {
956  // ...
957 }
958
959 export const module = {
960  // ...
961 }
962
963 export const namespace = {
964  // ...
965 }
966
967 export const module = {
968  // ...
969 }
970
971 export const namespace = {
972  // ...
973 }
974
975 export const module = {
976  // ...
977 }
978
979 export const namespace = {
980  // ...
981 }
982
983 export const module = {
984  // ...
985 }
986
987 export const namespace = {
988  // ...
989 }
990
991 export const module = {
992  // ...
993 }
994
995 export const namespace = {
996  // ...
997 }
1000
1001 export const module = {
1002  // ...
1003 }
1004
1005 export const namespace = {
1006  // ...
1007 }
1008
1009 export const module = {
1010  // ...
1011 }
1012
1013 export const namespace = {
1014  // ...
1015 }
1016
1017 export const module = {
1018  // ...
1019 }
1020
1021 export const namespace = {
1022  // ...
1023 }
1024
1025 export const module = {
1026  // ...
1027 }
1028
1029 export const namespace = {
1030  // ...
1031 }
1032
1033 export const module = {
1034  // ...
1035 }
1036
1037 export const namespace = {
1038  // ...
1039 }
1040
1041 export const module = {
1042  // ...
1043 }
1044
1045 export const namespace = {
1046  // ...
1047 }
1048
1049 export const module = {
1050  // ...
1051 }
1052
1053 export const namespace = {
1054  // ...
1055 }
1056
1057 export const module = {
1058  // ...
1059 }
1060
1061 export const namespace = {
1062  // ...
1063 }
1064
1065 export const module = {
1066  // ...
1067 }
1068
1069 export const namespace = {
1070  // ...
1071 }
1072
1073 export const module = {
1074  // ...
1075 }
1076
1077 export const namespace = {
1078  // ...
1079 }
1080
1081 export const module = {
1082  // ...
1083 }
1084
1085 export const namespace = {
1086  // ...
1087 }
1088
1089 export const module = {
1090  // ...
1091 }
1092
1093 export const namespace = {
1094  // ...
1095 }
1096
1097 export const module = {
1098  // ...
1099 }
1100
1101 export const namespace = {
1102  // ...
1103 }
1104
1105 export const module = {
1106  // ...
1107 }
1108
1109 export const namespace = {
1110  // ...
1111 }
1112
1113 export const module = {
1114  // ...
1115 }
1116
1117 export const namespace = {
1118  // ...
1119 }
1120
1121 export const module = {
1122  // ...
1123 }
1124
1125 export const namespace = {
1126  // ...
1127 }
1128
1129 export const module = {
1130  // ...
1131 }
1132
1133 export const namespace = {
1134  // ...
1135 }
1136
1137 export const module = {
1138  // ...
1139 }
1140
1141 export const namespace = {
1142  // ...
1143 }
1144
1145 export const module = {
1146  // ...
1147 }
1148
1149 export const namespace = {
1150  // ...
1151 }
1152
1153 export const module = {
1154  // ...
1155 }
1156
1157 export const namespace = {
1158  // ...
1159 }
1160
1161 export const module = {
1162  // ...
1163 }
1164
1165 export const namespace = {
1166  // ...
1167 }
1168
1169 export const module = {
1170  // ...
1171 }
1172
1173 export const namespace = {
1174  // ...
1175 }
1176
1177 export const module = {
1178  // ...
1179 }
1180
1181 export const namespace = {
1182  // ...
1183 }
1184
1185 export const module = {
1186  // ...
1187 }
1188
1189 export const namespace = {
1190  // ...
1191 }
1192
1193 export const module = {
1194  // ...
1195 }
1196
1197 export const namespace = {
1198  // ...
1199 }
1200
1201 export const module = {
1202  // ...
1203 }
1204
1205 export const namespace = {
1206  // ...
1207 }
1208
1209 export const module = {
1210  // ...
1211 }
1212
1213 export const namespace = {
1214  // ...
1215 }
1216
1217 export const module = {
1218  // ...
1219 }
1220
1221 export const namespace = {
1222  // ...
1223 }
1224
1225 export const module = {
1226  // ...
1227 }
1228
1229 export const namespace = {
1230  // ...
1231 }
1232
1233 export const module = {
1234  // ...
1235 }
1236
1237 export const namespace = {
1238  // ...
1239 }
1240
1241 export const module = {
1242  // ...
1243 }
1244
1245 export const namespace = {
1246  // ...
1247 }
1248
1249 export const module = {
1250  // ...
1251 }
1252
1253 export const namespace = {
1254  // ...
1255 }
1256
1257 export const module = {
1258  // ...
1259 }
1260
1261 export const namespace = {
1262  // ...
1263 }
1264
1265 export const module = {
1266  // ...
1267 }
1268
1269 export const namespace = {
1270  // ...
1271 }
1272
1273 export const module = {
1274  // ...
1275 }
1276
1277 export const namespace = {
1278  // ...
1279 }
1280
1281 export const module = {
1282  // ...
1283 }
1284
1285 export const namespace = {
1286  // ...
1287 }
1288
1289 export const module = {
1290  // ...
1291 }
1292
1293 export const namespace = {
1294  // ...
1295 }
1296
1297 export const module = {
1298  // ...
1299 }
1300
1301 export const namespace = {
1302  // ...
1303 }
1304
1305 export const module = {
1306  // ...
1307 }
1308
1309 export const namespace = {
1310  // ...
1311 }
1312
1313 export const module = {
1314  // ...
1315 }
1316
1317 export const namespace = {
1318  // ...
1319 }
1320
1321 export const module = {
1322  // ...
1323 }
1324
1325 export const namespace = {
1326  // ...
1327 }
1328
1329 export const module = {
1330  // ...
1331 }
1332
1333 export const namespace = {
1334  // ...
1335 }
1336
1337 export const module = {
1338  // ...
1339 }
1340
1341 export const namespace = {
1342  // ...
1343 }
1344
1345 export const module = {
1346  // ...
1347 }
1348
1349 export const namespace = {
1350  // ...
1351 }
1352
1353 export const module = {
1354  // ...
1355 }
1356
1357 export const namespace = {
1358  // ...
1359 }
1360
1361 export const module = {
1362  // ...
1363 }
1364
1365 export const namespace = {
1366  // ...
1367 }
1368
1369 export const module = {
1370  // ...
1371 }
1372
1373 export const namespace = {
1374  // ...
1375 }
1376
1377 export const module = {
1378  // ...
1379 }
1380
1381 export const namespace = {
1382  // ...
1383 }
1384
1385 export const module = {
1386  // ...
1387 }
1388
1389 export const namespace = {
1390  // ...
1391 }
1392
1393 export const module = {
1394  // ...
1395 }
1396
1397 export const namespace = {
1398  // ...
1399 }
1400
1401 export const module = {
1402  // ...
1403 }
1404
1405 export const namespace = {
1406  // ...
1407 }
1408
1409 export const module = {
1410  // ...
1411 }
1412
1413 export const namespace = {
1414  // ...
1415 }
1416
1417 export const module = {
1418  // ...
1419 }
1420
1421 export const namespace = {
1422  // ...
1423 }
1424
1425 export const module = {
1426  // ...
1427 }
1428
1429 export const namespace = {
1430  // ...
1431 }
1432
1433 export const module = {
1434  // ...
1435 }
1436
1437 export const namespace = {
1438  // ...
1439 }
1440
1441 export const module = {
1442  // ...
1443 }
1444
1445 export const namespace = {
1446  // ...
1447 }
1448
1449 export const module = {
1450  // ...
1451 }
1452
1453 export const namespace = {
1454  // ...
1455 }
1456
1457 export const module = {
1458  // ...
1459 }
1460
1461 export const namespace = {
1462  // ...
1463 }
1464
1465 export const module = {
1466  // ...
1467 }
1468
1469 export const namespace = {
1470  // ...
1471 }
1472
1473 export const module = {
1474  // ...
1475 }
1476
1477 export const namespace = {
1478  // ...
1479 }
1480
1481 export const module = {
1482  // ...
1483 }
1484
1485 export const namespace = {
1486  // ...
1487 }
1488
1489 export const module = {
1490  // ...
1491 }
1492
1493 export const namespace = {
1494  // ...
1495 }
1496
1497 export const module = {
1498  // ...
1499 }
1500
1501 export const namespace = {
1502  // ...
1503 }
1504
1505 export const module = {
1506  // ...
1507 }
1508
1509 export const namespace = {
1510  // ...
1511 }
1512
1513 export const module = {
1514  // ...
1515 }
1516
1517 export const namespace = {
1518  // ...
1519 }
1520
1521 export const module = {
1522  // ...
1523 }
1524
1525 export const namespace = {
1526  // ...
1527 }
1528
1529 export const module = {
1530  // ...
1531 }
1532
1533 export const namespace = {
1534  // ...
1535 }
1536
1537 export const module = {
1538  // ...
1539 }
1540
1541 export const namespace = {
1542  // ...
1543 }
1544
1545 export const module = {
1546  // ...
1547 }
1548
1549 export const namespace = {
1550  // ...
1551 }
1552
1553 export const module = {
1554  // ...
1555 }
1556
1557 export const namespace = {
1558  // ...
1559 }
1560
1561 export const module = {
1562  // ...
1563 }
1564
1565 export const namespace = {
1566  // ...
1567 }
1568
1569 export const module = {
1570  // ...
1571 }
1572
1573 export const namespace = {
1574  // ...
1575 }
1576
1577 export const module = {
1578  // ...
1579 }
1580
1581 export const namespace = {
1582  // ...
1583 }
1584
1585 export const module = {
1586  // ...
1587 }
1588
1589 export const namespace = {
1590  // ...
1591 }
1592
1593 export const module = {
1594  // ...
1595 }
1596
1597 export const namespace = {
1598  // ...
1599 }
1600
1601 export const module = {
1602  // ...
1603 }
1604
1605 export const namespace = {
1606  // ...
1607 }
1608
1609 export const module = {
1610  // ...
1611 }
1612
1613 export const namespace = {
1614  // ...
1615 }
1616
1617 export const module = {
1618  // ...
1619 }
1620
1621 export const namespace = {
1622  // ...
1623 }
1624
1625 export const module = {
1626  // ...
1627 }
1628
1629 export const namespace = {
1630  // ...
1631 }
1632
1633 export const module = {
1634  // ...
1635 }
1636
1637 export const namespace = {
1638  // ...
1639 }
1640
1641 export const module = {
1642  // ...
1643 }
1644
1645 export const namespace = {
1646  // ...
1647 }
1648
1649 export const module = {
1650  // ...
1651 }
1652
1653 export const namespace = {
1654  // ...
1655 }
1656
1657 export const module = {
1658  // ...
1659 }
1660
1661 export const namespace = {
1662  // ...
1663 }
1664
1665 export const module = {
1666  // ...
1667 }
1668
1669 export const namespace = {
1670  // ...
1671 }
1672
1673 export const module = {
1674  // ...
1675 }
1676
1677 export const namespace = {
1678  // ...
1679 }
1680
1681 export const module = {
1682  // ...
1683 }
1684
1685 export const namespace = {
1686  // ...
1687 }
1688
1689 export const module = {
1690  // ...
1691 }
1692
1693 export const namespace = {
1694  // ...
1695 }
1696
1697 export const module = {
1698  // ...
1699 }
1700
1701 export const namespace = {
1702  // ...
1703 }
1704
1705 export const module = {
1706  // ...
1707 }
1708
1709 export const namespace = {
1710  // ...
1711 }
1712
1713 export const module = {
1714  // ...
1715 }
1716
1717 export const namespace = {
1718  // ...
1719 }
1720
1721 export const module = {
1722  // ...
1723 }
1724
1725 export const namespace = {
1726  // ...
1727 }
1728
1729 export const module = {
1730  // ...
1731 }
1732
1733 export const namespace = {
1734  // ...
1735 }
1736
1737 export const module = {
1738  // ...
1739 }
1740
1741 export const namespace = {
1742  // ...
1743 }
1744
1745 export const module = {
1746  // ...
1747 }
1748
1749 export const namespace = {
1750  // ...
1751 }
1752
1753 export const module = {
1754  // ...
1755 }
1756
1757 export const namespace = {
1758  // ...
1759 }
1760
1761 export const module = {
1762  // ...
1763 }
1764
1765 export const namespace = {
1766  // ...
1767 }
1768
1769 export const module = {
1770  // ...
1771 }
1772
1773 export const namespace = {
1774  // ...
1775 }
1776
1777 export const module = {
1778  // ...
```

```

1 import { IHouse } from '@/interfaces/house.interface'
2 import axiosApi from '../axios-api'
3
4 export const createHouse = async (info: any) => {
5     const { data } = await axiosApi.post('/house/create', info)
6     return data
7 }
8
9 export const getHouses = async () => {
10    const res = await axiosApi.get('/house/all')
11    const houses = res.data as IHouse[]
12    return houses
13 }
14
15 export const deleteHouse = async (_id: any) => {
16    const { data } = await axiosApi.delete('/house/delete', {
17        params: { houseId: _id },
18    })
19    return data
20 }
21
22 export const editHouse = async (info: any) => {
23    const { data } = await axiosApi.put('/house/update', info.body, {
24        params: { houseId: info._id },
25    })
26    return data
27 }
28
29 export const getOneHouse = async (_id: any) => {
30    try {
31        const res = await axiosApi.get('/house/by-id', {
32            params: { houseId: _id },
33        })
34
35        const house = res.data as IHouse
36        return house
37    } catch (error) {
38        console.log(error)
39
40        throw new Error('Internal Server Error')
41    }
42 }
43

```

```

1 import * as yup from 'yup'
2
3 export const budgetSchema = yup
4   .object({
5     name: yup.string().required('Nombre requerido'),
6     area: yup.number().required('Area requerida').typeError('Area debe ser un numero'),
7     numberFloors: yup.number().required('Numero de pisos requerido').typeError('Numero de pisos debe ser un numero'),
8     material: yup
9       .object({ value: yup.string(), label: yup.string() })
10      .default(null)
11      .required('Material requerido')
12      .typeError('Material debe ser un objeto'),
13
14     stats: yup.array().of(
15       yup.object({
16         label: yup.string().required('Etiqueta requerida'),
17         entry: yup.string().required('Rubro requerida'),
18       })
19     ),
20     isPaid: yup
21       .object({ value: yup.string(), label: yup.string() })
22       .default({
23         value: 'GRATIS',
24         label: 'GRATIS',
25       })
26       .required('Campo requerido')
27       .typeError('Debe ser un objeto'),
28
29     action: yup.string().required('Accion requerida').default('CALCULATE'),
30   })
31   .required()
32
33 export type BudgetFormData = yup.InferType<typeof budgetSchema>
34

```

```

1 'use client'
2
3 import { PropsWithChildren, useState } from 'react'
4 import { queryClientConfig } from '@/utils/query.config'
5 import { QueryClientProvider, QueryClient } from '@tanstack/react-query'
6 import { ReactQueryDevtools } from '@tanstack/react-query-devtools'
7 import { ToastContainer } from 'react-toastify'
8
9 const QueryProvider = ({ children }: PropsWithChildren) => {
10   const [client] = useState(new QueryClient(queryClientConfig))
11
12   return (
13     <
14       <ProgressBar height="4px" color="#6366f1" options={{ showSpinner: false }} shallowRouting appDirectory /> </>
15     <QueryClientProvider client={client}>
16       {children}
17       <ReactQueryDevtools initialIsOpen={false} position="bottom-right" />
18       <ToastContainer />
19     </QueryClientProvider>
20   </>
21 )
22 }
23
24 export default QueryProvider
25

```

```
1 #!/usr/bin/perl
2 #
3 # This script is a simple Perl program that prints out the
4 # contents of a file named 'data.txt' to the standard
5 # output. It uses the 'open' function to open the file
6 # and the 'print' function to output the contents.
7 #
8 # Usage: perl script.pl
9 #
10 # Author: Your Name
11 #
12 # License: Public Domain
13 #
14 # This script is provided as-is, without any warranty.
15 #
16 # Copyright (c) 2023, Your Name. All rights reserved.
17 #
18 # Redistribution and use in source and binary forms,
19 # with or without modification, are permitted provided
20 # that the following conditions are met:
21 #
22 # 1. Redistributions of source code must retain the
23 #    above copyright notice, this list of conditions
24 #    and the following disclaimer.
25 #
26 # 2. Redistributions in binary form must reproduce
27 #    the above copyright notice, this list of
28 #    conditions and the following disclaimer in the
29 #    documentation and/or other materials provided
30 #    with the distribution.
31 #
32 # 3. Neither the name of the copyright holder nor the
33 #    names of its contributors may be used to
34 #    endorse or promote products derived from this
35 #    software without specific prior written
36 #    permission.
37 #
38 # 4. If you have modified this code, you must
39 #    clearly indicate all modifications to the
40 #    source code.
41 #
42 # 5. In no event shall the copyright holder be
43 #    liable for any damages, including but not
44 #    limited to, direct, indirect, special,
45 #    incidental, or consequential damages, or any
46 #    losses or profits, arising out of the use of
47 #    this software, even if advised of the
48 #    possibility of such damage.
49 #
50 # IN NO EVENT SHALL THE COPYRIGHT HOLDER BE
51 # LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT
52 # LIMITED TO, DIRECT, INDIRECT, SPECIAL,
53 # INCIDENTAL, OR CONSEQUENTIAL DAMAGES, OR ANY
54 # LOSSES OR PROFITS, ARISING OUT OF THE USE OF
55 # THIS SOFTWARE, EVEN IF ADVISED OF THE
56 # POSSIBILITY OF SUCH DAMAGE.
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #
1001 #
1002 #
1003 #
1004 #
1005 #
1006 #
1007 #
1008 #
1009 #
1010 #
1011 #
1012 #
1013 #
1014 #
1015 #
1016 #
1017 #
1018 #
1019 #
1020 #
1021 #
1022 #
1023 #
1024 #
1025 #
1026 #
1027 #
1028 #
1029 #
1030 #
1031 #
1032 #
1033 #
1034 #
1035 #
1036 #
1037 #
1038 #
1039 #
1040 #
1041 #
1042 #
1043 #
1044 #
1045 #
1046 #
1047 #
1048 #
1049 #
1050 #
1051 #
1052 #
1053 #
1054 #
1055 #
1056 #
1057 #
1058 #
1059 #
1060 #
1061 #
1062 #
1063 #
1064 #
1065 #
1066 #
1067 #
1068 #
1069 #
1070 #
1071 #
1072 #
1073 #
1074 #
1075 #
1076 #
1077 #
1078 #
1079 #
1080 #
1081 #
1082 #
1083 #
1084 #
1085 #
1086 #
1087 #
1088 #
1089 #
1090 #
1091 #
1092 #
1093 #
1094 #
1095 #
1096 #
1097 #
1098 #
1099 #
1100 #
1101 #
1102 #
1103 #
1104 #
1105 #
1106 #
1107 #
1108 #
1109 #
1110 #
1111 #
1112 #
1113 #
1114 #
1115 #
1116 #
1117 #
1118 #
1119 #
1120 #
1121 #
1122 #
1123 #
1124 #
1125 #
1126 #
1127 #
1128 #
1129 #
1130 #
1131 #
1132 #
1133 #
1134 #
1135 #
1136 #
1137 #
1138 #
1139 #
1140 #
1141 #
1142 #
1143 #
1144 #
1145 #
1146 #
1147 #
1148 #
1149 #
1150 #
1151 #
1152 #
1153 #
1154 #
1155 #
1156 #
1157 #
1158 #
1159 #
1160 #
1161 #
1162 #
1163 #
1164 #
1165 #
1166 #
1167 #
1168 #
1169 #
1170 #
1171 #
1172 #
1173 #
1174 #
1175 #
1176 #
1177 #
1178 #
1179 #
1180 #
1181 #
1182 #
1183 #
1184 #
1185 #
1186 #
1187 #
1188 #
1189 #
1190 #
1191 #
1192 #
1193 #
1194 #
1195 #
1196 #
1197 #
1198 #
1199 #
1200 #
1201 #
1202 #
1203 #
1204 #
1205 #
1206 #
1207 #
1208 #
1209 #
1210 #
1211 #
1212 #
1213 #
1214 #
1215 #
1216 #
1217 #
1218 #
1219 #
1220 #
1221 #
1222 #
1223 #
1224 #
1225 #
1226 #
1227 #
1228 #
1229 #
1230 #
1231 #
1232 #
1233 #
1234 #
1235 #
1236 #
1237 #
1238 #
1239 #
1240 #
1241 #
1242 #
1243 #
1244 #
1245 #
1246 #
1247 #
1248 #
1249 #
1250 #
1251 #
1252 #
1253 #
1254 #
1255 #
1256 #
1257 #
1258 #
1259 #
1260 #
1261 #
1262 #
1263 #
1264 #
1265 #
1266 #
1267 #
1268 #
1269 #
1270 #
1271 #
1272 #
1273 #
1274 #
1275 #
1276 #
1277 #
1278 #
1279 #
1280 #
1281 #
1282 #
1283 #
1284 #
1285 #
1286 #
1287 #
1288 #
1289 #
1290 #
1291 #
1292 #
1293 #
1294 #
1295 #
1296 #
1297 #
1298 #
1299 #
1300 #
1301 #
1302 #
1303 #
1304 #
1305 #
1306 #
1307 #
1308 #
1309 #
1310 #
1311 #
1312 #
1313 #
1314 #
1315 #
1316 #
1317 #
1318 #
1319 #
1320 #
1321 #
1322 #
1323 #
1324 #
1325 #
1326 #
1327 #
1328 #
1329 #
1330 #
1331 #
1332 #
1333 #
1334 #
1335 #
1336 #
1337 #
1338 #
1339 #
1340 #
1341 #
1342 #
1343 #
1344 #
1345 #
1346 #
1347 #
1348 #
1349 #
1350 #
1351 #
1352 #
1353 #
1354 #
1355 #
1356 #
1357 #
1358 #
1359 #
1360 #
1361 #
1362 #
1363 #
1364 #
1365 #
1366 #
1367 #
1368 #
1369 #
1370 #
1371 #
1372 #
1373 #
1374 #
1375 #
1376 #
1377 #
1378 #
1379 #
1380 #
1381 #
1382 #
1383 #
1384 #
1385 #
1386 #
1387 #
1388 #
1389 #
1390 #
1391 #
1392 #
1393 #
1394 #
1395 #
1396 #
1397 #
1398 #
1399 #
1400 #
1401 #
1402 #
1403 #
1404 #
1405 #
1406 #
1407 #
1408 #
1409 #
1410 #
1411 #
1412 #
1413 #
1414 #
1415 #
1416 #
1417 #
1418 #
1419 #
1420 #
1421 #
1422 #
1423 #
1424 #
1425 #
1426 #
1427 #
1428 #
1429 #
1430 #
1431 #
1432 #
1433 #
1434 #
1435 #
1436 #
1437 #
1438 #
1439 #
1440 #
1441 #
1442 #
1443 #
1444 #
1445 #
1446 #
1447 #
1448 #
1449 #
1450 #
1451 #
1452 #
1453 #
1454 #
1455 #
1456 #
1457 #
1458 #
1459 #
1460 #
1461 #
1462 #
1463 #
1464 #
1465 #
1466 #
1467 #
1468 #
1469 #
1470 #
1471 #
1472 #
1473 #
1474 #
1475 #
1476 #
1477 #
1478 #
1479 #
1480 #
1481 #
1482 #
1483 #
1484 #
1485 #
1486 #
1487 #
1488 #
1489 #
1490 #
1491 #
1492 #
1493 #
1494 #
1495 #
1496 #
1497 #
1498 #
1499 #
1500 #
1501 #
1502 #
1503 #
1504 #
1505 #
1506 #
1507 #
1508 #
1509 #
1510 #
1511 #
1512 #
1513 #
1514 #
1515 #
1516 #
1517 #
1518 #
1519 #
1520 #
1521 #
1522 #
1523 #
1524 #
1525 #
1526 #
1527 #
1528 #
1529 #
1530 #
1531 #
1532 #
1533 #
1534 #
1535 #
1536 #
1537 #
1538 #
1539 #
1540 #
1541 #
1542 #
1543 #
1544 #
1545 #
1546 #
1547 #
1548 #
1549 #
1550 #
1551 #
1552 #
1553 #
1554 #
1555 #
1556 #
1557 #
1558 #
1559 #
1560 #
1561 #
1562 #
1563 #
1564 #
1565 #
1566 #
1567 #
1568 #
1569 #
1570 #
1571 #
1572 #
1573 #
1574 #
1575 #
1576 #
1577 #
1578 #
1579 #
1580 #
1581 #
1582 #
1583 #
1584 #
1585 #
1586 #
1587 #
1588 #
1589 #
1590 #
1591 #
1592 #
1593 #
1594 #
1595 #
1596 #
1597 #
1598 #
1599 #
1600 #
1601 #
1602 #
1603 #
1604 #
1605 #
1606 #
1607 #
1608 #
1609 #
1610 #
1611 #
1612 #
1613 #
1614 #
1615 #
1616 #
1617 #
1618 #
1619 #
1620 #
1621 #
1622 #
1623 #
1624 #
1625 #
1626 #
1627 #
1628 #
1629 #
1630 #
1631 #
1632 #
1633 #
1634 #
1635 #
1636 #
1637 #
1638 #
1639 #
1640 #
1641 #
1642 #
1643 #
1644 #
1645 #
1646 #
1647 #
1648 #
1649 #
1650 #
1651 #
1652 #
1653 #
1654 #
1655 #
1656 #
1657 #
1658 #
1659 #
1660 #
1661 #
1662 #
1663 #
1664 #
1665 #
1666 #
1667 #
1668 #
1669 #
1670 #
1671 #
1672 #
1673 #
1674 #
1675 #
1676 #
1677 #
1678 #
1679 #
1680 #
1681 #
1682 #
1683 #
1684 #
1685 #
1686 #
1687 #
1688 #
1689 #
1690 #
1691 #
1692 #
1693 #
1694 #
1695 #
1696 #
1697 #
1698 #
1699 #
1700 #
1701 #
1702 #
1703 #
1704 #
1705 #
1706 #
1707 #
1708 #
1709 #
1710 #
1711 #
1712 #
1713 #
1714 #
1715 #
1716 #
1717 #
1718 #
1719 #
1720 #
1721 #
1722 #
1723 #
1724 #
1725 #
1726 #
1727 #
1728 #
1729 #
1730 #
1731 #
1732 #
1733 #
1734 #
1735 #
1736 #
1737 #
1738 #
1739 #
1740 #
1741 #
1742 #
1743 #
1744 #
1745 #
1746 #
1747 #
1748 #
1749 #
1750 #
1751 #
1752 #
1753 #
1754 #
1755 #
1756 #
1757 #
1758 #
1759 #
1760 #
1761 #
1762 #
1763 #
1764 #
1765 #
1766 #
1767 #
1768 #
1769 #
1770 #
1771 #
1772 #
1773 #
1774 #
1775 #
1776 #
1777 #
1778 #
1779 #
1780 #
1781 #
1782 #
1783 #
1784 #
1785 #
1786 #
1787 #
1788 #
1789 #
1790 #
1791 #
1792 #
1793 #
1794 #
1795 #
1796 #
1797 #
1798 #
1799 #
1800 #
1801 #
1802 #
1803 #
1804 #
1805 #
1806 #
1807 #
1808 #
1809 #
1810 #
1811 #
1812 #
1813 #
1814 #
1815 #
1816 #
1817 #
1818 #
1819 #
1820 #
1821 #
1822 #
1823 #
1824 #
1825 #
1826 #
1827 #
1828 #
1829 #
1830 #
1831 #
1832 #
1833 #
1834 #
1835 #
1836 #
1837 #
1838 #
1839 #
1840 #
1841 #
1842 #
1843 #
1844 #
1845 #
1846 #
1847 #
1848 #
1849 #
1850 #
1851 #
1852 #
1853 #
1854 #
1855 #
1856 #
1857 #
1858 #
1859 #
1860 #
1861 #
1862 #
1863 #
1864 #
1865 #
1866 #
1867 #
1868 #
1869 #
1870 #
1871 #
1872 #
1873 #
1874 #
1875 #
1876 #
1877 #
1878 #
1879 #
1880 #
1881 #
1882 #
1883 #
1884 #
1885 #
1886 #
1887 #
1888 #
1889 #
1890 #
1891 #
1892 #
1893 #
1894 #
1895 #
1896 #
1897 #
1898 #
1899 #
1900 #
1901 #
1902 #
1903 #
1904 #
1905 #
1906 #
1907 #
1908 #
1909 #
1910 #
1911 #
1912 #
1913 #
1914 #
1915 #
1916 #
1917 #
1918 #
1919 #
1920 #
1921 #
1922 #
1923 #
1924 #
1925 #
1926 #
1927 #
1928 #
1929 #
1930 #
1931 #
1932 #
1933 #
1934 #
1935 #
1936 #
1937 #
1938 #
1939 #
1940 #
1941 #
1942 #
1943 #
1944 #
1945 #
1946 #
1947 #
1948 #
1949 #
1950 #
1951 #
1952 #
1953 #
1954 #
1955 #
1956 #
1957 #
1958 #
1959 #
1960 #
1961 #
1962 #
1963 #
1964 #
1965 #
1966 #
1967 #
1968 #
1969 #
1970 #
1971 #
1972 #
1973 #
1974 #
1975 #
1976 #
1977 #
1978 #
1979 #
1980 #
1981 #
1982 #
1983 #
1984 #
1985 #
1986 #
1987 #
1988 #
1989 #
1990 #
1991 #
1992 #
1993 #
1994 #
1995 #
1996 #
1997 #
1998 #
1999 #
2000 #
2001 #
2002 #
2003 #
2004 #
2005 #
2006 #
2007 #
2008 #
2009 #
2010 #
2011 #
2012 #
2013 #
2014 #
2015 #
2016 #
2017 #
2018 #
2019 #
2020 #
2021 #
2022 #
2023 #
2024 #
2025 #
2026 #
2027 #
2028 #
2029 #
2030 #
2031 #
2032 #
2033 #
2034 #
2035 #
2036 #
2037 #
2038 #
2039 #
2040 #
2041 #
2042 #
2043 #
2044 #
2045 #
2046 #
2047 #
2048 #
2049 #
2050 #
2051 #
2052 #
2053 #
2054 #
2055 #
2056 #
2057 #
2058 #
2059 #
2060 #
2061 #
2062 #
2063 #
2064 #
2065 #
2066 #
2067 #
2068 #
2069 #
2070 #
2071 #
2072 #
2073 #
2074 #
2075 #
2076 #
2077 #
2078 #
2079 #
2080 #
2081 #
2082 #
2083 #
2084 #
2085 #
2086 #
2087 #
2088 #
2089 #
2090 #
2091 #
2092 #
2093 #
2094 #
2095 #
2096 #
2097 #
2098 #
2099 #
2100 #
2101 #
2102 #
2103 #
2104 #
2105 #
2106 #
2107 #
2108 #
2109 #
2110 #
2111 #
2112 #
2113 #
2114 #
2115 #
2116 #
2117 #
2118 #
2119 #
2120 #
2121 #
2122 #
2123 #
2124 #
2125 #
2126 #
2127 #
2128 #
2129 #
2130 #
2131 #
2132 #
2133 #
2134 #
2135 #
2136 #
2137 #
2138 #
2139 #
2140 #
2141 #
2142 #
2143 #
2144 #
2145 #
2146 #
2147 #
2148 #
2149 #
2150 #
2151 #
2152 #
2153 #
2154 #
2155 #
2156 #
2157 #
2158 #
2159 #
2160 #
2161 #
2162 #
2163 #
2164 #
2165 #
2166 #
2167 #
2168 #
2169 #
2170 #
2171 #
2172 #
2173 #
2174 #
2175 #
2176 #
2177 #
2178 #
2179 #
2180 #
2181 #
2182 #
2183
```

```

1 import { createHouse, deleteHouse, editHouse } from '@/annie-api/queries/house.query'
2 import { useMutation, useQueryClient } from '@tanstack/react-query'
3
4 export const useCreateHouse = () => {
5   const queryClient = useQueryClient()
6   return useMutation(createHouse, {
7     onSuccess: () => {
8       queryClient.invalidateQueries(['houses'])
9     },
10  })
11 }
12
13 export const useDeleteHouse = () => {
14   const queryClient = useQueryClient()
15   return useMutation(deleteHouse, {
16     onSuccess: () => {
17       queryClient.invalidateQueries(['houses'])
18     },
19  })
20 }
21
22 export const useEditHouse = () => {
23   const queryClient = useQueryClient()
24   return useMutation(editHouse, {
25     onSuccess: (_, { _id }) => {
26       queryClient.invalidateQueries(['houses'])
27       queryClient.invalidateQueries({
28         queryKey: ['house', _id],
29       })
30     },
31  })
32 }
33

```

```

1 import { useDeleteBudget } from '@/hooks/useBudget'
2 import { useToast } from '@/hooks/useToast'
3 import { IBudget } from '@/interfaces/budget.interface'
4 import { AButton } from '../button/AButton'
5 import { Modal, ModalCoreProps } from './Modal'
6
7 interface ModalDeleteBudgetProps extends ModalCoreProps {
8   budget: IBudget | undefined
9 }
10
11 export const ModalDeleteBudget = (props: ModalDeleteBudgetProps) => {
12   const { showToast } = useToast()
13
14   const { mutate, isLoading } = useDeleteBudget()
15
16   const { budget, ...rest } = props
17
18   return (
19     <Modal {...rest} title="Eliminar una casa">
20       <div>
21         <p className="text-slate-700">
22           ¿Estas seguro que deseas eliminar el presupuesto{' '}
23           <span className="font-medium">{props.budget?.name}</span>?
24         </p>
25
26         <div className="mt-3 flex justify-end">
27           <AButton
28             label="Eliminar"
29             type="button"
30             isLoading={isLoading}
31             onClick={() => {
32               mutate(budget?._id, {
33                 onSuccess: () => {
34                   props.setIsOpen(false)
35                   showToast('Presupuesto eliminado correctamente', 'success')
36                 },
37                 onError: (e: any) => {
38                   showToast(e.response.data.message, 'error')
39                 },
40               })
41             }}
42           </>
43         </div>
44       </div>
45     </Modal>
46   )
47 }
48

```

```

1  'use client'
2
3  import { getHouses } from '@/annie-api/queries/house.query'
4  import { IHouse } from '@/interfaces/house.interface'
5  import { useQuery } from '@tanstack/react-query'
6  import { useState } from 'react'
7  import { ModalDeleteHouse } from '../modal/ModalDeleteHouse'
8  import { materials } from '@/assets/data/general.data'
9  import Link from 'next/link'
10
11 export const TableHouse = () => {
12   const [isDeleteOpen, setIsDeleteOpen] = useState(false)
13
14   const [house, setHouse] = useState<IHouse>()
15
16   const { data } = useQuery({
17     queryKey: ['houses'],
18     queryFn: () => getHouses(),
19   })
20
21   return (
22     <
23       <table className="w-full text-left text-sm text-gray-500 ">
24         <thead className="bg-gray-50 text-xs uppercase text-gray-700 ">
25           <tr>
26             <th scope="col" className="px-6 py-3">
27               Descripción
28             </th>
29             <th scope="col" className="px-6 py-3">
30               Área
31             </th>
32
33             <th scope="col" className="px-6 py-3">
34               Pisos
35             </th>
36
37             <th scope="col" className="px-6 py-3">
38               Material
39             </th>
40
41             <th scope="col" className="px-6 py-3">
42               Acción
43             </th>
44           </tr>
45         </thead>
46         <tbody>
47           {data?.map(({ _id, name, area, stats, material, numberFloors }) => (
48             <tr key={_id} className="border-b bg-white">
49               <th scope="row" className="whitespace-nowrap px-6 py-4 font-medium text-gray-900">
50                 {name}
51               </th>
52               <td className="px-6 py-4">{area}</td>
53               <td className="px-6 py-4">{numberFloors}</td>
54               <td className="px-6 py-4">{materials.find((m) => m.value === material)?.label}</td>
55
56               <td className="flex items-center space-x-3 px-6 py-4">
57                 <Link href={`/casas/${_id}`} className="font-medium text-indigo-600">
58                   Editar
59                 </Link>
60                 <button
61                   onClick={() => {
62                     setHouse({ _id, name, area, stats, material, numberFloors })
63                     setIsDeleteOpen(true)
64                   }}
65                   className="font-medium text-red-600"
66                 >
67                   Eliminar
68                 </button>
69               </td>
70             </tr>
71           )})}
72
73           {!data?.length && (
74             <tr className="border-b bg-white">
75               <td className="px-6 py-4 text-center" colspan={5}>
76                 No hay casas registradas
77               </td>
78             </tr>
79           )}
80         </tbody>
81       </table>
82       <ModalDeleteHouse isOpen={isDeleteOpen} setIsOpen={setIsDeleteOpen} house={house} />
83     </
84   )
85 }
86

```

## Código del servidor de la base de datos

```
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3 import { ValidationPipe, BadRequestException, RequestMethod } from '@nestjs/common';
4
5 import { ValidationError } from 'class-validator';
6
7 async function bootstrap() {
8   const app = await NestFactory.create(AppModule);
9
10  app.enableCors();
11  app.setGlobalPrefix('api', {
12    exclude: [{ path: '/', method: RequestMethod.GET }],
13  });
14
15  app.useGlobalPipes(
16    new ValidationPipe({
17      whitelist: true,
18      stopAtFirstError: true,
19      transform: true,
20      enableDebugMessages: true,
21      exceptionFactory: (errors: ValidationError[]) => {
22        let onlyError: string | undefined;
23
24        while (errors.length > 0) {
25          const error = errors.shift();
26
27          if (error?.children?.length) {
28            errors.push(...error.children);
29          } else {
30            onlyError = error?.constraints?.[Object.keys(error.constraints)[0]];
31            break;
32          }
33        }
34        return new BadRequestException(onlyError);
35      },
36    }),
37  );
38
39  await app.listen(process.env.PORT || 4004);
40  console.log(`Application is running on: http://localhost:${process.env.PORT || 4004}`);
41 }
42
43 bootstrap();
44
```

```

1 import { Module } from '@nestjs/common';
2 import { AppController } from './app.controller';
3 import { AppService } from './app.service';
4 import { ConfigModule } from '@nestjs/config';
5 import { EnvConfiguration } from './config/app.config';
6 import { MongooseModule } from '@nestjs/mongoose';
7 import { ConfigService } from '@nestjs/config';
8 import { EntryModule } from './entry/entry.module';
9 import { HouseModule } from './house/house.module';
10 import { CommonModule } from './common/common.module';
11 import { BudgetModule } from './budget/budget.module';
12
13 @Module({
14   imports: [
15     ConfigModule.forRoot({ load: [EnvConfiguration], isGlobal: true }),
16
17     MongooseModule.forRootAsync({
18       imports: [ConfigModule],
19       useFactory: async (configService: ConfigService) => ({
20         uri: configService.get('mongoUri'),
21       }),
22       inject: [ConfigService],
23     }),
24
25     EntryModule,
26
27     HouseModule,
28
29     CommonModule,
30
31     BudgetModule,
32   ],
33   controllers: [AppController],
34   providers: [AppService],
35 })
36 export class AppModule {}
37

```

```

1 import { Body, Controller, Delete, Get, Post, Put, Query } from '@nestjs/common';
2 import { ParseMongoIdPipe } from 'src/common/pipes/parse-mongo-id.pipe';
3 import { CreateHouseDto } from './dto/create-house.dto';
4 import { UpdateHouseDto } from './dto/update-house.dto';
5 import { HouseService } from './house.service';
6
7 @Controller('house')
8 export class HouseController {
9   constructor(private readonly houseService: HouseService) {}
10
11   @Post('create')
12   create(@Body() createHouseDto: CreateHouseDto) {
13     return this.houseService.create(createHouseDto);
14   }
15
16   @Get('update-all')
17   updateAll() {
18     return this.houseService.updateAll();
19   }
20
21   @Get('all')
22   findAll() {
23     return this.houseService.findAll();
24   }
25
26   @Get('by-id')
27   findOne(@Query('houseId', ParseMongoIdPipe) id: string) {
28     return this.houseService.findOne(id);
29   }
30
31   @Put('update')
32   update(@Query('houseId', ParseMongoIdPipe) id: string, @Body() updateHouseDto: UpdateHouseDto) {
33     return this.houseService.update(id, updateHouseDto);
34   }
35
36   @Delete('delete')
37   remove(@Query('houseId', ParseMongoIdPipe) id: string) {
38     return this.houseService.remove(id);
39   }
40 }
41

```

```
1 import { Type } from 'class-transformer';
2 import { IsArray, IsNotEmpty, IsNumber, IsString, ValidateNested } from 'class-validator';
3 import { StatDto } from './entry.dto';
4
5 export class CreateHouseDto {
6   @IsNotEmpty()
7   @IsString()
8   name: string;
9
10  @IsNotEmpty()
11  @IsNumber()
12  area: number;
13
14  @IsNotEmpty()
15  @IsNumber()
16  numberFloors: number;
17
18  @IsNotEmpty()
19  @IsString()
20  material: string;
21
22  @IsArray()
23  @ValidateNested({ each: true })
24  @Type(() => StatDto)
25  stats: StatDto[];
26 }
27
```

```

1 import { Prop, Schema, SchemaFactory, raw } from '@nestjs/mongoose';
2 import mongoose from 'mongoose';
3 import { Entry } from 'src/entry/entities/entry.entity';
4
5 @Schema({ timestamps: true, versionKey: false, collection: 'Houses' })
6 export class House extends mongoose.Document {
7   @Prop({ type: String, required: true, trim: true })
8   name: string;
9
10  @Prop({ type: Number, required: true })
11  area: number;
12
13  @Prop({ type: Number, required: true })
14  numberFloors: number;
15
16  @Prop({ type: String, required: true })
17  material: string;
18
19  @Prop(
20    raw([
21      {
22        entry: { type: mongoose.Schema.Types.ObjectId, ref: 'Entry' },
23        value: { type: Number, required: true },
24      },
25    ]),
26  )
27  stats: string[] & Entry[];
28
29  @Prop({ type: Boolean, default: false })
30  comesBudget: boolean;
31 }
32
33 export const HouseSchema = SchemaFactory.createForClass(House);
34

```

```

1 import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
2 import mongoose from 'mongoose';
3 import { Entry } from 'src/entry/entities/entry.entity';
4
5 @Schema({ timestamps: true, versionKey: false, collection: 'Budgets' })
6 export class Budget extends mongoose.Document {
7   @Prop({ type: String, required: true, trim: true })
8   name: string;
9
10  @Prop([{ type: mongoose.Schema.Types.ObjectId, required: true, ref: 'Entry' }])
11  stats: string[] & Entry[];
12
13  @Prop({ type: Number, required: true })
14  total: number;
15
16  @Prop({ type: Boolean, default: false })
17  isPaid: boolean;
18
19  @Prop({ type: Number, required: true })
20  area: number;
21
22  @Prop({ type: Number, required: true })
23  numberFloors: number;
24
25  @Prop({ type: String, required: true })
26  material: string;
27 }
28
29 export const BudgetSchema = SchemaFactory.createForClass(Budget);
30

```

```

1 import { Injectable, InternalServerErrorException, NotFoundException } from '@nestjs/common';
2 import { InjectModel } from '@nestjs/mongoose';
3 import { AggregatePaginateModel } from 'mongoose';
4 import { Budget } from 'src/budget/entities/budget.entity';
5 import { House } from 'src/house/entities/house.entity';
6 import { CreateEntryDto } from './dto/create-entry.dto';
7 import { UpdateEntryDto } from './dto/update-entry.dto';
8 import { Entry } from './entities/entry.entity';
9
10 @Injectable()
11 export class EntryService {
12   constructor(
13     @InjectModel(House.name)
14     private readonly houseModel: AggregatePaginateModel<House>,
15
16     @InjectModel(Entry.name)
17     private readonly entryModel: AggregatePaginateModel<Entry>,
18
19     @InjectModel(Budget.name)
20     private readonly budgetModel: AggregatePaginateModel<Budget>,
21   ) {}
22
23   async create(createEntryDto: CreateEntryDto) {
24     try {
25       const entry = new this.entryModel(createEntryDto);
26       await entry.save();
27       return { message: 'Rubro creado correctamente' };
28     } catch (error) {
29       console.log(error);
30       throw new InternalServerErrorException('Ocurrio un error al crear el rubro');
31     }
32   }
33
34   async findAll() {
35     try {
36       const entries = await this.entryModel.find({}, { createdAt: 0, updatedAt: 0 });
37
38       return entries;
39     } catch (error) {
40       console.log(error);
41       throw new InternalServerErrorException('Ocurrio un error al obtener los rubros');
42     }
43   }
44
45   findOne(id: number) {
46     return `This action returns a #${id} entry`;
47   }
48
49   async update(id: string, updateEntryDto: UpdateEntryDto) {
50     try {
51       await this.exists(id);
52
53       await this.entryModel.updateOne({ _id: id }, { $set: { ...updateEntryDto } });
54
55       return { message: 'Rubro actualizado correctamente' };
56     } catch (error) {
57       console.log(error);
58       throw new InternalServerErrorException('Ocurrio un error al actualizar el rubro');
59     }
60   }
61
62   async remove(id: string) {
63     try {
64       await this.exists(id);
65
66       await this.entryModel.deleteOne({ _id: id });
67
68       // Delete the entry from all houses in stats.entry
69       await this.houseModel.updateMany({}, { $pull: { stats: { entry: id } } });
70
71       // Delete the entry from all budgets in stats
72       await this.budgetModel.updateMany({}, { $pull: { stats: id } });
73
74       return { message: 'Rubro eliminado correctamente' };
75     } catch (error) {
76       console.log(error);
77       throw new InternalServerErrorException('Ocurrio un error al actualizar el rubro');
78     }
79   }
80
81   private async exists(id: string) {
82     if (!(await this.entryModel.exists({ _id: id }).lean()))
83       throw new NotFoundException(`No existe el rubro con el ID: ${id}`);
84   }
85 }
86

```

```
1 import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
2 import mongoose from 'mongoose';
3
4 @Schema({ timestamps: true, versionKey: false, collection: 'Entries' })
5 export class Entry extends mongoose.Document {
6   @Prop({ type: String, required: true, trim: true })
7   name: string;
8
9   @Prop({ type: String, required: true })
10  measure: string;
11
12  @Prop({ type: String, required: true })
13  chapter: string;
14
15  @Prop({ type: Number, required: true })
16  unitPrice: number;
17
18  @Prop({ type: String, required: true })
19  material: string;
20
21  @Prop({ type: Number, required: true })
22  parameterA: number;
23
24  @Prop({ type: Number, required: true })
25  parameterB: number;
26
27  @Prop({ type: Number, required: true })
28  parameterC: number;
29
30  @Prop({ type: Number, required: true })
31  parameterD: number;
32
33  @Prop({ type: Boolean, default: false })
34  grayWork: boolean;
35 }
36
37 export const EntrySchema = SchemaFactory.createForClass(Entry);
38
```

```
1 import { IsBoolean, IsNotEmpty, IsNumber, IsOptional, IsString } from 'class-validator';
2
3 export class CreateEntryDto {
4   @IsString()
5   @IsNotEmpty()
6   name: string;
7
8   @IsString()
9   @IsNotEmpty()
10  measure: string;
11
12  @IsString()
13  @IsNotEmpty()
14  chapter: string;
15
16  @IsNumber()
17  unitPrice: number;
18
19  @IsString()
20  @IsNotEmpty()
21  material: string;
22
23  @IsNumber()
24  parameterA: number;
25
26  @IsNumber()
27  parameterB: number;
28
29  @IsNumber()
30  parameterC: number;
31
32  @IsNumber()
33  parameterD: number;
34
35  @IsBoolean()
36  @IsOptional()
37  grayWork?: boolean;
38 }
39
```

```

1 import { Module, forwardRef } from '@nestjs/common';
2 import { MongooseModule } from '@nestjs/mongoose';
3 import { BudgetModule } from 'src/budget/budget.module';
4 import { HouseModule } from 'src/house/house.module';
5 import { Entry, EntrySchema } from './entities/entry.entity';
6 import { EntryController } from './entry.controller';
7 import { EntryService } from './entry.service';
8
9 @Module({
10   controllers: [EntryController],
11
12   providers: [EntryService],
13
14   imports: [
15     MongooseModule.forFeatureAsync([
16       {
17         name: Entry.name,
18         useFactory: async () => {
19           const schema = EntrySchema;
20           schema.plugin(require('mongoose-aggregate-paginate-v2'));
21           return schema;
22         },
23       },
24     ]),
25
26     HouseModule,
27
28     forwardRef(() => BudgetModule),
29   ],
30
31   exports: [MongooseModule],
32 })
33 export class EntryModule {}
34

```

```

1 {
2   "name": "annie-backend",
3   "version": "0.0.1",
4   "description": "",
5   "author": "",
6   "private": true,
7   "license": "UNLICENSED",
8   "scripts": {
9     "build": "nest build",
10    "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
11    "start": "nest start",
12    "start:dev": "nest start --watch",
13    "start:debug": "nest start --debug --watch",
14    "start:prod": "node dist/main",
15    "lint": "eslint \"{src,apps,libs,test}/**/*.ts\" --fix",
16    "test": "jest",
17    "test:watch": "jest --watch",
18    "test:cov": "jest --coverage",
19    "test:debug": "node --inspect-brk -r tsconfig-paths/register -r ts-node/register node_modules/.bin/jest --runInBand",
20    "test:e2e": "jest --config ./test/jest-e2e.json"
21  },
22  "dependencies": {
23    "@nestjs/common": "^10.0.0",
24    "@nestjs/config": "^3.0.0",
25    "@nestjs/core": "^10.0.0",
26    "@nestjs/mapped-types": "*",
27    "@nestjs/mongoose": "^10.0.0",
28    "@nestjs/platform-express": "^10.0.0",
29    "class-transformer": "^0.5.1",
30    "class-validator": "^0.14.0",
31    "mongoose": "^7.3.0",
32    "mongoose-aggregate-paginate-v2": "^1.0.6",
33    "reflect-metadata": "^0.1.13",
34    "rxjs": "^7.8.1"
35  },
36  "devDependencies": {
37    "@nestjs/cli": "^10.0.0",
38    "@nestjs/schematics": "^10.0.0",
39    "@nestjs/testing": "^10.0.0",
40    "@types/express": "^4.17.17",
41    "@types/jest": "^29.5.2",
42    "@types/mongoose-aggregate-paginate-v2": "^1.0.9",
43    "@types/node": "^20.3.1",
44    "@types/supertest": "^2.0.12",
45    "@typescript-eslint/eslint-plugin": "^5.59.11",
46    "@typescript-eslint/parser": "^5.59.11",
47    "eslint": "^8.42.0",
48    "eslint-config-prettier": "^8.8.0",
49    "eslint-plugin-prettier": "^4.2.1",
50    "jest": "^29.5.0",
51    "prettier": "^2.8.8",
52    "source-map-support": "^0.5.21",
53    "supertest": "^6.3.3",
54    "ts-jest": "^29.1.0",
55    "ts-loader": "^9.4.3",
56    "ts-node": "^10.9.1",
57    "tsconfig-paths": "^4.2.0",
58    "typescript": "^5.1.3"
59  },
60  "jest": {
61    "moduleFileExtensions": [
62      "js",
63      "json",
64      "ts"
65    ],
66    "rootDir": "src",
67    "testRegex": ".*\\.spec\\.ts$",
68    "transform": {
69      "^.+\\.\\.?(t|j)s?$": "ts-jest"
70    },
71    "collectCoverageFrom": [
72      "**/*.\\.?(t|j)s"
73    ],
74    "coverageDirectory": "../coverage",
75    "testEnvironment": "node"
76  }
77 }
78

```

Link de la aplicación MyHousingCosts:

<https://myhousingcosts.vercel.app/rubros>

Link donde está alojada la base de datos:

<https://www.mongodb.com/es/atlas>

## AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Anabel Solange Soto Merino portadora de la cédula de ciudadanía N° 1104069669. En calidad de autora y titular de los derechos patrimoniales del proyecto de titulación “Presupuestos para una vivienda de hasta dos pisos en hormigón o acero a través de una aplicación”, de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, así mismo; autorizo a la Universidad para que realice la publicación de éste proyecto de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, 06 de septiembre de 2023



F: .....

Anabel Solange Soto Merino

C.I. 1104069669