

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**“SISTEMA PRESUPUESTARIO DE INFORMACIÓN PARA PROYECTOS DE  
CONSTRUCCIÓN. CASO: EMPRESA MULTIDISCIPLINARIA ADU S.R.L.”**

**PARA OPTAR AL GRADO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA EN SISTEMAS INFORMÁTICOS**

**POSTULANTE:** MAURICIO APAZA BILBAO

**TUTOR DE METODOLÓGICO:** PHD. JAVIER HUGO GUSTAVO REYES PACHECO

**ASESOR:** LIC. VICTOR PABLO POZO DIAZ

**La Paz – Bolivia**

**2021**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## DEDICATORIA

El presente proyecto de grado está dedicado a todas las personas que me apoyaron incondicionalmente.

Gracias

## AGRADECIMIENTOS

Agradezco a la Universidad Mayor de San Andrés y a la Carrera de Informática por todos los conocimientos para mi formación.

A mi tutor metodológico Phd. Javier Hugo Gustavo Reyes Pacheco por el tiempo, los consejos y la orientación del proyecto.

A mi asesor Lic. Víctor Pablo Pozo Díaz por haberme brindado su tiempo y conocimiento para el avance del proyecto.

## RESUMEN

El presente proyecto fue realizado para la empresa ADU S.R.L. una empresa multidisciplinaria dedicada a la construcción y a la consultoría. Esta empresa no cuenta con una adecuada sistematización de los cálculos que efectúa día a día, con la realización del proyecto se pretende optimizar la realización de los cálculos del presupuesto para una obra de construcción y de esta forma ayudar a mejorar los tiempos de realización de los presupuesto.

Este proyecto ha sido realizado con la ayuda de la metodología RUP, esta metodología nos ayuda con el análisis y el diseño del sistema, cuenta con cuatro fases, además complementamos con la metodología para sistemas web UWE esta metodología nos ayuda con los modelos necesarios para el sistema.

La calidad del sistema fue medida basándonos en la normativa ISO 9126 aplicando sus conceptos y midiendo las características que tiene, los resultados son evaluados para determinar si el sistema es de calidad.

Se hizo un análisis de la seguridad tomando en cuenta la calidad ya que si no hay calidad no hay seguridad y se tomaron medidas para proteger el sistema de ataques maliciosos, protegiendo la base de datos, asignando roles y restringiendo el acceso a usuarios.

Se calculó el costo de nuestro sistema utilizando el método de COCOMO tomando en cuenta varias variables que afectan al costo final del sistema

Palabras clave: Metodología, cálculo, sistema, calidad, costo.

## **ABSTRACT**

This project was carried out for the company ADU S.R.L. a multidisciplinary company dedicated to construction and consulting. This company does not have an adequate systematization of the calculations that it carries out day by day, with the completion of the project it is intended to optimize the performance of the budget calculations for a construction site and in this way help to improve the times of completion of the budget.

This project has been carried out with the help of the RUP methodology, this methodology helps us with the analysis and design of the system, it has four phases, we also complement the methodology for UWE web systems, this methodology helps us with the necessary models for the system.

The quality of the system was measured based on the ISO 9126 standard, applying its concepts and measuring the characteristics it has, the results are evaluated to determine if the system is of quality.

A security analysis was made taking into account the quality since if there is no quality there is no security and measures were taken to protect the system from malicious attacks, protecting the database, assigning roles and restricting access to users.

The cost of our system was calculated using the COCOMO method taking into account several variables that affect the final cost of the system.

Keywords: Methodology, calculation, system, quality, cost.

## ÍNDICE

<b>CAPÍTULO I: MARCO INTRODUCTORIO</b>	<b>1</b>
<b>1.1 INTRODUCCIÓN</b>	<b>1</b>
<b>1.2 ANTECEDENTES</b>	<b>2</b>
<b>1.3 PROBLEMÁTICA</b>	<b>3</b>
1.3.1 PROBLEMA PRINCIPAL	3
1.3.2 PROBLEMAS SECUNDARIOS	4
<b>1.4 OBJETIVO</b>	<b>4</b>
1.4.1 OBJETIVO GENERAL	4
1.4.2 OBJETIVOS ESPECÍFICOS	4
<b>1.5 JUSTIFICACIÓN</b>	<b>5</b>
<b>1.6 METODOLOGÍA</b>	<b>5</b>
<b>CAPÍTULO II: MARCO TEÓRICO</b>	<b>6</b>
<b>2.1 ANÁLISIS DE PRECIOS UNITARIOS Y SU ESTRUCTURA</b>	<b>6</b>
2.1.3 MANO DE OBRA	8
2.1.4 EQUIPO, MAQUINARÍA Y HERRAMIENTA	8
2.1.4.1 EQUIPO Y MAQUINARIA	8
2.1.4.2 HERRAMIENTA	8
2.1.5 GASTOS GENERALES Y ADMINISTRATIVOS	9
2.1.6 UTILIDAD	9
2.1.7 IMPUESTOS	9
2.1.7.1 IMPUESTO AL VALOR AGREGADO (IVA)	10
2.1.7.2 IMPUESTO A LAS TRANSACCIONES (IT)	10
<b>2.2 INGENIERÍA DE SOFTWARE</b>	<b>10</b>
2.2.1 METODOLOGÍA ORIENTADA A OBJETOS	11
2.2.2 METODOLOGÍA RUP RATIONAL UNIFIED PROCESS	12
2.2.2.1 HISTORIA	12
2.2.2.2 FASES DEL PROCESO UNIFICADO	13
a) FASE DE CONCEPCIÓN	14
b) FASE DE ELABORACIÓN	14
c) FASE DE CONSTRUCCIÓN	15
d) FASE DE TRANSICIÓN	15
e) FASE DE PRODUCCIÓN	15

<b>2.3 INGENIERÍA WEB</b>	<b>16</b>
2.3.1 ETAPAS DE LA INGENIERÍA WEB	17
2.3.2 PROCESO DE LA INGENIERÍA WEB	18
2.3.3 METODOLOGÍA DE MODELADO UWE	19
2.3.3.1 MODELO DE CONTENIDO	20
2.3.3.2 MODELO DE NAVEGACION	20
2.3.3.3 MODELO DE PRESENTACION	21
2.3.3.4 MODELO DE PROCESO	22
2.3.3.5 FASES DE UWE	22
<b>2.4 EL LENGUAJE UNIFICADO PARA LA CONSTRUCCIÓN DE MODELOS (UML)</b>	<b>23</b>
2.4.1 DIAGRAMA DE CASOS DE USO	24
2.4.2 DIAGRAMA DE ACTIVIDADES	24
2.4.3 DIAGRAMA DE SECUENCIAS	24
2.4.4 DIAGRAMA DE CLASES	24
<b>CAPÍTULO III: MARCO APLICATIVO</b>	<b>26</b>
<b>3.1 INTRODUCCIÓN</b>	<b>26</b>
<b>3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL</b>	<b>26</b>
<b>3.3 PROCESO UNIFICADO RATIONAL RUP</b>	<b>26</b>
3.3.1 FASE 1: Concepción	26
3.3.1.1 Actores	27
3.3.1.2 Requisitos funcionales	27
3.3.1.3 Requerimientos no funcionales	28
3.3.2 FASE 2: Elaboración	28
3.3.2.1 Casos de uso	28
Administración de usuario	29
Administración del proyecto	30
Administración de la planilla de precios	32
Cálculo del presupuesto	33
Reporte del presupuesto	34
3.3.2.2 Diagrama de actividades	36
3.3.2.3 Diagrama de secuencias	41
3.3.2.1 Diagrama de clases	46
3.3.3 FASE 3: Construcción	47
3.3.3.1 Diagrama entidad relación	47
3.3.3.2 Modelo navegacional	47



3.3.3.3 Modelo presentación	48
3.3.4 FASE 4: Transición	52
3.3.4.1 PRUEBAS UNITARIAS	52
<b>CAPÍTULO IV: CALIDAD, SEGURIDAD Y COSTOS</b>	<b>54</b>
<b>4.1 INTRODUCCIÓN</b>	<b>54</b>
<b>4.2 FUNCIONALIDAD</b>	<b>54</b>
4.2.1 PUNTO FUNCIÓN	54
<b>4.3 CONFIABILIDAD</b>	<b>57</b>
<b>4.4 MANTENIBILIDAD</b>	<b>58</b>
<b>4.5 PORTABILIDAD</b>	<b>59</b>
4.5.1 PORTABILIDAD DE HARDWARE Y SOFTWARE	59
<b>4.6 USABILIDAD</b>	<b>59</b>
<b>4.7 SEGURIDAD</b>	<b>60</b>
4.7.1 SEGURIDAD EN BASE DE DATOS	61
4.7.1.1 COPIAS DE SEGURIDAD	61
4.7.2 SEGURIDAD DEL USUARIO	61
4.7.2.1 CONTROL DE ACCESO	62
<b>4.8 COSTOS</b>	<b>62</b>
4.8.1 COCOMO	62
<b>CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES</b>	<b>67</b>
<b>5.1 CONCLUSIONES</b>	<b>67</b>
<b>5.2 RECOMENDACIONES</b>	<b>67</b>
<b>BIBLIOGRAFÍA</b>	<b>68</b>
<b>ANEXOS</b>	<b>69</b>

## **CAPÍTULO I: MARCO INTRODUCTORIO**

### **1.1 INTRODUCCIÓN**

La tecnología tiene grandes avances y las empresas son beneficiadas gracias a estos avances, las empresas dedicadas a la construcción hacen el uso de equipos de computación además de usar sistemas informáticos para poder competir en el mercado y manejar la información que generan para la toma de decisiones y el respaldo de sus datos en nubes virtuales.

Dentro del área de construcción las empresas tienen éxito debido a la correcta ejecución de los proyectos la cual depende, de un análisis financiero el cual muestra el volumen de la inversión, en este influyen muchas variables para garantizar el proyecto y se determina si es viable financieramente, en la etapa de planeación se realiza el presupuesto en el cual se distribuyen los recursos.

Cada empresa constructora tiene sus propios métodos de cómo realizar sus presupuestos, no están del todo informadas sobre los beneficios que ofrece la informática, la forma de administrar los negocios ofreciéndoles mayor seguridad en su información, mejores registro de los procesos, dentro de la empresa se optimizará los recursos tanto materiales como humanos. En los presupuestos de obras se evitará la pérdida de información al estar toda centralizada en un software dentro de una nube virtual.

En el presente proyecto de grado se pretende describir el proceso de desarrollo de un sistema de información web para la gestión presupuestal en el área de la construcción, una vez elaborado el presupuesto se mostrará mediante formularios los resultados del proceso de la cotización de presupuestos.

Debido a esto se realizará el siguiente proyecto, el cual su objetivo es ayudar a la empresa constructora a mejorar sus servicios y tener ventaja con el uso de tecnologías de información. En el proyecto se desarrollara los siguientes capítulos  
El capítulo 1. Desarrollaremos el propósito del proyecto de grado para entender :el planteamiento de este proyecto, mostraremos las características generales que se pretende alcanzar.

El capítulo 2. Recopilaremos información sobre los conceptos de costos de presupuestos de obras para entender el funcionamiento, usaremos ingeniería de software para planificar el proyecto.

El capítulo 3. Marco práctico aplicaremos la metodología de ingeniería de software, mediante la interpretación y comprensión de las definiciones obtendremos procesos y resultados.

El capítulo 4. Calidad, seguridad y costos en estos capítulos mediante análisis y cálculos obtendremos los resultados, estos resultados nos indicarán si es de calidad, si es seguro y nos dará el costo del sistema.

El capítulo 5. Analizando los datos de los procesos y resultados del Capítulo 3, y del capítulo 4 tendremos la conclusión.

## **1.2 ANTECEDENTES**

En la búsqueda de otros proyectos relacionados con la construcción y con el tema de precios unitarios se encontró en el repositorio de la Universidad Mayor de San Andrés (U.M.S.A.) proyectos de grado relacionados

Sistema web para el diseño y cálculo de presupuesto de obras de arte hidráulicas, realizado por Maribel Pachacuti Blanco (2015) caso: Consultora Argos S.R.L. con el objetivo de diseñar y calcular el presupuesto de obras de arte hidráulica.

Tutor inteligente para la elaboración de un presupuesto de construcción de una vivienda, realizado por Roberto Carlos Mamani Mamani (2015) en este el cálculo de presupuestos de una vivienda con un tutor inteligente.

Sistema de cálculo de costos presupuestarios, realizado por Sarah Condori Kapa (2010) caso: honorable cámara de diputados, la realización de este proyecto es para facilitar al usuario la elaboración de cálculos para costos.

Sistema web de costos por servicios portuarios, realizado por Rai Enzo Xabier Tito Condori (2017) caso: empresa “administración de servicios portuarios –bolivia (aspb)” brindar información de costos permitiendo una mejor atención al cliente.

### **1.3 PROBLEMÁTICA**

La empresa ADU maneja los procesos de cálculos de manera escrita o en hojas de cálculo para presentar el presupuesto, pero de esta forma no puede dar toda la información, no es posible cambiar o modificar y es necesario poder realizar la modificación durante el proyecto.

Aunque la empresa cuenta con un personal experimentado es difícil proveer a tiempo los presupuestos de construcción al estar en hojas de cálculo o en papel no se pueden modificar, ya que el costo final del proyecto termina aumentando o disminuyendo provocando una variación en el presupuesto inicial.

Al realizar el presupuesto de la construcción debe estar bien desarrollado para obtener y mostrar buenos resultados tenemos que considerar el costo de inversión y los factores que van interviniendo, en los proyectos de construcción los factores van afectando al presupuesto. Por este motivo es importante mantener una actualización de las cantidades y precios en el esquema del presupuesto para realizar proyecciones y reducir el costo final.

Dentro de la empresa el personal con la experiencia para realizar los presupuestos hace el uso de herramientas de software para realizar los presupuesto de construcción, pero al ser un software muy general y no todo el personal lo usa por desconfiar su procedencia por ello no cumple las expectativas de la empresa, el software brinda información innecesaria o no brinda información de otras áreas de la construcción de manera que afecta a la toma de decisiones.

#### **1.3.1 PROBLEMA PRINCIPAL**

En la empresa el problema principal es la optimización del tiempo en los cálculos presupuestarios.

¿De qué manera el Sistema Presupuestario de Información para Proyectos de Construcción puede facilitar el cálculo de los presupuestos para la empresa multidisciplinaria ADU S.R.L.?

### **1.3.2 PROBLEMAS SECUNDARIOS**

- a. El manejo de presupuestos es un problema debido a cantidades de gastos de obra.
- b. El cálculo y elaboración del presupuesto tarda demasiado.
- c. Los costos finales no se actualizan de acuerdo a las modificaciones hechas en el presupuesto ya que se realiza un nuevo cálculo al hacer dicha modificación .
- d. Un mal cálculo del presupuesto producirá otros gastos innecesarios.
- e. Al momento de implementar un presupuesto se debe tener en cuenta lo anterior para brindar la información útil y así tomar la decisión una vez planteado el presupuesto.

## **1.4 OBJETIVO**

### **1.4.1 OBJETIVO GENERAL**

Desarrollar un Sistema Presupuestario de Información para Proyectos de Construcción para facilitar el cálculo de presupuestos de la empresa multidisciplinaria ADU S.R.L.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Recopilar los requisitos de la empresa para realizar el sistema de información web.
- Automatizar los cálculos matemáticos en los precios unitarios para disminuir el tiempo que se tarda en elaborar dichos cálculos.
- Diseñar la interfaz gráfica del sistema de información web de manera que sea comprensible para el usuario.
- Elaborar una base de datos que cumpla con los requerimientos necesarios.
- Verificar el funcionamiento y calidad del sistema de información web en la empresa constructora y asegurar la información de posibles daños.

## **1.5 JUSTIFICACIÓN**

Con el fin de ayudar a mejorar a la constructora se realiza el sistema de información web el cual será seguro, confiable y actualizado debido a que la empresa ADU maneja manualmente los proyectos de presupuestos se tiene problemas financieros.

Dificulta la preparación de los presupuesto además de la búsqueda de elementos para su modificación esto afecta el costo final y no es posible una buena toma de decisiones, la forma de trabajo actual no es eficiente se hace tedioso el crear presupuestos de manera detallada por ello surge la necesidad de crear un sistema informático web moderno brindando la posibilidad de generar los presupuestos detalladamente.

## **1.6 METODOLOGÍA**

Para el desarrollo del sistema se usará el método científico, con la finalidad de entender el problema se hará uso de la investigación mediante entrevistas y cuestionarios los cuales serán revisados. La investigación descriptiva implica la observación y descripción del comportamiento de los registros.

La metodología de software para el desarrollo del proyecto, el proceso unificado de rational (RUP) con sus 4 fases elaboración, construcción, transición y producción, realizando el desarrollo de software, se aplicarán las herramientas del lenguaje unificado el Modelo UML.

En la parte técnica se hará uso de UWE esta metodología está enfocada al desarrollo de aplicaciones web y basada en técnicas del UML, esta metodología ayuda al desarrollo, simplifica y agiliza, además de estar basada en las técnicas UML.

En cuanto a la programación se usarán herramientas de desarrollo web como NodeJS para la parte del backend para conectar la base de datos y el servidor, se usará Angular un framework para la parte del frontend la cual estará relacionada con el backend de manera que ambos trabajen juntos.

## **CAPÍTULO II: MARCO TEÓRICO**

### **2.1 ANÁLISIS DE PRECIOS UNITARIOS Y SU ESTRUCTURA**

Es un modelo matemático que adelanta el resultado, expresado en moneda, de una situación relacionada con una actividad sometida a estudio. También es una unidad dentro del concepto "Costo de Obra", ya que una Obra puede contener varios Presupuestos.

Los análisis de precios unitarios, que se calculan para cada uno de los ítems de una obra en base a las especificaciones y planos, juntamente con los correspondientes volúmenes de obra sirven para determinar el presupuesto de una obra. Los elementos que componen un Precio Unitario son:

- 1) Materiales
- 2) Mano de obra
- 3) Equipo, maquinaria y herramientas
- 4) Gastos generales y administrativos
- 5) Utilidad
- 6) Impuestos

La estructura de un análisis de precios unitarios componentes y la manera en que se organizan y calculan para la construcción de una obra pública, hace unos años atrás no estaba normada y por tanto se encontraba sujeta al formato que el proponente manejase, tal cual sucede en las contrataciones privadas. Actualmente, el Modelo de Documento Base de Contratación para la Contratación de Obras en la modalidad de Licitación Pública, aprobado por el Ministerio de Economía y Finanzas Públicas mediante Resolución Ministerial N° 262 en el marco del Decreto Supremo N° 181 de Normas Básicas del Sistema de Administración de Bienes y Servicios, normativa que es aplicable a contrataciones con financiamiento nacional por montos de Bs. 1.000.001.- adelante, establece un formato de análisis de precios unitarios como se ve en la tabla 2.1, de uso obligatorio, que permite homogeneizar la forma de presentación de las propuestas y coadyuvar en el cumplimiento de las obligaciones sociales y tributarias (Ferrufino, 1997).

Tabla 2.1 Formulario B 2

DATOS GENERALES				
Proyecto :				
Actividad :				
Cantidad :				
Unidad :				
Moneda :				
1. MATERIALES				
DESCRIPCIÓN	UNIDAD	CANTIDAD	PRECIO PRODUCTIVO	COSTO TOTAL
1				
2				
...				
N				
<b>TOTAL MATERIALES</b>				
2. MANO DE OBRA				
DESCRIPCIÓN	UNIDAD	CANTIDAD	PRECIO PRODUCTIVO	COSTO TOTAL
1				
2				
...				
N				
<b>SUBTOTAL MANO DE OBRA</b>				
CARGAS SOCIALES = (% DEL SUBTOTAL DE MANO DE OBRA) (55% al 71,18%)				
IMPUESTOS IVA MANO DE OBRA = (% DE SUMA DE SUBTOTAL DE MANO DE OBRA + CARGAS SOCIALES)				
<b>TOTAL MANO DE OBRA</b>				
3. EQUIPO, MAQUINARIA Y HERRAMIENTAS				
DESCRIPCIÓN	UNIDAD	CANTIDAD	PRECIO PRODUCTIVO	COSTO TOTAL
1				
2				
...				
N				
* HERRAMIENTAS = (% DEL TOTAL DE MANO DE OBRA)				
<b>TOTAL EQUIPO, MAQUINARIA Y HERRAMIENTAS</b>				
4. GASTOS GENERALES Y ADMINISTRATIVOS				
				<b>COSTO TOTAL</b>
* GASTOS GENERALES = % DE 1 + 2 + 3				
<b>TOTAL GASTOS GENERALES Y ADMINISTRATIVOS</b>				
5. UTILIDAD				
				<b>COSTO TOTAL</b>
* UTILIDAD = % DE 1 + 2 + 3 + 4				
<b>TOTAL UTILIDAD</b>				
6. IMPUESTOS				
				<b>COSTO TOTAL</b>
* IMPUESTOS IT = % DE 1 + 2 + 3 + 4 + 5				
<b>TOTAL IMPUESTOS</b>				
<b>TOTAL PRECIO UNITARIO (1 + 2 + 3 + 4 + 5 + 6)</b>				
<b>TOTAL PRECIO UNITARIO ADOPTADO (Con dos (2) decimales)</b>				
(*) El proponente deberá señalar los porcentajes pertinentes a cada rubro				
<b>NOTA.-</b> El Proponente declara que el presente Formulario ha sido llenado de acuerdo con las especificaciones técnicas, aplicando las leyes sociales y tributarias vigentes, y es consistente con el Formulario B-3.				

Fuente: (Ferrufino, 1997)

### 2.1.2 MATERIALES

Es el primer componente de precios unitarios, en el cual se debe tener especial cuidado en la estimación del rendimiento y del precio. En este último aspecto, que puede variar sustancialmente, se debe considerar de manera individual para



cada material, algunos o todos los siguientes factores: fecha de cotización, proveedor, marca, calidad, sistema tecnológico, disponibilidad, volumen de compra, presentación, condiciones de entrega, manipuleo, forma de pago (credito/contado), impuestos de ley, tasa de cambio, extracción/elaboración a cuenta propia, transporte entre el centro de ventas/producción y la obra, etc (Ferrufino, 1997).

### **2.1.3 MANO DE OBRA**

El costo de la mano de obra es dependiente de dos factores: rendimiento y precio pagado por la misma. Donde el rendimiento es función del grado de especialización del trabajador así como de los métodos y sistemas constructivos empleados; y el precio pagado por la misma, que es muy variable y se halla regulado por la oferta y la demanda, que puede variar también en función del grado de especialización, disponibilidad, carga horaria/turnos de trabajo, bonos de producción, alimentación, consideraciones especiales (Ferrufino, 1997).

### **2.1.4 EQUIPO, MAQUINARÍA Y HERRAMIENTA**

Se analizaron los siguientes elementos:

#### **2.1.4.1 EQUIPO Y MAQUINARIA**

El costo horario de equipo y maquinaria puede ser determinado en el mercado, por ejemplo, en el caso de que el equipo no sea propio de la empresa constructora. El costo horario del equipo y maquinaria, como se puede observar, depende de elementos como: propiedad del equipo (propio/alquilado), marca/tipo/modelo, capacidad, potencia, rendimiento (horas de uso/estado), accesorios, uso de combustible/lubricantes, traslado a obra cantidad de trabajo a ejecutarse, impuestos de ley, etc. (Ferrufino, 1997).

#### **2.1.4.2 HERRAMIENTA**

La incidencia de equipos y herramientas de la menores se obtiene de un análisis de la intervención de las mismas, sus costos y periodo de duración.

Calculando finalmente su incidencia como un porcentaje del total de la mano de obra. Se realiza a continuación un cálculo referencial para una obra de 30 (treinta) trabajadores con duración aproximada de un año. El valor calculado se tomará en cuenta como un porcentaje fijo del monto total de la mano de obra (costo directo de la mano de obra + cargas sociales + impuestos) (Ferrufino, 1997).

### **2.1.5 GASTOS GENERALES Y ADMINISTRATIVOS**

Los gastos generales incluyen aquellos gastos que siendo imputables a la obra no pueden ser asignados dentro los costos directos y también aquellos que siendo independientes se erogan exista o no trabajo para la empresa constructora (Ferrufino, 1997).

### **2.1.6 UTILIDAD**

Es el beneficio o ganancia de la empresa por la ejecución de la obra. Se asumió como referencia un porcentaje del 10% para los precios unitarios. Sin embargo, es potestad de la empresa y está en función a su expectativa de ganancia (Ferrufino, 1997).

Incidencia estimada por Utilidad = 10,00 %

### **2.1.7 IMPUESTOS**

Comprende los impuestos fijados por ley con sus respectivas alícuotas. En la estructura de precios unitarios el Impuesto a las Transacciones (IT) se aplica sobre todos los componentes y el Impuesto al Valor Agregado (IVA) se aplica solamente sobre la mano de obra conforme señala el formulario B-2 del Modelo del Documento Base de Contrataciones, es decir, se interpreta de esta situación que la aplicación del correspondiente impuesto es sólo sobre la mano de obra porque no presenta crédito fiscal y que para el resto de los componentes se entiende que son ingresados a la estructura de precios con sus precios contemplando ya los impuestos de ley (con factura) (Ferrufino, 1997).

### **2.1.7.1 IMPUESTO AL VALOR AGREGADO (IVA)**

El Impuesto al Valor Agregado (IVA) tiene una alícuota de 13% a aplicarse sobre el costo total de la mano de obra, es decir, sobre el costo de la mano de obra más sus cargas sociales (Ferrufino, 1997).

### **2.1.7.2 IMPUESTO A LAS TRANSACCIONES (IT)**

El Impuesto a las Transacciones (IT) tiene una alícuota de 3% a aplicarse sobre el monto total del contrato de la obra. El IT puede ser compensado con el importe pagado por el Impuesto sobre las Utilidades de las Empresas (IUE) en la gestión anterior, sin embargo, para fines del análisis del precio unitario se lo considerará sin compensación alguna.

## **2.2 INGENIERÍA DE SOFTWARE**

La ingeniería del software es una disciplina que implica el uso de estructuras, herramientas y técnicas para construir programas informáticos.

Así mismo, incluye el análisis previo de la situación, la redacción del proyecto, la creación del software y las pruebas necesarias para garantizar el correcto funcionamiento del software antes de poner el sistema en funcionamiento.

Esta ingeniería aborda todas las fases del ciclo de vida de desarrollo de cualquier tipo de sistema de información y es aplicable a una amplia gama de ámbitos de la informática y la ciencia de los ordenadores, como el diseño de compiladores, sistemas operativos y tecnologías de Intranet/Internet: La empresa, la investigación científica, la medicina, la fabricación, la logística, la banca, el control del tráfico y la meteorología son sólo algunos de los campos en los que puede trabajar.

Dentro de la ingeniería de software entendemos que también se encuentra todo el proceso de elaboración del software, que se denomina ciclo de vida. Está formado por cuatro etapas:

Concepción en esta primera fase se desarrolla el modelo de negocio. Es decir, conocemos las necesidades que debe de tener un software y empezamos a buscar las herramientas para cubrirlas.

Elaboración se detalla las características de la estructura del software.

Construcción tal y como su nombre indica en este paso empezamos a elaborar de forma tangible todo aquello que, de momento, solo hemos plasmado en forma de ideas.

Transición es el momento de la implementación y el desarrollo para los clientes o usuarios. Deben tener tiempo para familiarizarse con el nuevo software.

Una vez se realiza todo este ciclo, entramos en otra fase conocida como mantenimiento. Es una de las etapas más importantes ya que se solucionan los problemas o errores que puedan surgir durante su implementación y también su posterior puesta en marcha. Además, se incorporan actualizaciones teniendo en cuenta los requisitos del cliente con el objetivo de que puedan cumplir la mayor cantidad de tareas.

### **2.2.1 METODOLOGÍA ORIENTADA A OBJETOS**

Una metodología orientada a objetos es un proceso para producir software de una manera organizada, usando convenciones y técnicas de notación predefinidas. Desde que la comunidad de programación orientada a objetos tuvo la noción de incorporar el pensamiento de que los objetos son entidades coherentes con identidad estado y conducta, estos objetos pueden ser organizados por sus similitudes y sus diferencias, puestas en uso en herencia y polimorfismo, las metodologías orientadas a objetos incorporan estos conceptos para definir sus reglas, normas, procedimientos, guías y notaciones para alcanzar un producto de calidad que satisfaga las necesidades del cliente.

UML El lenguaje unificado de diagrama o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos.

UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema (MOO, 2015) .

## **2.2.2 METODOLOGÍA RUP RATIONAL UNIFIED PROCESS**

El proceso unificado es un intento por obtener los mejores rasgos y características de los modelos tradicionales del proceso del software, pero en forma que implemente muchos de los mejores principios del desarrollo ágil de software.

Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo que está enfocada hacia diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal. El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica pueda acceder a la misma base de datos incluyendo sus conocimientos. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar un software.

Hace énfasis en la importancia de la arquitectura del software y ayuda a que el arquitecto se centre en las metas correctas, tales como que sea comprensible, permite cambios futuros y la reutilización.

### **2.2.2.1 HISTORIA**

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con

Boehm en la investigación. En 1995, Rational Software compró una compañía sueca llamada Objectory AB, fundada por Ivar Jacobson, famoso por haber incorporado los casos de uso a los métodos de desarrollo orientados a objetos. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory (el proceso de la empresa Objectory AB). El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten.

En 2006, IBM creó un subconjunto de RUP ajustado para proyectos de desarrollo ágil - publicado como un método libre, llamado OpenUP a través del sitio de Eclipse.

### 2.2.2.2 FASES DEL PROCESO UNIFICADO

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

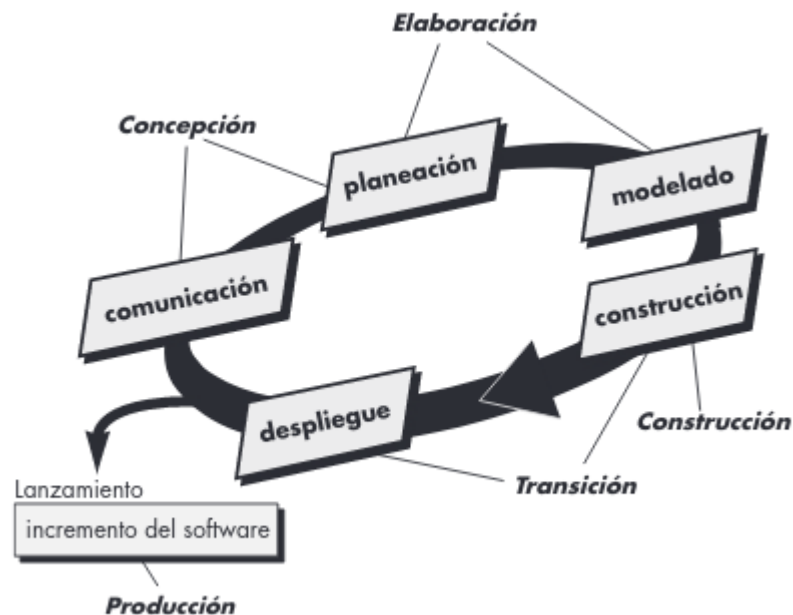


Figura 2.2 Fases del proceso unificado  
Fuente: (Pressman, 2003)

### **a) FASE DE CONCEPCIÓN**

La fase de concepción del PU agrupa actividades tanto de comunicación con el cliente como de planeación. Al colaborar con los participantes, se identifican los requerimientos del negocio, se propone una arquitectura aproximada para el sistema y se desarrolla un plan para la naturaleza iterativa e incremental del proyecto en cuestión. Los requerimientos fundamentales del negocio se describen por medio de un conjunto de casos de uso preliminares que detallan las características y funciones que desea cada clase principal de usuarios. En este punto, la arquitectura no es más que un lineamiento tentativo de subsistemas principales y la función y rasgos que tienen. La arquitectura se mejorará después y se expandirá en un conjunto de modelos que representarán distintos puntos de vista del sistema. La planeación identifica los recursos, evalúa los riesgos principales, define un programa de actividades y establece una base para las fases que se van a aplicar a medida que avanza el incremento del software.

### **b) FASE DE ELABORACIÓN**

La fase de elaboración incluye las actividades de comunicación y modelado del modelo general del proceso. La elaboración mejora y amplía los casos de uso preliminares desarrollados como parte de la fase de concepción y aumenta la representación de la arquitectura para incluir cinco puntos de vista distintos del software: los modelos del caso de uso, de requerimientos, del diseño, de la implementación y del despliegue. En ciertos casos, la elaboración crea una "línea de base de la arquitectura ejecutable" que representa un sistema ejecutable de "primer corte". La línea de base de la arquitectura demuestra la viabilidad de ésta, pero no proporciona todas las características y funciones que se requieren para usar el sistema. Además, al terminar la fase de elaboración se revisa con cuidado el plan a fin de asegurar que el alcance, riesgos y fechas de

entrega siguen siendo razonables. Es frecuente que en este momento se hagan modificaciones al plan.

### **c) FASE DE CONSTRUCCIÓN**

La fase de construcción del PU es idéntica a la actividad de construcción definida para el proceso general del software. Con el uso del modelo de arquitectura como entrada, la fase de construcción desarrolla o adquiere los componentes del software que harán que cada caso de uso sea operativo para los usuarios finales. Para lograrlo, se completan los modelos de requerimientos y diseño que se comenzaron durante la fase de elaboración, a fin de que reflejen la versión final del incremento de software. Después se implementan en código fuente todas las características y funciones necesarias para el incremento de software. A medida de que se implementan los componentes, se diseñan y efectúan pruebas unitarias para cada uno. Además, se realizan actividades de integración. Se emplean casos de uso para obtener un grupo de pruebas de aceptación que se ejecutan antes de comenzar la siguiente fase del PU.

### **d) FASE DE TRANSICIÓN**

La fase de transición del PU incluye las últimas etapas de la actividad general de construcción y la primera parte de la actividad de despliegue general (entrega y retroalimentación). Se da el software a los usuarios finales para las pruebas beta, quienes reportan tanto los defectos como los cambios necesarios. Además, el equipo de software genera la información de apoyo necesaria que se requiere para el lanzamiento. Al finalizar la fase de transición, el software incrementado se convierte en un producto utilizable que se lanza.

### **e) FASE DE PRODUCCIÓN**

La fase de producción del PU coincide con la actividad de despliegue del proceso general. Durante esta fase, se vigila el uso que



se da al software, se brinda apoyo para el ambiente de operación y se reportan defectos y solicitudes de cambio para su evaluación. Es probable que al mismo tiempo que se llevan a cabo las fases de construcción, transición y producción, comience el trabajo sobre el siguiente incremento del software. Esto significa que las cinco fases del PU no ocurren en secuencia sino que concurren en forma escalonada. El flujo de trabajo de la ingeniería de software está distribuido a través de todas las fases del PU. En el contexto de este, un flujo de trabajo es análogo al conjunto de tareas. Es decir, un flujo de trabajo identifica las tareas necesarias para completar una acción importante de la ingeniería de software y los productos de trabajo que se generan como consecuencia de la terminación exitosa de aquéllas. Debe notarse que no toda tarea identificada para el flujo de trabajo del PU es realizada en todos los proyectos de software. El equipo adapta el proceso a fin de que cumpla sus necesidades (Pressman, 2003).

### **2.3 INGENIERÍA WEB**

La ingeniería Web hace alusión a los procedimientos, tecnología y herramientas que se emplean en el desarrollo de aplicaciones Web complicadas y de gran capacidad que sirve de apoyo a la evaluación, al proyecto, crecimiento, ejecución y progreso de dichas aplicaciones.

Las aplicaciones elaboradas para la Web se caracterizan especialmente por hacer que las herramientas de ingeniería a utilizar sean diversas.

La metodología de la Ingeniería web consiste en un procedimiento evolutivo, de seis etapas que son las herramientas necesarias para transformar un modelo eficaz para el desarrollo de programas para la web.

Las necesidades actuales de desarrollo de software están altamente influenciadas por el impacto del Internet y de la World Wide Web (www). Cada vez más, las empresas e instituciones necesitan y dependen de aplicaciones basadas en la web, lo cual requiere por parte de los desarrolladores un amplio conocimiento de tecnologías de vanguardia que le permitan desarrollarlas, sin embargo, esto no es

suficiente, también se requiere del conocimiento de las metodologías y herramientas que permitan analizar, planear, modelar e implementar un sistema de calidad que responda a los requerimientos del usuario en forma, tiempo y costos y que además permita en forma sencilla su mantenimiento, crecimiento y evolución. Estas consideraciones se encuentran en la Ingeniería Web, que promueve una forma de trabajo disciplinada y sistemática y el uso de herramientas que faciliten el desarrollo, implantación y mantenimiento de aplicaciones web de alta calidad. La Ingeniería Web, es una gran área de oportunidad para la investigación y desarrollo de herramientas que permitan desarrollar aplicaciones web de alta calidad.

### **2.3.1 ETAPAS DE LA INGENIERÍA WEB**

#### **Formulación**

Consiste en identificar las metas y los objetivos del sistema, constituyendo de esta manera el motivo del progreso del sistema, su importancia y los usuarios potenciales.

#### **Planificación**

Consiste en el cálculo del costo integral del proyecto y se determinan las amenazas que se relacionan con el impulso del desarrollo además se determina un plan muy detallado para el desarrollo y progresos de la aplicación.

#### **Análisis**

El análisis es una actividad técnica que identifica los datos y requisitos funcionales y de comportamiento para la WebApp (Pressman, 2006). Durante la IWeb se realizan cuatro tipos de análisis diferentes: Modelo de contenido Web. Se trata de la identificación del espectro completo de contenido que se va a proporcionar. Análisis de la interacción. Se trata de la descripción detallada de la interacción del usuario y la WebApp. Análisis funcional. Los escenarios de utilización (casos de uso) creados como parte del análisis de interacción definen las operaciones que se aplicarán en el contenido de la WebApp e implicarán otras funciones de procesamiento. Análisis de la configuración. Se efectúa una descripción detallada del entorno y de la

infraestructura en donde reside la WebApp. La WebApp puede residir en Internet, en una intranet o en una Extranet. Además, se deberá identificar la infraestructura (es decir, la infraestructura de los componentes y el grado de utilización de la base de datos para generar el contenido) de la WebApp.

### **Ingeniería**

Esta etapa consiste en la realización de diseños tanto del tema como el de fabricación, en paralelo con los diseños arquitectónicos, navegación e interfaz. Incorpora dos tareas paralelas el diseño de contenido y la producción son tareas llevadas a cabo por personas no técnicas del equipo IWeb. El objetivo de estas tareas es diseñar, producir y/o adquirir todo el contenido de texto, gráfico y video que se vayan a integrar en la WebApp.

#### **Generación de páginas**

Esta etapa consiste en la realización de la estructura, empleando herramientas para el desarrollo de aplicaciones web. Está relacionado con el diseño arquitectónico, de navegación y de interfaz para la elaboración de web dinámica.

### **Pruebas**

Esta etapa sirve para encontrar las fallas y permite garantizar que la aplicación web funciona perfectamente en distintos campos, utilizando tácticas y tecnologías que son sugeridas para otros sistemas.

#### **Evaluación del cliente**

En este punto, se efectúan todas las modificaciones y variaciones que se encontraron en la etapa de pruebas y se incorporan al sistema para el siguiente incremento, de tal modo que se asegure la satisfacción por parte del cliente, según los requerimientos solicitados.

### **2.3.2 PROCESO DE LA INGENIERÍA WEB**

El proceso de ingeniería web tiene sus características como inmediatez y transformación y el crecimiento continuos, que posibilita que el usuario se involucre, facilitando el desarrollo de productos que se ajustan mucho a lo que éste busca y necesita.

Los pasos más resaltantes del proceso de la ingeniería web: enunciación, planificación análisis, modelización, generación de páginas, test y evaluación del cliente.

La formulación determina metas y fija la relevancia de la primera entrega. La Planificación genera la estimación del costo general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.

El Análisis determina los requerimientos e identifica el contenido. La Modelización se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario (Ingeniería Web, 2018).

### **2.3.3 METODOLOGÍA DE MODELADO UWE**

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web.

El enfoque UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de la "lengua franca" UML, sino que también utiliza XMI como formato de intercambio de modelos, MOF para metamodelo, los principios basados en modelos de el enfoque MDA, el lenguaje de transformación de modelos QVT y XML.

Las principales razones para utilizar los mecanismos de extensión de UML en lugar de técnicas de modelado propietarias son la aceptación de UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio web: un llamado perfil UML, y

amplio soporte de modelado visual mediante herramientas UML CASE existentes.

UWE utiliza notación UML "pura" y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones web, es decir, sin extensiones de ningún tipo. Para las características específicas de la Web, como los nodos y enlaces de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre aspectos de navegación, presentación, procesos comerciales y adaptación. La notación UWE se define como una extensión "ligera" de UML.

### **2.3.3.1 MODELO DE CONTENIDO**

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

### **2.3.3.2 MODELO DE NAVEGACION**

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

- Clases de navegación, que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- Links de navegación, que muestran el vínculo directo entre las clases de navegación.
- Caminos de navegación alternativos, los cuales son visualizados con el estereotipo <<menu>> ( ).

- Primitivas de acceso, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación(<<index>> o <<guided tour>> ) o para seleccionar ítems (<<query>> ).
- Clases de procesos ( ), las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- Links de procesos, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de estructura de navegación se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

### **2.3.3.3 MODELO DE PRESENTACION**

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

Los elementos básicos del modelo de presentación son:

- Clases de presentación, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación ( ) está compuesta por elementos de UI tales como, texto (<<text>> ), vínculo (<<anchor>> ), botón (<<button>> ), imagen (<<image>> ), formulario (<<form>> ), y colección de vínculos (<<anchored collection>> )
- Páginas web (<<page>>), que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- Grupo de presentación (<<presentation group>>), el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación

### **2.3.3.4 MODELO DE PROCESO**

El modelo de proceso o tareas integra los procesos de negocios al modelo de UWE, especificando los comportamientos de cada proceso y de las interfaces que permiten manejar a cada uno de ellos. Representa la parte dinámica de la aplicación Web, especificando la funcionalidad de las transiciones y de los flujos de trabajo complejos de las actividades, contrario al modelo de navegación, que representa la parte estática de la información.

### **2.3.3.5 FASES DE UWE**

#### **Captura, análisis y especificación de requisitos**

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

#### **Diseño del sistema**

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- Diagrama de Casos de Usos
- Diagrama Conceptual
- Diagrama Físico
- Diagrama de Clases
- Modelo Navegacional
- Modelo de Presentación

#### **Codificación del software**

Durante esta etapa se realizan las tareas que se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

### **Pruebas**

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

### **La Instalación o Fase de Implementación**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Implementación y Lanzamiento: En la implementación de la Página Web es recomendable utilizar estándares (HTML, XHTML...) para asegurar la futura compatibilidad y escalabilidad del sitio. Una vez implementada la página web y aprobada su funcionalidad se procede al lanzamiento del sitio.

### **El Mantenimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

Mantenimiento y Seguimiento: Una vez puesta la Página Web a disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio.

## **2.4 EL LENGUAJE UNIFICADO PARA LA CONSTRUCCIÓN DE MODELOS (UML)**

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce



como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

#### **2.4.1 DIAGRAMA DE CASOS DE USO**

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

#### **2.4.2 DIAGRAMA DE ACTIVIDADES**

Un diagrama de actividades ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

#### **2.4.3 DIAGRAMA DE SECUENCIAS**

Los diagramas de clases y los de objetos representan información estática. No obstante, en un sistema funcional, los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.

#### **2.4.4 DIAGRAMA DE CLASES**

Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos

(propiedades) y acciones similares. Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre sí (UML, 2014).

## **CAPÍTULO III: MARCO APLICATIVO**

### **3.1 INTRODUCCIÓN**

En el presente capítulo se realizará el análisis y diseño del “Sistema presupuestario de información para proyectos de construcción” caso Empresa Multidisciplinaria ADU S.R.L. Para el sistema ha desarrollar se usará la metodología de desarrollo RUP

### **3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL**

La empresa constructora ADU S.R.L. realiza los presupuestos de proyectos además de supervisar, hacer seguimientos y evaluar los proyectos. En la evaluación de proyectos se hace la parte presupuestaria.

Los presupuestos de proyectos manejan cálculos de costos, mediante los precios unitarios. La búsqueda de materiales, la generación del presupuesto, y la actualización de costos se hace mediante excel.

Es necesario que los costos de presupuestos se manejen de manera confiable, el resultado además debe ser exacto y se deben actualizar los cambios en costos. En la elaboración del presupuesto se cometen errores al momento de buscar los materiales y al momento de actualizar los costos.

### **3.3 PROCESO UNIFICADO RATIONAL RUP**

En esta sección procederemos a realizar la metodología proceso unificado rational la cual consta de las siguientes fases a continuación.

#### **3.3.1 FASE 1: Concepción**

En esta fase recopilaremos la información acerca de los procesos mediante casos de uso como se ve en la figura 3.1 se muestra el caso de uso del negocio.

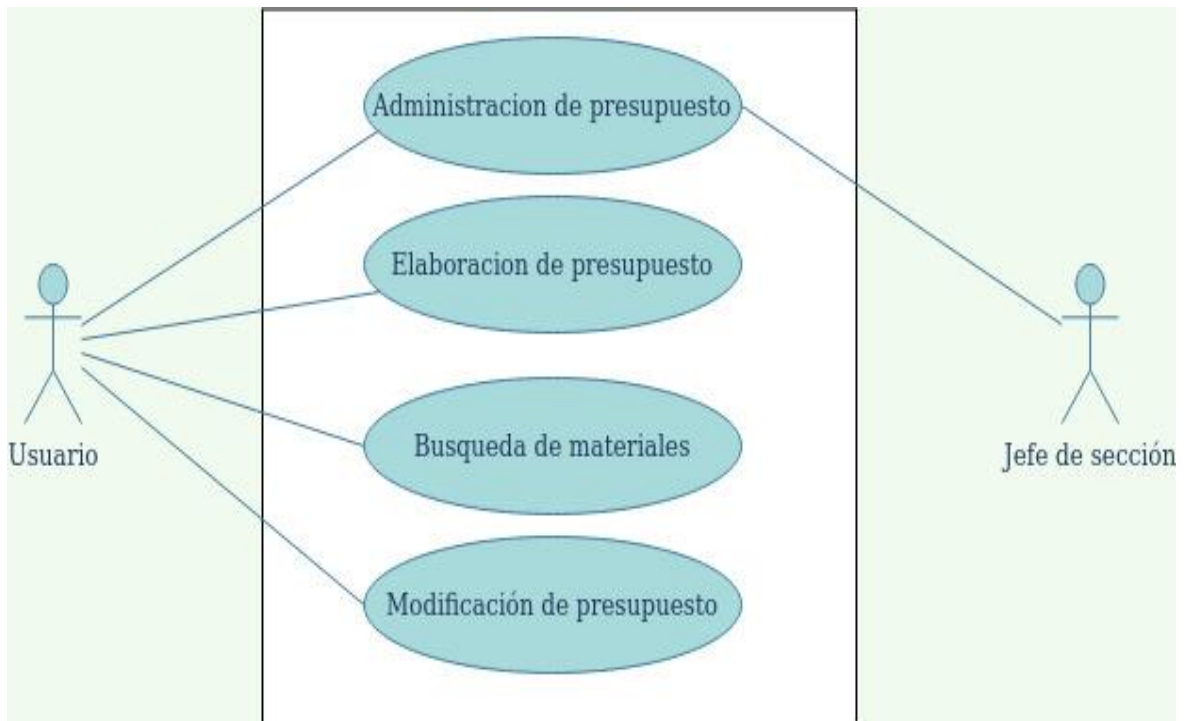


Figura 3.1 Diagrama caso de uso del negocio  
Fuente: Elaboración propia

### 3.3.1.1 Actores

Jefe de sección

Es el encargado del sistema en general, esta persona asigna roles, actualiza los perfiles de ser necesario además de modificar los presupuestos.

Usuario

Es la persona que se encarga de realizar los presupuestos puede crear, actualizar, eliminar y ver la información que genera

### 3.3.1.2 Requisitos funcionales

- Registro de usuario, se debe poder registrar usuarios y almacenarlos en la base de datos.
- Administración de usuarios, la aplicación permite la asignación de roles y controlar el acceso.
- Realizar modificación de usuario, la aplicación debe permitir dependiendo del rol modificar los datos de un usuario.

- Actualizar los precios de materiales, la aplicación tiene que tener la opción de modificar los costos.
- Permitir el inicio de sesión, la aplicación web debe permitir al usuario iniciar sesión al sistema presupuestario.
- Control de los componentes del APU, la aplicación debe controlar los componentes de manera organizada.
- Generación de presupuestos, se debe generar un presupuesto con los detalles necesarios.
- Actualización de registros, permite la actualización de las modificación del presupuesto.

### **3.3.1.3 Requerimientos no funcionales**

- Procesos rápidos, dependerá del equipo y la conexión.
- Usabilidad, será usado por personas que tengan y no tengan conocimientos de computación.
- Rendimiento, el sistema debe responder de manera inmediata a las peticiones del usuario.
- Seguridad, el sistema debe controlar el acceso solo a usuarios registrados y aprobados.

### **3.3.2 FASE 2: Elaboración**

En esta fase implementaremos casos de uso preliminares con su respectiva historia de usuario, elaboramos además los diagramas de actividades, diagrama de secuencias, diagrama de clases.

#### **3.3.2.1 Casos de uso**

Analizando los requerimientos funcionales del sistema los expresaremos en casos de uso.

## Administración de usuario

Es la gestión de usuarios al crear una cuenta el administrador autoriza la solicitud y le agrega un rol además de mostrar a los demás usuarios puede actualizar a los usuarios y eliminarlos como se ve en la figura 3.2.

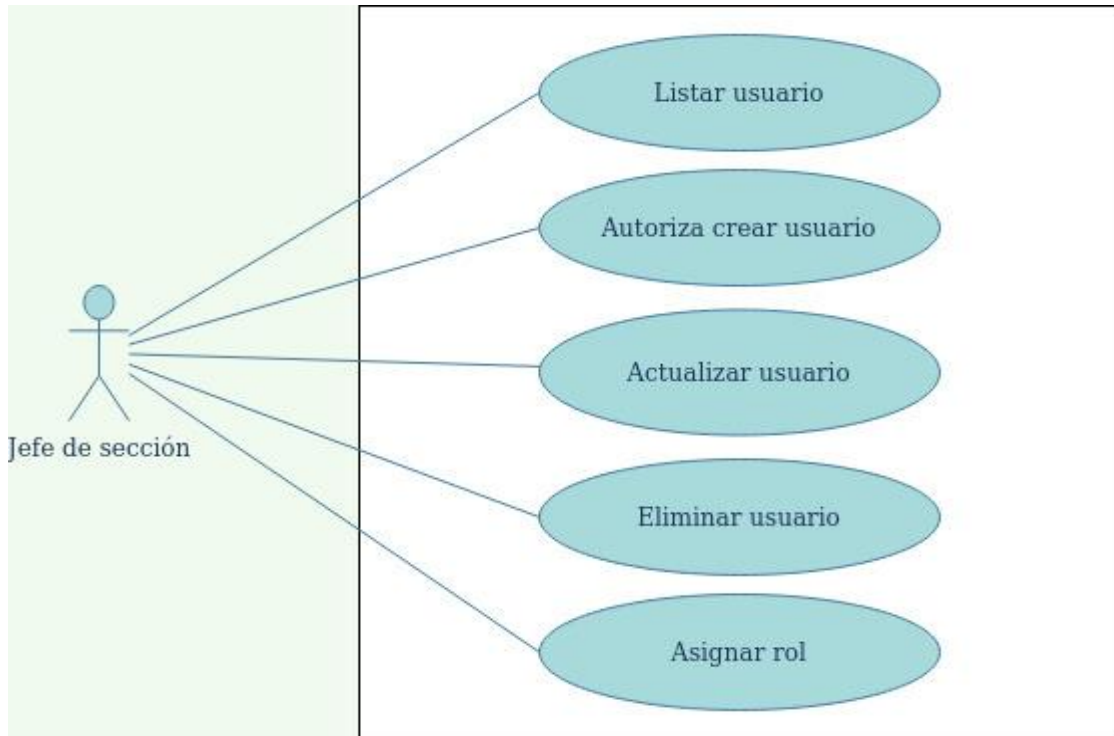


Figura 3.2 Diagrama caso de uso: Administración de usuario  
Fuente: Elaboración propia

En la tabla 3.1 describimos la historia de usuario.

Tabla 3.1 Historia de usuario: Administración de usuario

<b>Caso de uso</b>	Administración de usuario
<b>Actor principal</b>	Jefe de sección, Usuario
<b>Objetivo en contexto</b>	Preparar el sistema para crear, actualizar, eliminar, mostrar usuario
<b>Precondición</b>	El usuario o Jefe de sección ingresa al sistema
<b>Disparador</b>	El Jefe de sección autoriza o niega la creación de usuario
<b>Escenario</b>	
1. El usuario inicia sesión	
2. Se verifica el usuario	
3. El jefe de sección lista los usuarios	
4. El jefe de sección autoriza la creación de usuario	
5. El jefe de sección elimina a los usuarios	
6. El jefe de sección actualizar a los usuarios	
7. El jefe de sección asigna roles	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	Varias al día
<b>Canal Para el actor</b>	Mediante el panel de control

Fuente: Elaboración propia

### Administración del proyecto

La administración del proyecto es la gestión de crear, actualizar, listar y eliminar. El usuario puede crear el proyecto y el jefe de sección puede listar.

La interacción del administrador y del usuario se ve en la figura 3.3.

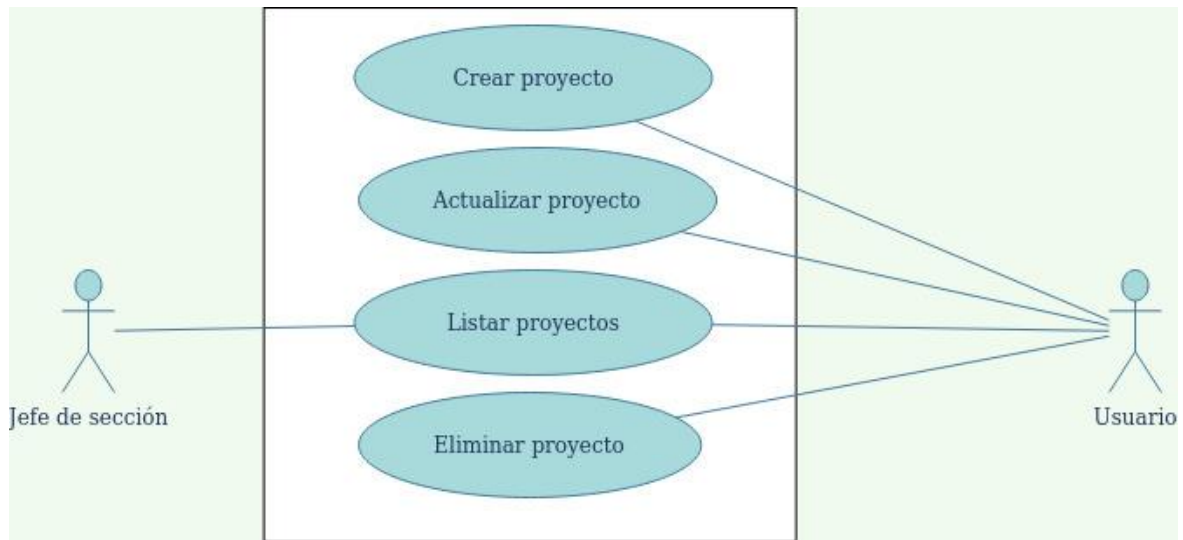


Figura 3.3 Diagrama caso de uso : Administración del proyecto  
Fuente: Elaboración propia

En la tabla 3.2 describimos la historia de usuario

Tabla 3.2 Historia de usuario: Administración del proyecto

<b>Caso de uso</b>	Administración del proyecto
<b>Actor principal</b>	Jefe de sección, Usuario
<b>Objetivo en contexto</b>	Elaboración del proyecto
<b>Precondición</b>	El usuario o Jefe de sección ingresa al sistema
<b>Escenario</b>	
1. El usuario o Jefe de sección inicia sesión	
2. Se verifica al usuario	
3. El usuario ingresa a proyectos	
4. El usuario crea el proyecto	
5. Se muestra los proyecto y la información de cada proyecto	
6. El usuario puede actualizar los proyectos	
7. El usuario puede eliminar los proyectos	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	Varias al día
<b>Canal Para el actor</b>	Mediante el panel de control

Fuente: Elaboración propia



## Administración de la planilla de precios

El usuario puede administrar precios unitarios como se ve en la figura 3.4.

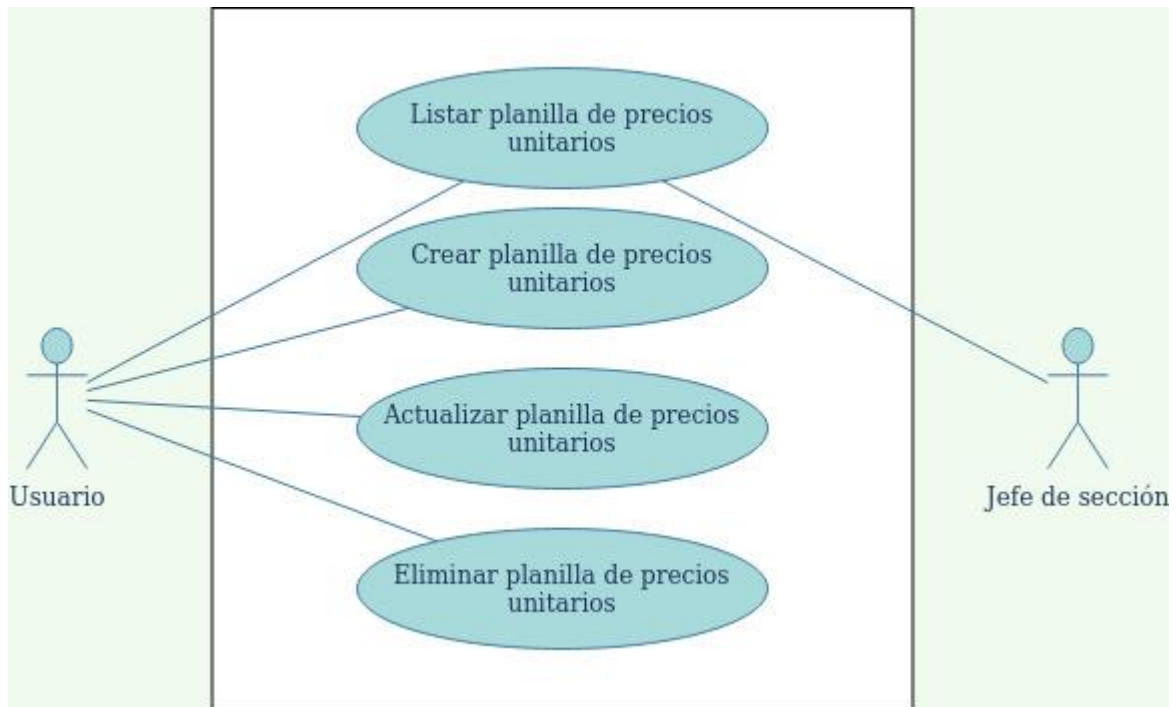


Figura 3.4 Diagrama caso de uso : Administración de la planilla de precios unitarios  
Fuente: Elaboración propia

En la tabla 3.3 describimos la historia de usuario.

Tabla 3.3 Historia de usuario: Administración de la planilla de precios unitarios

<b>Caso de uso</b>	Administración de la planilla de precios unitarios
<b>Actor principal</b>	Usuario
<b>Objetivo en contexto</b>	Preparar el sistema para crear, actualizar, eliminar, mostrar
<b>Precondición</b>	El usuario o Jefe de sección ingresa al sistema
<b>Escenario</b>	
	1. El usuario inicia sesión
	2. Se verifica el usuario
	3. El usuario lista la planilla de los precios unitarios
	4. El usuario crea una planilla de precio unitario
	5. El usuario actualizar la planilla de precio unitario

6. el usuario elimina la planilla de precio unitario	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	Varias al dia
<b>Canal Para el actor</b>	Mediante el panel de control

Fuente: Elaboración propia

### Cálculo del presupuesto

Realiza el cálculo mediante fórmulas una vez se agreguen los items necesarios mostrara los calculos por su totalidad y de manera independiente como se ve en la figura 3.5.

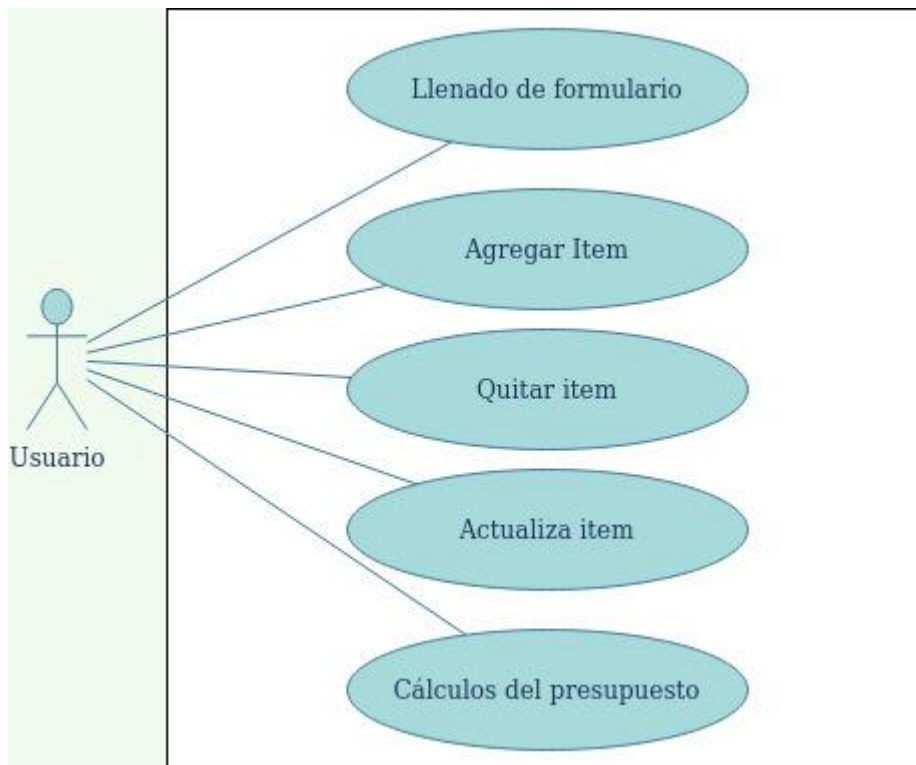


Figura 3.5 Diagrama caso de uso : Cálculo del presupuesto  
Fuente: Elaboración propia

En la tabla 3.4 se describe la historia de usuario.

Tabla 3.4 Historia de usuario: Cálculo del presupuesto

<b>Caso de uso</b>	Cálculo del presupuesto
<b>Actor principal</b>	Jefe de sección, Usuario
<b>Objetivo en contexto</b>	Preparar el sistema para el cálculo del presupuesto

<b>Precondición</b>	El usuario o Jefe de sección ingresa al sistema, agregar precios unitarios
<b>Escenario</b>	
1. El usuario inicia sesión	
2. Se verifica el usuario	
3. Se llena el formulario	
4. Se agrega el ítem de precio unitario de acuerdo al pedido	
5. Se quita el ítem de precio unitario	
6. Se actualiza el ítem de precio unitario	
7. Se realiza el cálculo y se obtiene el costo	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	Varias al día
<b>Canal Para el actor</b>	Mediante el panel de control

Fuente: Elaboración propia

### Reporte del presupuesto

Antes de la generación del presupuesto el jefe de sección debe verificar además puede modificar y pedir este reporte, el usuario genera el reporte como se ve en la figura 3.6.

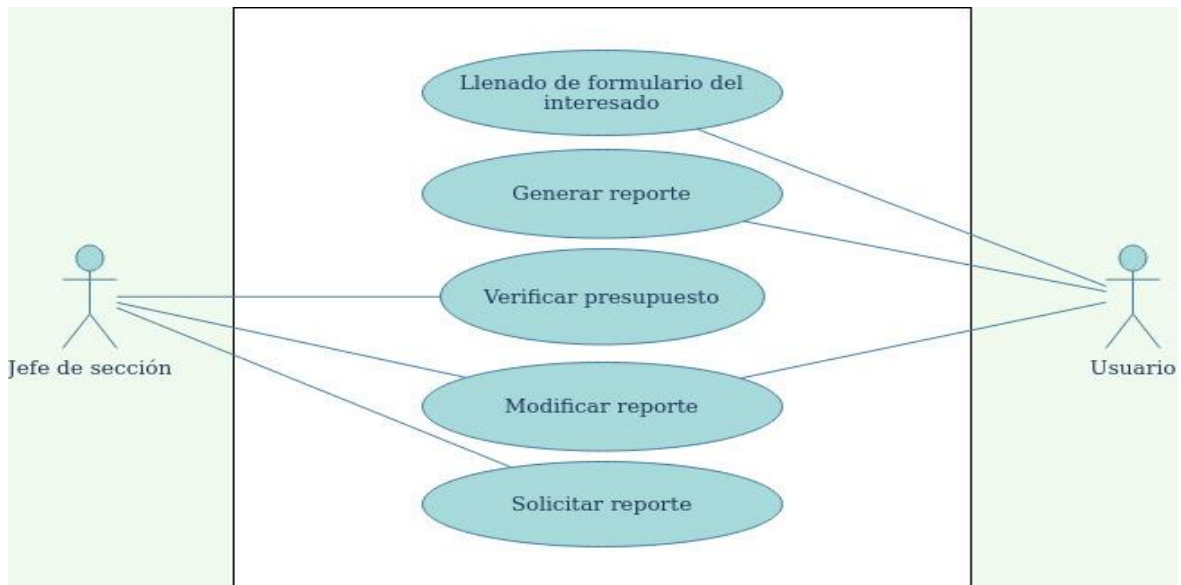


Figura 3.6 Diagrama caso de uso : Reporte del presupuesto  
Fuente: Elaboración propia

En la tabla 3.5 se describe la historia de usuario.

Tabla 3.5 Historia de usuario: Reporte del presupuesto

<b>Caso de uso</b>	Reporte del presupuesto
<b>Actor principal</b>	Jefe de sección, Usuario
<b>Objetivo en contexto</b>	Preparar el sistema para generar el reporte
<b>Precondición</b>	El usuario o Jefe de sección ingresa al sistema y se debe tener los cálculos del presupuesto
<b>Escenario</b>	
1. El usuario inicia sesión	
2. Se verifica el usuario	
3. El usuario ingresa a generación de reporte	
4. El usuario llena el formulario con datos del interesado	
5. Se genera el reporte	
6. Se verifica presupuesto	
7. Se puede modificar los ítems de precios unitarios	
8. Solicitar reporte	
<b>Prioridad</b>	Alta
<b>Frecuencia de uso</b>	Varias al día
<b>Canal Para el actor</b>	Mediante el panel de control

Fuente: Elaboración propia

### 3.3.2.2 Diagrama de actividades

#### Administrar usuarios

Vemos la interacción del Jefe de sección con el sistema como se ve en la figura 3.7.

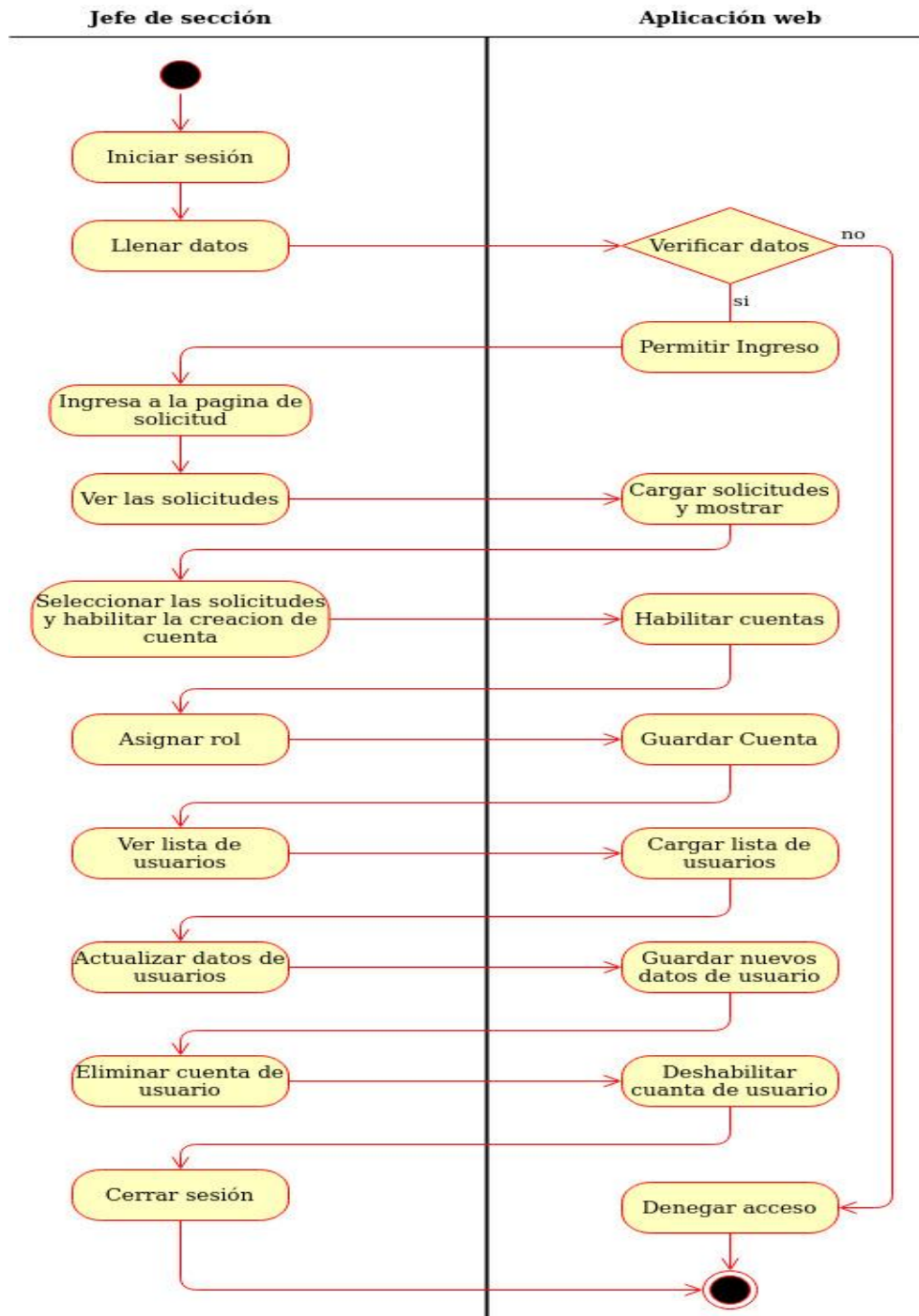


Figura 3.7 Diagrama actividad: Administrar usuario  
Fuente: Elaboración propia

## Administrar proyecto

Vemos al usuario y al jefe de sección interactuar con el sistema el jefe de sección podrá listar los proyectos mientras el usuario está más dedicado a la creación, modificación y eliminación de los proyectos mostrados en la figura 3.8.

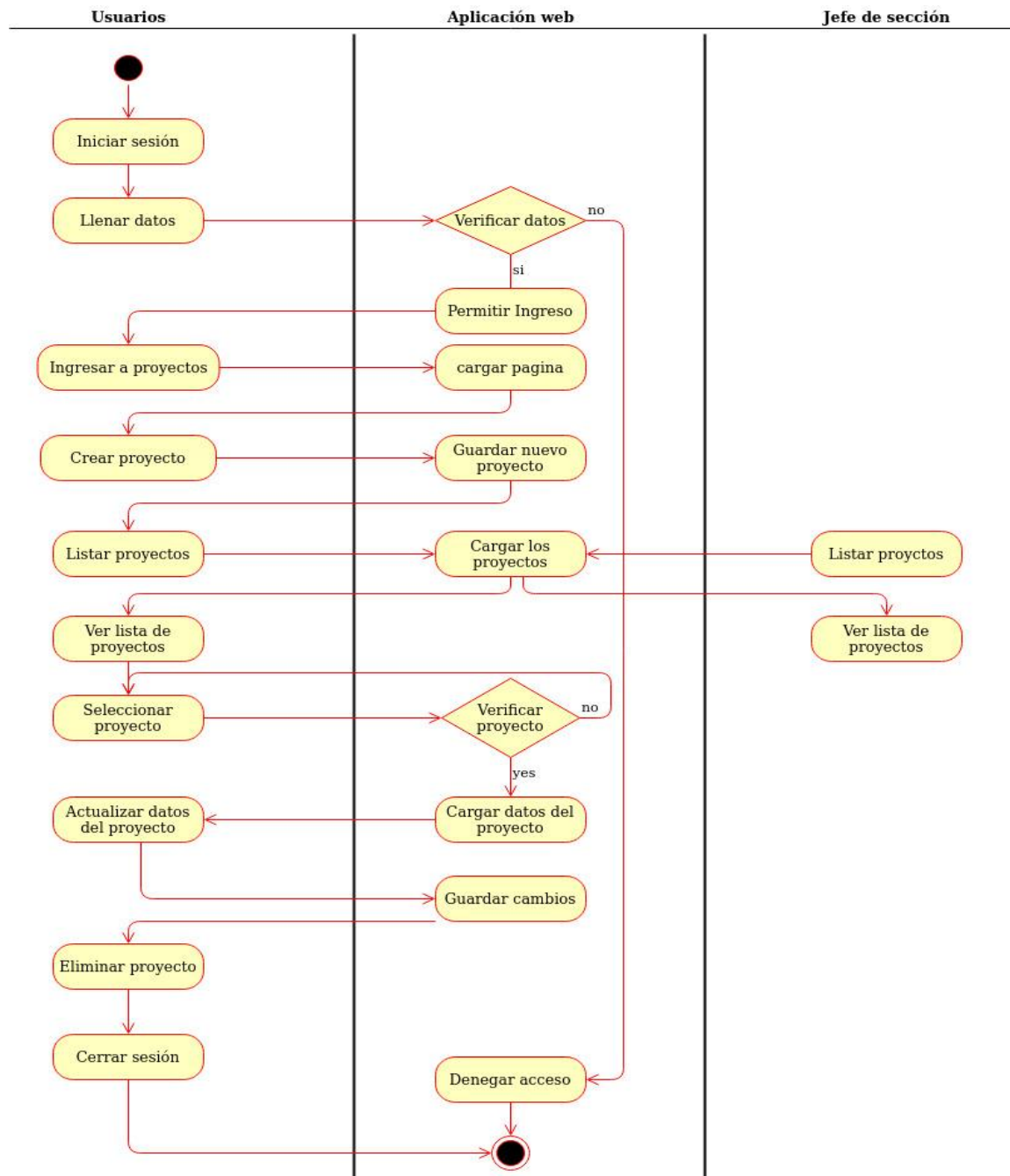


Figura 3.8 Diagrama actividad: Administrar proyecto  
Fuente: Elaboración propia

## Planilla de precios

Los actores principales interactúan con el sistema y el administrador sólo podrá listar y ver las planillas de precios como se ve en la figura 3.9.

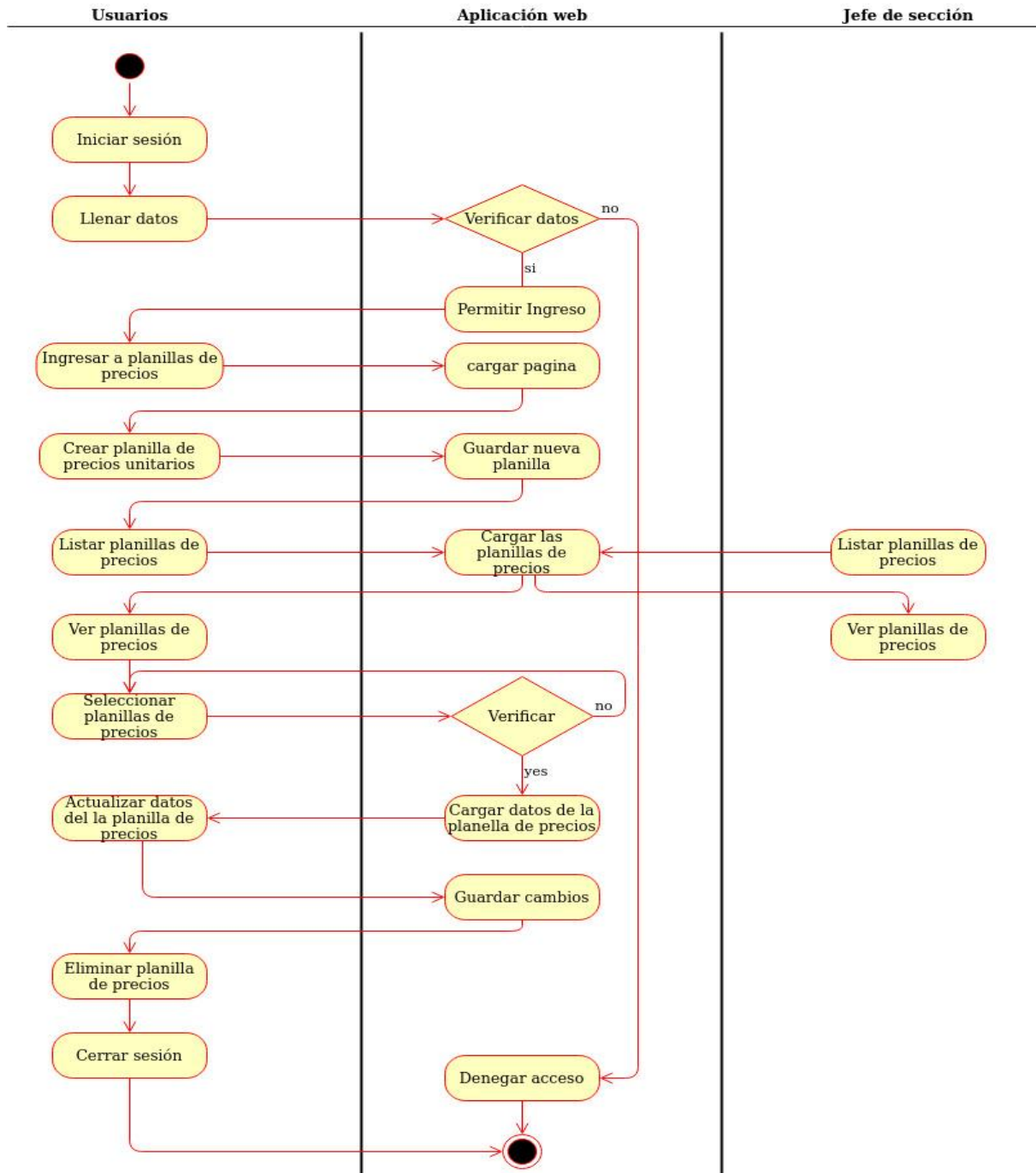


Figura 3.9 Diagrama actividad: Planilla de precios  
Fuente: Elaboración propia

## Cálculo de presupuestos

Para el cálculo del presupuesto el jefe de sección es el encargado de interactuar con el sistema como se ve en la figura 3.10.

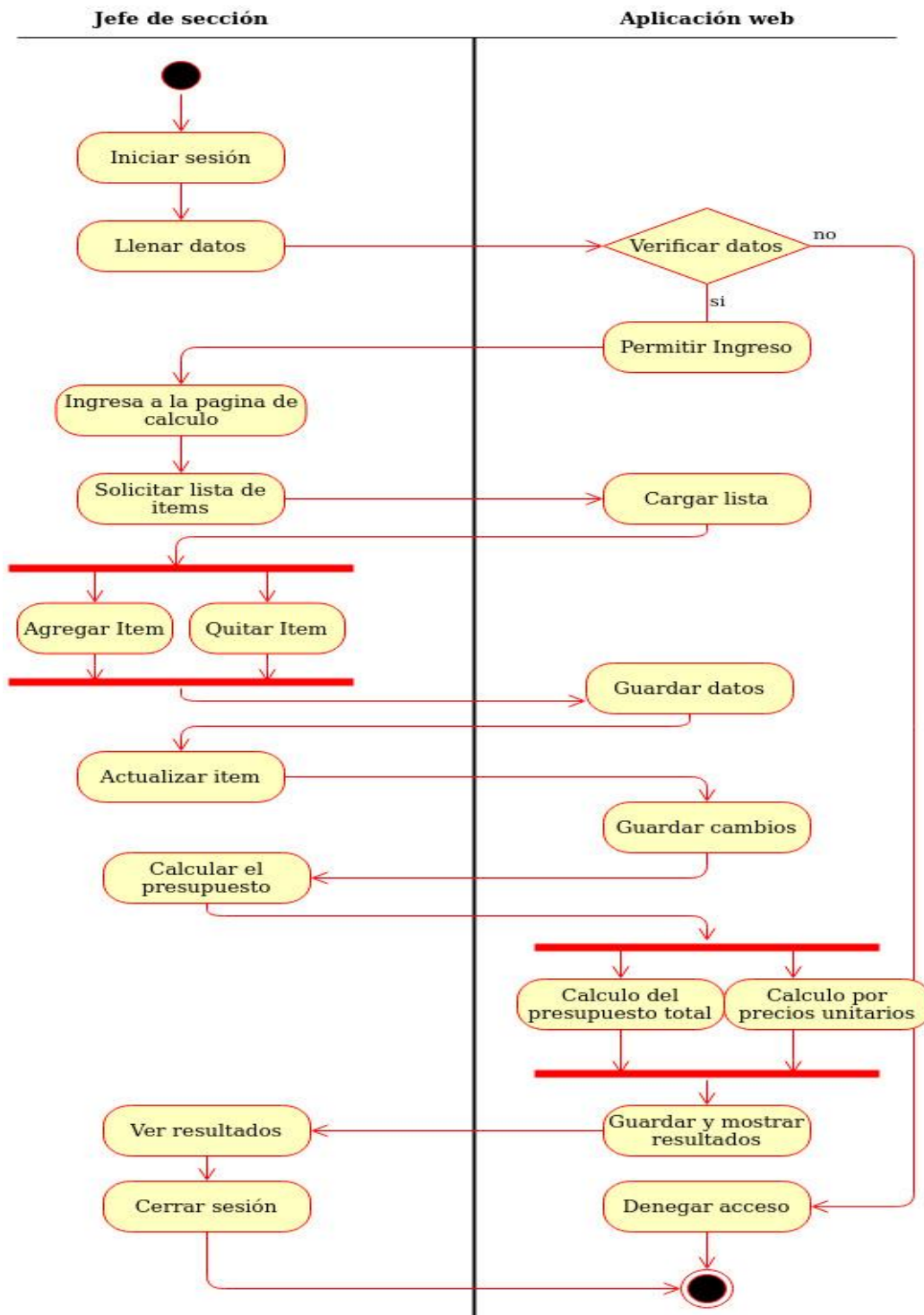


Figura 3.10 Diagrama actividad: Cálculo de presupuesto  
Fuente: Elaboración propia



## Reporte presupuestos

El reporte del presupuesto interactúa el usuario y el jefe de sección siendo aquí responsabilidad del jefe de sección verificar y ver los reportes como se muestra en la figura 3.11

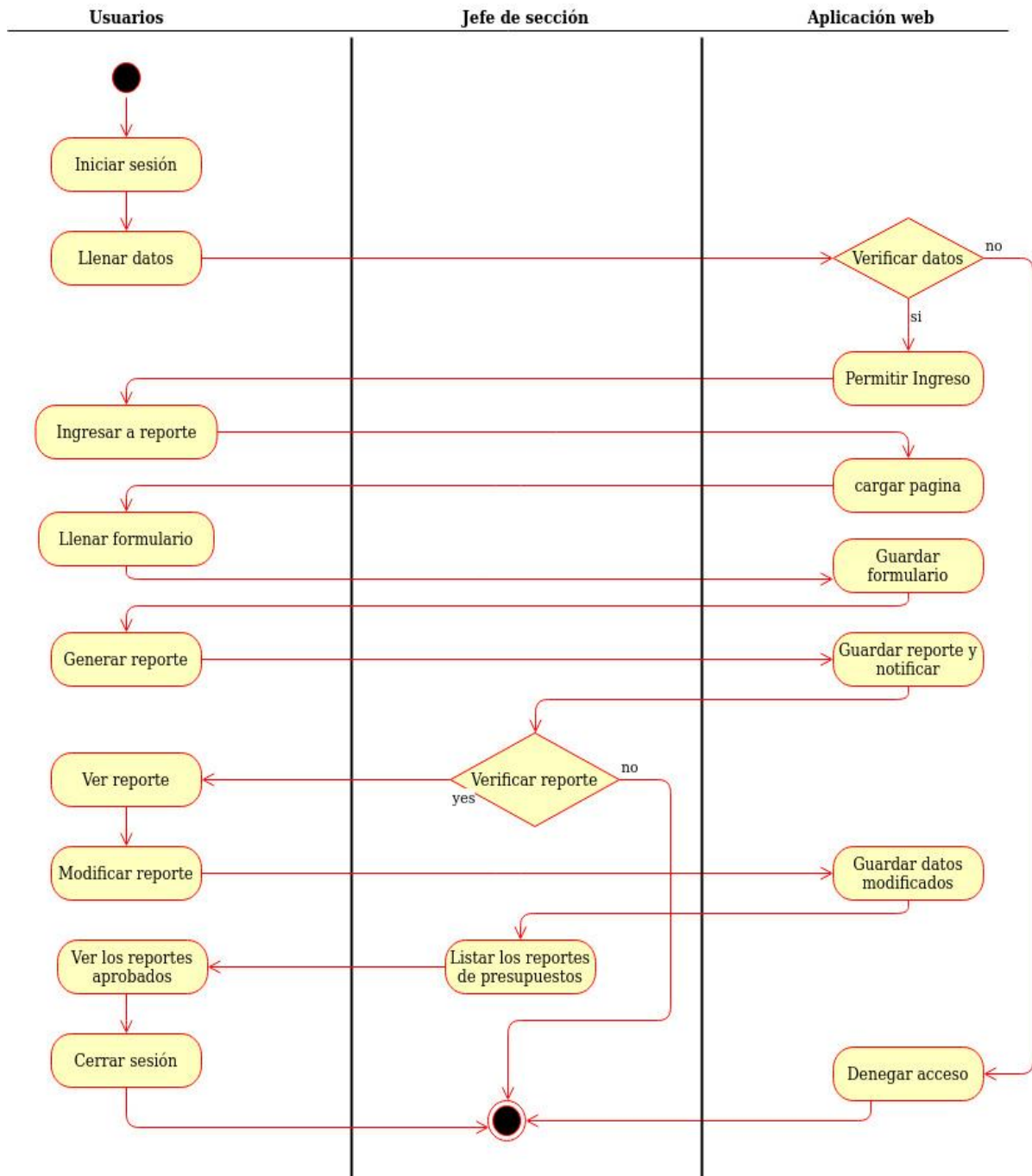


Figura 3.11 Diagrama actividad: Reporte de presupuesto  
Fuente: Elaboración propia

### 3.3.2.3 Diagrama de secuencias

#### Administrar usuarios

Vemos como el jefe de sección interactúa con el sistema con los diferentes procesos para realizar acciones como se ve en la figura 3.12

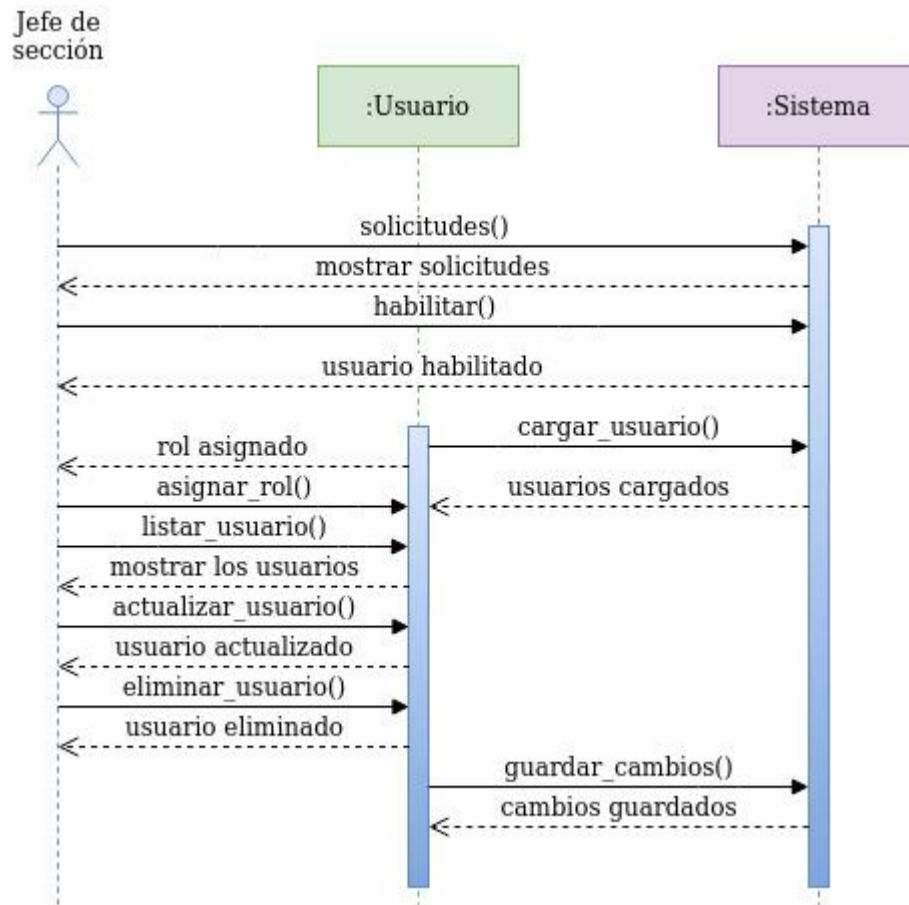


Figura 3.12 Diagrama de secuencias: Administrar usuario  
Fuente: Elaboración propia

#### Administrar proyecto

Vemos las acciones que el usuario y el jefe de sección deben tomar para administrar el proyecto este proceso se describe en la figura 3.13.

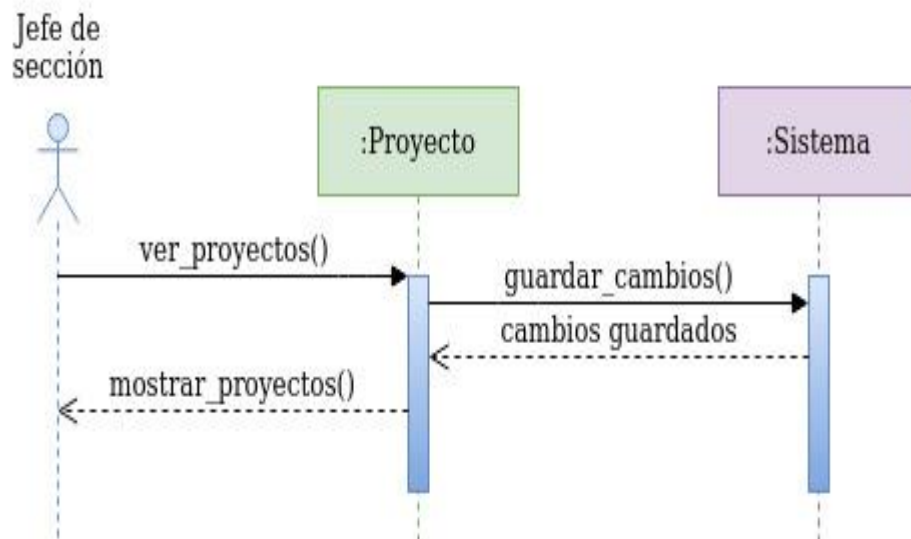
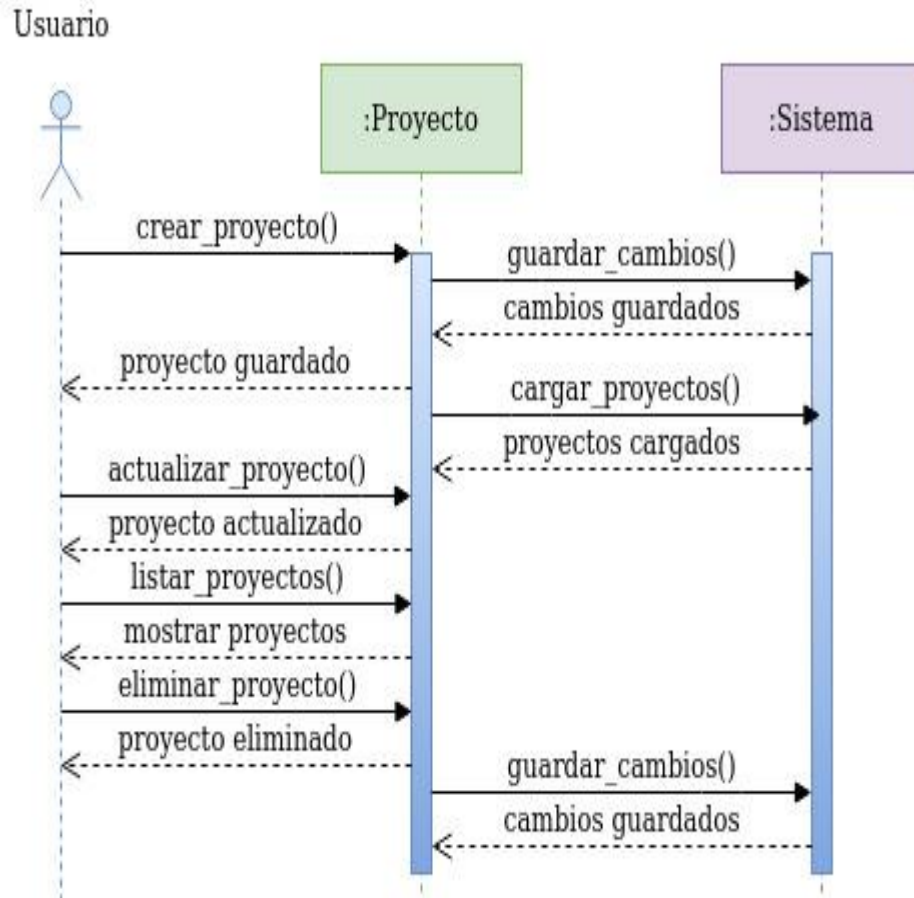


Figura 3.13 Diagrama de secuencias: Administrar proyecto  
Fuente: Elaboración propia

## Planilla de precios

Se describe los procesos de creacion, modificacion y eliminacion de planillas mediante el usuario y el sistema como se muestra en la figura 3.14

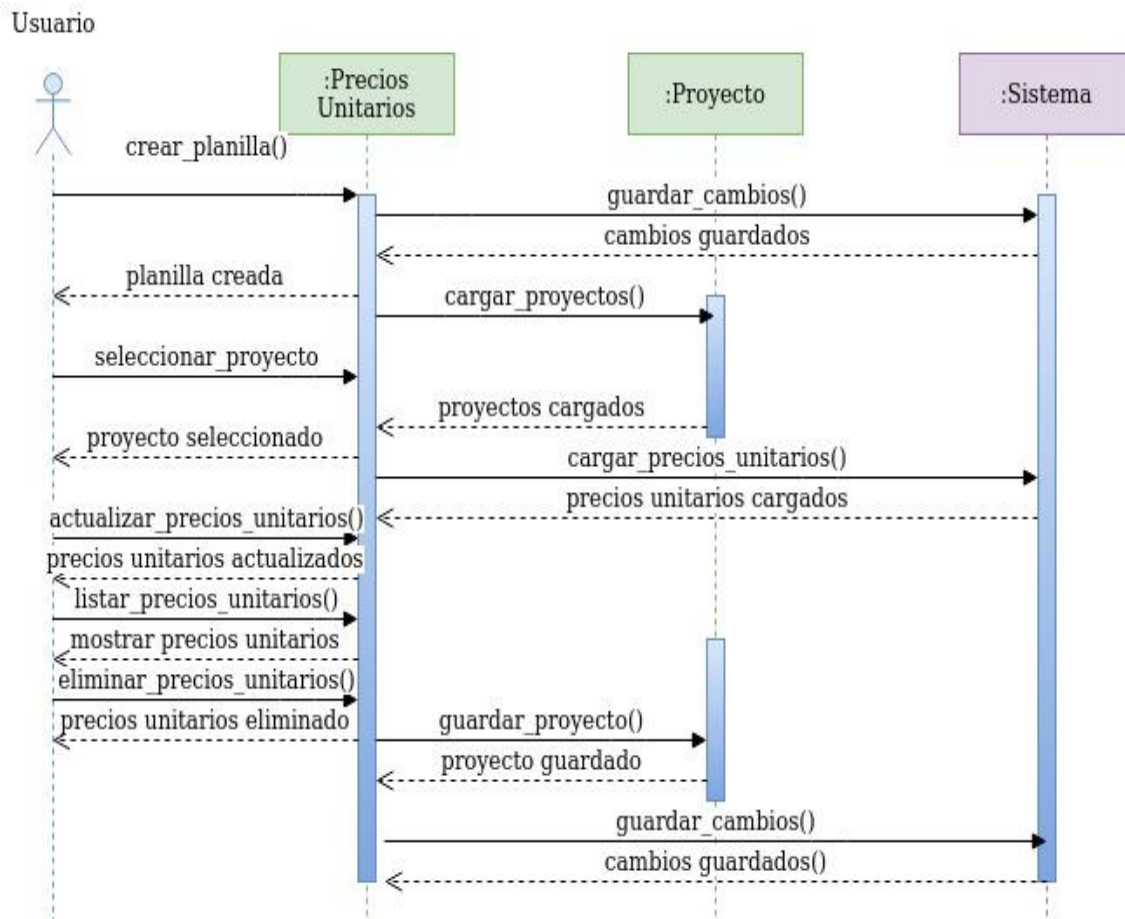


Figura 3.14 Diagrama de secuencias: Planilla de precios  
Fuente: Elaboración propia

## Cálculo de presupuestos

El usuario interactúa con gran parte del sistema y en la figura 3.5 se describe.

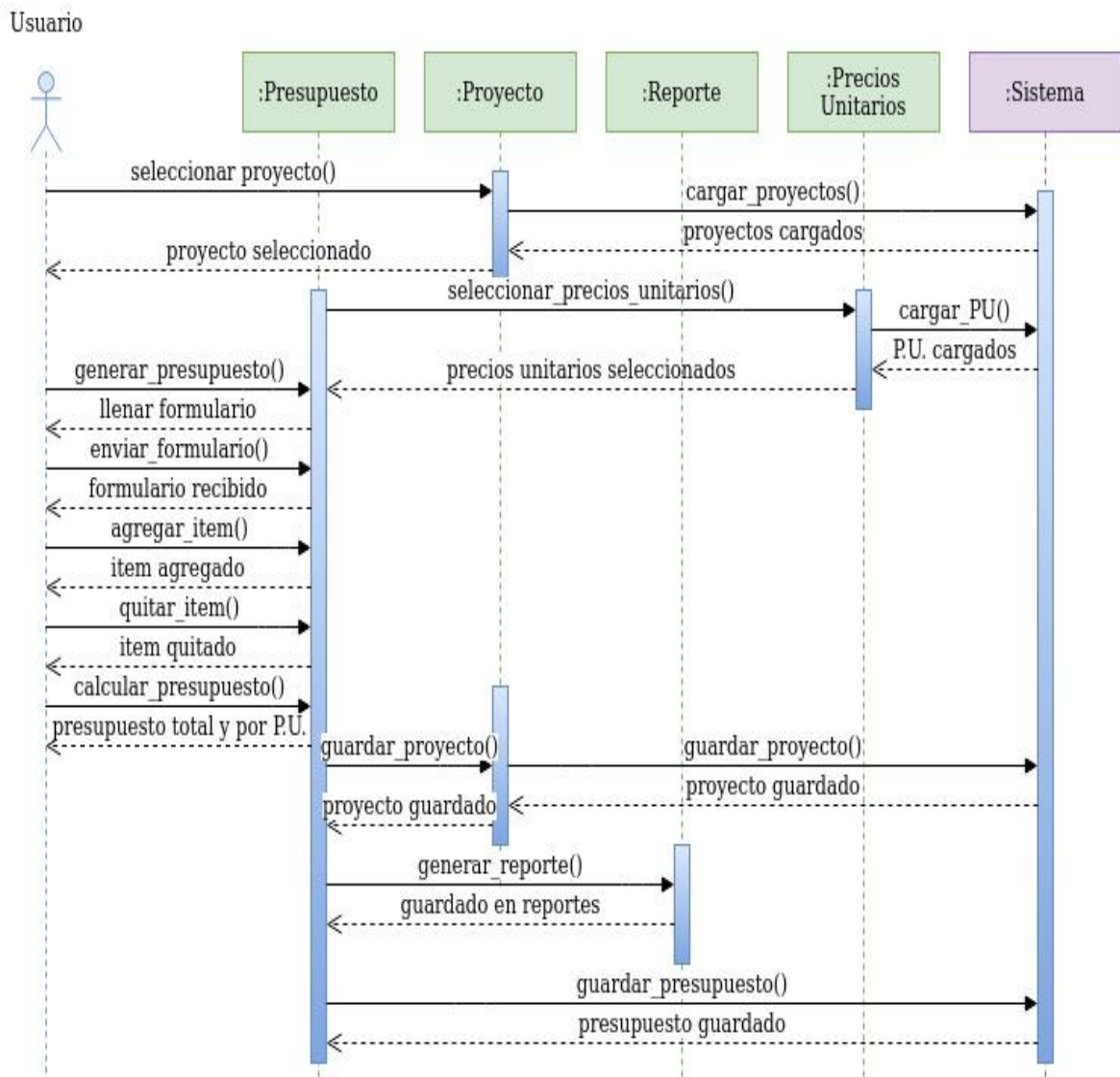


Figura 3.15 Diagrama de secuencias: Cálculo de presupuesto  
Fuente: Elaboración propia

### Reporte presupuestos

Los reportes generados por el usuario deben ser visualizados por el jefe de sección en la figura 3.16 describimos la interacción de ambos.

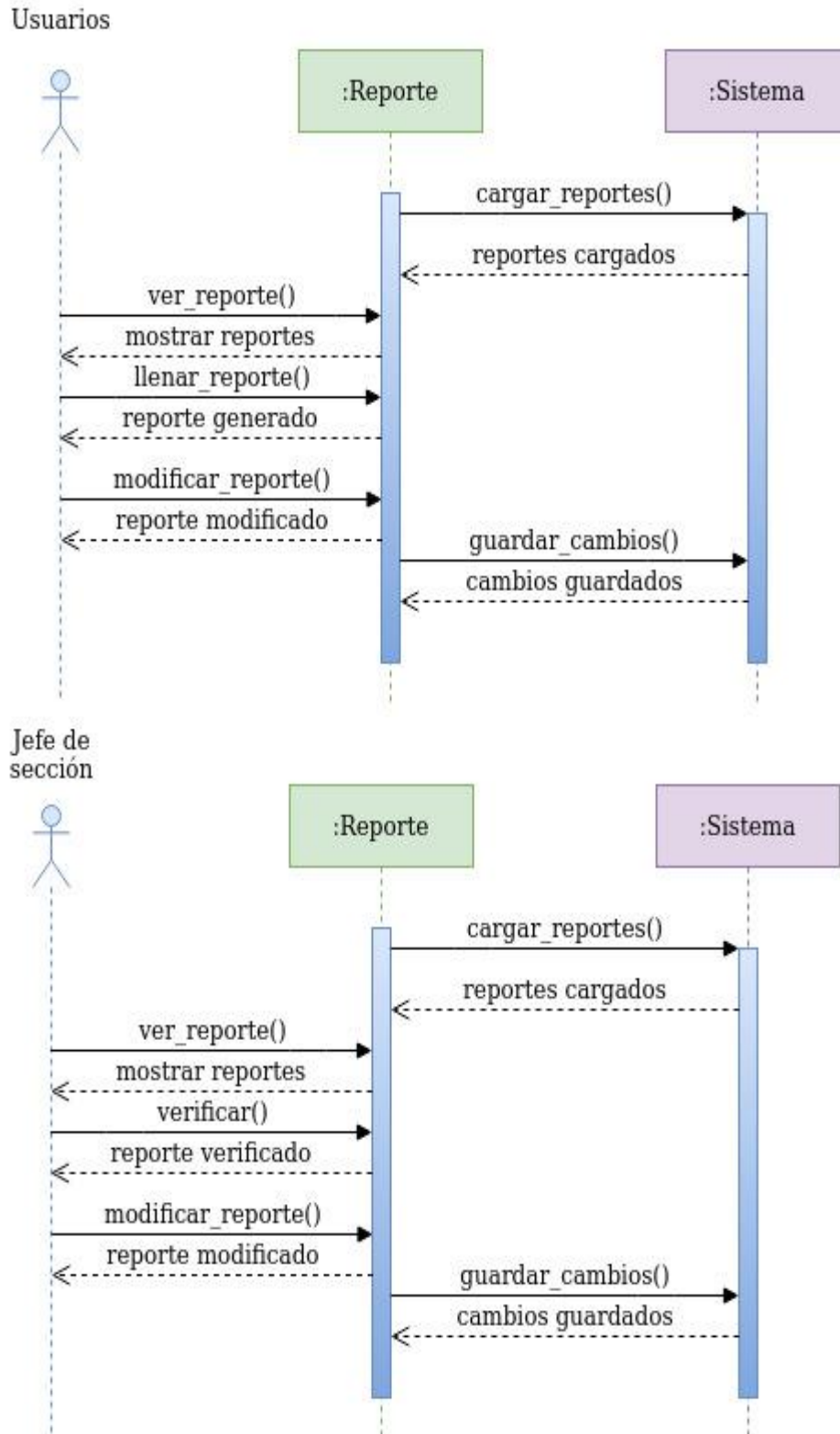


Figura 3.16 Diagrama de secuencias: Reporte de presupuesto  
 Fuente: Elaboración propia

### 3.3.2.1 Diagrama de clases

De acuerdo a los datos obtenidos para el sistema proponemos el siguiente diagrama de clases como se ve en la figura 3.17.

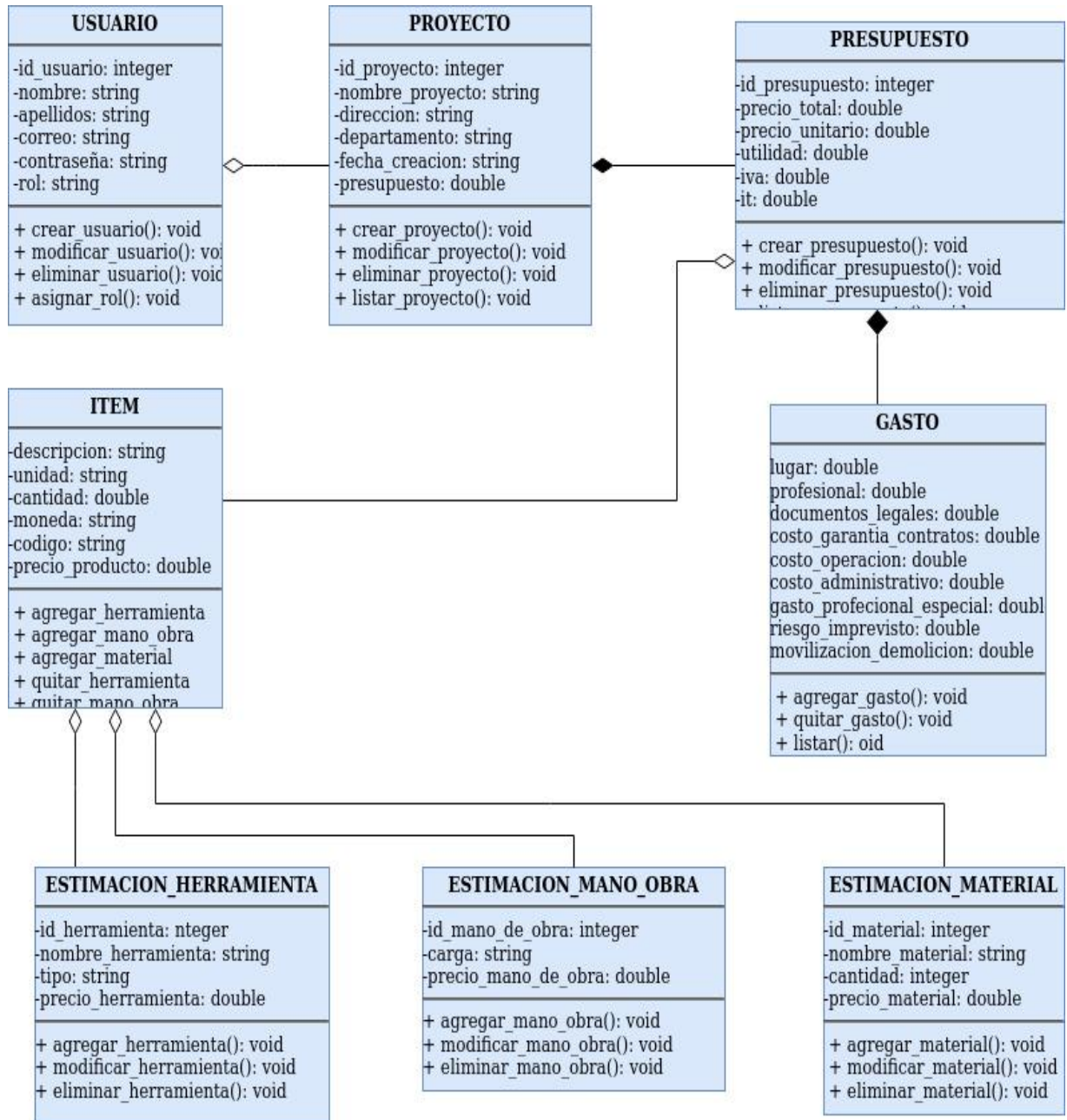


Figura 3.17 Diagrama de clases  
Fuente: Elaboración propia

### 3.3.3 FASE 3: Construcción

En esta fase lo más importante es comprender los requerimientos y así desarrollar el sistema.

#### 3.3.3.1 Diagrama entidad relación

Para la base de datos debemos proponer una entidad relación así obtenemos en la figura 3.18 el siguiente diagrama.

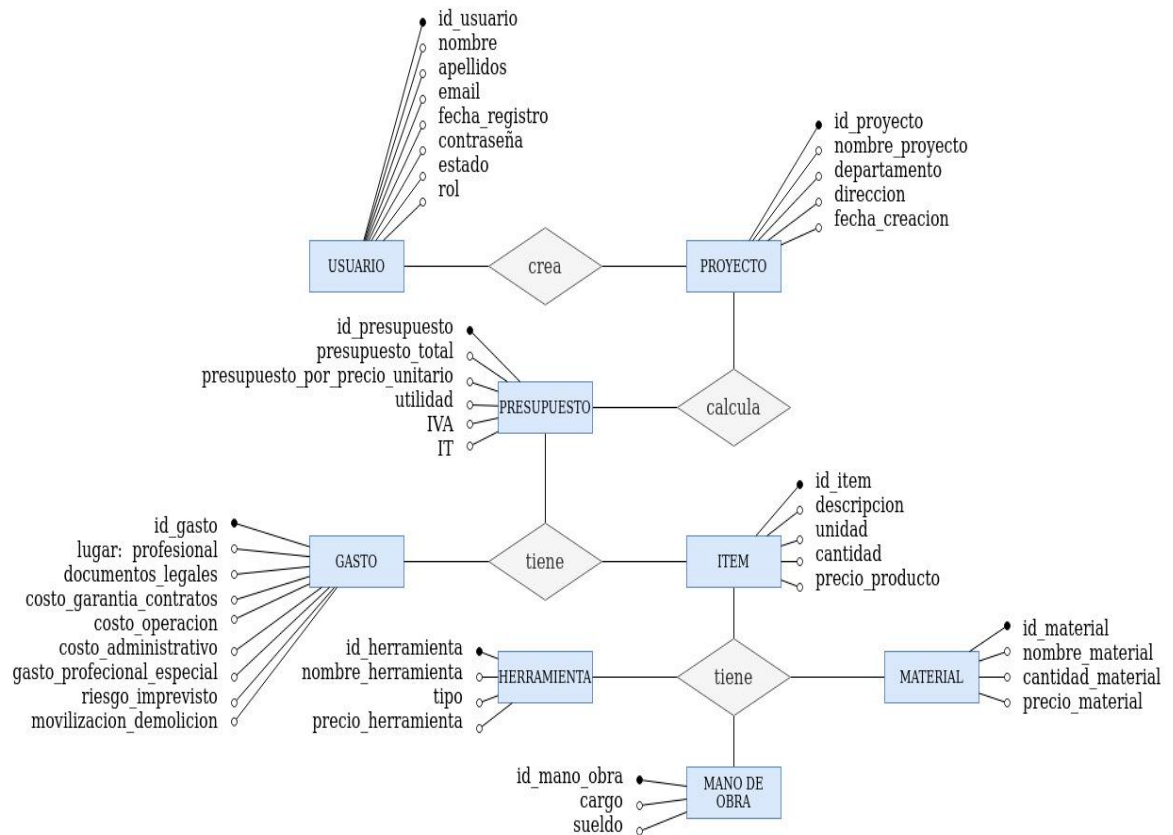


Figura 3.18 Diagrama entidad relación  
Fuente: Elaboración propia

#### 3.3.3.2 Modelo navegacional

Vamos proponiendo modelos de navegación para nuestros usuarios.

##### Modelo navegacional para el jefe de sección

En la figura 3.19 se aprecia la navegación del jefe de sección.



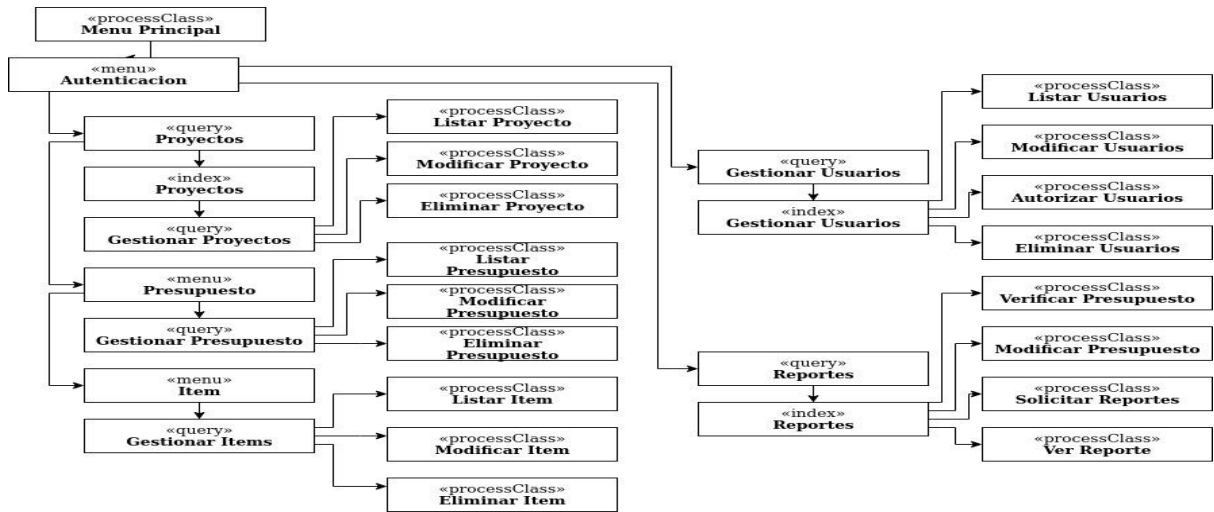


Figura 3.19 Modelo navegacional: Jefe de sección  
Fuente: Elaboración propia

### Modelo navegacional para el usuario

En la figura 3.20 vemos la navegación del usuario.

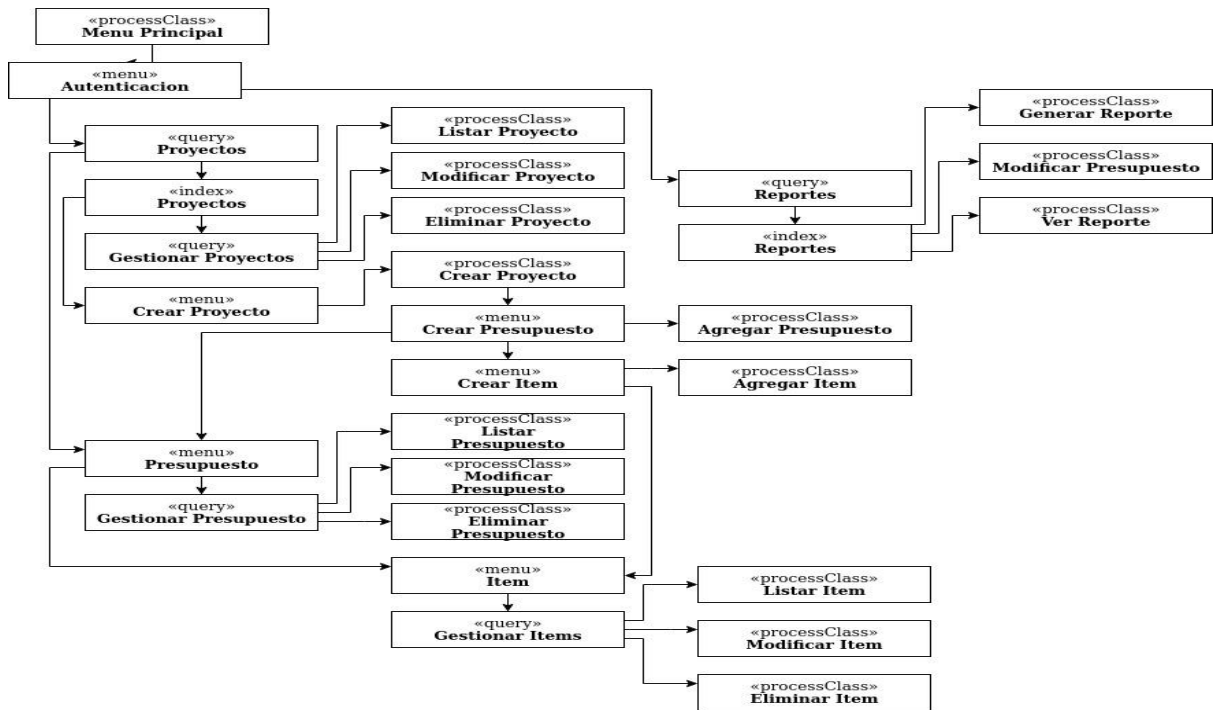


Figura 3.20 Modelo navegacional: Usuario  
Fuente: Elaboración propia

### 3.3.3.3 Modelo presentación

Propondremos tres modelos de presentación

modelo presentación para la página de inicio

En la figura 3.21 se muestra como quedará la página de inicio.

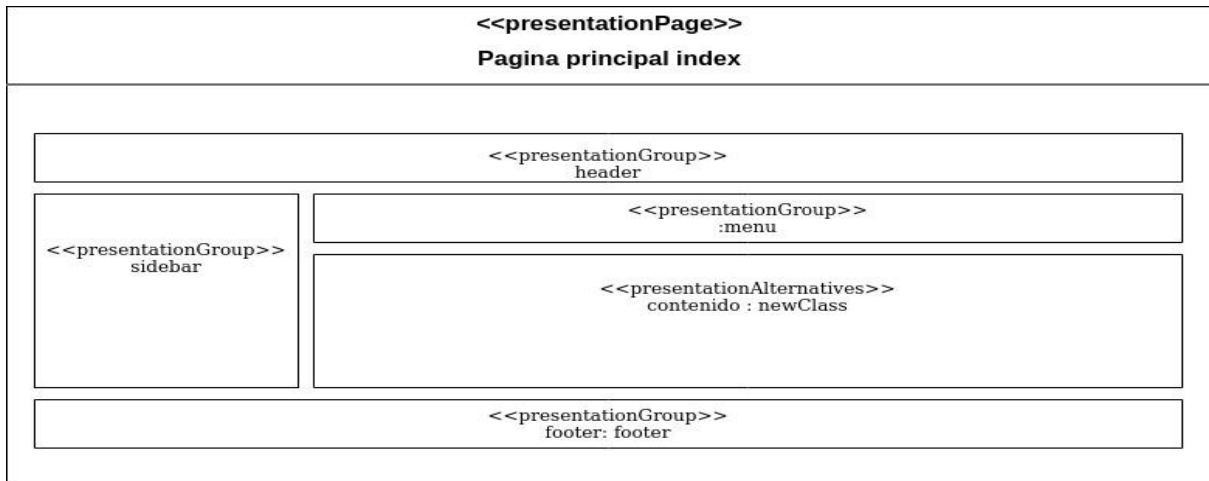


Figura 3.21 Modelo de presentación:Página principal  
Fuente: Elaboración propia

### modelo presentación para la página del jefe de sección

Para el jefe de sección debemos tener en cuenta que tendrá diferentes acciones que un usuario de esta forma armamos el modelo de presentación para el jefe de sección como se ve en la figura 3.22

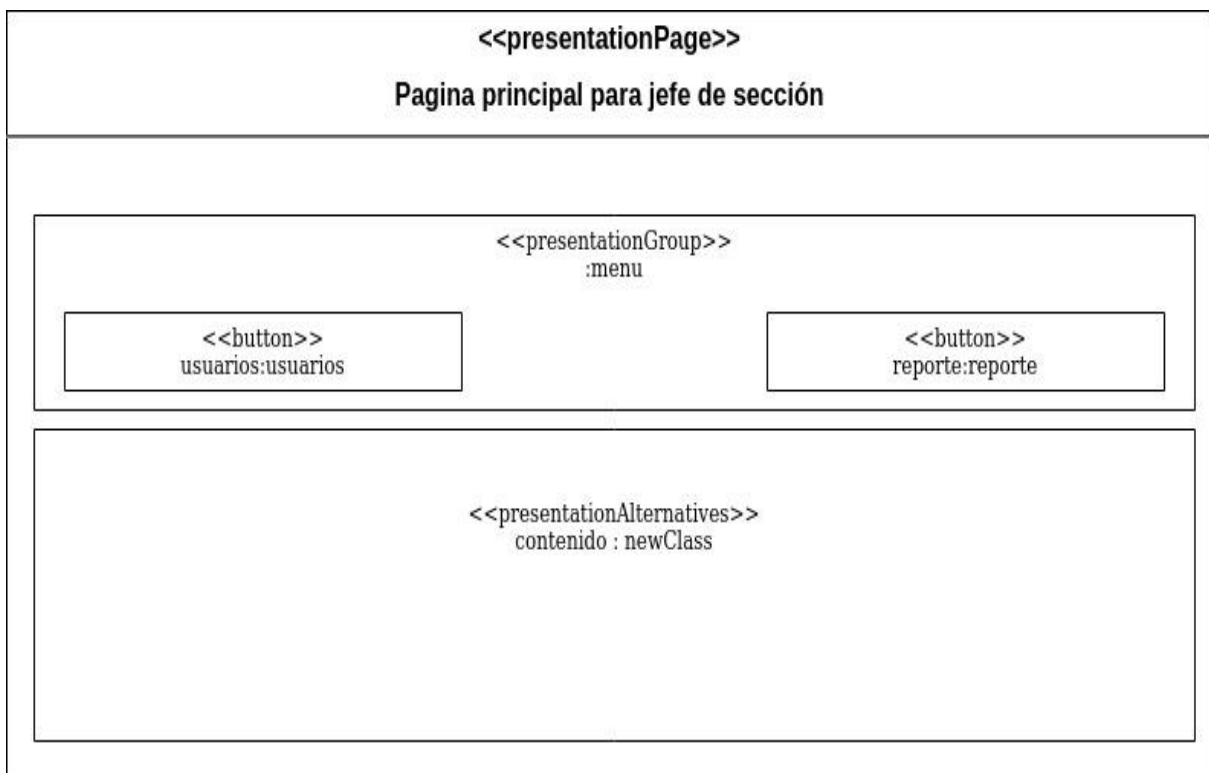


Figura 3.22 Modelo de presentación:Pagina Jefe de sección  
Fuente: Elaboración propia

## modelo presentación para la página del usuarios

Así como el jefe de sección tiene su propio modelo el usuario también lo tiene y se muestra en la figura 3.23.

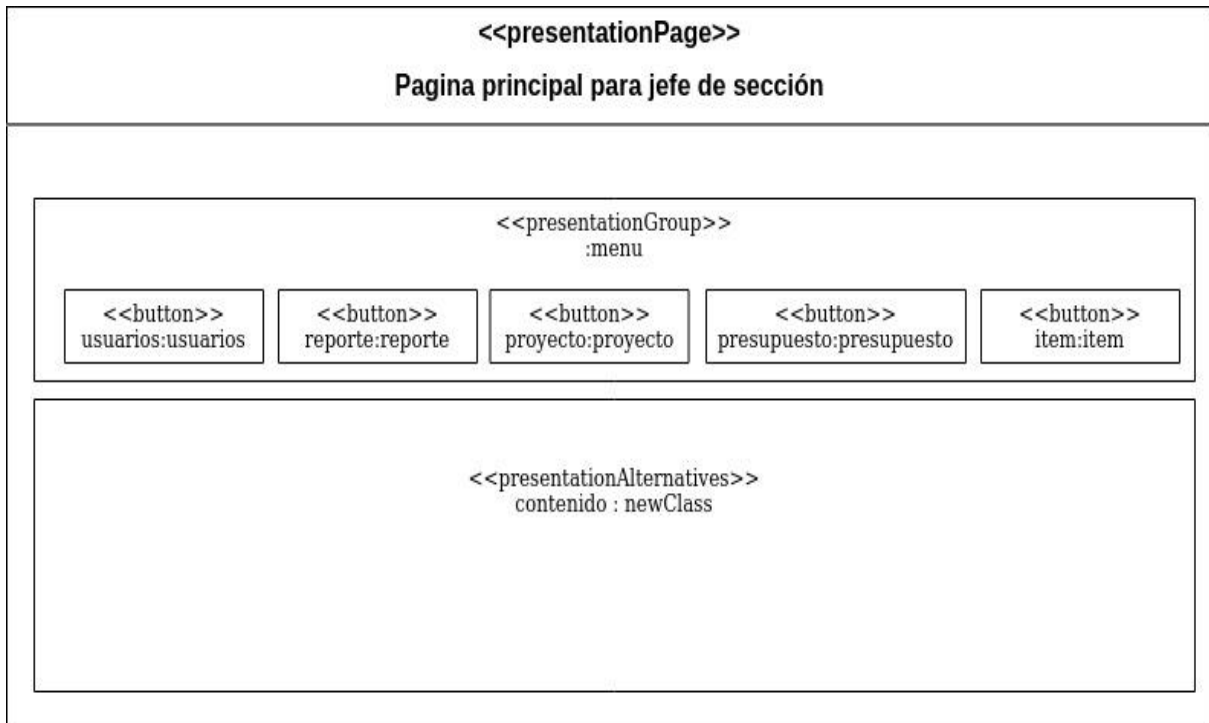


Figura 3.23 Modelo de presentación:Pagina Usuario  
Fuente: Elaboración propia

## Diseño de interfaz

El diseño de interfaz de usuario debe ser el medio de comunicación entre el usuario y sistema de manera que debe ser entendible, ordenado el diseño identifica así los objetivos y acciones para después hacer una plantilla que forma el prototipo de la interfaz de usuario.

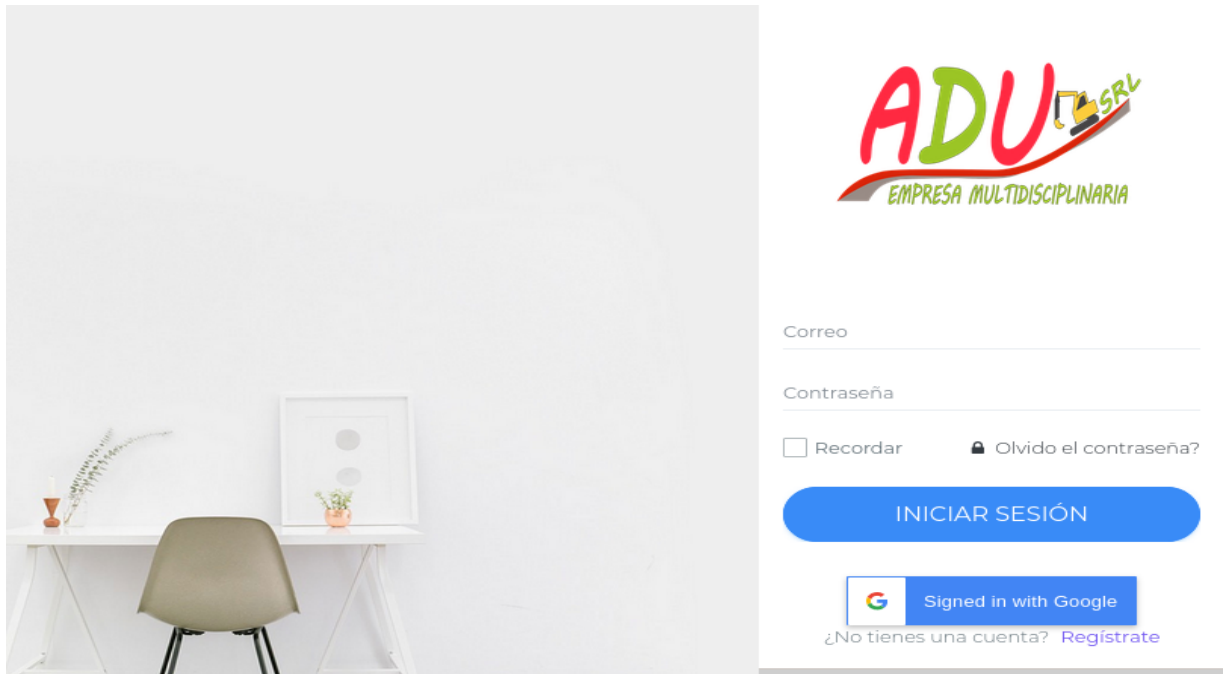


Figura 3.24 Login  
Fuente: Elaboración propia

En la figura 3.24 se ve la parte del login como se ingresará al sistema.

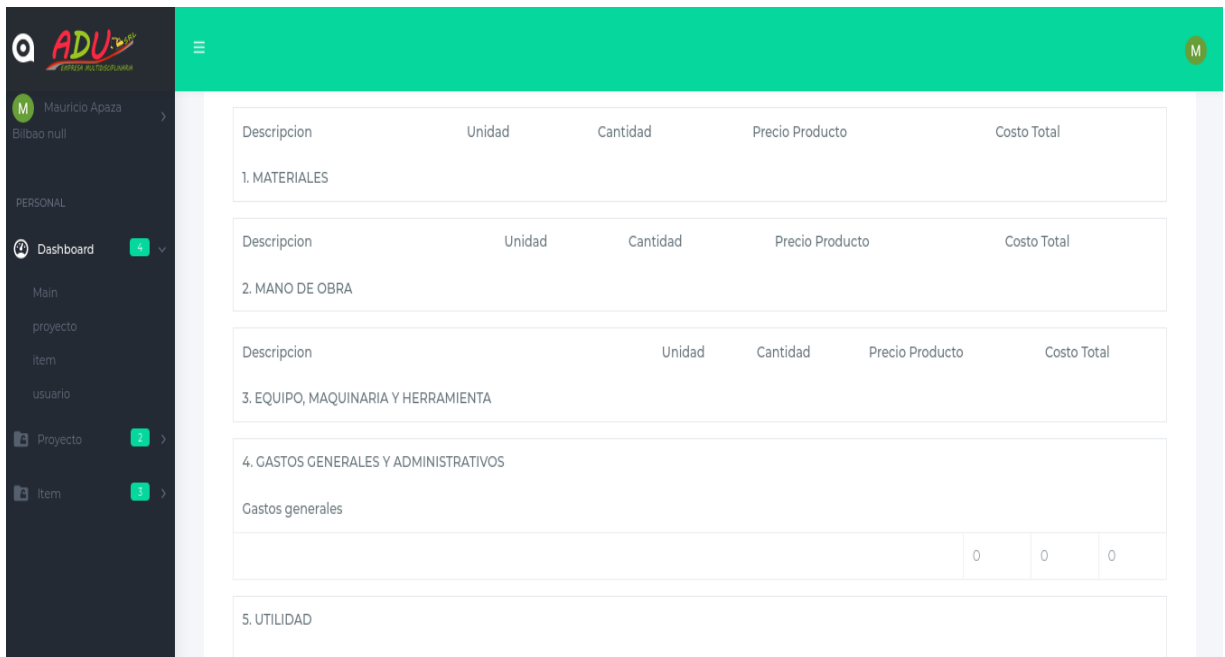


Figura 3.25 Formulario Precio Unitario  
Fuente: Elaboración propia

En la figura 3.25 mostramos el formulario del precio unitario para elaborarlo y posteriormente agregar items.

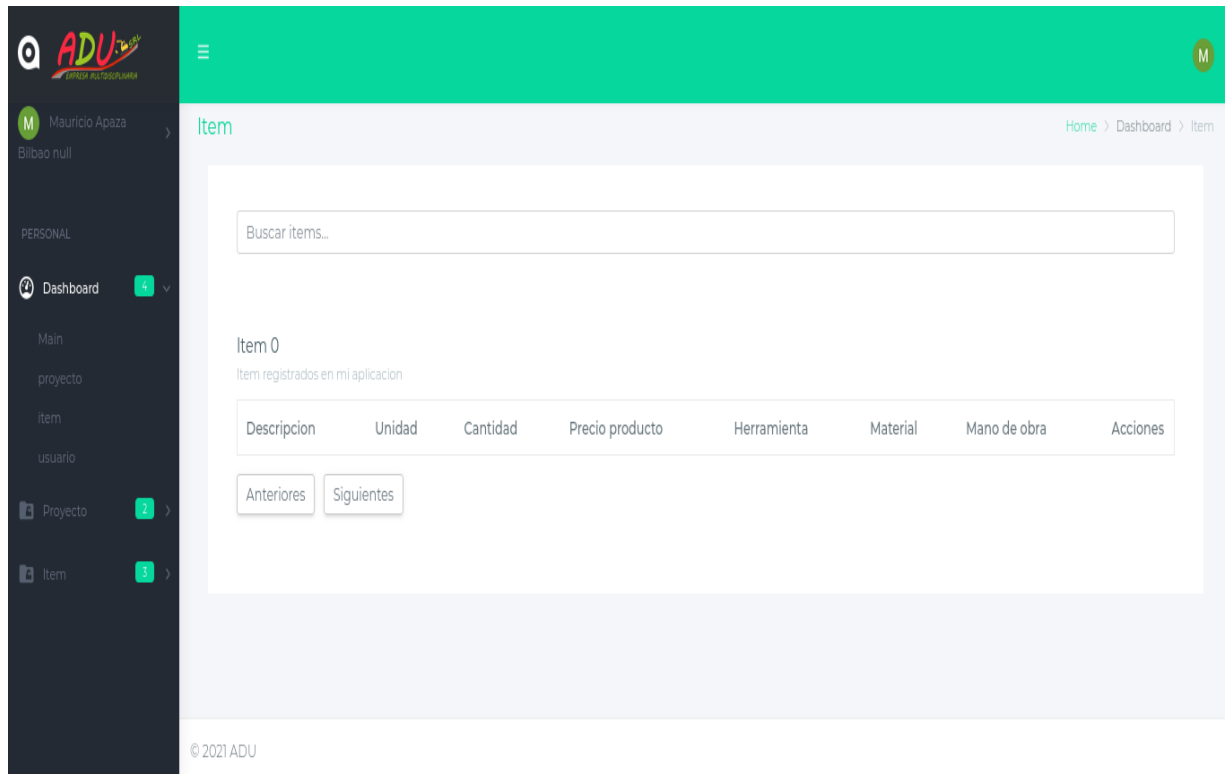


Figura 3.26 Item  
Fuente: Elaboración propia

En la figura 3.26 se muestra el ítem que debe agregarse al precio unitario tomando en cuenta la mano de obra, herramienta y el material.

### 3.3.4 FASE 4: Transición

En esta fase se tiene varias pruebas, y se analizan los defectos.

#### 3.3.4.1 PRUEBAS UNITARIAS

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el

tipo de lo que se devuelve, que si el estado inicial es válido, entonces el estado final es válido también. La idea es escribir casos de prueba para cada función no trivial o método en el módulo, de forma que cada caso sea independiente del resto. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

## CAPÍTULO IV: CALIDAD, SEGURIDAD Y COSTOS

### 4.1 INTRODUCCIÓN

La calidad es un aspecto importante del desarrollo de software y en este capítulo veremos la medición de calidad para ello aplicaremos métricas de calidad con la finalidad de evaluar el sistema. La norma ISO 9126 define un conjunto de métricas [ISO9126, 2019].

En los siguientes puntos describiremos cómo evaluar la calidad.

### 4.2 FUNCIONALIDAD

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas usaremos la métrica de punto función.

#### 4.2.1 PUNTO FUNCIÓN

El principio de este enfoque es que software comprende un número de puntos de funciones que encajan dentro de uno de los cinco tipos. A cada punto función se le da un marcador (el multiplicador) dependiendo de su grado de complejidad. Los marcadores relativos mostrados aquí abajo están basados en el análisis del proyecto [Punto Función, 2016]. Analizamos el punto función como se ve en la tabla 4.1

Tabla 4.1 Punto función

Valor dominio de información	Factor ponderado				
	Conteo	Simple	Promedio	Complejo	Total
Entradas externas (EE)	9	3	4	6	27
Salidas externas (SE)	9	4	5	7	36
Consultas externas (CE)	3	3	4	6	9

Archivos lógicos externos (ALI)	9	7	10	15	63
Archivo de interfaz externa (AIE)	0	5	7	10	0
<b>Conteo total</b>	<b>135</b>				

Fuente: (Pressman, 2003)

Los factores de ajuste de valor se usan para proporcionar un indicio de la complejidad del problema como se ve en la tabla 4.2.

Tabla 4.2 Factores de ajuste de valor

N r o	Factores de ajuste de valor (FAV)						
		1	2	3	4	5	
1	¿El sistema requiere respaldo y recuperación confiables?						5
2	¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?						4
3	¿Existen funciones de procesamiento distribuidas?						3
4	¿El desempeño es crucial?						4
5	¿El sistema correrá en un entorno operativo existente enormemente utilizado?						5
6	¿El sistema requiere entrada de datos en línea?						5
7	¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?						3
8	¿Los ALI se actualizan en línea?						2
9	¿Las entradas, salidas, archivos o consultas son complejos?						4
10	¿El procesamiento interno es complejo?						4
11	¿El código se diseña para ser reutilizable?						5
12	¿La conversión y la instalación se incluyen en el diseño?						2



1 3	¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?								3
1 4	¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?								4
	$\Sigma(F_i)$								53

Fuente: (Pressman, 2003)

Calculando el punto función con la siguiente ecuación

$$PF = \text{Conteo Total} * (0.65 + 0.01 * \Sigma(F_i)) \quad (\text{Ecu 4.1})$$

Donde el conteo total es la suma de todas las entradas PF obtenidas de la tabla 4.1.

Los  $F_i$  ( $i = 1$  a  $14$ ) son factores de ajuste de valor (FAV) con base en respuestas a las anteriores preguntas reemplazando los valores en la ecuación (Ecu 4.1)

$$PF = 135 * (0.65 + 0.01 * 53)$$

$$PF = 159.3$$

Considerando los  $F_i$  al 100% es decir a 70 que es la suma del valor máximo de los  $F_i$ .

Reemplazamos nuevamente en la ecuación (Ecu 4.1) considerando los valores al 100%.

$$PF_{max} = 135 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 182.25$$

Para calcular la funcionalidad aplicamos la siguiente ecuación.

$$\text{Funcionalidad} = PF / PF_{max} \quad (\text{Ecu 4.2})$$

Reemplazando en la ecuación (Ecu 4.2) los valores obtenidos con los diferentes factores de ajuste de valor obtenemos.

$$Funcionalidad = 159.3 / 182.25$$

$$Funcionalidad = 0.87$$

$$Funcionalidad = 87\%$$

Concluimos que la funcionalidad es del 87%

### 4.3 CONFIABILIDAD

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando se está utilizando en condiciones específicas. En este caso la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.

Calcularemos la probabilidad de ejecución del sistema libre de fallos en un determinado tiempo y usaremos la siguiente ecuación:

$$F(t) = f * e^{(-\lambda*t)} \quad (\text{Ecu 4.3})$$

Donde  $f$  es la funcionalidad encontrada a partir del punto función,  $t$  es el tiempo y  $\lambda$  es la probabilidad de error del sistema.

Reemplazando en la ecuación (Ecu 4.3) los valores obtenidos de la ecuación (Ecu 4.2) a un tiempo de 12 meses y lambda obtenemos del siguiente cálculo

$$\lambda = \text{Nro ingresos fallidos} / \text{Nro de ingresos}$$

$$\lambda = 3 / 20$$

$$F(t) = 0.84 * e^{(-0.15*12)}$$

$$F(t) = 0.14$$

$$F(t) = 14\%$$

El resultado nos indica que hay una probabilidad del 14% de fallar

#### 4.4 MANTENIBILIDAD

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, cambios en el entorno, y especificaciones de requerimientos funcionales.

IEEE Std. 982.1-1988 [IEE93] sugiere un índice de madurez de software (IMS) que proporcione un indicio de la estabilidad de un producto de software (con base en cambios que ocurran para cada liberación del producto). Para ello, se determina la siguiente información:

$M_t$  número de módulos en la liberación actual

$F_c$  número de módulos en la liberación actual que cambiaron

$F_a$  número de módulos en la liberación actual que se agregaron

$F_d$  número de módulos de la liberación anterior que se borraron en la liberación actual

El índice de madurez del software se calcula de la forma siguiente con la ecuación (Ecu 4.4)

$$IMS = \frac{M_t - (F_a + F_c + F_d)}{M_t} \quad (\text{Ecu 4.4})$$

Reemplazando en la ecuación (Ecu 4.4) obtenemos.

$$IMS = \frac{7 - (0 + 2 + 0)}{7}$$

$$IMS = 0.71$$

$$IMS = 71\%$$

Este resultado de 0.71 tiene a 1.00 lo cual nos indica que el sistema puede estabilizarse es de esta forma que mantenimiento es posible.

## 4.5 PORTABILIDAD

La portabilidad se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma. La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

Se toma en cuenta los recursos de hardware y software

### 4.5.1 PORTABILIDAD DE HARDWARE Y SOFTWARE

La portabilidad de hardware, puede ser implementado en un servidor y puede ser transferido a otro servidor con más o menos recursos esto no afecta al sistema pero si al rendimiento.

Se debe tener en cuenta la base de datos, esta debe permitir recuperar datos y además ser accesible a diferentes entornos de programación.

El sistema puede correr en cualquier plataforma web

## 4.6 USABILIDAD

La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su funcionalidad y eficiencia.

Para determinar la usabilidad del sistema nos apoyamos en la encuesta a dos usuarios. Usamos el cuestionario de la tabla 4.3 y valoramos del 1 al 5

Tabla 4.3 Cuestionario

Nro	Cuestionario	Respuestas	
1	¿El sistema es comprensible ?	4	5
2	¿El sistema le permite realizar su trabajo sin contratiempos?	5	4
3	¿El sistema satisface sus requerimientos?	5	4

4	¿La emisión de reportes corresponde a sus expectativas?	4	5
5	¿El tiempo en realizar un proceso es menor al realizado sin el sistema?	5	5
6	¿Considera que el ingreso de parámetros al realizar alguna tarea le provoca contratiempos ?	3	2
7	¿Considera que el acceso al sistema es seguro?	5	5
8	¿Le resulta sencillo el manejo del sistema?	3	2
9	¿Es comprensible el formulario de registro de datos?	4	3
10	¿El registro de datos es efectivo?	5	3
	<b>Totales</b>	<b>43</b>	<b>38</b>
	<b>Total</b>		<b>40.5</b>

Fuente: Elaboración propia

Usamos la siguiente ecuación (Ecu 4.5)

$$F = p_u/p \quad (\text{Ecu 4.5})$$

Donde

F es la usabilidad del sistema

$p_u$  es la suma de las respuestas de los usuarios

$p$  el total de todas las preguntas ponderada al máximo

Reemplazando datos en la ecuación (Ecu 4.5).

$$F = 40.5/50$$

$$F = 0.81$$

$$F = 81\%$$

Este resultado nos indica que tenemos una usabilidad del 81%

#### 4.7 SEGURIDAD

La seguridad de software es utilizada para proteger el software contra ataques maliciosos y otros riesgos de forma que el software siga funcionando correctamente.

La seguridad de software está relacionada con la calidad confiabilidad y portabilidad el software que no tiene alta calidad es fácil de penetrar su seguridad.

#### **4.7.1 SEGURIDAD EN BASE DE DATOS**

Los ataques a la base de datos son las consultas que se le hace a la base de datos. Se hace uso de las debilidades del diseño de la base de datos para manipular la información.

Los accesos no autorizados, para ello la asignación de roles es importante, los accesos a formularios y listas deben estar restringidas.

Se deben proteger todos los datos restringiendo el acceso, proteger los datos mediante copias de seguridad cada cierto tiempo, cifrar las contraseñas.

##### **4.7.1.1 COPIAS DE SEGURIDAD**

El gestor de base de datos cuenta con la opción de hacer una copia de seguridad de la base de datos, esta configuración puede ser modificada y puede realizarse cada cierto tiempo.

##### **4.7.1.2 TRANSACCIONES**

Con este mecanismo nos aseguramos de realizar una consulta a la base de datos, esta transacción solo se ejecuta cuando es preciso y luego la consulta se acaba.

De esta forma evitamos cargar todas las consultas, evitando fallas de almacenamiento en la base de datos, evitando mostrar todos los datos.

#### **4.7.2 SEGURIDAD DEL USUARIO**

Para la seguridad del sistema se aplica la autenticación del usuario la cual requiere el nombre de usuario y su contraseña. Además el usuario registrado necesita ser habilitado por el administrador y se le asigna un rol. El usuario contará con la opción de cambiar su contraseña si es que lo desea.

#### 4.7.2.1 CONTROL DE ACCESO

Para este control de acceso es necesario asignar roles y solo el administrador podrá asignar los roles correspondientes a los usuarios también podrá cambiar los roles.

Así cada usuario con su rol podrá ingresar al sistema y acceder solo a los módulos que le corresponden.

### 4.8 COSTOS

Antes de que el proyecto pueda comenzar, debemos estimar el trabajo que se va a realizar, los recursos que se requerirán y el tiempo que transcurrirá de principio a fin. La técnica de análisis de costo beneficio tiene el propósito de proporcionar las medidas de los costos. Mostraremos los beneficios y su costo de desarrollo para realizar esto se hará uso de COCOMO II una herramienta que nos ayudará a estimar el costo. Luego de realizar los cálculos se determinará si el proyecto es viable, sostenible y comprobar que es una buena opción invertir en el proyecto.

#### 4.8.1 COCOMO

El modelo de estimación de software COCOMO Constructive Cost Model modelo constructivo de costos es uno de los modelos más utilizados, nos permite estimar el costo y el esfuerzo, consta de tres modelos.

**Modo Orgánico (Organic):** En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar es relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware.

**Modo Semi Acoplado (Semi Detached):** Es un modelo para productos de software de tamaño y complejidad media. Las características de los proyectos se consideran intermedias a las de los modos Orgánico y Empotrado [COCOMO, 2012].

**Modo Empotrado (Embedded):** En esta clasificación están incluidos proyectos de gran envergadura que operan en un ambiente complejo con altas restricciones

de hardware, software y procedimientos operacionales, tales como los sistemas de tráfico aéreo.

### Estimación del esfuerzo

Para hacer la estimación usaremos la siguiente ecuación

$$E = a * KLDC^b \quad (\text{Ecu 4.6})$$

Donde

*E*: Es el esfuerzo estimado. Representa los meses-persona necesarios para ejecutar el proyecto.

*KLDC*: Es el tamaño del software a desarrollar en miles de líneas de código.

*a, b*: Son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semi-acoplado, Empotrado)

Tabla 4.4 Coeficientes de estimación de a, b, c y d de COCOMO

Proyecto de software	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi-Acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Fuente:(Pressman, 2003)

En total se obtiene 274.72 del punto función

Tabla 4.5 Factor de correlación

Lenguaje	Factor LDC/DF
C	128
Ansi Basic	64
Java	53
JavaScript	47
Visual Basic	46
ASP	36
PHP	29
Visual C++	34

Fuente: (COCOMO, 2012)



Se tiene la siguiente ecuación para calcular el número de líneas de código o LDC.

$$LDC = PF * FactorLDC \text{ (Ecu 4.7)}$$

Donde:

*LDC*: Líneas de código

*PF*: Punto función

*FactorLDC*: Es el factor de líneas de código el cual está dado por la tabla 4.5

Se usará el factor de correlación de Javascript de esta forma reemplazando en la ecuación 4.7

$$LDC = 159.3 * 47$$

$$LDC = 7487.1$$

Obteniendo el valor del *LDC* vamos a obtener el valor del *KLDC* con la siguiente ecuación 4.8.

$$KLDC = LDC/1000 \quad \text{(Ecu 4.8)}$$

$$KLDC = 7487.1/1000$$

$$KLDC = 7.48$$

Ya tenemos todos los valores necesarios para reemplazar en la ecuación 4.6 y así obtener el cálculo del esfuerzo en personas-mes usaremos el modo orgánico ya que *KLDC* no supera los 50 *KLDC*

$$E = 2.4 * 7.48^{1.05}$$

$$E = 19.87$$

Ahora calculamos el tiempo requerido para lo cual usaremos la siguiente ecuación 4.9.

$$D = c(E)^d \quad (\text{Ecu 4.9})$$

Donde:

$D$ : Tiempo requerido para el proyecto

$c, d$ : Son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado)

$E$ : Es el esfuerzo estimado. Representa los meses-persona necesarios para ejecutar el proyecto.

$$D = 2.5(19.87)^{0.38}$$

$$D = 7.79$$

$$D = 8$$

Reemplazando el esfuerzo y el tiempo en la siguiente ecuación 4.10 obtenemos el número de personas necesarias para realizar el proyecto

$$P = \frac{E}{D} \quad (\text{Ecu 4.10})$$

Donde:

$P$ : Número de personas necesarias para realizar el proyecto

$E$ : Es el esfuerzo estimado. Representa los meses-persona necesarios para ejecutar el proyecto.

$D$ : Tiempo requerido para el proyecto

$$P = \frac{19.87}{7.79}$$

$$P = 2.55$$

$$P = 3$$

El costo mensual se obtiene del personal y de la sueldo mensual de 450\$ así la ecuación queda

$$C_{mes} = P * S_{regular} \quad (\text{Ecu 4.11})$$

Donde

$C_{mes}$ : Es el costo mensual de desarrollo

$P$ : Número de personas necesarias

$S_{regular}$ : Es el sueldo regular

$$C_{mes} = 3 * 450$$

$$C_{mes} = 1350\$$$

El costo total se obtiene con la ecuación 4.12

$$C_{total} = C_{mes} * D \quad (\text{Ecu 4.12})$$

Donde:

$C_{total}$ : Es el costo total del proyecto

$C_{mes}$ : El costo mensual de desarrollo

$D$ : Tiempo requerido para el proyecto

$$C_{total} = 1350 * 8$$

$$C_{total} = 10800\$$$

El costo total de la aplicación llegará a ser 10800\$

## **CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES**

### **5.1 CONCLUSIONES**

Finalizando el presente proyecto se obtuvieron las siguientes conclusiones.

- Se logró desarrollar el Sistema Presupuestario de Información para Proyectos de Construcción caso: empresa multidisciplinaria ADU S.R.L., haciendo uso de la tecnología.
- Se recopilaron los requerimientos de la empresa, fueron realizados satisfactoriamente mediante el uso de la metodología RUP. Esta metodología ayudó a realizar un sistema web de alta calidad.
- La automatización de los precios unitarios además de otros cálculos se realizó esto ayudó al tiempo y costo de elaboración de estos presupuestos.
- Con el nuevo sistema se logró optimizar el tiempo haciendo los cálculos de costos más eficientes y fáciles de modificar ya que el sistema con el cual trabajaban no era del todo automatizado.
- Se diseñaron las interfaces necesarias de manera que sean amigables es decir que sean fáciles de usar.
- Se elaboró e implementó una base de datos la cual cumple con los requerimientos de la empresa.
- Se verificó el funcionamiento y calidad del sistema mediante los métodos implementados en el capítulo 4.

### **5.2 RECOMENDACIONES**

- Se recomienda hacer copias de seguridad de manera constante para evitar la pérdida de información.
- Si el usuario desea realizar algún cambio es necesario revisar la documentación y cada componente del sistema
- Se recomienda mantener actualizado los datos para el sistema para evitar malos cálculos en el presupuesto.

## BIBLIOGRAFÍA

COCOMO. (2012). El modelo Cocomo.

<https://blogadmi1.files.wordpress.com/2010/11/cocomo01full.pdf>

Fernández, C (2014) Metodología de la investigación.

Ferrufino, A. M. (1997) Análisis de precios unitarios para la construcción.

Ingeniería Software. (2014). Ingeniería de software,

[https://es.slideshare.net/efecto\\_006/ingeniera-de-software-52089007](https://es.slideshare.net/efecto_006/ingeniera-de-software-52089007)

Ingeniería Web. (2018). Ingeniería Web: Que es, características y todo lo que

debes saber <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-web/>

ISO9126. (2019). Calidad en la Industria del Software. La Norma ISO-9126,

<http://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

Koch, A. K. (2002). The expressive Power of UML-based Web Engineering.

Larman G. (1999). UML y patrones, Introducción Al análisis y diseño orientado a

objetos, Prentice Hall México

Ludwing. (2000). Metodología UWE,

<http://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html>

MOO. (2015) Metodología orientada a objetos,

[https://www.academia.edu/15943563/Que\\_es\\_la\\_Metodolog%C3%ADa\\_orientada\\_a\\_objetos\\_Introducci%C3%B3n\\_a\\_UML](https://www.academia.edu/15943563/Que_es_la_Metodolog%C3%ADa_orientada_a_objetos_Introducci%C3%B3n_a_UML)

Pressman R. (2003). Ingeniería de Software – Un enfoque práctico.

Punto Funcion. (2016) Análisis de punto función,

<https://www.praxisframework.org/es/library/function-point-analysis>

RUP. (2020). Metodología RUP, <https://metodoss.com/metodologia-rup/>

UML. (2014) Diagramas UML,

[https://www.teatroabadia.com/es/uploads/documentos/iagramas\\_del\\_uml.pdf](https://www.teatroabadia.com/es/uploads/documentos/iagramas_del_uml.pdf)

UWE. (2015) Metodología UWE <https://metodologiauwe.wordpress.com/>

ANEXOS

ACCIÓN		MAYO	JUNIO	JULIO	AGOS TO	SEPT IEMB RE	OCT UBRE	NOVI EMB RE	DICIE MBRE
	Capítulo I Marco Introducctorio	█							
	Capitulo II Marco Teorico	█							
	Recopilacion de informacion	█							
	Análisis de la información	█	█						
	Desarrollo de la documentación		█	█					
	Capítulo III Marco Aplicativo			█					
	Análisis de la información obtenida			█	█				
	Metodología				█				
	Requerimientos				█				
	Análisis					█			
	Diseño					█			



## Documentación