

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
"BOLIVIA"

## **MARCO PRÁCTICO DE TRABAJO DE GRADO**



### **SISTEMA WEB COLABORATIVO DE CONTROL DE INVENTARIO DE PROYECTOS. CASO DE ESTUDIO: CONSTRUCTORA SUAREZ**

**GERALD MICHAEL GUTIERREZ MALDONADO**

**COCHABAMBA, 2015**

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

## **MARCO PRÁCTICO DE TRABAJO DE GRADO**

### **SISTEMA WEB COLABORATIVO DE CONTROL DE INVENTARIO DE PROYECTOS. CASO DE ESTUDIO: CONSTRUCTORA SUAREZ**

**GERALD MICHAEL GUTIERREZ MALDONADO**

**Modalidad: Proyecto de Grado  
Presentado como requisito  
Parcial para optar al título de  
Licenciatura de Ingeniería de  
Sistemas**

**TUTOR: SAMUEL LUJAN NAVARRO**

**COCHABAMBA, 2015**

<b>1</b>	<b>GENERALIDADES .....</b>	<b>1</b>
<b>1.1</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.2</b>	<b>ANTECEDENTES .....</b>	<b>3</b>
<b>1.3</b>	<b>PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>4</b>
<b>1.3.1</b>	<b>Identificación del problema.....</b>	<b>4</b>
<b>1.3.1.1</b>	<b>Identificación de la situación problemática .....</b>	<b>5</b>
<b>1.3.1.2</b>	<b>Identificación de las causas .....</b>	<b>5</b>
<b>1.3.2</b>	<b>Formulación del problema .....</b>	<b>6</b>
<b>1.3.3</b>	<b>Análisis Causa-efecto.....</b>	<b>6</b>
<b>1.3.3.1</b>	<b>Causas.....</b>	<b>6</b>
<b>1.3.3.2</b>	<b>Efecto.....</b>	<b>6</b>
<b>1.4</b>	<b>OBJETIVOS Y ACCIONES .....</b>	<b>6</b>
<b>1.4.1</b>	<b>Objetivo General.....</b>	<b>6</b>
<b>1.4.2</b>	<b>Objetivos Específicos y Acciones.....</b>	<b>6</b>
<b>1.5</b>	<b>JUSTIFICACION .....</b>	<b>9</b>
<b>1.5.1</b>	<b>Justificación Técnica.....</b>	<b>9</b>
<b>1.5.2</b>	<b>Justificación Económica.....</b>	<b>9</b>
<b>1.5.3</b>	<b>Justificación Operativa.....</b>	<b>10</b>
<b>1.6</b>	<b>ALCANCE .....</b>	<b>10</b>
<b>1.6.1</b>	<b>Alcance Temático .....</b>	<b>10</b>
<b>1.6.2</b>	<b>Alcance Institucional .....</b>	<b>10</b>
<b>1.6.3</b>	<b>Alcance Temporal .....</b>	<b>10</b>
<b>1.6.4</b>	<b>Alcance Proyecto .....</b>	<b>10</b>
<b>1.6.5</b>	<b>Alcance Geográfico.....</b>	<b>11</b>
<b>1.7</b>	<b>HIPOTESIS.....</b>	<b>11</b>
<b>1.7.1</b>	<b>Análisis de Variables .....</b>	<b>11</b>
<b>1.7.1.1</b>	<b>Variables Independientes.....</b>	<b>11</b>
<b>1.7.1.2</b>	<b>Variables Dependientes.....</b>	<b>11</b>
<b>1.7.2</b>	<b>Definición Conceptual .....</b>	<b>11</b>
<b>1.7.2.1</b>	<b>Variables Independientes.....</b>	<b>11</b>
<b>1.7.2.2</b>	<b>Variables Dependientes.....</b>	<b>12</b>
<b>1.7.3</b>	<b>Operativización de variables .....</b>	<b>13</b>
<b>1.8</b>	<b>MATRIZ DE CONSISTENCIA .....</b>	<b>14</b>
<b>2</b>	<b>MARCO TEÓRICO.....</b>	<b>15</b>

2.1	TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN.....	15
2.1.2	Metodología de la investigación cuantitativa .....	15
2.1.2.1	Técnicas de Recolección de datos cuantitativos .....	15
2.2	SISTEMAS DE INVENTARIOS.....	17
2.2.2	Método FIFO.....	18
2.2.3	Método LIFO.....	18
2.2.4	Método de Promedio Ponderado.....	19
2.3	SISTEMAS WEB .....	19
2.3.1	Servicios Web .....	20
2.4	SISTEMAS COLABORATIVOS.....	23
2.4.1	GROUPWARE .....	24
2.4.2	WORKFLOW.....	25
2.5	INGENIERA DE SOFTWARE. ....	26
2.5.1	Unified Modeling Language (UML).....	27
2.5.1.1	Desarrollo de un diagrama de actividades .....	27
2.5.1.2	Diagrama de Casos de Uso.....	29
2.5.1.3	Diagrama de Colaboración.....	31
2.5.1.4	Diagrama basado en clases .....	33
2.5.2	Modelos del Proceso .....	35
2.5.2.1	Proceso Ágil.....	35
2.5.2.2	Programación Extrema (XP).....	36
2.5.2.3	SCRUM.....	38
2.5.2.4	Desarrollo adaptativo de software (DAS).....	41
2.6	ARQUITECTURA DE SOFTWARE .....	43
2.6.1	Arquitectura Cliente-Servidor .....	43
2.7	TECNOLOGÍAS DE DESARROLLO.....	46
2.7.1	Lenguajes de programación del lado servidor .....	46
2.7.1.1	JAVA .....	46
2.7.1.2	RUBY.....	49
2.7.1.3	C# .....	50
2.7.2	Lenguajes de programación del lado Cliente .....	52
2.7.2.1	JavaScript .....	52
2.7.2.2	Applets de Java .....	53
2.7.2.3	Visual Basic Script.....	53

2.7.3	Frameworks del lado Servidor.....	55
2.7.3.1	Sinatra.....	55
2.7.3.2	Padrino .....	55
2.7.3.3	Rails .....	56
2.8	SISTEMAS DE GESTIÓN DE BASE DE DATOS .....	57
2.8.1	SQL Server.....	57
2.8.2	PostgreSQL.....	58
2.8.3	MySQL .....	59
3	MARCO PRÁCTICO .....	61
3.1	ANÁLISIS DE LOS PROCESOS DE PLANILLAS DEL SECTOR DE COSTOS DIRECTOS Y DISEÑO DEL MODELADO DE NEGOCIO ACTUAL.....	61
3.1.1	Recopilación de información relacionada con el seguimiento de proyectos empleando Sistemas Colaborativos .....	61
3.1.2	Elaboración de entrevistas con el personal encargado de los registros de los procesos de planillas.....	61
3.1.3	Elaboración del diagrama de flujo que represente el modelado de negocio actual de los procesos de movimiento de planilla.....	62
3.2	DISEÑO DEL MODELADO DE NEGOCIO ALTERNATIVO .....	63
3.2.1	Elaborar un diagrama de flujo de negocio alternativo. ....	63
3.2.2	Identificar las falencias de los procedimientos manuales.....	64
3.2.3	Seleccionar la metodología de desarrollo de Software.....	64
3.2.4	Planificar el desarrollo de software en base a la metodología seleccionada	66
3.2.5	Relación entre product backlog y objetivos específicos .....	70
3.3	DISEÑAR E IMPLEMENTAR MÓDULO DE GESTIÓN DE USUARIOS PARA LA CONSTRUCTORA.....	71
3.3.1	Seleccionar el lenguaje de programación y sistema de gestor de base de datos	71
3.3.2	Realizar análisis de requerimientos para el módulo de usuarios.....	76
3.3.3	Descripción de Usuarios .....	76
3.3.4	Sprint 1: Administración de usuarios.....	77
3.3.5	Elaborar los diagramas de casos de uso.....	79
3.3.6	Diagrama de colaboración .....	79
3.3.7	Diagrama de clases.....	80
3.3.8	Diseñar la base de datos.....	81
3.3.9	Diccionario de datos .....	81

3.3.10	Codificar módulo de gestión de usuarios .....	85
3.4	<b>DISEÑAR E IMPLEMENTAR UN MÓDULO DE REGISTRO DE PROYECTOS EN GENERAL.</b> ....	85
3.4.1	Identificar actores y objetivos involucrados .....	85
3.4.2	Sprint 2: Administración de proyectos.....	85
3.4.3	Diagrama de caso de uso.....	87
3.4.4	Diagrama de colaboración. ....	87
3.4.5	Diagrama de clases.....	89
3.4.6	Diagrama de Base de datos .....	89
3.4.7	Diccionario de datos .....	89
3.4.8	Codificar módulo de administración de proyectos.....	91
3.5	<b>DESARROLLAR EL MODULO DE SOLICITUD DE PEDIDOS Y COMPRA DE MATERIAL</b> .....	93
3.5.1	Identificar actores y objetivos involucrados .....	93
3.5.2	Sprint 3: Solicitud de pedidos y compra de material.....	93
3.5.3	Diagrama de caso de uso.....	95
3.5.4	Diagrama de colaboración .....	95
3.5.5	Diagrama de clases.....	97
3.5.6	Diagrama de base de datos.....	98
3.5.7	Diccionario de datos .....	99
3.5.8	Codificar módulo de solicitud de compras .....	104
3.6	<b>DESARROLLAR EL MODULO DE MOVIMIENTOS DE INVENTARIOS</b> .....	105
3.6.1	Identificar actores y objetivos involucrados .....	105
3.6.2	Sprint 4: Movimiento de inventarios .....	105
3.6.3	Diagrama de caso de uso.....	107
3.6.4	Diagrama de colaboración .....	107
3.6.5	Diagrama de clases.....	109
3.6.6	Diagrama de base de datos.....	110
3.6.7	Diccionario de datos .....	111
3.6.8	Codificar módulo de solicitud de movimiento de inventarios .....	117
3.7	<b>DISEÑAR E IMPLEMENTAR EL MODULO COLABORATIVO DEL SISTEMA PARA GENERAR EL SEGUIMIENTO DE LA INFORMACION QUE FLUYE DE UN PERSONAL A OTRO.</b> .....	118
3.7.1	Identificar actores y objetivos involucrados .....	118
3.7.2	Sprint 5: Colaboración del sistema.....	118
3.7.3	Diagrama de clases.....	119

3.7.4	Diagrama de base de datos.....	119
3.7.5	Diccionario de datos.....	120
3.7.6	Diagrama de estados .....	121
3.7.7	Codificar módulo de colaboración del sistema para generar seguimiento de la información que fluye de un personal a otro.....	122
3.8	REALIZAR PRUEBAS AL SISTEMA FINAL .....	122
3.9	DEMOSTRACION DE HIPOTESIS.....	122
3.9.1	Demostración de la primera variable dependiente .....	122
3.9.2	Demostración de la segunda variable dependiente.....	123
3.9.3	Demostración de la tercera variable dependiente .....	123
3.9.4	Demostración de la variable independiente .....	124
3.9.5	Definición de la hipótesis.....	125
3.9.6	Calculo de estadístico T.....	126
	Bibliografía.....	128

# ÍNDICE DE TABLAS

<b>CONTENIDO</b>	<b>Página</b>
<b>Tabla 1:</b> Objetivos Específicos y Accione.....	7
<b>Tabla 2:</b> Operativización de variables.....	13
<b>Tabla 3:</b> Metodologías de desarrollo de software.....	68
<b>Tabla 4:</b> Bitácora del Product Owner.....	71
<b>Tabla 5:</b> Lenguajes de Programación.....	72
<b>Tabla 6:</b> Gestores de Base de Datos .....	75
<b>Tabla 7:</b> Historias de usuario para módulo de usuarios.....	80
<b>Tabla 8:</b> Historias de usuario para módulo de proyectos.....	83



# ÍNDICE DE FIGURAS

<b>CONTENIDO</b>	<b>Página</b>
<b>Figura 1:</b> Matriz de Consistencia.....	14
<b>Figura 2:</b> Ejemplo de uso de Servicios Web.....	22
<b>Figura 3:</b> Elementos clase de un proceso de negocio.....	26
<b>Figura 4:</b> Diagrama de actividades.....	28
<b>Figura 5:</b> Casos de uso.....	29
<b>Figura 6:</b> Actores con caso de uso.....	30
<b>Figura 7:</b> Diagrama de casos de uso.....	30
<b>Figura 8:</b> Clases entidad.....	31
<b>Figura 9:</b> Clase Interfaz de usuario.....	31
<b>Figura 10:</b> Clase Interfaz responsable.....	32
<b>Figura 11:</b> Clase Interfaz.....	32
<b>Figura 12:</b> Diagrama de Clase.....	34
<b>Figura 13:</b> Diagrama de Clases.....	35
<b>Figura 14:</b> Proceso XP.....	37
<b>Figura 15:</b> Proceso SCRUM.....	40
<b>Figura 16:</b> Proceso DAS.....	42
<b>Figura 17:</b> Cliente-Servidor.....	44
<b>Figura 18:</b> Modelo Cliente/Servidor.....	44
<b>Figura 19:</b> Diagrama de flujo de negocio actual.....	66

<b>Figura 20:</b> Diagrama de flujo de negocio actual.....	66
<b>Figura 21:</b> Diagrama de casos de uso.....	78
<b>Figura 22:</b> Diagrama de clases.....	79
<b>Figura 23:</b> Diagrama de base de datos.....	79
<b>Figura 24:</b> Interfaz de entrada de usuario .....	81
<b>Figura 25:</b> Diagrama de base de datos.....	82
<b>Figura 26:</b> Tabla de entidad pública.....	82
<b>Figura 27:</b> Entrada de proyectos.....	84
<b>Figura 28:</b> Registro de proyectos.....	84

# 1 GENERALIDADES

## 1.1 INTRODUCCIÓN

Los sistemas web son sistemas que funcionan a través de internet o intranet y cuyo contenido se ve únicamente por solicitud del usuario, además debe disponer de una base de datos. Los sistemas web son ampliamente utilizados por las empresas ya que facilitan el modo de acceso a sus empleados o clientes ya que solo es necesaria una conexión a internet. Además los sistemas web son dinámicos lo que favorece la adaptabilidad de la empresa a cualquier cambio que pueda ocurrir. (Andariel, 2013), una arquitectura web es una forma de diseñar y desarrollar en la web, permite ordenar todos los elementos que se utilizan para tener un resultado mucho más elaborado. Hay distintas arquitecturas que tienen enfoques diferentes, la interfaz, la funcionalidad, la conexión a las bases de datos, etc. (Mariño y Godoy, 2003).

Un sistema de información efectivo, proporciona a los usuarios la información necesaria para responder a los requerimientos de su entorno. Sin embargo, debido a que la demanda de información cambia, estos sistemas también deben evolucionar. Estas demandas van desde pequeñas piezas de información que un individuo puede generar hasta sistemas complejos que unifican información de muchas organizaciones. Por lo tanto, existe una demanda de sistemas que posibiliten compartir información y coordinar el flujo de trabajo. A estos sistemas que permiten el uso de Tecnologías de Información avanzadas para ayudar a coordinar el flujo de información entre los diversos actores involucrados en un determinado proceso, se les denomina Sistemas Colaborativos. Un sistema de colaboración permite compartir información entre las personas, en una organización y también entre organizaciones.

Los Sistemas colaborativos o Sistemas Cooperativos, son un tipo de Sistema de Información, y nacieron de la necesidad de realizar actividades que requieren trabajar en grupo. Es así como se crearon software orientados a la colaboración, los Groupware y Workflow.

Los Groupware son sistemas computacionales que permiten a un grupo de personas trabajar en conjunto en una tarea en común. Por lo tanto un groupware tiene como características que sea un ambiente de colaboración, con la información en un solo lugar, y que se pueda interactuar con los usuarios. Pueden ser además en sincrónico, es decir que se envían mensajes en tiempo real, como es sesiones de chat o pizarras compartidas, o asíncrono, como es el caso de los blogs y correos electrónicos. Un ejemplo de groupware es Moodle, el cual es una aplicación educativa que permite compartir recursos y administrar cursos.

Los workflow son sistemas que autorizan e integran los procesos de negocios de una empresa, de acuerdo a determinadas estrategias. Algunas de sus actividades pueden ser asignar tareas, avisar de tareas pendientes, automatizar secuencias de negocios y optimizarlas entre otras. La principal diferencia de un workflow con respecto a un groupware es que el primero no necesariamente implica colaboración de otras personas, sino que se puede utilizar de forma individual. Un ejemplo de un sistema workflow es FlowMind, el cual permite administrar recursos, ver las diferentes etapas de un determinado proceso, como mejorar los procesos, entre otras funcionalidades, todo de forma gráfica. (Universidad Tecnológica de la Mixteca, 2002).

La ineficiencia en el manejo de la información, la mala comunicación y la ausencia del trabajo en equipo en la constructora, son problemáticos para su buen funcionamiento, para poder romper estos problemas existen los sistemas web que nos dan comunicación y los sistemas colaborativos nos demuestran el trabajo en equipo y el control del estado de cómo va la información administrada.

El proyecto utilizará un modelo de Sistema Colaborativo que beneficiará en la interacción de proyectos entre las áreas técnica y administrativa para un mejor control de los inventarios en base a lo que se necesite en los proyectos.

## 1.2 ANTECEDENTES

La Consultora Constructora Suarez antes solamente Consultora con manejo de maquinaria pesada es ahora una pequeña empresa que tiene por objeto realizar oferta de servicios de construcciones y de consultoría.

Solo cuenta con una oficina la cual está dividida en dos áreas: el área técnica y el área de administración, mismas que trabajan en conjunto para cualquiera de los dos servicios que brinda la empresa.

Los proyectos que realizan las empresas públicas tales como Semapa, la Alcaldía, y otros, necesitan de contratistas para realizar los proyectos o en otro caso consultorías tales como el diseño en detalles de proyectos, evaluación sobre factibilidad de proyectos, planificación y diseño, etc. para determinar si algún proyecto es viable o no.

Una vez obtenido un contrato de consultoría o servicio de construcción con alguna empresa pública la Constructora Suarez debe realizar un seguimiento estricto del avance del proyecto debido a que las empresas públicas requieren tener evidencias del avance de los proyectos para realizar los desembolsos en la medida en que estos avancen o bien, que se solicite un aumento de presupuesto (por ejemplo: debido al aumento del costo de materiales). De esta manera, es fundamental para la Constructora, el contar con documentación fiable del avance del proyecto, conocer la utilización de materiales en los mismos, y otras evidencias tales como documentación que verifiquen el soporte de sus actividades, esta información es almacenada en documentos Excel y en papel porque no existe un sistema.

Una de las evidencias más importantes del avance de los proyectos, son las Planillas Presupuestarias, en Bolivia, existen normas, para su aplicación, estas van desde la formulación de planillas de cálculos métricos, análisis de precios unitarios, planillas de ejecución de proyectos, entre otros. Dentro de la constructora las planillas pasan por varias manos, primero llegan a la parte administrativa ya que son quienes reciben los proyectos con los contratos que realizan a las entidades públicas, luego ciertas planillas a contabilidad como los presupuestos y otras

planillas son derivados a la parte técnica donde hacen los cálculos métricos. De la parte técnica se adjuntan papeles de cómputo de cálculos de la obra que se está realizando de acuerdo al proyecto destinado a una cierta área ver Anexo B. De la parte administrativa se adjuntan costos directos, uno de los componentes de estos costos directos son los Inventarios ver Anexo A y Anexo C. Cada vez que la parte técnica necesita material para continuar con la obra el departamento de contabilidad debe revisar los presupuestos y el inventario para hacer el pedido según la disponibilidad de recursos económicos y financieros, en caso de que no se disponga de suficiente capital disponible se debe solicitar un nuevo desembolso a la Empresa Pública, para lo cual se entrega esa la planilla al Contratante con la finalidad de obtener más recursos para la compra de material.

Luego de la compra, todo material va a almacenes con la finalidad de realizar un control de inventariado para evitar que el material desaparezca.

Al momento de realizar las planillas en Excel tanto de la parte técnica como la parte administrativa, se confunden los inventarios y la planilla de costos generales porque la información es gestionada de manera aislada a nivel de cada área de trabajo, aparte de que tanto la parte administrativa como la contable no tienen un control adecuado de Inventarios que actualmente solo es pasado en papel escrito a mano que es lo que necesitan para la obra de la parte técnica a la administrativa y luego se registra en un documento Excel, debido a que la empresa, que es pequeña y está en crecimiento, no cuenta con un Sistema automatizado de Control de Inventarios y no tiene una organización sistemática en la parte de planillas para un mejor control de los proyectos.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

#### **1.3.1 Identificación del problema**

El almacenamiento de información en tablas impresas, el uso de documentos Excel que contienen información sobre los proyectos, y demás información, conllevan a la inoportuna comunicación y organización que ocasiona demora y confusión al

momento de elaborar las planillas de inventarios y la planilla de seguimiento de avance ya que la información registrada tiene posibles errores de incoherencia.

#### **1.3.1.1 Identificación de la situación problemática**

- El almacenamiento de información en planillas manuales y archivos Excel ocasionan que la búsqueda de datos de un proyecto sea compleja y tediosa.
- Los procesos manuales actualmente empleados generan pérdida de tiempo en el registro de nueva información de proyectos en las planillas utilizadas.
- La comunicación inoportuna entre las áreas técnicas y administrativa genera demora.
- El desconocimiento de los estados de los proyectos provoca confusión en el registro de datos de los proyectos
- La información gestionada de manera aislada solo a nivel de cada área de trabajo provoca demora en tiempo de entrega de documentación de planillas.
- La ineficiencia del manejo de la información de cada proyecto provoca retrasos en los desembolsos de capital para continuar con el proyecto.

#### **1.3.1.2 Identificación de las causas**

- El almacenamiento de información en planillas manuales, archivos Excel y archivos de Word.
- Los procesos manuales actualmente empleados.
- La comunicación inoportuna entre las áreas técnicas y administrativa.
- El desconocimiento de los estados de los proyectos.
- La información gestionada de manera aislada solo a nivel de cada área de trabajo.
- La ineficiencia del manejo de la información de cada proyecto.

### **1.3.2 Formulación del problema**

La inoportuna comunicación entre las áreas técnica y administrativa y el ineficiente manejo de información de proyectos basados en el uso de tablas de Excel empleados en la gestión de proyectos provocan demora en la búsqueda de planillas de costos, pérdida de información registrada y desconocimiento del estado de los proyectos.

### **1.3.3 Análisis Causa-efecto**

#### **1.3.3.1 Causas**

- La inoportuna comunicación entre las áreas técnica y administrativa y el ineficiente manejo de información de proyectos basados en el uso de tablas de Excel.

#### **1.3.3.2 Efecto**

- Demora en la búsqueda de planillas de costos, redundancia de información registrada y desconocimiento del estado de los proyectos.

## **1.4 OBJETIVOS Y ACCIONES**

### **1.4.1 Objetivo General**

Desarrollar un sistema web colaborativo de control de inventario de proyectos para coadyuvar en el abastecimiento de material y a la presentación de informes para la solicitud de desembolsos.

### **1.4.2 Objetivos Específicos y Acciones**

- Analizar los procesos de planillas del sector de costos directos y diseñar el modelado de negocio actual
- Diseñar el modelado de negocio alternativo.
- Diseñar e implementar módulo de gestión de usuarios para la constructora.
- Diseñar e implementar un módulo de Administración de proyectos en general.



- Desarrollar el módulo de solicitud de compra y pedidos de material.
- Desarrollar el módulo de movimientos de inventarios.
- Diseñar e implementar el módulo colaborativo del sistema para generar el seguimiento de la información que fluye de un personal a otro.
- Realizar pruebas al sistema final.

**Tabla 1:** Objetivos Específicos y Acciones

OBJETIVOS ESPECIFICOS	ACCIONES
Analizar los procesos de planillas del sector de costos directos y diseñar el modelado de negocio actual	<ul style="list-style-type: none"> <li>• Recopilar información relacionada con el seguimiento de proyectos empleando Sistemas Colaborativos.</li> <li>• Elaborar entrevista con el personal encargado de los registros de los procesos de planilla.</li> <li>• Elaborar el diagrama de flujo que represente el modelado de negocio actual de los procesos de movimiento de planilla.</li> <li>• Realizar la verificación del modelado de negocio con los usuarios involucrados.</li> </ul>
Diseñar el modelado de negocio alternativo.	<ul style="list-style-type: none"> <li>• Identificar las falencias de los procedimientos manuales que se realizan en la constructora.</li> <li>• Elaborar un diagrama de flujo el cual refleje el modelado de negocio alternativo de los procesos de recorrido de planillas referidos a los inventarios.</li> <li>• Seleccionar la metodología de desarrollo de Software.</li> <li>• Planificación de desarrollo de software en base a la metodología seleccionada.</li> </ul>

<p>Diseñar e implementar módulo de gestión de usuarios para la Constructora</p>	<ul style="list-style-type: none"> <li>• Seleccionar el lenguaje de programación y sistema de gestor de base de datos.</li> <li>• Realizar análisis de requerimientos para el módulo de usuarios.</li> <li>• Identificar actores y objetivos involucrados en el sistema.</li> <li>• Elaborar los diagramas de casos de uso, de colaboración y de clases.</li> <li>• Diseñar la base de datos del sistema.</li> <li>• Codificar módulo de gestión de usuarios.</li> </ul>
<p>Diseñar e implementar un módulo de Administración de proyectos en general.</p>	<ul style="list-style-type: none"> <li>• Identificar actores y objetivos involucrados.</li> <li>• Elaborar los diagramas de colaboración.</li> <li>• Aumentar tablas del módulo de administración de proyectos, gestión de usuarios técnicos y administrativos a la base de datos.</li> <li>• Determinar entidades públicas y entidades que hayan solicitado los servicios de la constructora.</li> <li>• Codificar módulo de administración de proyectos.</li> </ul>
<p>Desarrollar el módulo de solicitud de pedidos y compra de material.</p>	<ul style="list-style-type: none"> <li>• Realizar análisis de requerimientos para el módulo de solicitud y pedidos de material</li> <li>• Elaborar los diagramas de colaboración.</li> <li>• Desarrollar proveedores de material.</li> </ul>
<p>Desarrollar el módulo de movimientos de inventarios.</p>	<ul style="list-style-type: none"> <li>• Analizar teoría de Sistema de inventarios y escoger un método.</li> <li>• Realizar análisis de requerimientos para el módulo de movimientos de inventarios</li> <li>• Elaborar los diagramas de colaboración.</li> <li>• Realizar formularios específicos para los inventarios.</li> </ul>

	<ul style="list-style-type: none"> <li>• Realizar pruebas funcionales.</li> </ul>
Diseñar e implementar el módulo Colaborativo del sistema para generar el seguimiento de la información que fluye de un personal a otro.	<ul style="list-style-type: none"> <li>• Realizar análisis de requerimientos para el módulo colaborativo del sistema</li> <li>• Elaborar los diagramas de colaboración.</li> <li>• Identificar los estados en los que se encuentran los proyectos</li> <li>• Codificar módulo de seguimiento</li> </ul>
Realizar pruebas al sistema final.	<ul style="list-style-type: none"> <li>• Elaborar escenarios de prueba</li> <li>• Aplicar las pruebas seleccionadas</li> </ul>

**Fuente:** Elaboración Propia

## **1.5 JUSTIFICACION**

### **1.5.1 Justificación Técnica**

Los sistemas web usados en empresas con un funcionamiento mediante internet o intranet facilitan el modo de acceso de los empleados de la empresa, los sistemas colaborativos que nacieron a partir de los sistemas de información beneficia a la empresa al realizar actividades de trabajo en grupo reduciendo tiempo y mejorando la comunicación del personal de la empresa.

### **1.5.2 Justificación Económica**

El sistema beneficiará a la Constructora evitando el uso excesivo de papel y en paralelo la pérdida de tiempo empleado en la impresión de planillas. Así mismo, al tener un sistema colaborativo, se evitara perdida de material o ítems que se necesiten para las obras.

### **1.5.3 Justificación Operativa**

El sistema desarrollado beneficiará a la constructora de manera que se podrá manejar los inventarios de forma oportuna y ordenada, para un mejor control de los costos directos de material que necesite la empresa para realizar los proyectos, además de tener actualizado los estados en los que se encuentran los proyectos y la comunicación oportuna entre las áreas técnica y administrativa.

## **1.6 ALCANCE**

### **1.6.1 Alcance Temático**

El sistema propuesto se desarrollará haciendo uso de Sistemas Colaborativos dentro de lo que es Ingeniería de Software aplicando análisis, diseño de sistemas, modelos de desarrollo de software, Sistema de gestión de Base de datos, Servicios Web.

### **1.6.2 Alcance Institucional**

El sistema será utilizado en las dos áreas de la Constructora, en la parte técnica y en la administrativa, ya que el proceso de planilla de Inventario comienza en el área técnica y acaba en la administrativa.

### **1.6.3 Alcance Temporal**

El tiempo de vida promedio del sistema desarrollado será de uno 1 año, será desarrollado en la presente gestión 2015 y primer semestre del 2016, dando un tiempo de vida promedio de 7 años gracias a la tecnología utilizada y al crecimiento de la empresa, usuarios e información de proyectos.

### **1.6.4 Alcance Proyecto**

El sistema concluido realizará el control de inventarios tanto como pedidos y compras de material para la empresa, podrá ofrecer una mejor comunicación, entendimiento y trabajo en equipo en la empresa

### **1.6.5 Alcance Geográfico**

El sistema tendrá un alcance de solo la ciudad de Cochabamba ya que la Constructora es una empresa nueva y está en crecimiento y solo cuenta con una sola oficina ubicada en el departamento de Cochabamba.

Además abarcara las áreas técnica y administrativa de la constructora ya que es necesario ambas áreas para el proceso de las planillas de costos directos de Inventarios.

## **1.7 HIPOTESIS**

El desarrollo de un sistema web colaborativo de control de inventario de proyectos permitirá conocer el estado en el que se encuentra el inventario, reducción de tiempo en la elaboración de planillas de inventarios registrada y el riesgo de la pérdida de información.

### **1.7.1 Análisis de Variables**

#### **1.7.1.1 Variables Independientes**

Sistema web colaborativo de control de inventario de proyectos.

#### **1.7.1.2 Variables Dependientes**

- Conocer el estado de los inventarios de proyectos.
- Tiempo en la elaboración de planillas de inventarios
- Información extraviada o perdida.

### **1.7.2 Definición Conceptual**

#### **1.7.2.1 Variables Independientes**

Sistema web colaborativo de control de inventario de proyectos para coadyuvar al abastecimiento y a la presentación de informes para la solicitud de desembolsos.

Es un sistema que permitirá reducir los trámites y procesos de las planillas de inventarios. El sistema colaborativo ayudará a tener una eficiente comunicación en la constructora y el conocimiento del estado en que se encuentran los proyectos.

#### **1.7.2.2 Variables Dependientes**

- Estado de los inventarios de proyectos.

Permite tener la información en la que se encuentran los proyectos, en qué estado se encuentran los inventarios.

- Tiempo en los trámites de las planillas de inventarios.

Permite una optimización de tiempo en el trámite de las planillas de inventarios en el sector que se necesite la planilla tanto como en la parte técnica o administrativa.

- Riesgo de pérdida de información de las planillas de inventarios.

Permite tener toda la información de las planillas en una base de datos, sin correr el riesgo de pérdida de documentos en papel.

### 1.7.3 Operativización de variables

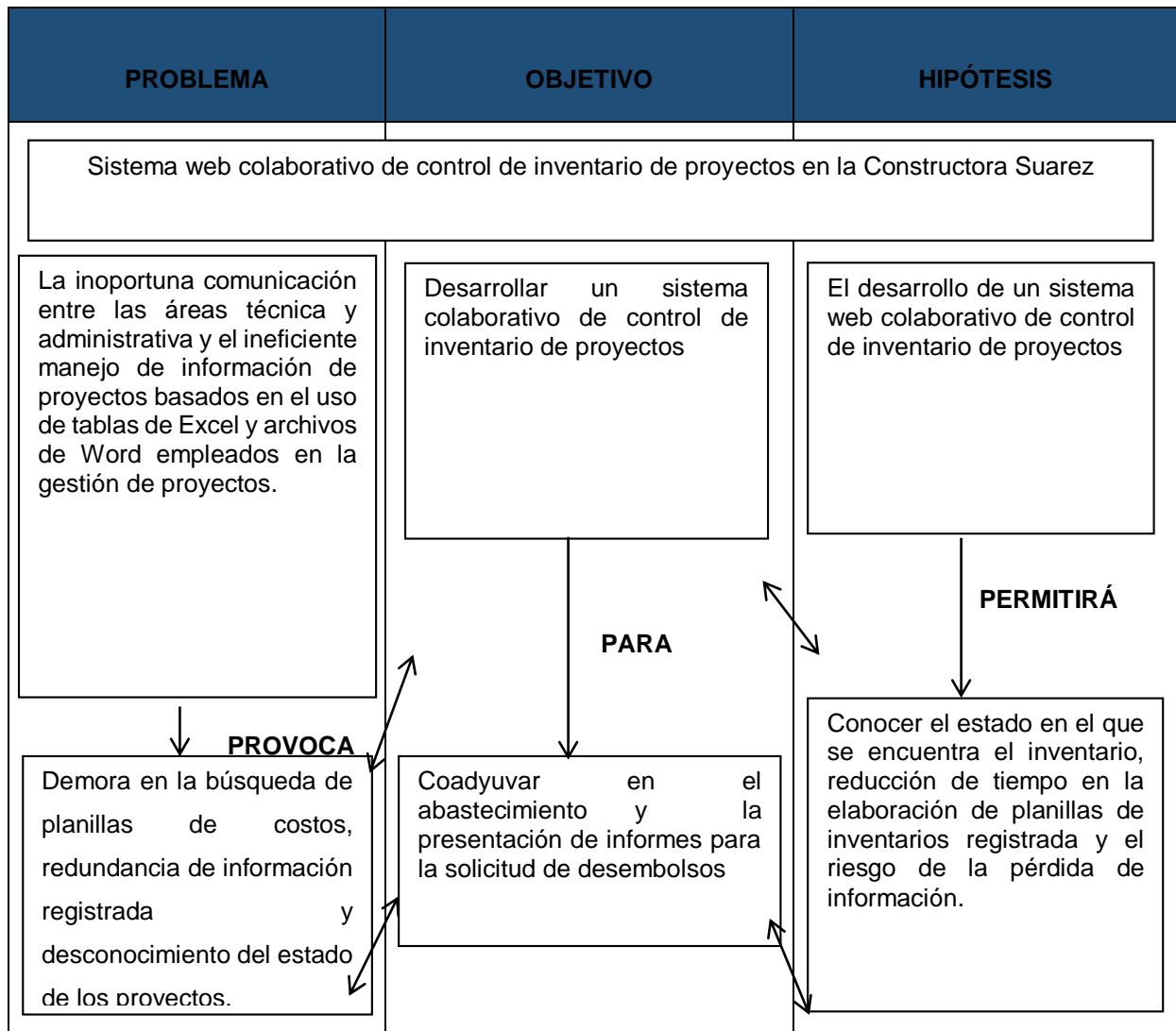
**Tabla 2:** Operativización de variables

VARIABLE	DIMENSIÓN	INDICADOR
<u>Variable Independiente</u>  Sistema web de control de inventario de proyectos aplicando un modelo de Sistema Colaborativo para coadyuvar al abastecimiento y a la presentación de informes para la solicitud de desembolsos.	  Sistema web desarrollado.	  Comparación de beneficios entre el sistema actual y el proyecto.
<u>Variable Dependiente</u>  Estado de los inventarios de proyectos.	 Reportes para determinar estado de inversión de proyecto	 Numero de pasos empleados
<u>Variable Dependiente</u>  Tiempo en los tramites de las planillas de inventarios	 Tiempo	 Minutos.
<u>Variable Dependiente</u>  Riesgo en pérdida y duplicidad de información de inventarios.	 Matriz de riesgos	 Nivel de riesgo

**Fuente:** Elaboración Propia

## 1.8 MATRIZ DE CONSISTENCIA

**Figura 1:** Matriz de Consistencia



**Fuente:** Elaboración Propia



## **2 MARCO TEÓRICO**

### **2.1 TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN.**

#### **2.1.2 Metodología de la investigación cuantitativa**

Para averiguar información sobre alguna investigación o de algún proyecto se necesitan recolectar pruebas para que respondan a nuestras preguntas. Con estos analizamos los datos utilizados elementos del análisis estadístico, tales como: medias, desviaciones, coeficientes y diversas pruebas psicométricas (Zapata, 2005).

En ciertos proyectos de investigación es necesario tener una guía de trabajo de campo; para su elaboración se pueden emprender los siguientes pasos, que a la vez permiten contar con la estrategia aplicable al trabajo de campo con metodología cuantitativa:

- a) Estudio previo o sondeo.
- b) Diseño de la muestra.
- c) Preparación de los materiales de recolección de datos.
- d) Equipo de trabajo necesario: grabadoras, cámaras fotográficas, filmadoras.
- e) Selección y entrenamiento de personal.
- f) Revisión de las etapas anteriores y prueba experimental piloto.
- g) Recolección de datos, ya sea primarios o secundarios.
- h) Análisis de los datos, conclusiones y generalizaciones.
- i) Elaboración del informe del trabajo de campo.
- j) Estimación del personal necesario y costos.
- k) Aspectos administrativos: aquí se deben ubicar los aspectos administrativos del proyecto que son importantes para obtener financiación total o parcial del proyecto. (Zapata, 2005)

##### **2.1.2.1 Técnicas de Recolección de datos cuantitativos**

Las técnicas más utilizadas son: el experimento, la encuesta o el sondeo y el análisis de contenido. (Zapata, 2005)

#### a. Experimentación

Está relacionado íntimamente con la observación; por tanto, la observación del fenómeno que deseamos investigar es previa a la experimentación. Por medio de ella hacemos una detallada descripción del fenómeno sustentada en los conceptos teóricos u, una vez que tenemos una exhaustiva descripción de él, empezamos a experimentar. (Zapata, 2005)

Las fases del experimento son por lo común:

- La elaboración de la hipótesis o de las diferentes hipótesis
- El análisis de las hipótesis
- La predicción de sus consecuencias
- La programación del experimento para comprobar las hipótesis
- El diseño del experimento
- La ejecución del experimento o experimentos
- La recolección de los resultados
- La confrontación de los resultados con las predicciones
- La interpretación de las conclusiones

#### b. Encuestas y Cuestionarios

Por medio de las encuestas se trata de recabar información sobre un sector denominado muestra para inferir el comportamiento del universo que se desea indagar. La encuesta puede definirse como un conjunto de técnicas destinadas a reunir, de manera sistemática, datos sobre determinado tema o temas relativos a una población, a través de contactos directos con los individuos o grupo de individuos que integran la población estudiada. Existen dos tipos de encuestas: las basadas en un relevamiento de todos los individuos o unidades de la población y las que solamente trabajan con datos de una pequeña parte de la población total, por medio de procedimientos especiales para asegurar que tal parte sea representativa del todo.

El cuestionario es un instrumento de investigación que consiste en una serie de preguntas y otras indicaciones con el propósito de obtener información de los consultados. Aunque a menudo están diseñados para poder realizar un análisis estadístico de las respuestas. El cuestionario debe incorporar y traducir los esquemas conceptuales del objeto de investigación y las hipótesis de la investigación. La elección del tipo de cuestiones que formara parte del cuestionario estará directamente asociada con el objetivo de investigación, sus dimensiones y sus relaciones. El cuestionario puede estructurarse específicamente como un instrumento de exploración de ideas sobre la vida cotidiana o algún fenómeno social. (Zapata, 2005)

En lo que respecta a las preguntas del cuestionario, deben ser construidas y estudiadas cuidadosamente para evitar lo siguiente:

- Dificultad de comprensión por parte del encuestado.
- Que incorporen la visión y tendencias del encuestador.
- Que generen reacciones negativas del encuestado.
- Contar con una correcta secuencia en la formulación de las preguntas.

c. Análisis de contenido

El análisis de contenido es una técnica de la investigación cuantitativa o experimental que se puede realizar por múltiples propósitos, pero que la base de sus procedimientos se ha diseñado para estudiar cualquier tipo de problema en el cual se trate de analizar el contenido de la comunicación.

El contenido de la comunicación resulta de gran complejidad y sus causas y efectos son muy diversos; a su vez, el análisis de contenido trata de describir las diferentes características del contenido de una comunicación en forma precisa y concisa.

## **2.2 SISTEMAS DE INVENTARIOS.**

Un sistema de inventarios es un conjunto de normas, métodos y procedimientos aplicados de manera sistemática para planificar y controlar los materiales y productos que se emplean en una organización. Este sistema puede ser manual o

automatizado. Para el control de los costos, elemento clave de la administración de cualquier, existen sistemas que permiten estimar los costos de las mercancías que son adquiridas y luego procesadas o vendidas.

Varios métodos de valuación han sido desarrollados para determinar los costos de inventarios, pudiendo citarse entre los más usuales los siguientes:

- Métodos FIFO, del inglés (first in, first out), primeras entradas, primeras salidas.
- Método LIFO, del inglés (last in, first out), últimas entradas, primeras salidas.
- Método del promedio ponderado.
- Método del precio más bajo de costo o de mercado.
- Método del costo estándar. (Palenque, 2005)

### **2.2.2 Método FIFO**

Bajo este método, las mercaderías más antiguamente adquiridas, son las primeras que se venden, quedando en inventarios las de más reciente adquisición. El término FIFO quiere decir: lo primero que entra, es lo primero que sale, proviene de las primeras letras de la expresión en inglés: " first in, first out ". Con los datos ofrecidos, se puede establecer el valor del inventario final y el costo de mercaderías vendidas, ya sea por el sistema de inventario periódico o de inventario permanente.

### **2.2.3 Método LIFO**

A diferencia del anterior, cuando se emplea en método LIFO, las mercaderías ingresadas al almacén últimamente, son las primeras que salen, quedando el inventario final valorizado a los precios más antiguos. El término LIFO, significa, últimas entradas, primeras salidas y que corresponde a las primeras letras de la expresión inglesa: " last in, first out ".

Si tratamos de establecer por este método los valores del Inventario final y del costo de Mercaderías vendidas suponiendo que el control de inventarios se efectúa por inventario periódico, los valores a consignarse no serán exactos como los establecidos por el sistema de inventario permanente, a menos que se efectúe una

labor de análisis ítem por ítem, de todo el movimiento, lo que en realidad resulta impracticable y muy costoso. (Palenque, 2005)

#### **2.2.4 Método de Promedio Ponderado**

Este método supone la mezcla física de los productos determina el valor del inventario final y de los productos vendidos por el precio promedio entre una compra y otra, para que la venta siguiente se efectúe a dicho promedio. Cuando se realiza inventario periódico, la suma total de las entradas, es dividida por el número total de unidades para obtener el costo del inventario. (Palenque, 2005)

### **2.3 SISTEMAS WEB**

Los "sistemas Web" o también conocido como "aplicaciones Web" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los 'sistemas Web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares. (Baez, 2012)

Los sistemas Web se pueden utilizar en cualquier navegador Web (chrome, Firefox, Internet Explorer, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

Las aplicaciones Web trabajan con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario.

Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema.

Este tipo de diferencias se ven reflejada en los costos, en la rapidez de obtención de la información, en la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable.

Un sistema web a medida le permitirá ahorrar costos, tiempo y Recursos humanos. Por tanto, la rentabilidad de su empresa aumentará y gestionará mejor a sus clientes, proveedores, distribuidores, etc. (Baez, 2012)

Los Sistemas Web han demostrado mejores resultados para las empresas frente a los Sistemas Tradicionales Cliente/Servidor, pues le brindan beneficios que estos no pueden tales como:

- No tiene que pagar Licencias por cada Computadora con el Sistema Instalado pues está en un sólo Servidor Web.
- Facilidad para acceder al Sistema desde cualquier punto con conexión a Internet
- Puede Integrar todas sus tiendas y sucursales; manteniendo información actualizada de stock de productos y de los ingresos y los egresos.
- Le permite comunicarse con sus clientes de manera más fluída las 24 horas del día

### **2.3.1 Servicios Web**

Un Servicio Web es un componente software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. (SCS, 2008)

- Permiten exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos WEB estándar.
- Cada Servicio Web se responsabiliza de realizar un conjunto de funciones concretas y bien definidas.
- Servicios Web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

Según la W3C (World Wide Web Consortium) tiene la siguiente definición:

“Un Servicio Web (Web Service [WS]) es una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre “agentes” software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.” (SCS, 2008)

Sus puntos clave son:

- Interoperabilidad
- Uso de estándares abiertos
- Mínimo acoplamiento

➤ Interoperabilidad

Distintas aplicaciones, en lenguajes de programación diferentes, ejecutadas sobre cualquier plataforma, pueden utilizar los Servicios Web para intercambiar datos

- La interoperabilidad se consigue mediante el uso de estándares abiertos.
- Servicios Web se asientan sobre protocolos y estándares ya existentes y muy difundidos (HTTP, XML, etc.)
- Uso de protocolos específicos extensibles ⇒ no imponen restricciones sobre las aplicaciones a las que dan acceso ni sobre las tecnologías que las implementan (independencia de lenguaje y de plataforma)
- OASIS y W3C: organizaciones responsables de definir la arquitectura y estándares para los Servicios Web

➤ Pueden verse como una evolución de los mecanismos RPC

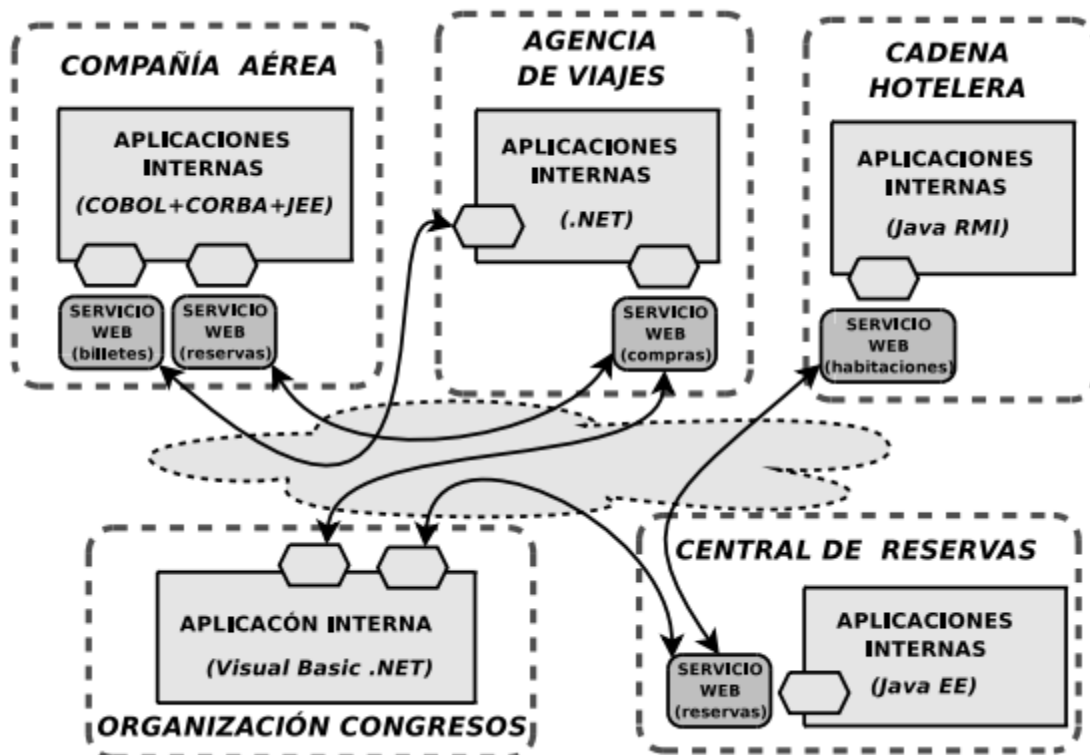
- Uso de protocolos estándar de internet (HTTP, SMTP) como mecanismo para el transporte de los mensajes, estos intercambiados se encapsulan dentro de mensajes HTTP (o SMTP) y evitan problemas con firewalls y filtrado de puertos no privilegiados Para la red el tráfico de Servicios Web es tráfico HTTP (o SMTP) normal

➤ Escenario típico

Integración de un conjunto de aplicaciones de distintas empresas/organizaciones

- Aplicaciones distribuidas convencionales se basan en el uso de un middleware común y centralizado (ORBs en CORBA, RMI en Java, etc.)
- Servicio Web permiten superar esa restricción porque no exigen middleware único común ni tampoco middleware abierto y no centralizado
- Servicios Web ofrecen un punto de entrada a los sistemas de información locales.
- Encapsulan una o más aplicaciones ofreciendo un interfaz único accesible por la Web
- Ofrecen un interfaz público y estable, independiente de su implementación concreta
- Facilitan la automatización de las interacciones entre los procesos internos de una organización con el exterior. (SCS, 2008)

**Figura 2:** Ejemplo de uso de Servicios Web



**Fuente:** (SCS, 2008)



## 2.4 SISTEMAS COLABORATIVOS.

A partir de la llegada de las computadoras personales al ambiente empresarial se inició una nueva revolución. Uno de los cambios más drásticos fue el incremento en la velocidad de procesamiento de los datos.

Desde los años 60s se tenía la necesidad de compartir recursos de cómputo, como la memoria, las unidades de almacenamiento y principalmente el procesador; pero no se compartía la información, ya que existían diferentes barreras que lo impedían, como las distancias entre oficinas o los diferentes sistemas operativos. Con la llegada de Internet algunas barreras se rompieron, con este avance tecnológico se logró compartir información pero no se podían realizar las actividades que necesitan colaboración, a partir de esta necesidad se comenzó a trabajar en un nuevo tipo de tecnología: el software colaborativo. (Chaffey, 1998)

Según (et al, 1991) se pueden definir los sistemas colaborativos como: **“Sistemas basados en computadoras que soportan grupos de personas involucradas en una tarea común (u objetivo) y que proveen de una interfaz a un ambiente compartido”** (et al, 1991) citado por Luis Mariano Bibbó en **“Modelado de Sistemas Colaborativos”**.

Bibbó hace una clara diferenciación entre un sistema multiusuario y uno colaborativo haciendo énfasis en dos conceptos clave. Las nociones de una “tarea común” y “ambiente compartido” son cruciales en la definición de sistema colaborativo. A partir de ellas, se excluyen, por ejemplo, a los sistemas multiusuarios, donde los usuarios comparten algunos recursos – como espacio en disco, tiempo de procesamiento, memoria, etc. – pero no comparten en realidad una tarea común (Bibbó, 2010) .

A lo largo de este trabajo, se emplearán términos específicos para referirse a ciertos aspectos relevantes, los mismos pueden ser vistos en el ANEXO E.

Para soportar colaboración, un sistema de información requiere ser capaz de proveer mecanismos para:

- Comunicación
- Administración de contenidos
- Workflow

Algunos autores dividen el software colaborativo en dos grandes grupos: el Groupware y el Workflow:

#### **2.4.1 GROUPWARE**

El Groupware es un tipo de software colaborativo que ayuda a grupos de trabajo a realizar sus actividades a través de una red. Formalmente se puede definir al groupware de la siguiente manera:

“Sistemas basados en computadoras que apoyan a grupos de personas que trabajan en una tarea común y que proveen una interfaz para un ambiente compartido (Chaffey, 1998)”

Las características más importantes de los groupware son:

- Proveer de un ambiente de colaboración, en el que realmente se perciba que el trabajo en grupo se lleva a cabo.
- Mantener la información en un solo sitio común para todos los miembros
- Interactuar con otros usuarios, de forma escrita, voz o video.

Además si se necesita tomar una decisión urgente y las personas se encuentran en diferentes partes del mundo, el groupware resuelve este problema ya que se pueden tomar decisiones sin importar la distancia entre cada miembro del equipo.

Es por esto que los groupware deben proporcionar tres funciones esenciales dentro de un grupo, llamadas las tres C's:

- La Comunicación, es la función más importante del groupware, ya que es el medio en que la información es compartida.
- La Colaboración, utilizada para unir cooperación y resolver problemas de negocios o alguna actividad empresarial. Proporciona la ventaja de resolver problemas de las asambleas tradicionales como: lugar y tiempo para la

realización de la misma o la disponibilidad de información. Además de mejorar la eficiencia en la toma de decisiones con la contribución de todos los miembros del grupo.

- La Coordinación, es la acción de asegurar que el equipo está trabajando eficientemente y en conjunto para alcanzar una meta. Esto incluye la distribución de tareas y revisión de su ejecución.

Es muy confuso distinguir entre groupware y workflow, esto sugiere desde que los workflow's son considerados como una función o un subconjunto de los groupware. Una definición estricta dice que todos los tipos de groupware deben incluir un elemento de colaboración, pero esto no es necesario en los sistemas workflow, algunas veces son utilizados para tareas individuales que no están directamente en colaboración.

## **2.4.2 WORKFLOW**

Los Workflows son sistemas que ayudan a administrar y automatizar procesos de negocios. Un workflow puede ser descrito como el flujo y control en un proceso de negocio.

La WfMC (Workflow Management Coalition) define a los workflows como:

“La automatización de un proceso de negocio, total o parcial en la cual documentos, información o tareas son pasadas de un participante a otro a los efectos de su procesamiento, de acuerdo a un conjunto de reglas establecidas.”

También definen lo que es un proceso de negocio:

“Es un conjunto de uno o más procedimientos o actividades directamente ligadas, que colectivamente realizan un objetivo del negocio normalmente dentro del contexto de una estructura organizacional que define roles funcionales y relaciones entre los mismos”

Entre los ejemplos de proceso de negocios tenemos: procesamiento de órdenes, reportes de gastos, procedimientos de producción, etc.

Los workflows son solo un camino para la información, para reducir tiempo, dinero y esfuerzo en la ejecución de un proceso de negocio.

Sus funciones más comunes son:

- Asignación de tareas al personal
- Aviso al personal de tareas pendientes
- Permitir la colaboración en la tareas comunes
- Optimización de recursos humanos y técnicos, alienándolos a la estrategia de la empresa.
- Automatización de las secuencias de los procesos de negocio y optimización de las mismas.
- Agilización de los procesos de negocio y como resultado un mejor servicio al cliente.
- Control y seguimiento de dichos procesos.

Para poder identificar cada elemento dentro de cada workflow se puede utilizar el modelo de componentes de proceso de negocio.

**Figura 3:** Elementos clave de un proceso de negocio.



**Fuente:** (Chaffey, 1998)

## 2.5 INGENIERIA DE SOFTWARE.

La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la

especificación del sistema, hasta el mantenimiento de este después de que se utiliza. En esta definición, existen dos frases clave:

1. Disciplina de la ingeniería: Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos.
2. Todos los aspectos de producción de software: La ingeniería del software no solo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software. (Ian Sommerville, 2005)

### **2.5.1 Unified Modeling Language (UML)**

El Unified Modeling Language (UML) es un modelo para la construcción de software orientado a objetos que ha sido propuesto como estándar de ISO por el OMG. Consta de un conjunto de tipos de diagramas interrelacionados, dentro de los cuales se utilizan elementos del modelo, que sirven para describir distintos aspectos de la estructura y la dinámica del software. (Campderrich Falgueras, 2003)

UML es el resultado de una cierta unificación de los modelos utilizados en tres métodos preexistentes de desarrollo de software orientado a objetos hechos por sus autores en colaboración. Estos métodos son los siguientes:

- El método de Grady Booch;
- El OMT, de Jim Rumbaugh y otros;
- El OOSE, de Ivar Jacobson

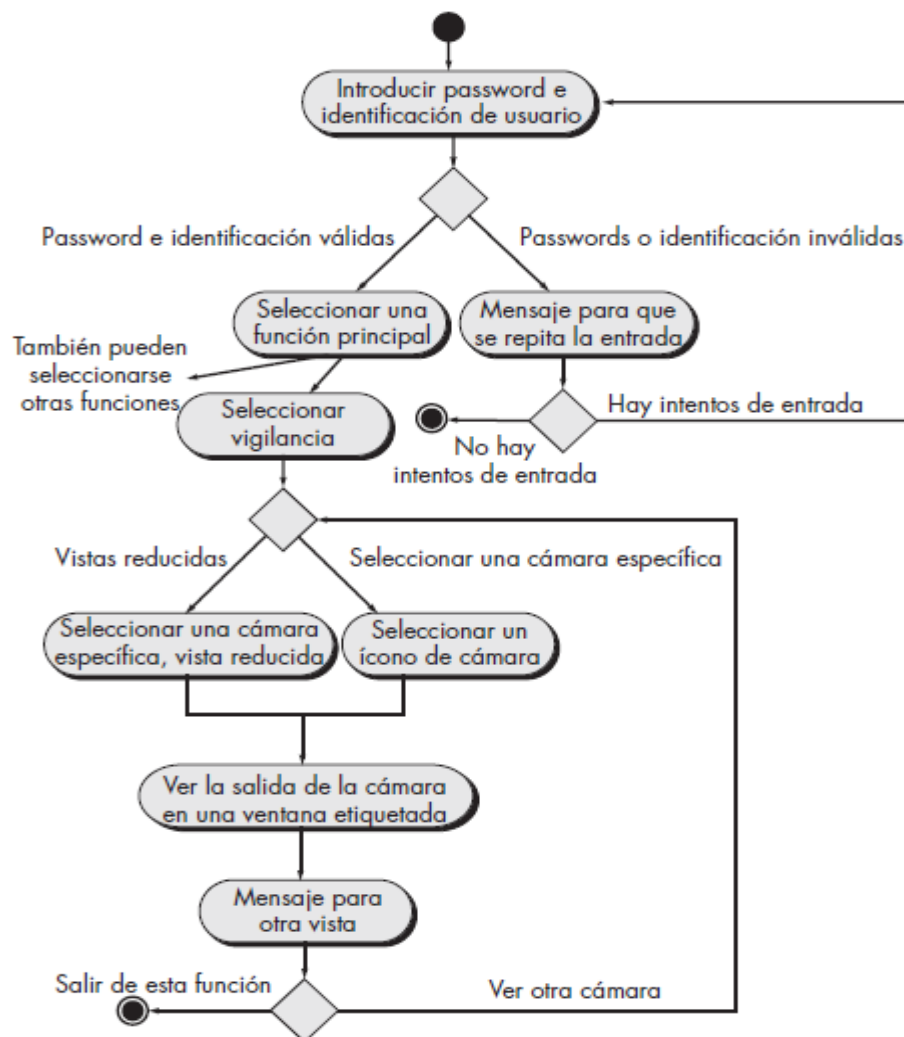
#### **2.5.1.1 Desarrollo de un diagrama de actividades**

El diagrama de actividad UML enriquece el caso de uso al proporcionar una representación gráfica del flujo de interacción dentro de un escenario específico. Un diagrama de actividades es similar a uno de flujo, y utiliza rectángulos redondeados

para denotar una función específica del sistema, flechas para representar flujo a través de éste, rombos de decisión para ilustrar una ramificación de las decisiones (cada flecha que salga del rombo se etiqueta) y líneas continuas para indicar que están ocurriendo actividades en paralelo.

Por ejemplo, un usuario quizá sólo haga algunos intentos de introducir su **identificación** y **password**. Esto se representa por el rombo de decisión debajo de “Mensaje para que se repita la entrada”. (S. Pressman, 2010)

**Figura 4:** Diagrama de actividades



**Fuente:** (S. Pressman, 2010)

### 2.5.1.2 Diagrama de Casos de Uso

El modelado de casos de uso es un método orientado a los usuarios para identificar necesidades funcionales de un nuevo sistema de información. El modelado de casos de uso es una técnica que permite modelar las funciones de un sistema en términos de eventos, de quien inicia los eventos y de cómo responde el sistema a estos eventos. (Fernández Alarcón, 2006)

Un diagrama de casos de uso representa las interacciones entre el sistema y los sistemas externos y los usuarios. En otras palabras, describe gráficamente quien utiliza el sistema y la forma en que los usuarios esperan interactuar con el sistema.

Los diagramas de caso de usos están compuestos por tres elementos:

- Casos de uso

Un caso de uso representa un objetivo sencillo de un sistema y describe una secuencia de actividades y de interacciones con el usuario para alcanzar el objetivo. Los casos de uso proporcionan una sólida base para el desarrollo de manuales y sistemas de ayuda para los usuarios, así como para la creación de documentación sobre el desarrollo del sistema.

**Figura 5:** Casos de uso.



**Fuente:** (Fernández Alarcón, 2006)

- Actores

Un actor es un elemento externo que interactúa con el sistema de información. Los actores son los encargados de iniciar los casos de uso que representan las actividades que el sistema de información debe realizar.

Un actor no equivale a un individuo o un sistema de información externo. Un actor representa un papel, mientras que un individuo o sistema externo puede representar uno o varios papeles al mismo tiempo. (Fernández Alarcón, 2006)

**Figura 6:** Actores con caso de uso.

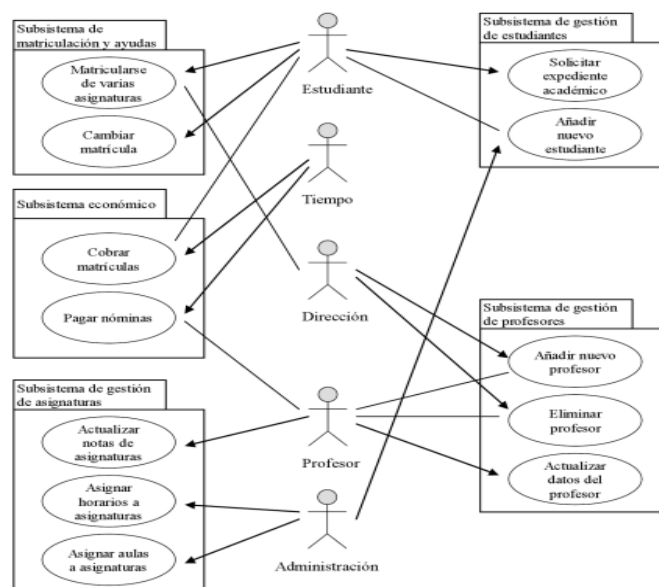


**Fuente:** (Fernández Alarcón, 2006)

- Relaciones

En un diagrama de casos de uso, los actores y los casos se interconectan a través de diversos tipos de relaciones. Las relaciones se representan a través de líneas, y su significado depende del tipo de línea y los elementos que interconectan. (Fernández Alarcón, 2006)

**Figura 7:** Diagrama de casos de uso.



**Fuente:** (Fernández Alarcón, 2006)



### 2.5.1.3 Diagrama de Colaboración

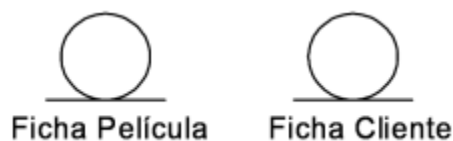
Para analizar los casos de uso se procede a identificar clases de análisis y a definir para cada caso de uso, las interacciones entre objetos de análisis. (Fernando Alono, 2005)

El primer paso a realizar en este análisis es Identificar las Clases de Análisis de entidad, de interfaz y de control de cada caso de uso. No es difícil hacerlo si se siguen las reglas y se utilizan los Atributos del Caso de Uso como base:

#### a) Clases Entidad

Las clases Entidad se refieren a información que se manipula en la relación del caso de uso. Hay que tener cuidado con separar dicha información de la que puede estar relacionada con las clases interfaz o control.

**Figura 8:** clases entidad.

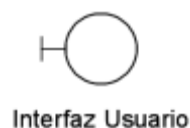


**Fuente:** (Fernández Alarcón, 2006)

#### b) Clases Interfaz (boundary)

Identificar una clase interfaz por cada actor humano, y que sea esa clase la que representa la ventana o lo que sea su interfaz. Si dicho actor tiene ya una clase interfaz habría que estudiar la posibilidad de reutilizarla para minimizar el número de ventanas con las que interactúa el actor.

**Figura 9:** Clase Interfaz de usuario.



**Fuente:** (Fernández Alarcón, 2006)

### c) Clases Control

Identificar una clase de control responsable de llevar el control y la coordinación de la realización del caso de uso. Una clase de control puede actuar para varios casos de uso y puede ocurrir que el control se encapsule en una clase interfaz, especialmente si el actor maneja gran parte del control.

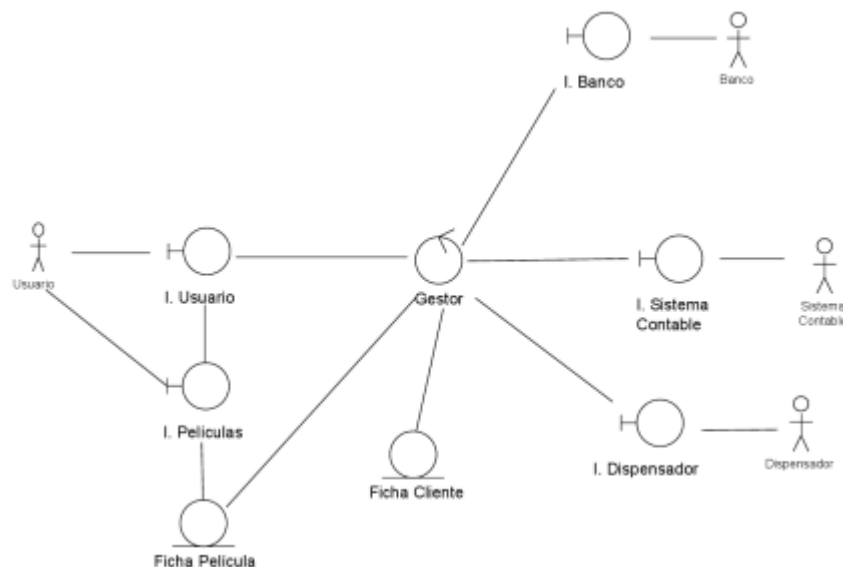
**Figura 10:** Clase Interfaz responsable.



**Fuente:** (Fernández Alarcón, 2006)

A partir de las clases identificadas se debe crear un Diagrama de Clases de Análisis que muestre las relaciones que se utilizan en la realización del caso de uso.

**Figura 11:** Clase Interfaz.



**Fuente:** (Fernández Alarcón, 2006)

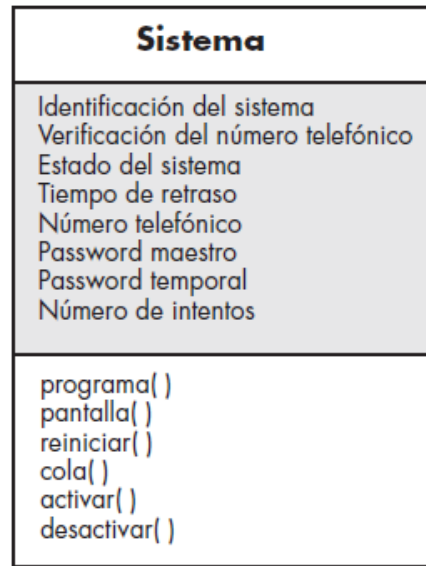
#### 2.5.1.4 Diagrama basado en clases

El modelado basado en clases representa los objetos que manipulará el sistema, las operaciones (también llamadas *métodos* o *servicios*) que se aplicarán a los objetos para efectuar la manipulación, las relaciones (algunas de ellas jerárquicas) entre los objetos y las colaboraciones que tienen lugar entre las clases definidas. Los elementos de un modelo basado en clases incluyen las clases y los objetos, atributos, operaciones, modelos clase-responsabilidad-colaborador (CRC), diagramas de colaboración y paquetes. En las secciones siguientes se presenta una serie de lineamientos informales que ayudarán a su identificación y representación. (S. Pressman, 2010)

Los *atributos* describen una clase que ha sido seleccionada para incluirse en el modelo de requerimientos.

Las *operaciones* definen el comportamiento de un objeto. Aunque existen muchos tipos distintos de operaciones, por lo general se dividen en cuatro categorías principales: 1) operaciones que manipulan datos en cierta manera (por ejemplo, los agregan, eliminan, editan, seleccionan, etc.), 2) operaciones que realizan un cálculo, 3) operaciones que preguntan sobre el estado de un objeto y 4) operaciones que vigilan un objeto en cuanto a la ocurrencia de un evento de control. Estas funciones se llevan a cabo con operaciones sobre los atributos o sobre asociaciones de éstos. Por tanto, una operación debe tener “conocimiento” de la naturaleza de los atributos y de las asociaciones de la clase. (S. Pressman, 2010)

**Figura 12:** Diagrama de Clase.



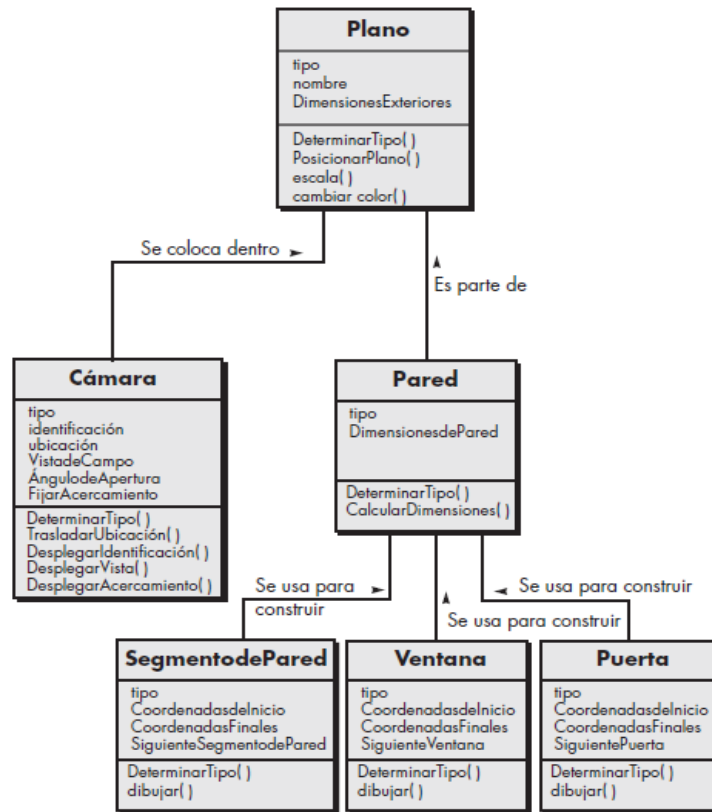
**Fuente:** (S. Pressman, 2010)

- Modelado clase-responsabilidad-colaborador (CRP)

*El modelado clase-responsabilidad-colaborador (CRC)* [Wir90] proporciona una manera sencilla de identificación y organización de las clases que son relevantes para los requerimientos de un sistema o producto. Ambler [Amb95] describe el modelado CRC en la siguiente forma:

Un modelo CRC en realidad es un conjunto de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha, los colaboradores. (S. Pressman, 2010)

**Figura 13:** Diagrama de Clases.



**Fuente:** (S. Pressman, 2010)

## 2.5.2 Modelos del Proceso

En el proceso como la colección de actividades de trabajo, acciones y tareas que se realizan cuando va a crearse algún producto terminado. Cada una de las actividades, acciones y tareas se encuentra dentro de una estructura o modelo que define su relación tanto con el proceso como entre sí. (S. Pressman, 2010)

### 2.5.2.1 Proceso Ágil

Cualquier proceso del software ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave acerca de la mayoría de proyectos de software:

1. Es difícil predecir qué requerimientos de software persistirán y cuáles cambiarán. También es difícil pronosticar cómo cambiarán las prioridades del cliente a medida que avanza el proyecto.

2. Para muchos tipos de software, el diseño y la construcción están imbricados. Es decir, ambas actividades deben ejecutarse en forma simultánea, de modo que los modelos de diseño se prueben a medida que se crean. Es difícil predecir cuánto diseño se necesita antes de que se use la construcción para probar el diseño.

3. El análisis, el diseño, la construcción y las pruebas no son tan predecibles como nos gustaría (desde un punto de vista de planeación).

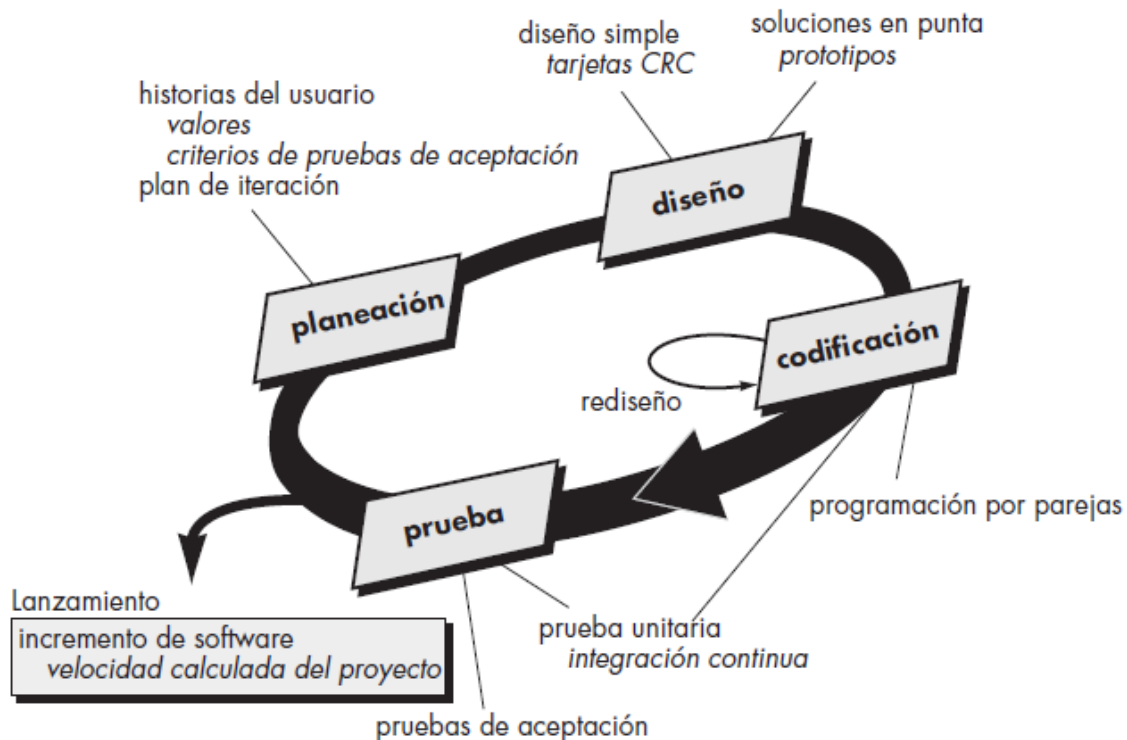
Dadas estas tres suposiciones, surge una pregunta importante: ¿cómo crear un proceso que pueda manejar lo *impredecible*? La respuesta, como ya se dijo, está en la adaptabilidad del proceso (al cambio rápido del proyecto y a las condiciones técnicas). Por tanto, un proceso ágil debe ser *adaptable*. (S. Pressman, 2010)

#### **2.5.2.2 Programación Extrema (XP)**

Beck define un conjunto de cinco *valores* que establecen el fundamento para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP.

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. La figura ilustra el proceso XP y resalta algunas de las ideas y tareas clave que se asocian con cada actividad estructural. En los párrafos que siguen se resumen las actividades de XP clave. (S. Pressman, 2010)

**Figura 14:** Proceso XP.



**Fuente:** (S. Pressman, 2010)

Los nuevos modelos y métodos de proceso han motivado análisis provechosos y en ciertas instancias debates acalorados. La programación extrema desencadena ambos. En un libro interesante que examina la eficacia de XP, Stephens y Rosenberg afirman que muchas prácticas de XP son benéficas, pero que otras están sobreestimadas y unas más son problemáticas.

Los autores sugieren que la naturaleza codependiente de las prácticas de XP constituye tanto su fortaleza como su debilidad. Debido a que muchas organizaciones adoptan sólo un subconjunto de prácticas XP, debilitan la eficacia de todo el proceso. Los defensores contradicen esto al afirmar que la XP está en evolución continua y que muchas de las críticas que se le hacen han llevado a correcciones conforme maduran sus prácticas. (S. Pressman, 2010)

➤ Ventajas

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.
- Solución de errores de programas
- Versiones nuevas
- Implementa una forma de trabajo donde se adapte fácilmente a las circunstancias

➤ Desventajas

- Es recomendable emplearlo solo en proyectos a corto plazo
- Altas comisiones en caso de fallar
- Imposible prever todo antes de programar
- Demasiado costoso e innecesario

### 2.5.2.3 SCRUM

Scrum es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. En años recientes, Schwaber y Beedle han desarrollado más los métodos Scrum. (S. Pressman, 2010)

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso (que se estudia en el párrafo siguiente) llamado *sprint*. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define —y con frecuencia se modifica— en tiempo real por parte del equipo Scrum. (S. Pressman, 2010)



*Retraso:* lista de prioridades de los requerimientos o características del proyecto que dan al cliente un valor del negocio. Es posible agregar en cualquier momento otros aspectos al retraso (ésta es la forma en la que se introducen los cambios).

*Sprints:* consiste en unidades de trabajo que se necesitan para alcanzar un requerimiento definido en el retraso que debe ajustarse en una caja de tiempo predefinida (lo común son 30 días). Durante el sprint no se introducen cambios (por ejemplo, aspectos del trabajo retrasado). Así, el sprint permite a los miembros del equipo trabajar en un ambiente de corto plazo pero estable.

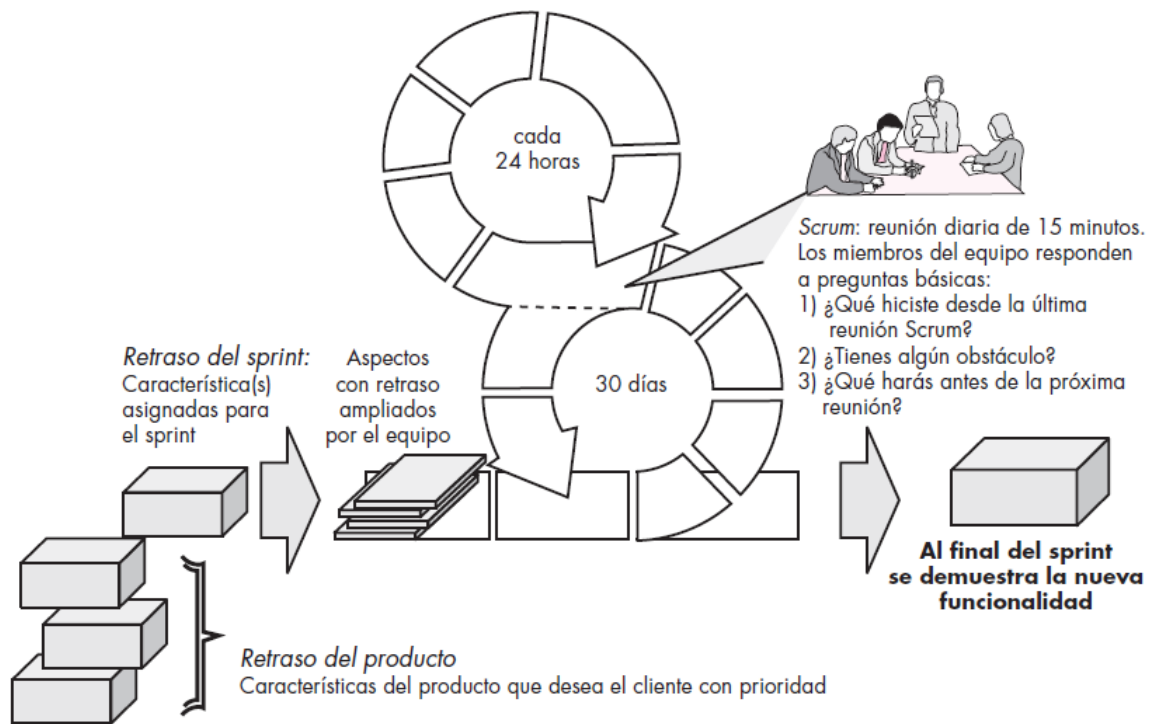
*Reuniones Scrum:* son reuniones breves (de 15 minutos, por lo general) que el equipo Scrum efectúa a diario. Hay tres preguntas clave que se pide que respondan todos los miembros del equipo [Noy02]:

- ¿Qué hiciste desde la última reunión del equipo?
- ¿Qué obstáculos estás encontrando?
- ¿Qué planeas hacer mientras llega la siguiente reunión del equipo?

Un líder del equipo, llamado *maestro Scrum*, dirige la junta y evalúa las respuestas de cada persona. La junta Scrum ayuda al equipo a descubrir los problemas potenciales tan pronto como sea posible. Asimismo, estas juntas diarias llevan a la “socialización del conocimiento” [Bee99], con lo que se promueve una estructura de equipo con organización propia.

*Demostraciones preliminares:* entregar el incremento de software al cliente de modo que la funcionalidad que se haya implementado pueda demostrarse al cliente y éste pueda evaluarla. (S. Pressman, 2010)

**Figura 15: Proceso SCRUM.**



**Fuente:** (S. Pressman, 2010)

➤ **Ventajas**

- Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.
- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.
- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva.
- Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento

➤ Desventajas

- Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.
- Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas.

#### 2.5.2.4 Desarrollo adaptativo de software (DAS)

El *desarrollo adaptativo de software* (DAS) fue propuesto por Jim Highsmith como una técnica para elaborar software y sistemas complejos. Los fundamentos filosóficos del DAS se centran en la colaboración humana y en la organización propia del equipo. (S. Pressman, 2010)

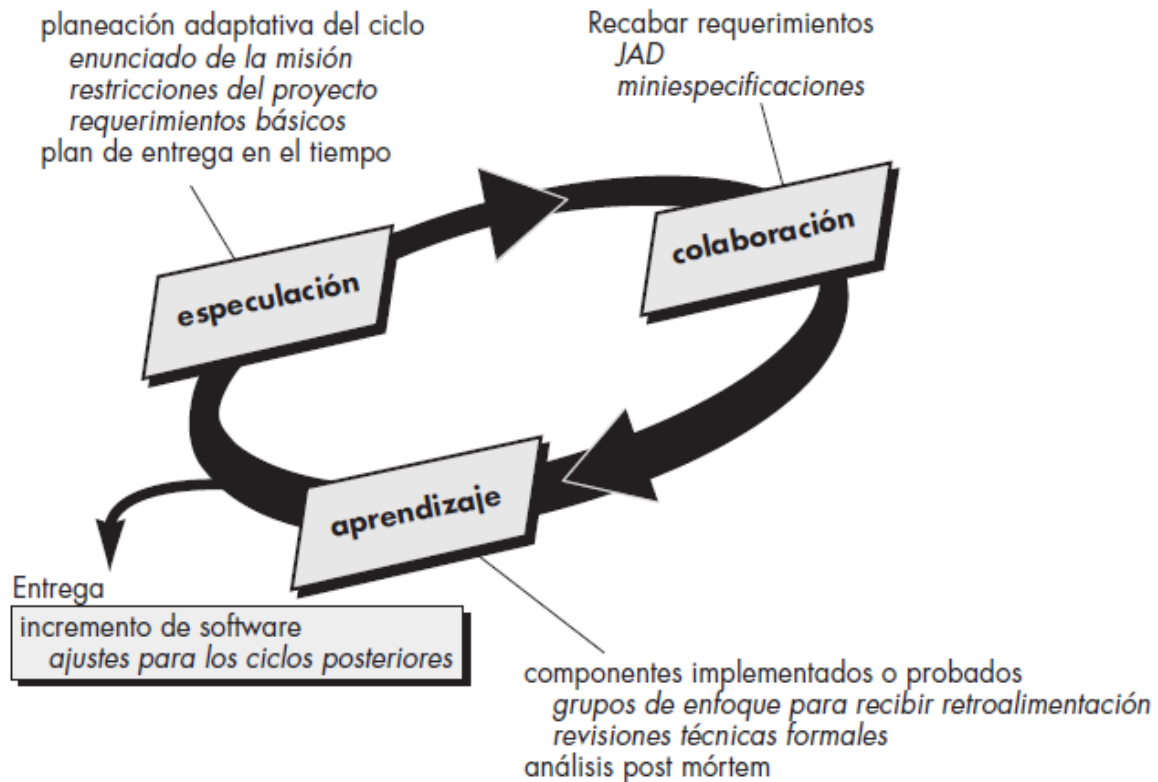
Durante la *especulación*, se inicia el proyecto y se lleva a cabo la *planeación adaptativa del ciclo*. La especulación emplea la información de inicio del proyecto —enunciado de misión de los clientes, restricciones del proyecto (por ejemplo, fechas de entrega o descripciones de usuario) y requerimientos básicos— para definir el conjunto de ciclos de entrega (incrementos de software) que se requerirán para el proyecto.

Las personas motivadas usan la *colaboración* de manera que multiplica su talento y producción creativa más allá de sus números absolutos. Este enfoque es un tema recurrente en todos los métodos ágiles. Sin embargo, la colaboración no es fácil. Incluye la comunicación y el trabajo en equipo, pero también resalta el individualismo porque la creatividad individual desempeña un papel importante en el pensamiento colaborativo. Es cuestión, sobre todo, de confianza. Las personas que trabajan juntas deben confiar una en otra a fin de: 1) criticarse sin enojo, 2) ayudarse sin resentimiento, 3) trabajar tan duro, o más, que como de costumbre, 4) tener el conjunto de aptitudes para contribuir al trabajo, y 5) comunicar los problemas o preocupaciones de manera que conduzcan a la acción efectiva.

Conforme los miembros de un equipo DAS comienzan a desarrollar los componentes que forman parte de un ciclo adaptativo, el énfasis se traslada al

“aprendizaje” de todo lo que hay en el avance hacia la terminación del ciclo. (S. Pressman, 2010)

**Figura 16:** Proceso DAS.



**Fuente:** (S. Pressman, 2010)

➤ **Ventajas**

- Se utiliza para poder aprender de los errores e iniciar nuevamente el ciclo de desarrollo.
- Utiliza información disponible acerca de todos los cambios para poder mejorar el comportamiento del Software.
- Difunde la colaboración de distintas personas.

➤ **Desventajas**

- Los errores y cambios que no son detectados con anterioridad afectan la calidad del producto y su costo total.

- Ya que esta es una metodología ágil, no permite realizar procesos que son requeridos en las metodologías tradicionales.

## **2.6 ARQUITECTURA DE SOFTWARE**

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.

La arquitectura no es el software operativo. Es una representación que permite 1) analizar la efectividad del diseño para cumplir los requerimientos establecidos, 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y 3) reducir los riesgos asociados con la construcción del software.

### **2.6.1 Arquitectura Cliente-Servidor**

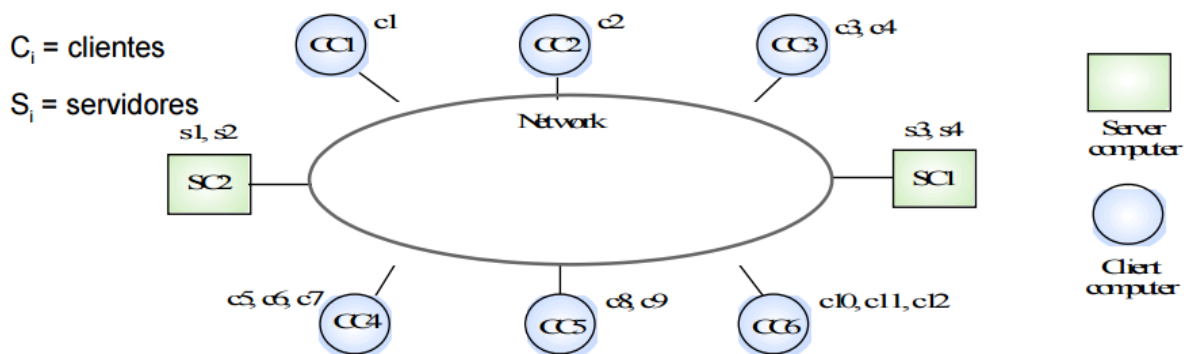
Este modelo Cliente/Servidor empezó a ser aceptado a finales de los 80's. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado. (Wesley & Sommerville, 2004)

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio) (Ver Figura 17). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe. Usualmente

la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario. (Wesley & Sommerville, 2004)

**Figura 17:** Cliente-Servidor.



**Fuente:** (Wesley & Sommerville, 2004)

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aun en entornos multiplataforma.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio).

**Figura 18:** Modelo Cliente-Servidor.



**Fuente:** (Wesley & Sommerville, 2004)

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC's a través de interfaces graficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente solo se ocupan de la interacción con el usuario. En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.

➤ Ventajas

- Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas.
- El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas Construidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario.
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

➤ Desventajas

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Es importante que los clientes y los servidores utilicen el mismo mecanismo (por ejemplo sockets o RPC), lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas.
- Además, hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad de un esquema Cliente/Servidor es otra preocupación importante. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor.
- El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente/Servidor. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc.

## **2.7 TECNOLOGÍAS DE DESARROLLO**

### **2.7.1 Lenguajes de programación del lado servidor**

#### **2.7.1.1 JAVA**

Java es un lenguaje orientado a objetos muy completo e independiente de las plataformas, diseñados por Sun Microsystems a principios de los años 90 al tratar de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La compañía Sun Microsystems describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”, lo que describe bastante bien el lenguaje Java, aunque en algunas de esas características el lenguaje sea todavía bastante mejorable. (Garcia & Rodríguez, 2000)

#### **a) Características de Java**



Algunas características notables son las siguientes:

- Es un lenguaje robusto.
- Gestiona la memoria automáticamente.
- No permite el uso de técnica de programación inadecuada.
- Multithreading
- Trabaja sobre una arquitectura cliente-servidor.
- Posee mecanismos de seguridad incorporados.
- Cuenta con herramientas de documentación incorporadas.

Existen distintos programas comerciales que permiten desarrollar código Java. La compañía Sun creadora de Java, distribuye gratuitamente el Java(tm) Development Kit (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java.

Los IDEs (Integrated Development Environment), tal como su nombre indica, son entornos de desarrollo integrados. En un mismo programa es posible escribir el código Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación.

El compilador de Java es una de las herramientas de desarrollo incluidas en el JDK. Realiza un análisis de sintaxis del código escrito en los ficheros fuente de Java (con extensión \*.java). Si no encuentra errores en el código genera los ficheros compilados (con extensión \*.class). En otros muestra la línea o líneas erróneas. En el JDK de Sun dicho compilador se llama javac.exe. Tiene numerosas opciones, algunas de las cuales varían de una versión a otra. (García & Rodríguez, 2000)

#### b) Java Virtual Machine

La existencia de distintos tipos de procesadores y ordenadores llevó a los ingenieros de Sun a la conclusión de que era muy importante conseguir un software que no dependiera del tipo de procesador utilizado. Se planteó la necesidad de conseguir un código capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado no debería ser necesaria ninguna modificación por el hecho de cambiar de procesador o de ejecutarlo en otra máquina. La clave consistió en desarrollar un

código “neutro” el cual estuviera preparado para ser ejecutado sobre una “maquina hipotética o virtual”, denominada Java Virtual Machine (JVM). Es esta JVM quien interpreta este código neutro convirtiéndolo a código particular de la CPU utilizada. Se evita tener que realizar un programa diferente para cada CPU o plataforma. La JVM es el intérprete de Java. Ejecuta los “bytecodes” (ficheros compilados con extensión \*.class) creados por el compilador de Java (java.exe). (Garcia & Rodríguez, 2000)

➤ Ventajas

- No debes volver a escribir el código si quieres ejecutar el programa en otra máquina. Un solo código funciona para todos los browsers compatibles con Java o donde se tenga una Máquina Virtual de Java
- Java es un lenguaje de programación orientado a objetos, y tiene todos los beneficios que ofrece esta metodología de programación
- Un browser compatible con Java deberá ejecutar cualquier programa hecho en Java, esto ahorra a los usuarios tener que estar insertando "plug-ins" y demás programas que a veces nos quitan tiempo y espacio en disco.
- Java es un lenguaje y por lo tanto puede hacer todas las cosas que puede hacer un lenguaje de programación: Cálculos matemáticos, procesadores de palabras, bases de datos, aplicaciones gráficas, animaciones, sonido, hojas de cálculo, etc.

➤ Desventajas

- Aunque la JVM es una buena idea, hace que la ejecución de las aplicaciones sea muy lenta.
- Sí, puede parecer complicado pero es el precio que hay que pagar para tener la portabilidad de las aplicaciones.
- El diseño de interfaces gráficas con awt y swing no es simple. Existen herramientas como el JBuilder que permiten generar interfaces gráficas de manera sencilla, pero tienen un costo adicional.

- Puede ser que no haya JDBC para bases de datos poco comerciales.  
o Algunas herramientas tienen un costo adicional
- Los programas hechos en Java no tienden a ser muy rápidos, supuestamente se está trabajando en mejorar esto. Como los programas de Java son interpretados nunca alcanzan la velocidad de un verdadero ejecutable.

### **2.7.1.2 RUBY**

Ruby es un lenguaje con balance cuidado. Su creador Yukihiro Matsumoto, mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la imperativa.

A menudo ha manifestado que está tratando de hacer que Ruby sea natural, no simple de una forma que se asemeje a la vida real. (Thomas, 2010)

Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando las capacidades de las conferencias relacionadas a Ruby.

#### **a) Características del lenguaje Ruby**

- Es un lenguaje de programación interpretado, reflexivo y orientado a objetos.
- Tiene una mezcla de diferentes lenguajes de programación tales como Perl, Smalltalk.
- Cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Manejo de excepciones.
- Altamente portable.
- Amplia librería estándar.
- Soporta alteración de objetos en tiempo de ejecución.

Con Ruby se pueden realizar Páginas Web, sistemas web y aplicaciones web con un alcance ilimitado, un lenguaje rápido y sin mucho código que escribir ya que este es inteligente.

➤ Ventajas

- Es un lenguaje multiplataforma.
- Es un lenguaje dinámico orientado a objetos.
- Se puede realizar cualquier tipo de cambio en tiempo de ejecución.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con sqlite3 y MySQL.
- Tiene claridad y simplicidad

➤ Desventajas

- Es un lenguaje interpretado y por ende acarrea con todos los conceptos relacionados al mismo.

### **2.7.1.3 C#**

El lenguaje de programación C# fue creado con el mismo espíritu que los lenguajes C y C++. Esto explica sus poderosas prestaciones y su fácil curva de aprendizaje. No se puede decir lo mismo de C y C++, pero como C# fue creado desde cero Microsoft se tomó la libertad de eliminar algunas de las prestaciones más pesadas. (Ferguson, Patterson, Beres, Boutquin, & Gupta, 2003)

C# es un nuevo lenguaje presentado en .NET Framework, procede de C++. Sin embargo C# es un lenguaje orientado a objetos, moderno y seguro.

a) Características del lenguaje C#

- Es un lenguaje orientado a objetos por eso todo código y datos deben ser incluidos en una clase.
- Es un lenguaje de programación simple pero eficaz, diseñada para escribir aplicaciones empresariales.
- es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores.
- presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria).

- proporciona acceso a los tipos de API más comunes: .NET Framework, COM, Automatización y estilo C. Asimismo, admite el modo unsafe, en el que se pueden utilizar punteros para manipular memoria que no se encuentra bajo el control del recolector de elementos no utilizados.

#### ➤ Ventajas

- Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- Atributos: cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.
- Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override
- Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.
- Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

#### ➤ Desventajas

- se tiene que conseguir una versión reciente de Visual Studio .NET
- se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, et

## 2.7.2 Lenguajes de programación del lado Cliente

### 2.7.2.1 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (Eguiluz, 2003)

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

#### a) Características de JavaScript

- Es simple, no hace falta tener conocimientos de programación para poder hacer un programa en JavaScript
- Maneja objetos dentro de nuestra página Web y sobre ese objeto podemos definir diferentes eventos.
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo

#### ➤ **Ventajas**

- Es un lenguaje sencillo y muy liviano.
- Recomendada para la creación de aplicaciones web
- Utiliza poca memoria útil para el desarrollo de páginas web dinámicas
- Tiene una gran cantidad de efectos visuales
- Es soportado por los más populares navegadores
- Puede agregar interactividad a elementos web

- Compatible con la gran mayoría de los navegadores modernos incluyendo iPhone, móviles y PS3.

➤ **Desventajas**

- Sus recursos no son tan extensos
- Sus opciones 3d son limitadas
- En sistemas no tan complejos puede generar errores
- Los usuarios pueden desactivar JavaScript en su navegador

### 2.7.2.2 Applets de Java

Una applet de java es realmente un programa pequeño en Java que se transfiere dinámicamente a través de la red, igual que una imagen archivo de sonido o animación. La diferencia importante es que una applet es un programa inteligente, no simplemente una animación o un formato de archivo de medios. En otras palabras, es un programa que puede reaccionar a la entrada del usuario y cambiar dinámicamente, no simplemente ejecutar la misma animación o sonido una y otra vez. (Naughton, 1996)

➤ **Ventajas**

- Son mucho menos dependientes del navegador que los scripts.

➤ **Desventajas**

- Son más lentos de procesar.
- Tienen espacio muy delimitado en la página donde se ejecutan.

### 2.7.2.3 Visual Basic Script

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de JavaScript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspirados en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero. El modo de funcionamiento de Visual Basic Script para construir efectos especiales

en páginas web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador. (Alvarez, 2002)

➤ **Ventajas**

- Es un lenguaje RAD.
- Integra el diseño e implementación de formularios de Windows.
- Permite usar con suma facilidad la plataforma de los sistemas Windows dado que tiene acceso prácticamente total a la API de Windows incluidas librerías actuales.
- El código en Visual Basic es fácilmente migrable a otros lenguajes.
- Es un lenguaje muy extendido por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.

➤ **Desventajas**

- No es multiplataforma (aunque la mayoría de las aplicaciones compiladas pueden correr nativamente en sistemas Linux sobre utilidades que emulan las librerías de Windows como por ejemplo Wine).
- Sin soporte oficial de Microsoft desde el 4 de abril de 2008
- Por defecto permite la programación sin declaración de variables.
- No permite programación a bajo nivel ni incrustar secciones de código en ASM
- Sólo soporta librerías dinámicas (DLL) que usen la convención de llamadas y componentes y librerías ActiveX.
- No soporta tratamiento de procesos como parte del lenguaje.
- No incluye operadores de desplazamiento de bits como parte del lenguaje.
- No permite el manejo de memoria dinámica, punteros, etc. como parte del lenguaje.
- No tiene instrucciones de pre procesamiento.



## 2.7.3 Frameworks del lado Servidor

### 2.7.3.1 Sinatra

Sinatra es un framework para aplicaciones web de software libre y código abierto para el lenguaje Ruby (Harris, 2009)

Se necesita de una gema para poder instalar Sinatra mediante Ruby el cual utiliza el puerto 4567, este framework no utiliza algún tipo de modelo lo que más sobresale son sus rutas que tienen los métodos de HTTP junto con el patrón URL y en base a estas rutas se coloca como función para que pueda hacer directo el código.

#### ➤ **Ventajas**

- Tiene una gran herramienta en cuestión de rutas.
- Muy ágil con funciones muy eficiente.

#### ➤ **Desventajas**

- En muy complicado de entender.
- Algunas veces difícil de añadir funciones.

### 2.7.3.2 Padrino

Padrino es un framework de Ruby edificado sobre el framework Sinatra que es DSL para la creación de aplicaciones web sencillas en Ruby con el mínimo de esfuerzo. Este framework hace que sea tan divertido y fácil posible de codificar aplicaciones web cada vez más avanzadas mediante la mejora de Sinatra mientras se mantiene fiel al espíritu que lo hace genial. ([www.padrino.com](http://www.padrino.com))

#### ➤ **Ventajas**

- Es fácil de aprender.
- Tiene generadores de modelos y controladores
- Reutilización de código, desarrollo ágil.
- Existe abundante documentación en la red.

#### ➤ **Desventajas**

- Se necesita primero aprender Sinatra para utilizarlo.

### 2.7.3.3 Rails

Ruby on Rails es un marco que hace que sea más fácil de desarrollar, implementar y mantener aplicaciones web. Durante los meses que siguieron a su lanzamiento inicial, Rails pasó de ser un juguete desconocido a ser un fenómeno mundial; más importante, se ha convertido en el marco de elección para la implementación de una amplia gama de los llamados aplicaciones Web 2.0. (Ruby, Thomas, & Heinemeier Hansson, 2013).

Un gran número de desarrolladores se vieron frustrados con las tecnologías que eran utilizando para crear aplicaciones web. No parece importar si utilizan Java, PHP o .NET hubo una creciente sensación de que sus trabajos eran demasiado condenadamente difícil. Y luego, de repente, llegó Rails, y Rails fue más fácil. Pero fácil por sí no es suficiente. Estamos hablando de los desarrolladores profesionales escribir sitios web del mundo real. Querían sentir que las aplicaciones estén disponibles estaban desarrollando sería resistir el paso del tiempo que se han diseñado e implementado el uso de técnicas modernas y profesionales. Así, estos desarrolladores excavado en Rails y descubrió que no era sólo una herramienta para hackear los sitios.

#### ➤ **Ventajas**

- Fácilmente integrable
- Flexibilidad
- Simplicidad
- Cercano al lenguaje natural
- Productividad

#### ➤ **Desventajas**

- Tipos dinámicos con problemas inesperados
- Los hilos nativos en lugar de Green threads.
- Sin soporte Unicode
- Falta especificación.

## 2.8 SISTEMAS DE GESTIÓN DE BASE DE DATOS

### 2.8.1 SQL Server.

SQL Server usa la arquitectura Cliente/Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo Servidor y tareas que corran en computadoras tipo Cliente: El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server. SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc.) entre las múltiples peticiones.

La arquitectura Cliente/Servidor permite desarrollar aplicaciones para realizar un una variedad de ambiente.

El DBMS de SQL Server es responsable de:

- Mantener las relaciones entre la información y la Base de Datos
- Asegurarse de que la información es almacenada correctamente, es decir, que las reglas que definen entre los datos no sean violadas.

#### ➤ Ventajas

- Es un sistema de gestión de base de datos.
- Es útil para manejar y obtener datos de la red de redes.
- Nos permite olvidarnos de los ficheros que forman la base de datos.
- Si trabajamos en una red social nos permite agregar otros servidores de SQL Server. Por ejemplo dos personas que trabajan con SQL Server, uno de ellos se puede conectar al servidor de su otro compañero y así se puede ver las bases de datos del otro compañero con SQL Server.

- SQL permite administrar permisos a todo. También permite que alguien conecte su SQL al nuestro pero sin embargo podemos decirle que no puede ver esta base de datos pero otro sí.

#### ➤ Desventajas

- Utiliza mucho la memoria RAM para las instalaciones y utilización de software.
- No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular.
- La relación, calidad y el precio está muy debajo comparado con Oracle.
- Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.

### **2.8.2 PostgreSQL.**

PostgreSQL es un Sistema Gestor de Base de Datos Orientadas a Objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo tiempo o a un mejor nivel que muchos SGDB comerciales. (Camps, y otros, 2005)

El origen de PostgreSQL se sitúa en el gestor de bases de datos POSTGRES desarrollado en la Universidad de Berkeley y que se abandonó en favor de PostgreSQL a partir de 1994. Ya entonces, contaba con prestaciones que lo hacían único en el mercado y que otros gestores de bases de datos comerciales han ido añadiendo durante este tiempo.

PostgreSQL se distribuye bajo licencia BSD, lo que permite su uso, redistribución, modificación con la única restricción de mantener el copyright del software a sus autores, en concreto el PostgreSQL Global Development Group y la Universidad de California.

#### ➤ Ventajas

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Footprint bajo de memoria, bastante poderoso con una configuración adecuada.
- Multiplataforma.
- Soporte empresarial disponible.

#### ➤ Desventajas

- Sin experticia, configurar llega a ser un caos.
- Es fácil de vulnerar sin protección adecuada.
- El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.
- InnoDB genera mucho footprint en memoria al indexar.
- El toolset empresarial tiene un costo adicional por suscripción anual.
- Realizar revisiones llegar a ser una labor manual y tediosa para el DBA.
- Reducida cantidad de tipos de datos.

### 2.8.3 MySQL

MySQL es un sistema gestor de base de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGDB del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos

importantes) contar con un alto grado de estabilidad y un rápido desarrollo. (Camps, y otros, 2005)

MySQL está disponible para múltiples plataformas, sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada en este caso es el cliente mysql-client, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto. (Camps, y otros, 2005)

#### ➤ Ventajas

- MySQL es software Open Source
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una maquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación. Soporta gran variedad de Sistemas operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL.

#### ➤ Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS)

### **3 MARCO PRÁCTICO**

#### **3.1 ANÁLISIS DE LOS PROCESOS DE PLANILLAS DEL SECTOR DE COSTOS DIRECTOS Y DISEÑO DEL MODELADO DE NEGOCIO ACTUAL.**

Para el análisis de los procesos de planillas se ha hecho una recopilación de información mediante el método de entrevistas personales con cada uno de los encargados en la constructora Suarez y con los antecedentes del proyecto se lograra hacer el modelo de negocio actual.

##### **3.1.1 Recopilación de información relacionada con el seguimiento de proyectos empleando Sistemas Colaborativos**

Según el marco teórico en el punto 2.4 se recopiló la información sobre los sistemas colaborativos en donde se escogerá el Groupware ya que va junto con los conceptos y terminologías se eligieron para realizar el sistema ver Anexo E.

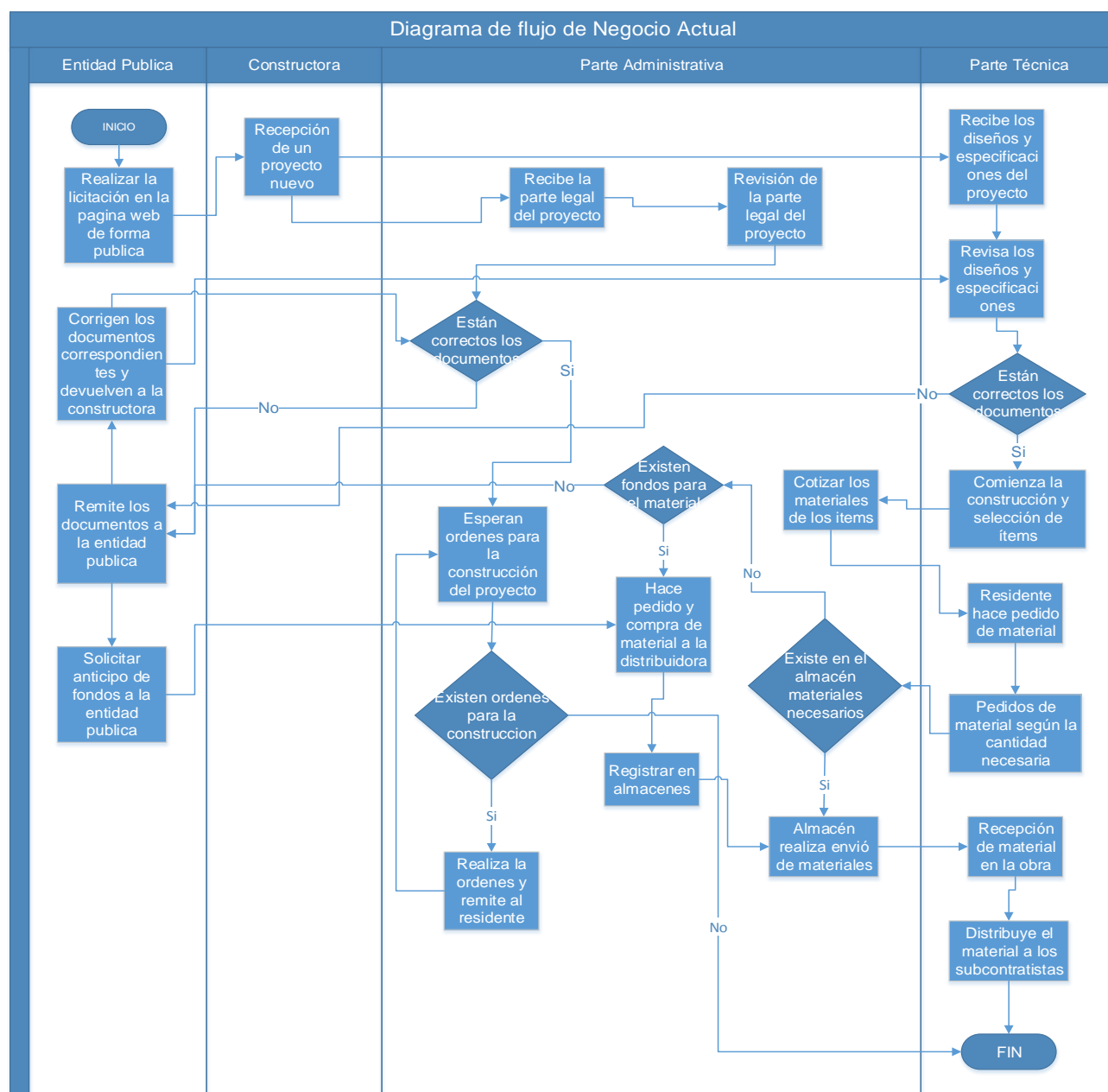
##### **3.1.2 Elaboración de entrevistas con el personal encargado de los registros de los procesos de planillas.**

Según las entrevistas realizadas al personal de la constructora pero en especial al de la parte administrativa se recolecto información sobre el flujo de los proyectos en la constructora ver Anexo D.

El modelo de negocio actual se elaboró a partir de las entrevistas y mediante observación participativa, siendo parte de todo el proceso de solicitar material para la obra teniendo como prioridad el control de inventario.

### 3.1.3 Elaboración del diagrama de flujo que represente el modelado de negocio actual de los procesos de movimiento de planilla

**Figura 19:** Diagrama de flujo de negocio actual.



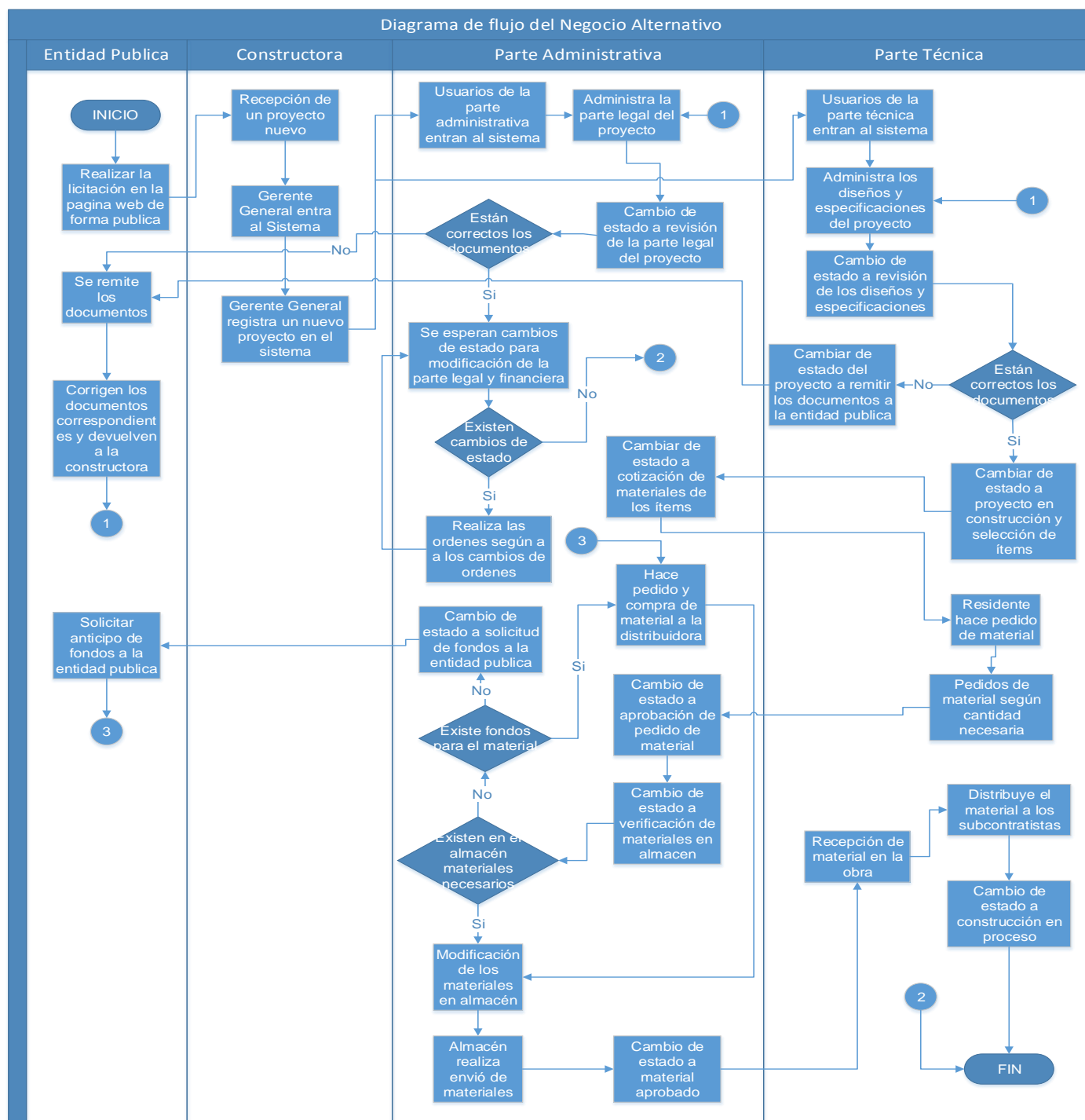
**Fuente:** Elaboración propia



## 3.2 DISEÑO DEL MODELADO DE NEGOCIO ALTERNATIVO

### 3.2.1 Elaborar un diagrama de flujo de negocio alternativo.

**Figura 20:** Diagrama de flujo de negocio alternativo.



**Fuente:** Elaboración propia

### 3.2.2 Identificar las falencias de los procedimientos manuales

Las falencias que existen en el proceso actual son:

- El registro manual de todos los datos del proyecto que genera riesgo de pérdida y no se pueden leer con claridad.
- La comunicación inoportuna entre las áreas técnica y administrativa.
- El desconocimiento de los estados en los que se encuentran los proyectos.
- La ineficiencia en el manejo de la información de cada proyecto que provoca retrasos en los desembolsos de capital.
- La demora de tiempo en los procesos de búsqueda de información de los proyectos.

### 3.2.3 Seleccionar la metodología de desarrollo de Software

**Tabla 3:** Metodologías de Desarrollo de Software

<b>METODOLOGIA DE DESARROLLO DE SOFTWARE</b>	<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Programación Extrema XP	<ul style="list-style-type: none"><li>• Programación organizada.</li><li>• Menor tasa de errores.</li><li>• Satisfacción del programador.</li><li>• Solución de errores de programas</li></ul>	<ul style="list-style-type: none"><li>• Es recomendable emplearlo solo en proyectos a corto plazo</li><li>• Altas comisiones en caso de fallar</li><li>• Imposible prever todo antes de programar</li><li>• Demasiado costoso e innecesario</li></ul>

	<ul style="list-style-type: none"> <li>• Versiones nuevas</li> <li>• Implementa una forma de trabajo donde se adapte fácilmente a las circunstancias</li> </ul>	
SCRUM	<ul style="list-style-type: none"> <li>• Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.</li> <li>• Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.</li> <li>• Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.</li> <li>• Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.</li> <li>• Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.</li> <li>• Permite producir software de una forma consistente, sostenida y competitiva.</li> </ul>	<ul style="list-style-type: none"> <li>• Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.</li> <li>• Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas.</li> </ul>

	<ul style="list-style-type: none"> <li>Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento</li> </ul>	
Desarrollo adaptativo de software (DAS)	<ul style="list-style-type: none"> <li>Se utiliza para poder aprender de los errores e iniciar nuevamente el ciclo de desarrollo.</li> <li>Utiliza información disponible acerca de todos los cambios para poder mejorar el comportamiento del Software.</li> <li>Difunde la colaboración de distintas personas.</li> </ul>	<ul style="list-style-type: none"> <li>Los errores y cambios que no son detectados con anterioridad afectan la calidad del producto y su costo total.</li> <li>No permite realizar procesos que son requeridos en las metodologías tradicionales.</li> </ul>

**Fuente:** Elaboración Propia

Por las características mencionadas anteriormente y de acuerdo a sus ventajas y desventajas se seleccionó como desarrollo de software SCRUM ya que es una de las metodologías más populares y también por la experiencia ganada al implementar este tipo de metodología, también porque es muy ágil al momento de estar interactuando constantemente con el cliente o dueño del producto.

### **3.2.4 Planificar el desarrollo de software en base a la metodología seleccionada**

Para la implementación de la metodología SCRUM se debe de primero tener en cuenta ciertos puntos:

- Tener muy claro los Roles que son tres:
  - 1) Dueño del producto (Product Owner)

El Product Owner es la persona que ve el producto final, en otras palabras en este caso es el cliente que contrata a un equipo de desarrolladores de software, esta persona hará una bitácora en base a las necesidades que desea como quiere su sistema finalizado, y a lo largo de los entregables llamados sprints el cambiará la bitácora según el desee.

#### 2) Scrum Master

Esta persona es el eje del equipo, es el que facilita y coordina todos los requerimientos que se tenga dentro del sprint, es el que habla con el Product Owner para así tener la bitácora para realizar los sprints, luego entrega la bitácora al equipo de trabajo para quedar en conjunto que historias de usuario tendrá cada sprint.

#### 3) Equipo de trabajo

El equipo de trabajo son un grupo de personas formadas de 3 a 5 personas máximo estos implementan las ideas y objetivos dentro del sprint.

- Los Sprints e historias de usuario

Los sprints son entregables hasta una cierta fecha y dentro de cada sprint se colocan las historias de usuario que se han de cumplir hasta la entrega y finalización del sprint, en otras palabras de toda la bitácora que nos da el Product Owner se escoge cual son los primordiales para el primer sprint.

**Tabla 4:** Roles de la metodología SCRUM

ROL	DESCRIPCIÓN	RESPONSABLE
Product owner	Conoce la lógica de negocio y así mismo se encarga de mediar entre el equipo de trabajo al cliente	Lic. Néstor Suarez
Scrum Master	Se encarga de supervisar el cumplimiento de la metodología	Tutor Samuel Lujan
Equipo	Desarrollo de los objetivos del proyecto	Estudiante Gerald Gutierrez

**Fuente:** Elaboración Propia

**Tabla 5:** Product Backlog del sistema

Prioridad	Como	Necesito	Para	Esfuerzo
1	Gerente general	Crear y modificar información de Entidades Públicas.	Saber qué proyectos asignar a estas Entidades Públicas.	1
2	Gerente general	Crear y modificar información de proyectos para las Entidades públicas.	Que el sistema me dé información de compras y ventas de material de obras por proyecto.	1
3	Superintendente	Guardar documentos de proyectos.	Compartir y trabajar en colaboración con el resto de empleados.	8
4	Superintendente	Revisar y autorizar cambios a los documentos de diseños y especificaciones.	Tener los documentos formalmente publicados.	2
5	Administrador	Administrar y asignar cuentas a los usuarios.	Que tengan acceso al sistema	8
6	Administrador	Establece los roles de trabajo dentro del sistema a los usuarios.	Que puedan trabajar solo con las tareas que les son asignadas.	5

<b>Prioridad</b>	<b>Como</b>	<b>Necesito</b>	<b>Para</b>	<b>Esfuerzo</b>
7	Administrador	Establece los flujos de trabajo y coordinación de los usuarios del sistema.	Que los usuarios que colaboran entre si coordinen sus tareas.	6
8	Administrador	Establece los permisos de usuario a las distintas opciones del sistema.	Que cada uno de ellos pueda realizar las operaciones que le son permitidas	7
9	Residente de obra	Realizar pedidos de material del proyecto que está encargado	Poder seguir con el proceso de construcción de la obra	5
10	Residente de obra	Realizar solicitudes de compra	Tener lo más antes posibles los materiales para la obra	6
11	Jefe de Almacén	Crear y modificar información de los materiales	Tener un catálogo de materiales con los cuales se pueda realizar compras y despachos de materiales.	8
12	Jefe de Almacén	Categorizar los materiales	Agrupados los materiales por categorías.	5
13	Jefe de Almacén	Tener información de cada uno de los materiales asignados a cada proyecto	Controlar existencias de materiales	5
14	Jefe de Almacén	Un reporte de inventario físico valorado por proyecto o por Entidad pública.	Entregar al contador y que él pueda actualizar su información financiera.	8
15	Jefe de Almacén	Saber todos los pedidos realizados por los residentes de obras.	Poder determinar todos los requerimientos de los ingenieros residentes.	7
16	Gerente General	Autorizar las solicitudes de compra de materiales para los proyectos	Controlar las compras que se realizan para los proyectos	5
17	Superintendente	Autorizar los pedidos de materiales para los proyectos	Controlas los envíos de materiales a las obras por proyecto	5
18	Jefe de contabilidad	Validar los costos de materiales	Tener una referencia del costo de los materiales	7
19	Jefe de contabilidad	Consultar inventarios físicos y valorados	Controlar el estado de los inventarios físicos y valorados	7
20	Auxiliar de contabilidad	Realizar seguimiento a la gestión de compras	Tener un control específico de todas las compras que se realizan	6

Prioridad	Como	Necesito	Para	Esfuerzo
21	Auxiliar de contabilidad	Consultar inventarios físicos y valorados	Saber que hay dentro de los inventarios así como sus valorados	7
22	Auxiliar de contabilidad	Consulta de proveedores de materiales	Saber que proveedores se tienen para cada material	6

**Fuente:** Elaboración Propia

### 3.2.5 Relación entre product backlog y objetivos específicos

Dentro lo que es el sistema, para cumplir con los objetivos trazados y aplicando la metodología ágil se logra aplicarlo mediante el sprint realizado en cada objetivo asignado mostrado a continuación mediante la tabla.

**Tabla 6:** Product Backlog del sistema

OBJETIVOS ESPECÍFICOS	SPRINT	
	ID	TIPO
Diseñar e implementar módulo de gestión de usuarios para la Constructora	1	Administración de Usuario
Diseñar e implementar un módulo de Administración de proyectos en general	2	Administración de Proyectos
Desarrollar el módulo de solicitud de pedidos y compra de material	3	Administración de materiales
	4	Pedidos y compra de materiales
Desarrollar el módulo de movimientos de inventarios	5	Administrar inventarios
	6	Realizar movimientos de inventarios
Diseñar e implementar el módulo Colaborativo del sistema para generar el seguimiento de la información que fluye de un personal a otro	7	Establecer flujos de trabajo
	8	Establecer coordinación entre los usuarios



**Fuente:** Elaboración Propia

### **3.3 DISEÑAR E IMPLEMENTAR MÓDULO DE GESTIÓN DE USUARIOS PARA LA CONSTRUCTORA.**

#### **3.3.1 Seleccionar el lenguaje de programación y sistema de gestor de base de datos**

**Tabla 7:** Lenguajes de Programación

<b>LENGUAJES DE PROGRAMACION</b>	<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
JAVA	<ul style="list-style-type: none"><li>• Un solo código funciona para todos los browsers compatibles con Java o donde se tenga una Máquina Virtual de Java</li><li>• Java es un lenguaje de programación orientado a objetos, y tiene todos los beneficios que ofrece esta metodología de programación</li><li>• Un browser compatible con Java deberá ejecutar cualquier programa hecho en Java.</li><li>• Java es un lenguaje y por lo tanto puede hacer todas las cosas que puede hacer un lenguaje de</li></ul>	<ul style="list-style-type: none"><li>• Aunque la JVM es una buena idea, hace que la ejecución de las aplicaciones sea muy lenta.</li><li>• El diseño de interfaces gráficas con awt y swing no es simple.</li><li>• Puede ser que no haya JDBC para bases de datos poco comerciales. o Algunas herramientas tienen un costo adicional</li></ul>

	<p>programación: Cálculos matemáticos, procesadores de palabras, bases de datos.</p>	
RUBY	<ul style="list-style-type: none"> <li>• Es un lenguaje multiplataforma.</li> <li>• Es un lenguaje dinámico orientado a objetos.</li> <li>• Se puede realizar cualquier tipo de cambio en tiempo de ejecución.</li> <li>• Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con sqlite3 y MySQL.</li> <li>• Tiene claridad y simplicidad</li> </ul>	<ul style="list-style-type: none"> <li>• Es un lenguaje interpretado y por ende acarrea con todos los conceptos relacionados al mismo.</li> </ul>
C#	<ul style="list-style-type: none"> <li>• Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.</li> <li>• Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros.</li> <li>• Métodos virtuales y redefiniciones: antes de que</li> </ul>	<ul style="list-style-type: none"> <li>• se tiene que conseguir una versión reciente de Visual Studio .NET</li> <li>• se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior, tener</li> </ul>

	<p>un método pueda ser redefinido en una clase base, debe declararse como virtual.</p> <ul style="list-style-type: none"> <li>• Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.</li> <li>• Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres.</li> </ul>	<p>alrededor de 4 gigas de espacio libre para la pura instalación, et</p>
--	---	---

Por las características mencionadas anteriormente y de acuerdo a sus ventajas y desventajas se elegirá como lenguaje de programación PHP ya que es un lenguaje más dedicado hacia las páginas web y ya que también es muy utilizado hoy en día por las empresas para tener un sistema web dinámico y muy eficiente al momento de usar solo se necesita de un browser y lo demás solo el contar con un servidor.

**Tabla 8:** Gestores de Base de Datos

GESTOR DE BASE DE DATOS	VENTAJAS	DESVENTAJAS
SQL Server	<ul style="list-style-type: none"> <li>• Es un sistema de gestión de base de datos.</li> <li>• Es útil para manejar y obtener datos de la red de redes.</li> <li>• Nos permite olvidarnos de los ficheros que forman la base de datos.</li> <li>• Si trabajamos en una red social nos permite agregar otros servidores de SQL Server.</li> </ul>	<ul style="list-style-type: none"> <li>• Utiliza mucho la memoria RAM para las instalaciones y utilización de software.</li> <li>• No se puede utilizar como practicas porque se prohíben muchas cosas.</li> <li>• La relación, calidad y el precio está muy debajo comparado con Oracle.</li> <li>• Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño.</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>• Ampliamente popular - Ideal para tecnologías Web.</li> <li>• Fácil de Administrar.</li> <li>• Su sintaxis SQL es estándar y fácil de aprender.</li> <li>• Footprint bajo de memoria, bastante poderoso</li> </ul>	<ul style="list-style-type: none"> <li>• Sin experticia, configurar llega a ser un caos.</li> <li>• Es fácil de vulnerar sin proteccion adecuada.</li> <li>• El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.</li> </ul>

	<p>con una configuración adecuada.</p> <ul style="list-style-type: none"> <li>• Multiplataforma.</li> <li>• Capacidades de replicación de datos.</li> <li>• Soporte empresarial disponible.</li> </ul>	<ul style="list-style-type: none"> <li>• InnoDB genera mucho footprint en memoria al indizar.</li> <li>• El toolset empresarial tiene un costo adicional por suscripción anual.</li> <li>• Realizar revisiones llegar a ser una labor manual y tediosa para el DBA.</li> </ul>
MySQL	<ul style="list-style-type: none"> <li>• MySQL software es Open Source</li> <li>• Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.</li> <li>• Bajo costo en requerimientos para la elaboración de bases de datos</li> <li>• Facilidad de configuración e instalación.</li> <li>• Baja probabilidad de corromper datos.</li> <li>• Su conectividad, velocidad, y seguridad</li> </ul>	<ul style="list-style-type: none"> <li>• Un gran porcentaje de las utilidades de MySQL no están documentadas.</li> <li>• No es intuitivo, como otros programas (ACCESS).</li> </ul>

	<p>hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet</p> <ul style="list-style-type: none"> <li>• El software MySQL usa la licencia GPL</li> </ul>	
--	--	--

**Fuente:** Elaboración Propia

Para este proyecto se utilizara el gestor de base de datos de MySQL ya que es open source y que además es muy veloz su conectividad y existe una baja probabilidad de corromper los datos de MySQL además señalar que es la base de datos más utilizada para PHP contiene un sinnúmero de beneficios cuando se programa con PHP.

### 3.3.2 Realizar análisis de requerimientos para el módulo de usuarios.

Para poder realizar el módulo de usuarios primero debemos de crear una base de datos para poder registrar a los usuarios según el tipo que sean, luego se debe de diseñar la interfaz gráfica para que el usuario pueda ingresar al sistema y poder realizar las funciones que le corresponden dentro del sistema.

### 3.3.3 Descripción de Usuarios

Respetando los roles y funciones definidos en los nuevos procedimientos propuestos para el sistema web colaborativo de inventario de proyectos, que han sido aprobados, se describe las funciones de los actores a continuación.

- **Gerente General:** Cumple las siguientes funciones: Gestiona proyectos y se encarga de asignar los proyectos a las entidades públicas, Determina niveles de autorización de compras y autoriza ciertas compras.
- **Superintendente de obra:** Gestiona documentos de diseños y especificaciones de proyectos.
- **Administrador:** Las funciones que cumplirá serán las siguientes:  
Establece los flujos de trabajo y coordinación del sistema colaborativo.

Se encargará de administrar y asignar cuentas a los usuarios.

Establece los roles de trabajo dentro del sistema.

Establece los niveles de seguridad para todo el sistema, es decir; establece los permisos de operación a las distintas opciones del sistema.

- **Residente de obra:** Se encargara de:  
Realizar pedidos de material para un proyecto en la obra donde se encuentre.  
Adicionalmente puede realizar solicitudes de compra.
- **Jefe de Contabilidad:** Las funciones de este actor serán:  
Validar los costos de materiales y registrar costos de mano de obra e indirectos para cada proyecto.  
Consultar inventarios físicos y valorados.
- **Auxiliar de Contabilidad:** Las funciones de este actor serán:  
Realiza seguimiento a la gestión de compras, para ello consulta inventarios físicos y valorados, así como posibles proveedores de materiales a comprar por proyecto.
- **Jefe de Almacén:** Se encargará de poder:  
Realiza todo tipo de movimientos de almacén, incluyendo ingreso por compras, salidas para atender pedidos, devoluciones de compras, devoluciones de pedidos, transferencia de materiales y ajustes físicos de materiales.

#### 3.3.4 Sprint 1: Administración de usuarios

Una vez elegidas las prioridades en el product backlog del sistema se debe realizar un sprint backlog. En la tabla 9 se muestra el sprint backlog de Administración de Usuario.

**Tabla 9:** Sprint backlog de Administración de Usuario.

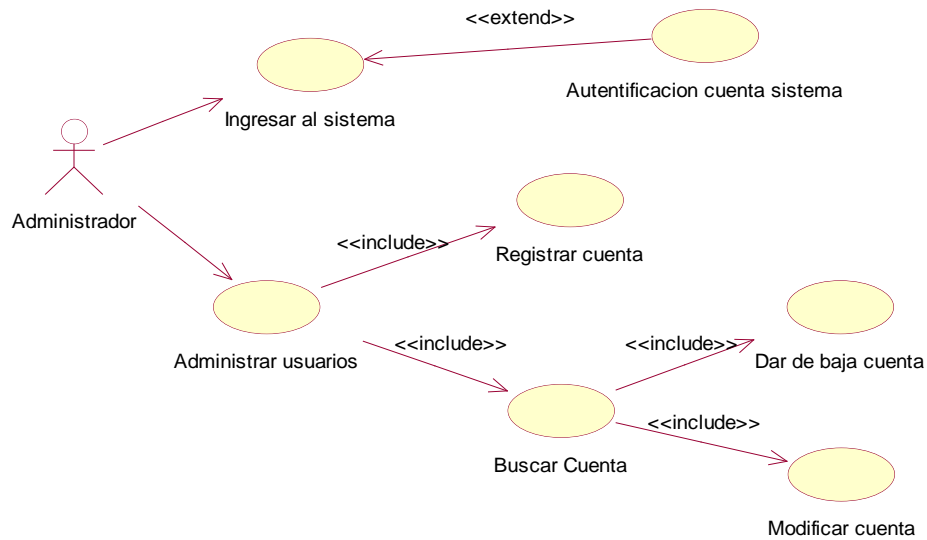
ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
1	Administrador	Administrar y asignar cuentas a los usuarios.	Que tengan acceso al sistema	8	<ul style="list-style-type: none"> <li>• Todos los campos de registro son obligatorios.</li> <li>• Cada usuario registrado se le asignara un login y contraseña.</li> <li>• Cada registro de usuario tiene que tener una fecha de alta.</li> <li>• La contraseña asignada no podrá ser mostrada en pantalla.</li> <li>• Las contraseñas estarán encriptados en la base de datos.</li> </ul>
2	Administrador	Establece los roles de trabajo dentro del sistema a los usuarios.	Que puedan trabajar solo con las tareas que les son asignadas.	5	<ul style="list-style-type: none"> <li>• Cada usuario registrado tendrá asignado un rol de trabajo.</li> <li>• Para cada usuario es obligatorio un rol.</li> </ul>
4	administrador	Establece los permisos de usuario a las distintas opciones del sistema.	Que cada uno de ellos pueda realizar las operaciones que le son permitidas	7	<ul style="list-style-type: none"> <li>• Cada usuario deberá tener permisos obligatorios</li> <li>• Los permisos deberán solo ser asignados por el administrador</li> <li>• Cada permiso solo mostrara una ventana diferente.</li> </ul>

**Fuente:** Elaboración propia.



### 3.3.5 Elaborar los diagramas de casos de uso.

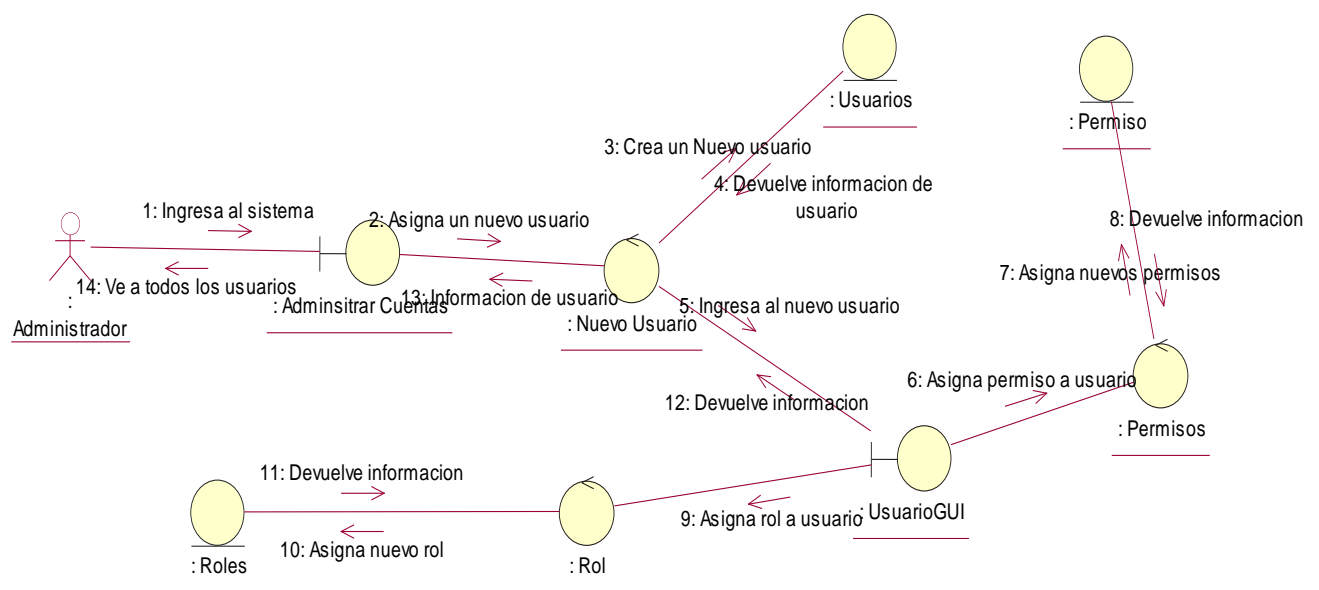
**Figura 21:** Diagrama de caso de uso del administrador.



**Fuente:** Elaboración Propia

### 3.3.6 Diagrama de colaboración

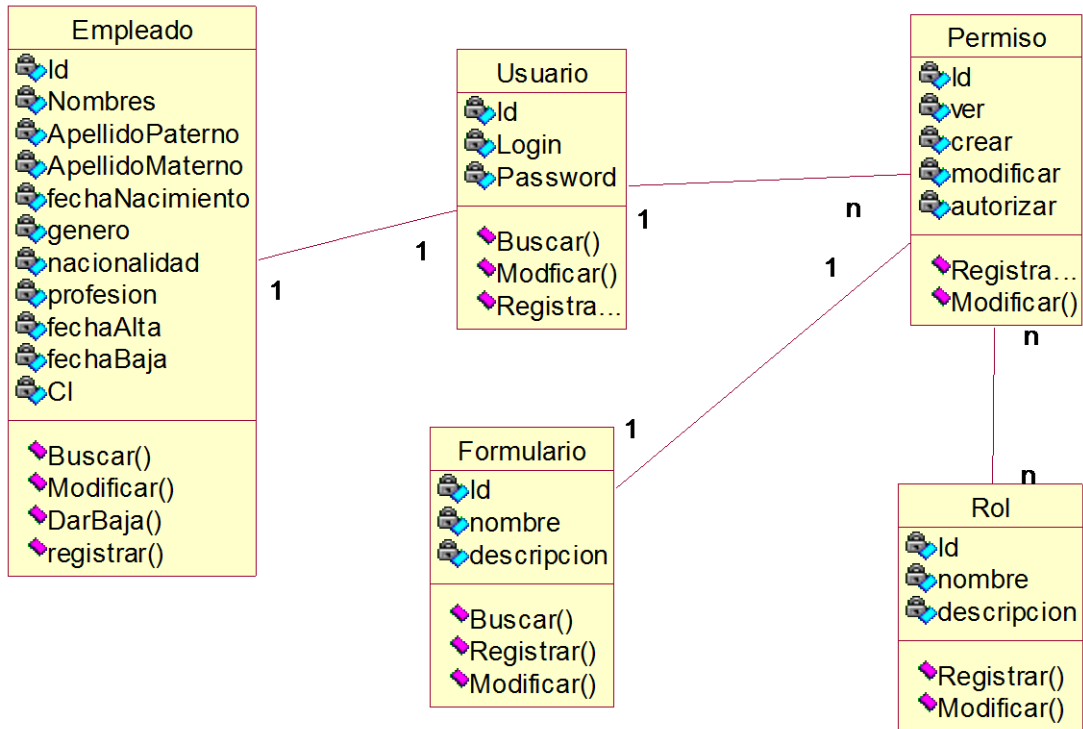
**Figura 22:** Diagrama de caso de colaboración.



**Fuente:** Elaboración Propia

### 3.3.7 Diagrama de clases

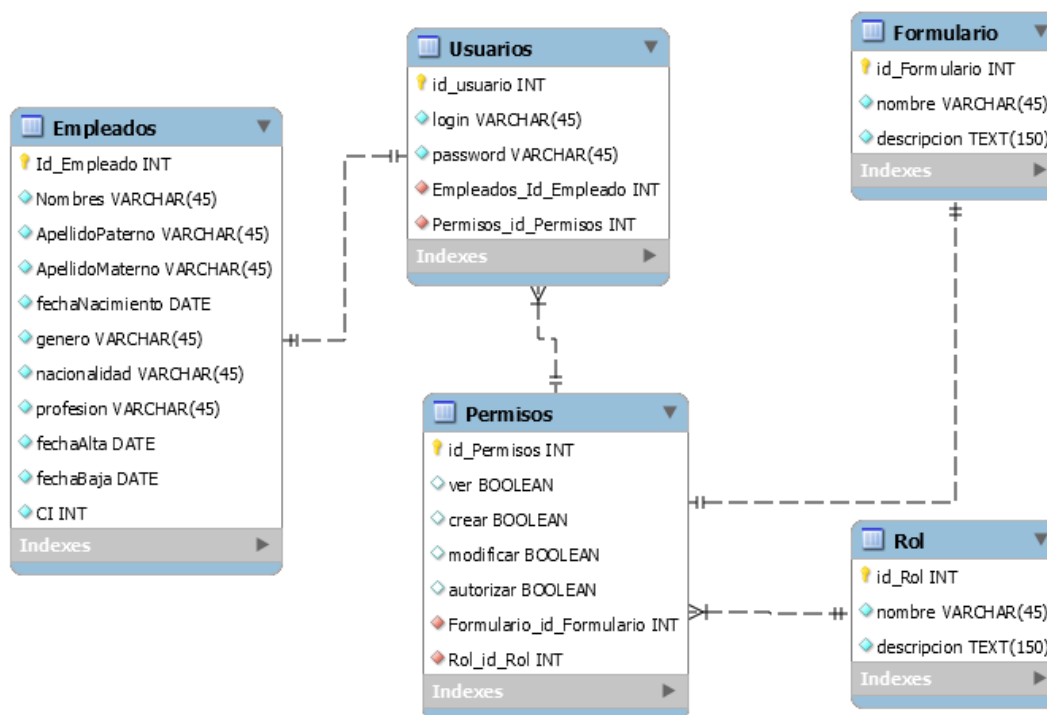
**Figura 23:** Diagrama de Clases.



**Fuente:** Elaboración Propia

### 3.3.8 Diseñar la base de datos.

**Figura 24:** Diagrama de base de datos.



**Fuente:** Elaboración Propia

### 3.3.9 Diccionario de datos

Se especificaran los campos de las tablas del diagrama de base de datos para la administración de usuarios con sus respectivas relaciones.

Nombre de archivo: BDInventario

Descripción: Tabla Empleados de la base de datos Inventario

**Tabla 10:** Diccionario de datos para Empleados.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Empleado	50	Entero	Clave Única de la tabla empleados
Nombres	50	Carácter	Nombres del empleado
ApellidoPaterno	20	Carácter	Apellido de parte del padre del empleado
ApellidoMaterno	20	Carácter	Apellido de parte de la madre del empleado
fechaNacimiento		Fecha	Fecha de nacimiento del empleado
Genero	9	Carácter	Genero del empleado
nacionalidad	20	Carácter	Nacionalidad del empleado
Profesión	20	Carácter	Profesión del empleado
fechaAlta	null	Fecha	Fecha que es contratado el empleado
fechaBaja	null	Fecha	Fecha que deja la empresa el empleado
CI	7	Entero	Carnet de identidad del empleado

**Fuente:** Elaboración Propia

**Tabla 11:** Diccionario de datos para Usuario.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Usuario	50	Entero	Clave Única de la tabla Usuario
Login	50	Carácter	El nombre con el que ingresa el usuario
Password	30	Carácter	Contraseña del usuario
Empleados_Id_empleado	50	Entero	Clave foránea de la tabla empleado
Permisos_id_permisos	50	Entero	Clave foránea de la tabla permiso

**Fuente:** Elaboración Propia

**Tabla 12:** Diccionario de datos para Permiso.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Permisos	50	Entero	Clave Única de la tabla Permiso
Ver		Booleano	Se da permiso solo para ver
Crear		Booleano	Se da permiso solo para crear
Modificar		Booleano	Se da permiso para modificar

autorizar		Booleano	Se da permiso para poder autorizar
Formulario_id_formulario	50	Entero	Clave foránea de formulario
Rol_id_rol	50	Entero	Clave foránea de rol

**Fuente:** Elaboración Propia

**Tabla 13:** Diccionario de datos para Rol.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Rol	50	Entero	Clave Única de la tabla Rol
Nombre	50	Carácter	Nombres del Rol
Descripción	120	Texto	Descripción de que hace cada rol

**Fuente:** Elaboración Propia

**Tabla 14:** Diccionario de datos para Formulario.

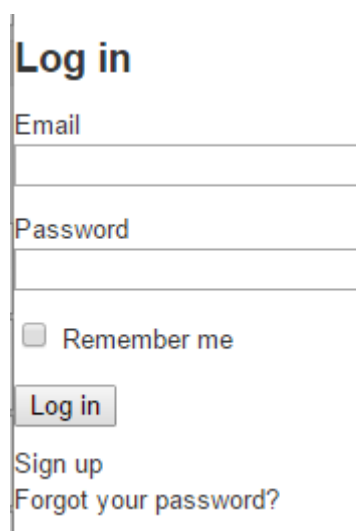
Campo	Tamaño	Tipo de Dato	Descripción
Id_Empleado	50	Entero	Clave Única de la tabla formulario
Nombres	50	Carácter	Nombres de cada formulario
Descripción	20	Carácter	Descripción de los formularios

**Fuente:** Elaboración Propia

### 3.3.10 Codificar módulo de gestión de usuarios.

Según los requerimientos de los usuarios y sus necesidades podemos hacer el desarrollo de algunas de las interfaces iniciales del sistema.

**Figura 25:** Interfaz de ingreso.



Log in

Email

Password

☐ Remember me

Log in

[Sign up](#)

[Forgot your password?](#)

**Fuente:** Elaboración Propia

Esta interfaz permite el ingreso de usuarios al sistema mediante un login y un password.

## 3.4 DISEÑAR E IMPLEMENTAR UN MÓDULO DE REGISTRO DE PROYECTOS EN GENERAL.

### 3.4.1 Identificar actores y objetivos involucrados

Para este módulo se identificaron los siguientes actores:

Administrador y Gerente General porque son los únicos actores que pueden

### 3.4.2 Sprint 2: Administración de proyectos.

Una vez elegidas las prioridades en el product backlog del sistema se debe realizar un sprint backlog. En la tabla 15 se muestra el sprint backlog de Administración de Usuario.

**Tabla 15:** Diccionario de datos para Formulario.

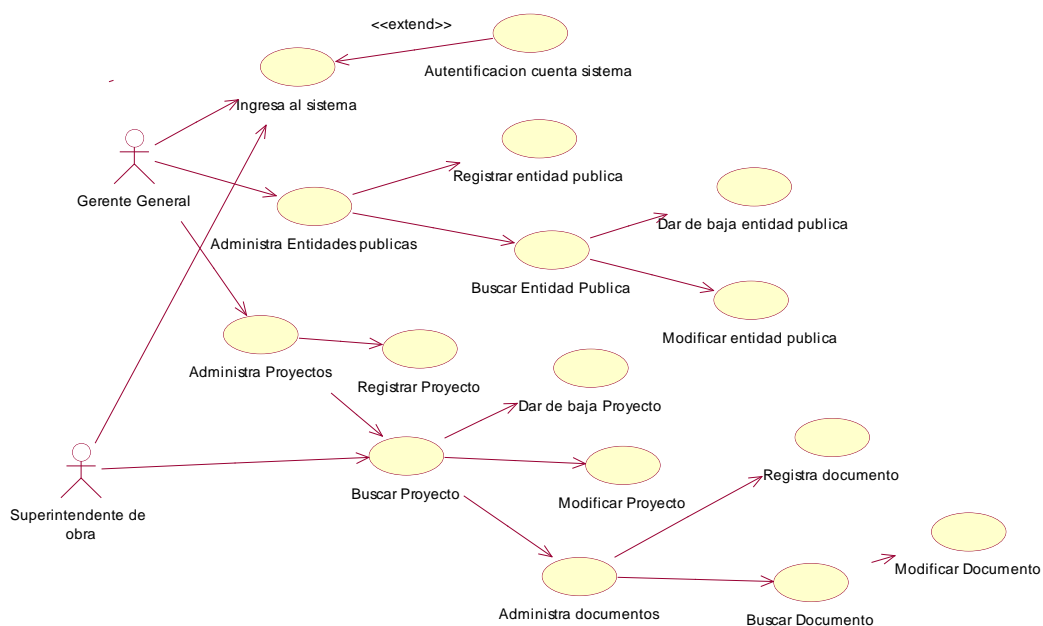
ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
1	Gerente General	Crear y modificar información de Entidades Públicas.	Saber qué proyectos asignar a estas Entidades Públicas.	8	<ul style="list-style-type: none"> <li>Todos los campos de registro son obligatorios.</li> <li>Solo el gerente general podrá registrar entidades publicas</li> </ul>
2	Gerente General	Crear y modificar información de proyectos para las Entidades públicas.	Que el sistema me dé información de compras y ventas de material de obras por proyecto.	5	<ul style="list-style-type: none"> <li>Los proyectos solo podrán ser registrados por el gerente general.</li> <li>Para cada usuario es obligatorio un número de control.</li> </ul>
	Superintendente	Guardar documentos de proyectos	Compartir y trabajar en colaboración con el resto de empleados		<ul style="list-style-type: none"> <li></li> </ul>
	Superintendente	Revisar y autorizar cambios a los documentos de diseños y especificaciones	Tener los documentos formalmente publicados		<ul style="list-style-type: none"> <li></li> </ul>

**Fuente:** Elaboración Propia



### 3.4.3 Diagrama de caso de uso

**Figura 26:** Diagrama de caso de Registro de Proyectos

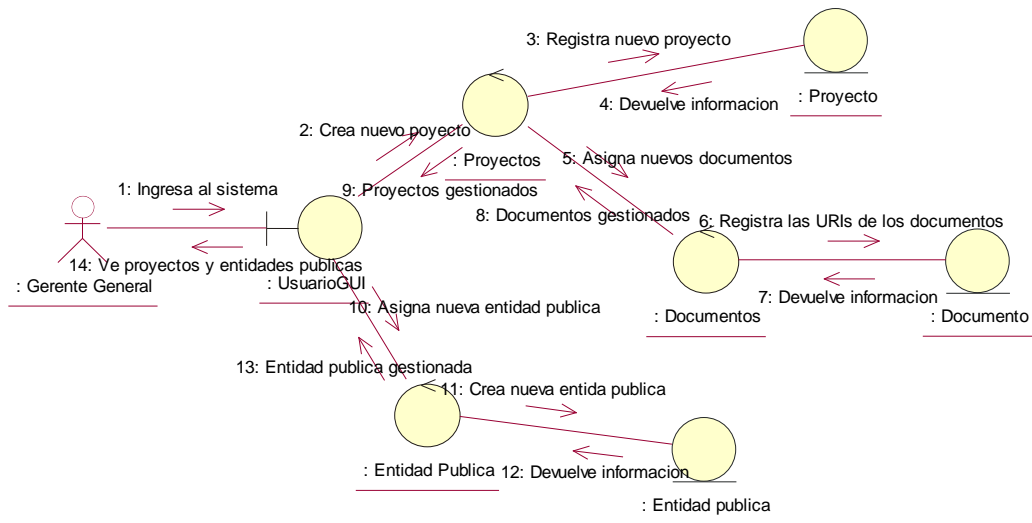


**Fuente:** Elaboración Propia

### 3.4.4 Diagrama de colaboración.

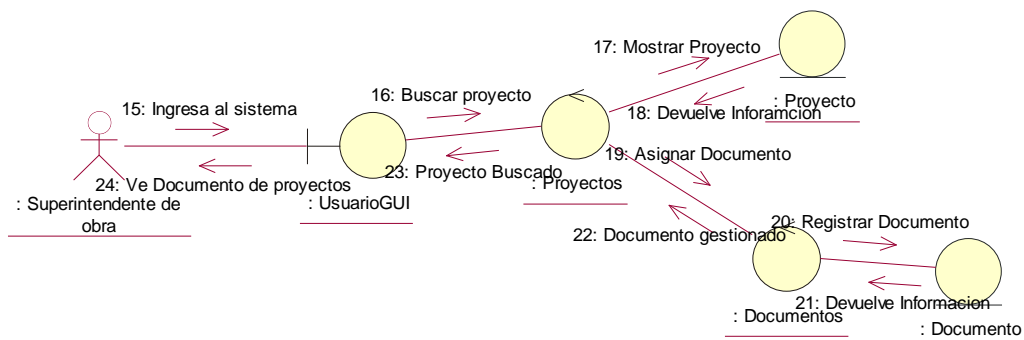
Para determinar las entidades públicas se debe de añadir la siguiente tabla a la base de datos y relacionarla con la tabla de proyectos.

**Figura 27:** Diagrama de colaboración Gerente General



**Fuente:** Elaboración Propia

**Figura 27:** Diagrama de colaboración Superintendente de obra

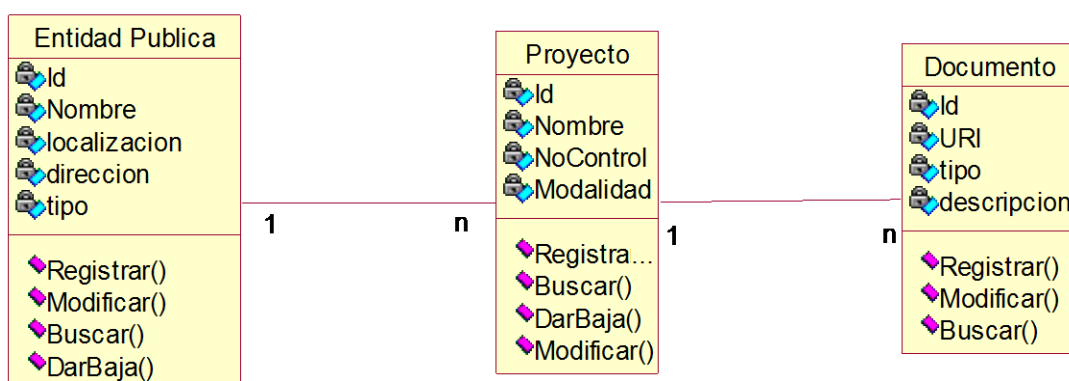


**Fuente:** Elaboración Propia

### 3.4.5 Diagrama de clases

Primero se debe de integrar las historias de usuario referentes a este módulo para y son los siguientes:

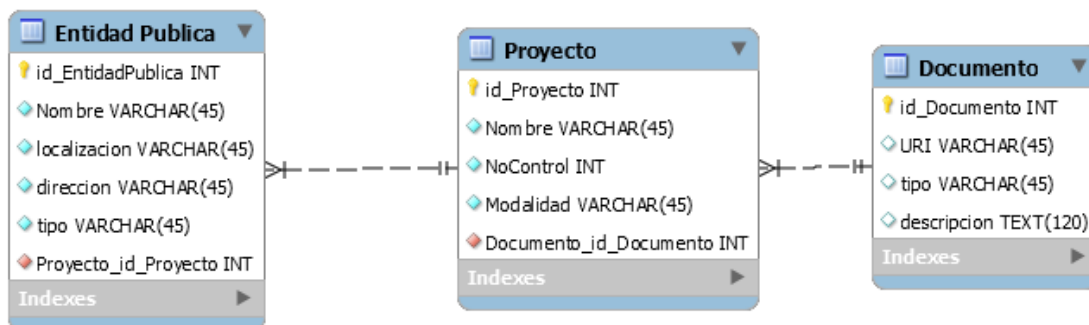
**Figura 28:** Diagrama de clases registro de proyectos



**Fuente:** Elaboración Propia

### 3.4.6 Diagrama de Base de datos

**Figura 29:** Diagrama de base de datos de registro de proyectos



**Fuente:** Elaboración Propia

### 3.4.7 Diccionario de datos

Se especificaran los campos de las tablas del diagrama de base de datos para la administración de usuarios con sus respectivas relaciones.

Nombre de archivo: BDInventario

Descripción: Tabla Entidad Pública de la base de datos Inventario

**Tabla 16:** Diccionario de datos para Entidad Publica.

Campo	Tamaño	Tipo de Dato	Descripción
Id_EntidadPublica	50	Entero	Clave Única de la tabla entidad publica
Nombre	50	Carácter	Nombres de la entidad publica
Localización	50	Carácter	Localización de la entidad publica
Dirección	50	Carácter	Dirección de la entidad publica
Tipo	50	carácter	Tipo de entidad
Proyecto_id_Proyecto	50	Entero	Clave foránea de proyecto

**Fuente:** Elaboración Propia

**Tabla 17:** Diccionario de datos para Proyecto.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Proyecto	50	Entero	Clave Única de la tabla proyecto
Nombre	50	Carácter	Nombres del proyecto
NoControl	20	Entero	Numero con el que se controla el proyecto

Modalidad	20	Carácter	Qué tipo de proyecto es
Documento_id_Documeto	50	Entero	Clave foránea de documento

**Fuente:** Elaboración Propia

**Tabla 18:** Diccionario de datos para Documento.

Campo	Tamaño	Tipo de Dato	Descripción
Id_Docuemnto	50	Entero	Clave Única de la tabla Documento
URI	50	Carácter	Dirección del servidor en la que se encuentran los documentos
Tipo	20	Carácter	Tipo de documento y formato
Descripción	120	Texto	Descripción de que tipos de documentos hay

**Fuente:** Elaboración Propia

#### 3.4.8 Codificar módulo de administración de proyectos

Según los requerimientos de los usuarios y sus necesidades podemos hacer el desarrollo de algunas de las interfaces de la parte de proyectos.

**Figura 30:** Interfaz de Proyectos

**SISTEMA DE INVENTARIO**

## Listing Proyectos

Nombre	Modalidad	Nrocontrol	Entidad publica	
Puentes	Construccion	877		Show Edit Destroy

[New Proyecto](#)

**Fuente:** Elaboración Propia

Esta interfaz muestra en primer lugar la lista de proyectos que están registrados en el caso de que exista alguno registrado tiene sus opciones de mostrar, editar y destruir dicho proyecto, sino existe ningún proyecto existe la opción de crear un nuevo proyecto.

**Figura 31:** Interfaz de Proyectos

## New Proyecto

Nombre

Modalidad

Nrocontrol

Entidad publica

[Back](#)

**Fuente:** Elaboración Propia

Esta interfaz es para registrar nuevos proyectos con sus campos específicos.

## 3.5 DESAROLLAR EL MODULO DE SOLICITUD DE PEDIDOS Y COMPRA DE MATERIAL

### 3.5.1 Identificar actores y objetivos involucrados

Para este módulo se identificaron los siguientes actores:

Residente de obra, Jefe de almacén, Gerente General

### 3.5.2 Sprint 3: Solicitud de pedidos y compra de material

Una vez elegidas las prioridades en el product backlog del sistema se debe realizar un sprint backlog para el módulo de solicitud de compras

**Tabla 19:** Sprint Backlog de solicitud de pedidos y compra de material

ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
1	Residente de obra	Realizar pedidos de material del proyecto que está encargado	Poder seguir con el proceso de construcción de la obra	8	<ul style="list-style-type: none"><li>Cada residente de obra podrá realizar pedido de material.</li><li>Cada residente de obra tendrá que esperar a la aprobación del pedido de material.</li></ul>
2	Residente de obra	Realizar solicitudes de compra	Tener lo más antes posibles los materiales para la obra	5	<ul style="list-style-type: none"><li>Cada residente de obra podrá realizar sus solicitudes de compra</li><li>Cada residente tendrá que esperar a la confirmación de la solicitud</li></ul>
4	Jefe de almacén	Saber todos los pedidos realizados por los residentes de obras.	Poder determinar todos los requerimientos de los ingenieros residentes.	7	<ul style="list-style-type: none"><li>Cada jefe de almacén podrá saber pedidos que de materiales que se necesiten para la obra</li><li>Cada jefe de almacén podrá ver reportes de pedidos de material</li><li>Cada jefe de almacén será notificado cada vez que exista nuevo pedido de material</li></ul>
3	Gerente General	Autorizar la solicitud de compra de materiales para los proyectos	Controlar las compras que se realizan para los proyectos	8	<ul style="list-style-type: none"><li>El gerente general podrá ver todos los reportes de solicitud de compras para materiales</li></ul>

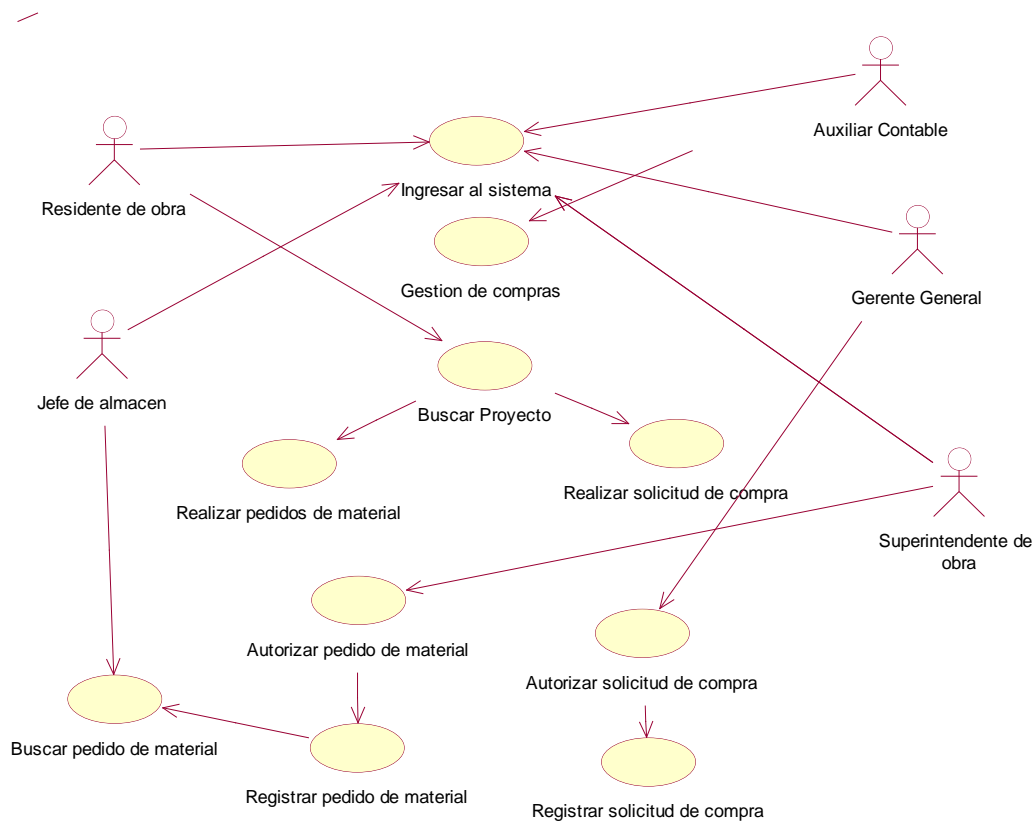
ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
					<ul style="list-style-type: none"> <li>Gerente general tendrá mensajes de advertencia para cada solicitud nueva que aparezca</li> </ul>
4	Superintendente	Autorizar los pedidos de materiales para los proyectos	Controlar los envíos de materiales a las obras por proyecto	8	<ul style="list-style-type: none"> <li>Superintendente tendrá la advertencia de autorizar pedido de materiales.</li> <li>Superintendente podrá ver que residente de obra está solicitando material.</li> </ul>
5	Auxiliar de contabilidad	Realizar seguimiento a la gestión de compras	Tener un control específico de todas las compras que se realizan	7	<ul style="list-style-type: none"> <li>Cada auxiliar de contabilidad podrá tener reportes constantemente actualizados.</li> <li>Cada auxiliar contable solo podrá ver los reportes no podrá modificar.</li> <li>Cada auxiliar contable tendrá será notificado cada vez que haya nueva actualización de reportes</li> </ul>

**Fuente:** Elaboración Propia



### 3.5.3 Diagrama de caso de uso

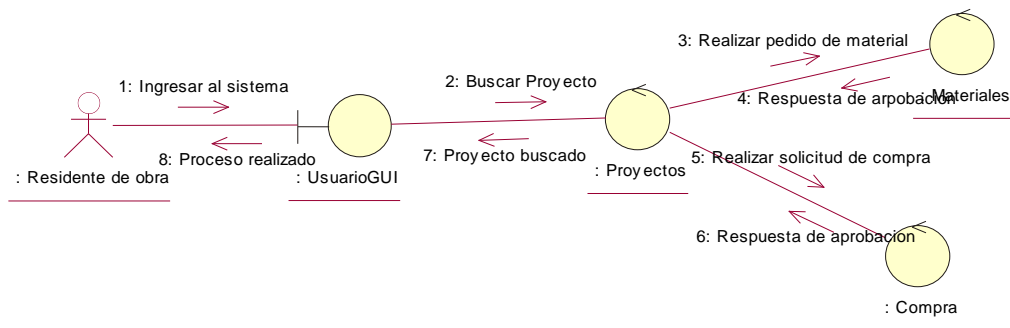
**Figura 32:** Diagrama de caso De solicitud de pedidos y compra de material



**Fuente:** Elaboración Propia

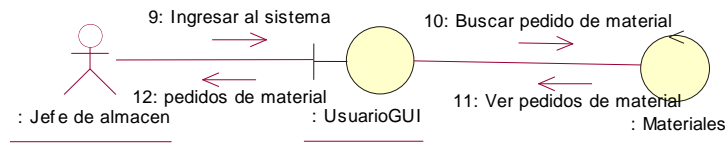
### 3.5.4 Diagrama de colaboración

**Figura 33:** Diagrama de colaboración de Residente de obra



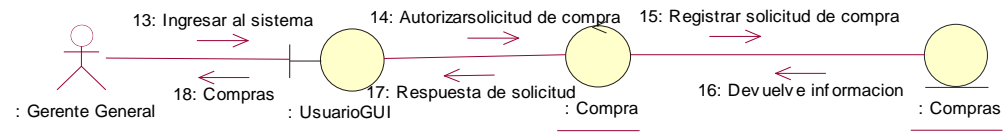
**Fuente:** Elaboración Propia

**Figura 34:** Diagrama de colaboración de jefe de almacén



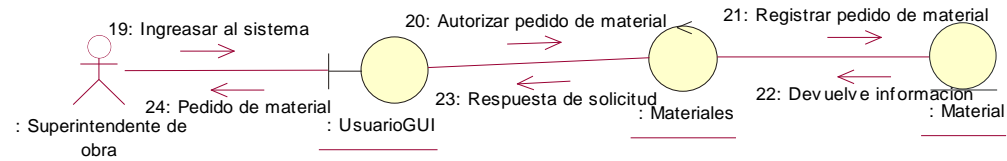
**Fuente:** Elaboración Propia

**Figura 35:** Diagrama de colaboración de Gerente General



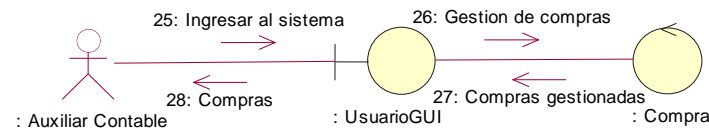
**Fuente:** Elaboración Propia

**Figura 36:** Diagrama de colaboración de Superintendente



**Fuente:** Elaboración Propia

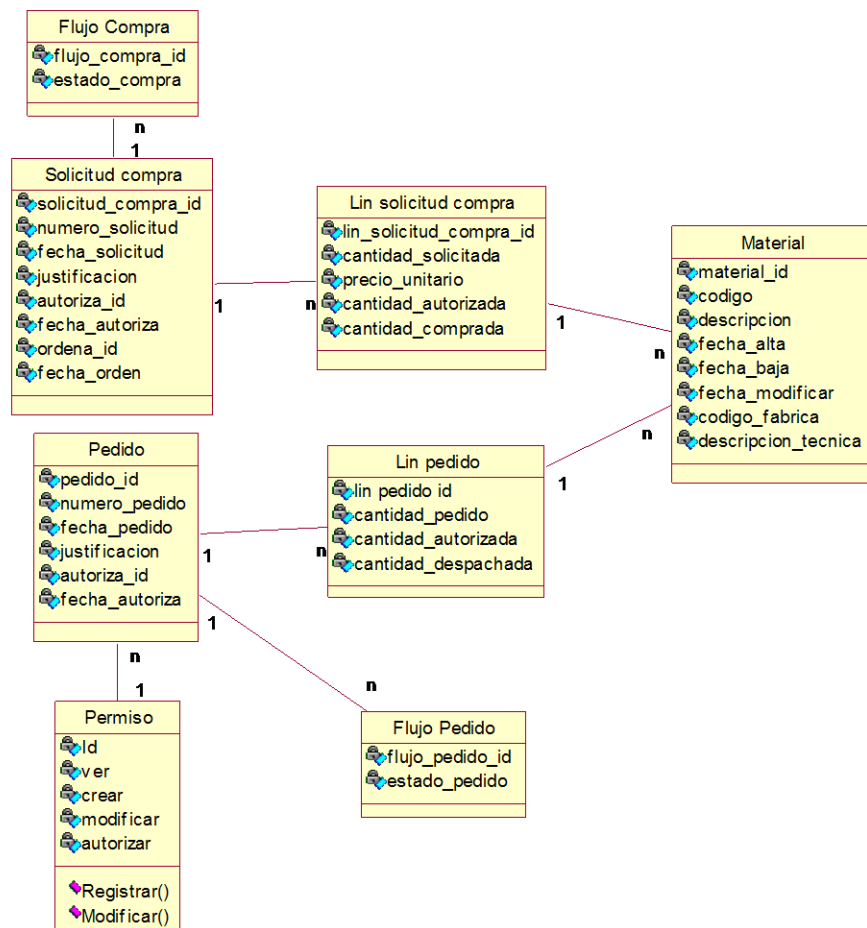
**Figura 37:** Diagrama de colaboración de Auxiliar contable



**Fuente:** Elaboración Propia

### 3.5.5 Diagrama de clases

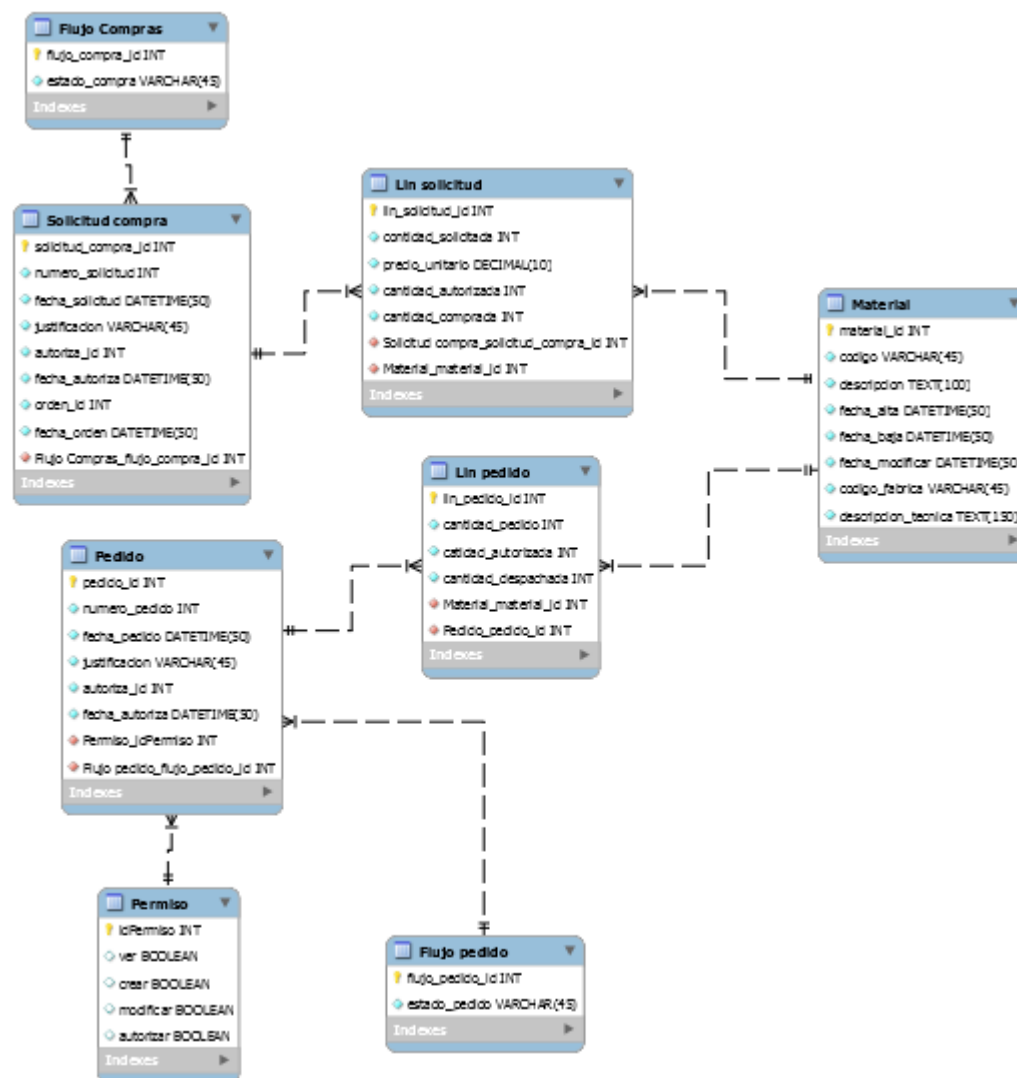
**Figura 38:** Diagrama de clases de solicitud de pedidos y compra de material



**Fuente:** Elaboración Propia

### 3.5.6 Diagrama de base de datos

**Figura 39:** Diagrama de base de datos de solicitud de pedidos y compra de material



**Fuente:** Elaboración Propia

### 3.5.7 Diccionario de datos

**Tabla 20:** Diccionario de datos flujo compra

Campo	Tamaño	Tipo de Dato	Descripción
Flujo_compra_id	50	Entero	Clave Única de la tabla flujo compra
Estado_compra	50	Carácter	Estado en el que se encuentra la compra

**Fuente:** Elaboración Propia

**Tabla 21:** Diccionario de datos solicitud de compras

Campo	Tamaño	Tipo de Dato	Descripción
Solicitud_compra_id	50	Entero	Clave Única de la tabla solicitud de compras
Numero_solicitud	50	Carácter	Numero con el que se registra la solicitud de compra
Fecha_solicitud	50	Fecha	Fecha en que se hace la solicitud
Justificación	120	Texto	Descripción de que tipos de documentos hay
Autoriza_id	50	Entero	Empleado que autoriza la solicitud de compra

Fecha_autoriza	50	Fecha	Fecha en la que se autoriza la solicitud de compra
Orden_id	50	Entero	Empleado que ordena la compra
Fecha_orden	50	Fecha	Fecha en la que se ordena la compra
Flujo_compra_id	50	Entero	Clave foránea de la tabla flujo compra

**Fuente:** Elaboración Propia

**Tabla 22:** Diccionario de datos Lin\_solicitud

Campo	Tamaño	Tipo de Dato	Descripción
Lin_solicitud_id	50	Entero	Clave Única de la tabla Documento
Cantidad_solicitada	100	Entero	Cantidad que se solicita en la compra
Precio_unitario	100	Decimal	Precio del material a comprar
Cantidad_autorizada	100	Entero	Cantidad que se autoriza para la compra
Cantidad_comprada	100	Entero	Cantidad que se compra

Solicitud_compra_id	20	Entero	Clave foránea de la tabla solicitud compra
Material_id	20	Entero	Clave foránea de la tabla material

**Fuente:** Elaboración Propia

**Tabla 23:** Diccionario de datos material

Campo	Tamaño	Tipo de Dato	Descripción
Material_id	50	Entero	Clave Única de la tabla material
Código	50	Carácter	Código del material que se le asigna en almacén
Descripción	50	Texto	Descripción del material
Fecha_alta	120	Fecha	Fecha en la que se registró el material
Fecha_baja	50	Fecha	Fecha en la que se eliminó material
Fecha_modificar	50	Fecha	Fecha en la que se modificó el material
Código_fabrica	50	Caracter	Código del material que viene por defecto desde la fabrica
Descripción_tecnica	50	Fecha	Descripción funcional del material

**Fuente:** Elaboración Propia

**Tabla 24:** Diccionario de datos pedido

Campo	Tamaño	Tipo de Dato	Descripción
Pedido_id	50	Entero	Clave Única de la tabla pedido
Numero_pedido	50	Carácter	Numero con el que se registra al pedido
Fecha_pedido	50	Fecha	Fecha en que se hace el pedido
Justificación	120	Texto	Breve descripción de la explicación del pedido
Autoriza_id	50	Entero	Empleado que autoriza la solicitud de compra
Fecha_autoriza	50	Fecha	Fecha en la que se autoriza la solicitud de compra
Flujo_pedido_id	50	Entero	Clave foránea de flujo_pedido

**Fuente:** Elaboración Propia



**Tabla 25:** Diccionario de Lin pedido

Campo	Tamaño	Tipo de Dato	Descripción
Lin_pedido_id	50	Entero	Clave Única de la tabla Lin pedido
Cantidad_pedido	50	Entero	Cantidad que se pidió
Cantidad_autorizada	20	Entero	Cantidad autorizada del pedido
Cantidad_despachada	120	Entero	Cantidad que se despacho
Material_id	20	Entero	Clave foránea de la tabla material
Pedido_id	20	Entero	Clave foránea de la tabla pedido

**Fuente:** Elaboración Propia**Tabla 26:** Diccionario de datos permiso

Campo	Tamaño	Tipo de Dato	Descripción
Permiso_id	50	Entero	Clave Única de la tabla permiso
ver	50	Booleano	Permiso para el usuario de poder ver
Crear	50	Booleano	Permiso para el usuario de poder crear
Modificar	120	Booleano	Permiso del usuario de poder modificar

Autorizar	50	Booleano	Permiso del usuario de poder autorizar
-----------	----	----------	--

**Fuente:** Elaboración Propia

**Tabla 27:** Diccionario de datos flujo pedido

Campo	Tamaño	Tipo de Dato	Descripción
Flujo_pedido_id	50	Entero	Clave Única de la tabla flujo pedido
Estado_pedido	50	Carácter	Estado en el cual está el flujo de pedido

**Fuente:** Elaboración Propia

### 3.5.8 Codificar módulo de solicitud de compras

## 3.6 DESARROLLAR EL MODULO DE MOVIMIENTOS DE INVENTARIOS

### 3.6.1 Identificar actores y objetivos involucrados

Para este módulo se identificaron los siguientes actores:

Jefe de almacén y Auxiliar de contabilidad

### 3.6.2 Sprint 4: Movimiento de inventarios

**Tabla 28:** Diccionario de datos permiso

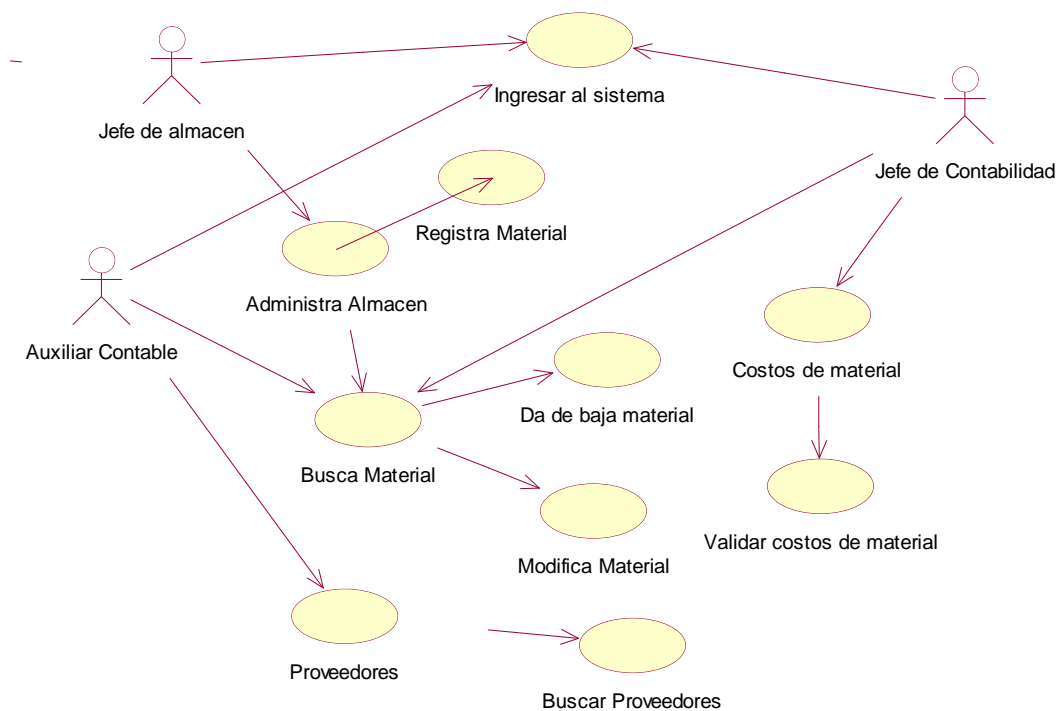
ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
1	Jefe de almacén	Crear y modificar información de los materiales	Tener un catálogo de materiales con los cuales se pueda realizar compras y despachos de materiales	8	<ul style="list-style-type: none"> <li>Cada jefe de almacén podrá gestionar la información de los materiales</li> <li>Cada jefe de almacén podrá tener un reporte de los materiales</li> <li>Cada jefe de almacén podrá ver un catálogo de los movimientos de inventarios</li> </ul>
2	Jefe de almacén	Categorizar los materiales	Agrupar los materiales por categorías	5	<ul style="list-style-type: none"> <li>Cada jefe de almacén podrá modificar la información de los materiales</li> <li>Cada jefe de almacén tendrá que agrupar en categorías a los materiales.</li> </ul>
4	Jefe de almacén	Tener información de cada uno de los materiales asignados a cada proyecto	Controlar existencias de materiales	7	<ul style="list-style-type: none"> <li>Cada jefe de almacén deberá poder acceder a la información de cada material asignado por obra para controlar almacén</li> <li>Cada jefe de almacén podrá solo ver la información de los proyectos asignados pero no poder modificarlo</li> </ul>
	Jefe de almacén	Un reporte de inventario físico valorado por proyecto o por Entidad pública	Entregar al contador y que él pueda actualizar su información financiera.	7	<ul style="list-style-type: none"> <li>Cada jefe de almacén podrá ver reportes de inventario con su valor respectivo respecto a cada proyecto</li> </ul>
	Jefe de contabilidad	Validar los costos de materiales	Tener una referencia del costo de los materiales	7	<ul style="list-style-type: none"> <li>Cada jefe de contabilidad podrá validar los costos de materiales</li> </ul>

ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
					<ul style="list-style-type: none"> <li>Cada jefe de contabilidad contara con una referencia para validar los costos</li> </ul>
	Jefe de contabilidad	Consultar inventarios físicos y valorados	Controlar el estado de los inventarios físicos y valorados	8	<ul style="list-style-type: none"> <li>Cada jefe de contabilidad podrá ver los inventarios.</li> <li>Cada jefe de contabilidad podrá verificar los estados de los cambios de inventarios</li> </ul>
	Auxiliar de contabilidad	Consultar inventarios físicos y valorados	Saber que hay dentro de los inventarios así como sus valorados	7	<ul style="list-style-type: none"> <li>Cada auxiliar contable podrá consultar los inventarios</li> <li>Cada auxiliar contable deberá tener seguimiento de inventarios.</li> </ul>
	Auxiliar de contabilidad	Consulta de proveedores de materiales	Saber que proveedores se tienen para cada material	7	<ul style="list-style-type: none"> <li>Cada auxiliar contable podrá consultar proveedores de material</li> <li>Cada auxiliar contable deberá estar actualizado en los proveedores</li> </ul>

**Fuente:** Elaboración Propia

### 3.6.3 Diagrama de caso de uso

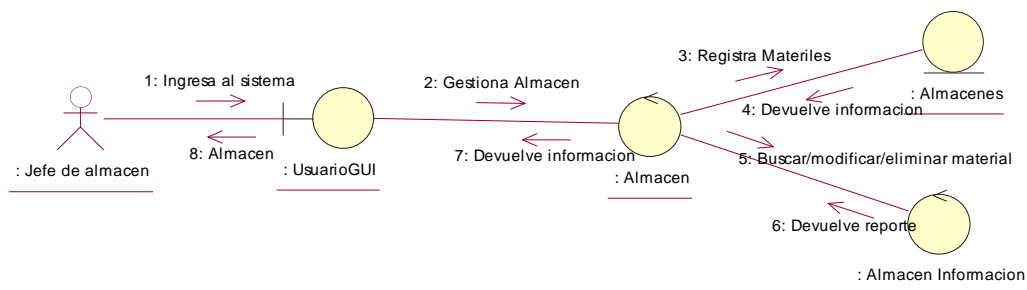
**Figura 40:** Diagrama de caso de uso de movimiento de inventarios



**Fuente:** Elaboración Propia

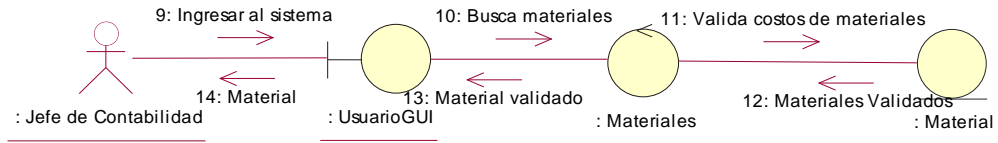
### 3.6.4 Diagrama de colaboración

**Figura 41:** Diagrama de colaboración de jefe de almacén



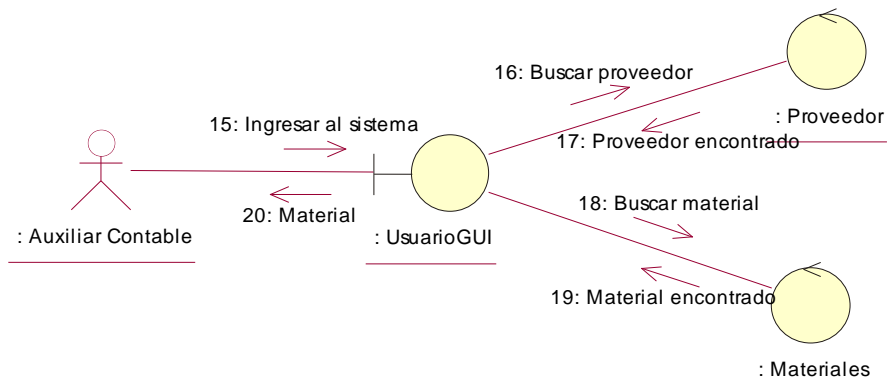
**Fuente:** Elaboración Propia

**Figura 42:** Diagrama de colaboración de jefe de contabilidad



**Fuente:** Elaboración Propia

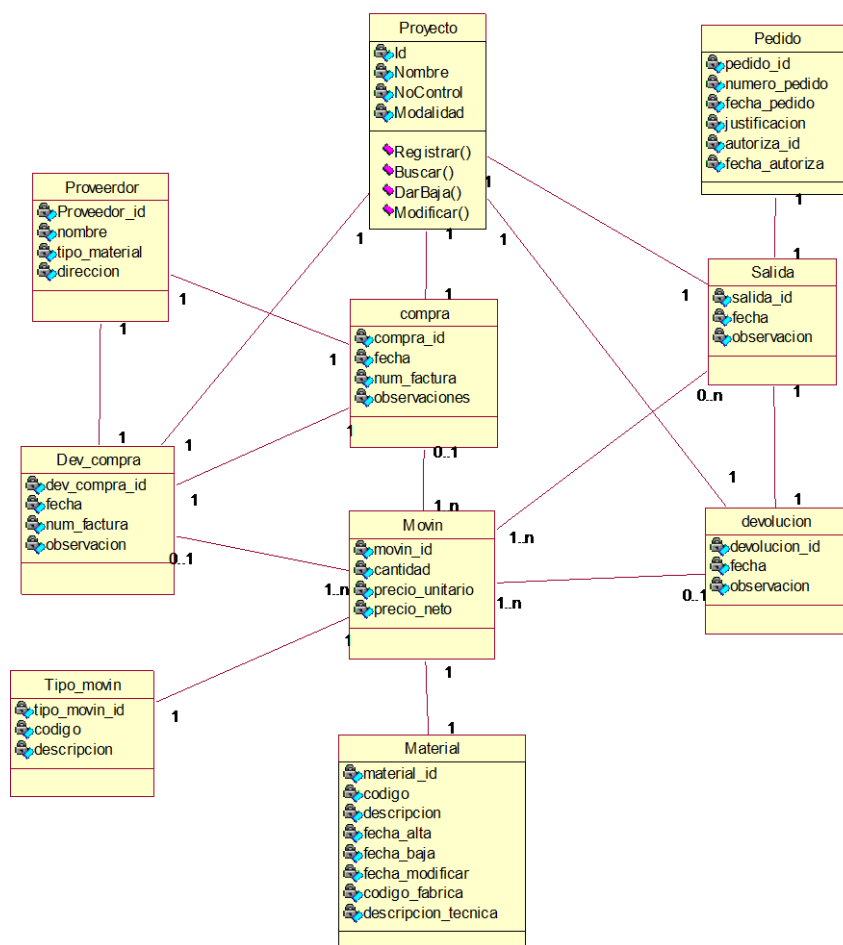
**Figura 43:** Diagrama de colaboración de auxiliar contable



**Fuente:** Elaboración Propia

### 3.6.5 Diagrama de clases

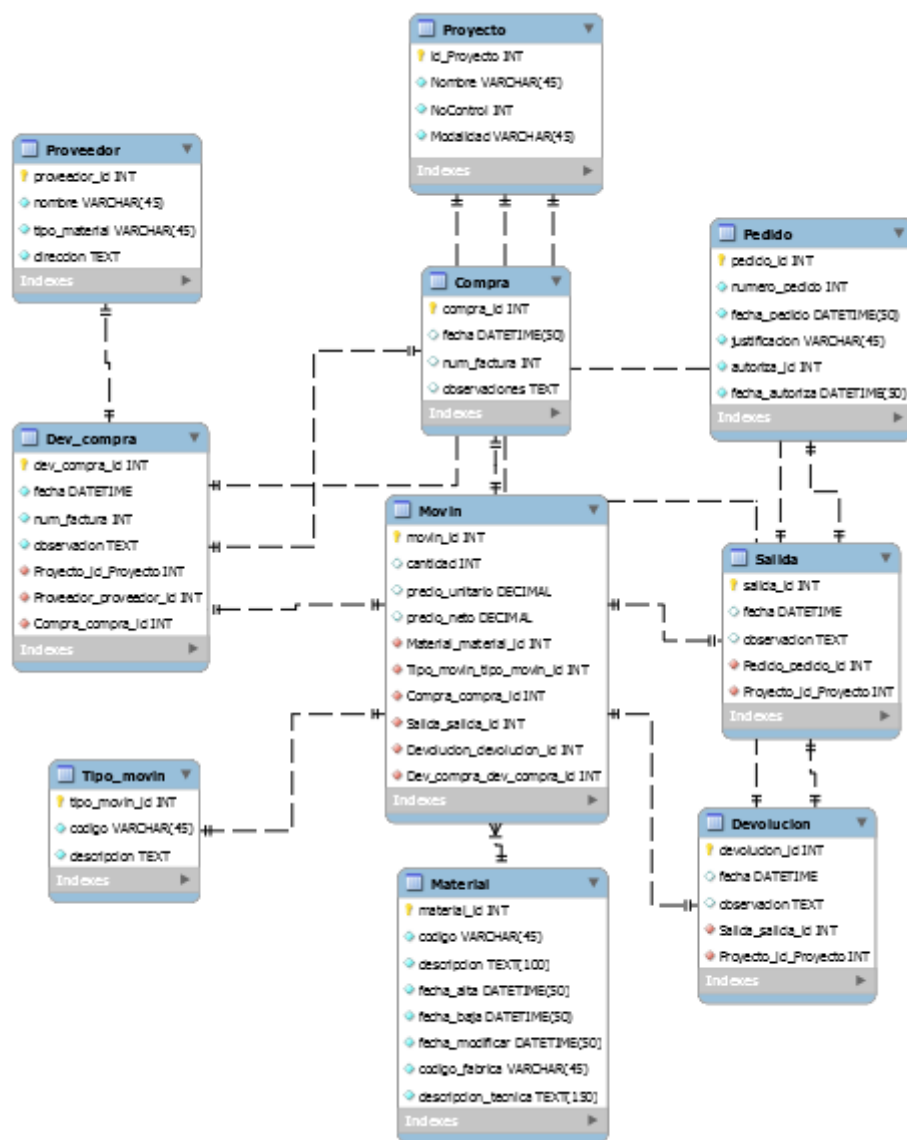
**Figura 44:** Diagrama de clases de movimiento de inventario



**Fuente:** Elaboración Propia

### 3.6.6 Diagrama de base de datos

**Figura 45:** Diagrama de base de datos de movimiento de inventario



**Fuente:** Elaboración Propia



### 3.6.7 Diccionario de datos

**Tabla 29:** Diccionario de datos proyectos

Campo	Tamaño	Tipo de Dato	Descripción
Proyecto_id	50	Entero	Clave Única de la tabla proyecto
Nombre	50	Carácter	Nombre del proyecto
Numero_control	50	Entero	Numero con el que se controla al proyecto
modalidad	120	Carácter	Modalidad en el que está el proyecto

**Fuente:** Elaboración Propia

**Tabla 30:** Diccionario de datos proveedor

Campo	Tamaño	Tipo de Dato	Descripción
Proveedor_id	50	Entero	Clave Única de la tabla proveedor
Nombre	50	Carácter	Nombre del proveedor
Tipo_material	50	Carácter	Tipo de material que reparte el proveedor
Dirección	120	Texto	Dirección del proveedor

**Fuente:** Elaboración Propia

**Tabla 31:** Diccionario de datos compra

Campo	Tamaño	Tipo de Dato	Descripción
Compra_id	50	Entero	Clave Única de la tabla compras
Fecha	50	Fecha	Fecha en el que se realiza la compra
Num_factura	50	Entero	Número de la factura de la compra
Observaciones	120	Texto	Descripción de las compra

**Fuente:** Elaboración Propia

**Tabla 32:** Diccionario de datos pedido

Campo	Tamaño	Tipo de Dato	Descripción
Pedido_id	50	Entero	Clave Única de la tabla pedido
Numero_pedido	50	Entero	Numero con el que se registra el pedido
Fecha_pedido	50	Fecha	Fecha en que se hace la solicitud de pedido
Justificación	120	Texto	Descripción de los pedidos que se realizan
Autoriza_id	50	Entero	Empleado que autoriza el pedido
Fecha_autoriza	50	Fecha	Fecha en la que se autoriza el pedido

**Fuente:** Elaboración Propia

**Tabla 33:** Diccionario de datos dev compra

Campo	Tamaño	Tipo de Dato	Descripción
Dev_compra_id	50	Entero	Clave Única de la tabla dev compra
Fecha	50	Fecha	Fecha de la devolución de la compra
Num_factura	50	Entero	Número de la factura de la compra
Observación	120	Texto	Descripción de la devolución de la compra
Proyecto_id	50	Entero	Clave foránea de la tabla proyecto
Proveedor_id	50	Entero	Clave foránea de la tabla proveedor
Compra_id	50	Entero	Clave foránea de la tabla compra

**Fuente:** Elaboración Propia

**Tabla 34:** Diccionario de datos movin

Campo	Tamaño	Tipo de Dato	Descripción
Movin_id	50	Entero	Clave Única de la tabla movin
Cantidad	50	Entero	Numero cuantificable de los movimientos
Precio_unitario	50	Decimal	Precio por unidad de movimiento
Precio_netto	120	Decimal	Precio casi total de movimiento
Material_id	50	Entero	Clave foránea de la tabla material
Tipo_movin_id	50	Entero	Clave foránea de la tabla tipo_mov
Compra_id	50	Entero	Clave foránea de la tabla compra
Salida_id	50	Entero	Clave foránea de la tabla salida
Devolución_id	50	Entero	Clave foránea de la tabla devolución
Dev_compra_id		Entero	Clave foránea de la tabla dev_compra

**Fuente:** Elaboración Propia

**Tabla 35:** Diccionario de datos salida

Campo	Tamaño	Tipo de Dato	Descripción
Salida_id	50	Entero	Clave Única de la tabla salida
Fecha	50	Fecha	Fecha de la salida
Observación	50	Texto	Descripción de la salida
Pedido_id	120	Entero	Clave foránea de la tabla pedido
Proyecto_id	50	Entero	Clave foránea de la tabla proyecto

**Fuente:** Elaboración Propia

**Tabla 36:** Diccionario de datos tipo movin

Campo	Tamaño	Tipo de Dato	Descripción
Tipo_movin_id	50	Entero	Clave Única de la tabla tipo movin
Código	50	Carácter	Código del tipo de movimiento
Descripción	50	texto	Descripción del tipo de movimiento

**Fuente:** Elaboración Propia

**Tabla 37:** Diccionario de datos material

Campo	Tamaño	Tipo de Dato	Descripción
Material_id	50	Entero	Clave Única de la tabla material
Código	50	Carácter	Código del material que se le asigna en almacén
Descripción	50	Texto	Descripción del material
Fecha_alta	120	Fecha	Fecha en la que se registró el material
Fecha_baja	50	Fecha	Fecha en la que se eliminó material
Fecha_modificar	50	Fecha	Fecha en la que se modificó el material
Código_fabrica	50	Caracter	Código del material que viene por defecto desde la fabrica
Descripción_tecnica	50	Fecha	Descripción funcional del material

**Fuente:** Elaboración Propia

**Tabla 38:** Diccionario de datos devolucion

Campo	Tamaño	Tipo de Dato	Descripción
Devolución_id	50	Entero	Clave Única de la tabla devolución
fecha	50	Fecha	Fecha de la devolución
Obseracion	50	Texto	Descripción de la devolución
Salida_id	120	entero	Clave foránea de la tabla salida
Proyecto_id	50	Entero	Clave foránea de la tabla proyecto

**Fuente:** Elaboración Propia

### 3.6.8 Codificar módulo de solicitud de movimiento de inventarios

### 3.7 DISEÑAR E IMPLEMENTAR EL MODULO COLABORATIVO DEL SISTEMA PARA GENERAR EL SEGUIMIENTO DE LA INFORMACION QUE FLUYE DE UN PERSONAL A OTRO.

#### 3.7.1 Identificar actores y objetivos involucrados

Para este módulo se identificaron los siguientes actores:

Administrador, Residente de obra, Jefe de almacén, Gerente General

#### 3.7.2 Sprint 5: Colaboración del sistema

**Tabla 39:** Product Backlog del módulo colaborativo

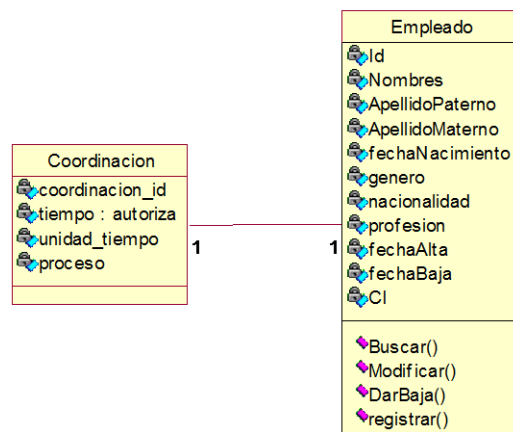
ENTREGA 1: Administración de Usuario					
Prioridad	Como	Necesito	Para	Esfuerzo	Criterios de Aceptación
1	Administrador	Establecer los flujos de trabajo y coordinación de los usuarios del sistema	Que los usuarios que colaboran entre si coordinen sus tareas.	8	<ul style="list-style-type: none"><li>El administrador dentro que establecer el trabajo y coordinación mediante notificaciones de los estados.</li><li>Los estados tendrán que ser en tiempo real</li><li></li></ul>

**Fuente:** Elaboración Propia



### 3.7.3 Diagrama de clases

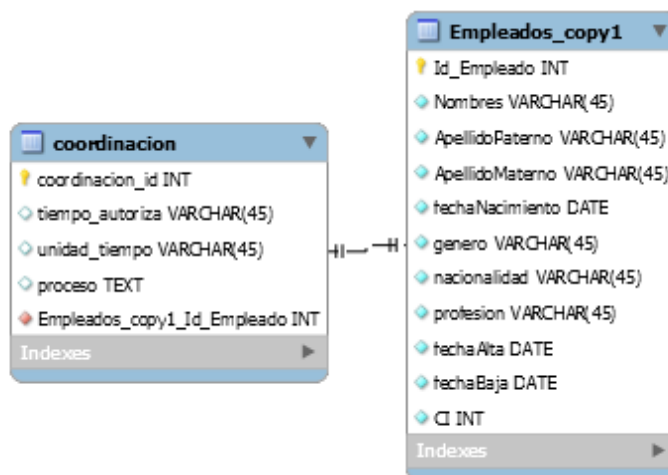
**Figura 46:** Diagrama de clases de módulo de colaboración



**Fuente:** Elaboración Propia

### 3.7.4 Diagrama de base de datos

**Figura 47:** Diagrama de clases de módulo de colaboración



**Fuente:** Elaboración Propia

### 3.7.5 Diccionario de datos

**Tabla 38:** Diccionario de coordinación

Campo	Tamaño	Tipo de Dato	Descripción
Coordinacion_id	50	Entero	Clave Única de la tabla coordinación
Tiempo_autoriza	50	Tiempo	Tiempo límite de autorización
Unidad_tiempo	50	Texto	Unidad de tiempo en el cual se autoriza
Proceso	120	Text	Proceso de coordinacion
Empleado_id	50	Entero	Clave foránea de la tabla empleado

**Fuente:** Elaboración Propia

**Tabla 38:** Diccionario de empleado

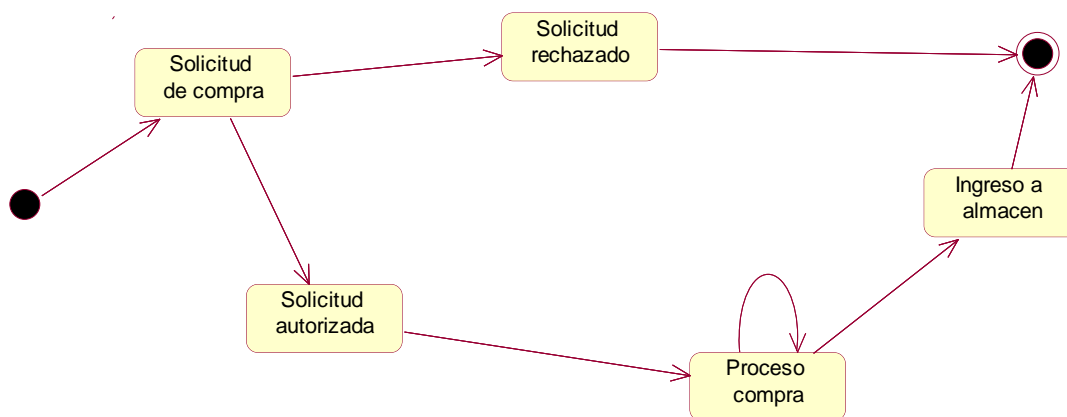
Campo	Tamaño	Tipo de Dato	Descripción
Id_Empleado	50	Entero	Clave Única de la tabla empleados
Nombres	50	Carácter	Nombres del empleado
ApellidoPaterno	20	Carácter	Apellido de parte del padre del empleado
ApellidoMaterno	20	Carácter	Apellido de parte de la madre del empleado

fechaNacimiento		Fecha	Fecha de nacimiento del empleado
Genero	9	Carácter	Genero del empleado
nacionalidad	20	Carácter	Nacionalidad del empleado
Profesión	20	Carácter	Profesión del empleado
fechaAlta	null	Fecha	Fecha que es contratado el empleado
fechaBaja	null	Fecha	Fecha que deja la empresa el empleado
CI	7	Entero	Carnet de identidad del empleado

**Fuente:** Elaboración Propia

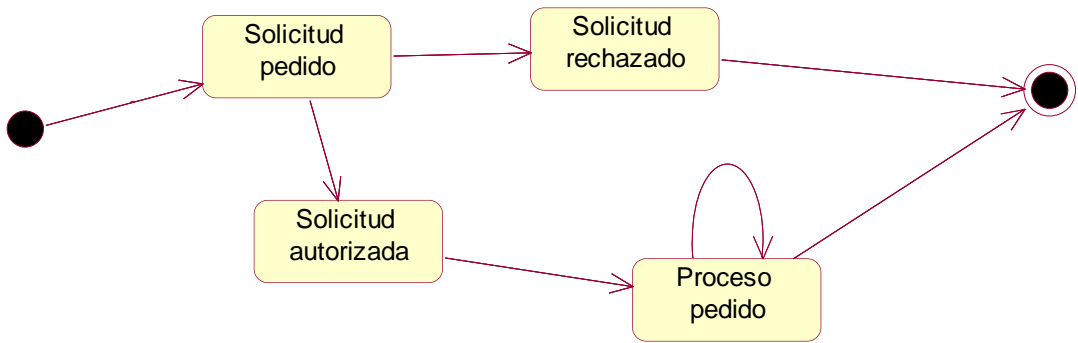
### 3.7.6 Diagrama de estados

Figura 48: Diagrama de estado de solicitud de compra



**Fuente:** Elaboración Propia

Figura 49: Diagrama de estado de solicitud de pedido



Fuente: Elaboración Propia

3.7.7 Codificar módulo de colaboración del sistema para generar seguimiento de la información que fluye de un personal a otro.

3.8 REALIZAR PRUEBAS AL SISTEMA FINAL

3.9 DEMOSTRACION DE HIPOTESIS

3.9.1 Demostración de la primera variable dependiente

Estado de los inventarios de proyectos

ESTADO DE PROYECTOS	PASOS SIN SISTEMA	PASOS CON SISTEMA
En estado de pedido de material	<ul style="list-style-type: none"><li>Residente hace pedido de material</li></ul>	
En estado de autorización	<ul style="list-style-type: none"><li>Gerente autoriza el pedido de material</li></ul>	

En estado de verificación	<ul style="list-style-type: none"> <li>Jefe de almacén verifica la llegada de material</li> </ul>	
---------------------------	---	--

### 3.9.2 Demostración de la segunda variable dependiente

Tiempo en los trámites de las planillas de inventarios

Procesos	Tiempo sin sistema	Tiempo con sistema
Revisión de material	2 horas	
Registro de material en planilla	2.5 horas	

### 3.9.3 Demostración de la tercera variable dependiente

Riesgo de pérdida y duplicidad de información de inventarios

AMENAZA	VULNERABILIDAD	P	I	R	CONTORL UTILIZADO	P	I	R
Administrativa registra parte de inventario	Información cambiada con datos falsos				Emplear permisos de acceso según cuenta de usuario			
Contabilidad registra parte de inventario	Información duplicada.				Emplear permisos de acceso según cuenta de usuario			
Manejo de reportes de inventarios	Pasa por varias manos				Emplear colaboración de trabajo			

### 3.9.4 Demostración de la variable independiente

Sistema web de control de inventario de proyectos aplicando un modelo de Sistema Colaborativo para coadyuvar al abastecimiento y a la presentación de informes para la solicitud de desembolsos

BENEFICIOS	SISTEMA ACTUAL	SISTEMA PROPUESTO
Permite registrar usuarios de cada parte de la empresa según su función con los inventarios	NO	si
Se realiza la asignación de roles a cada usuarios de acuerdo a la función que desempeña	NO	SI
Permite el control de estados de proyectos	NO	SI
Permite trabajo en equipo mediante sistema colaborativo	NO	SI
Permite la revisión en tiempo real de documentos	NO	SI

### 3.9.5 Definición de la hipótesis

Para definir la hipótesis se realiza un análisis de aceptación-rechazo, utilizando una variable  $P_n$  que representa una probabilidad. Para la hipótesis planteada tenemos lo siguiente:

$P_1$ : El procedimiento manual realizado para la solicitud de permisos, para conocer los días disponibles de vacación y el estado actual de las solicitudes.

$P_2$ : Los procesos realizados mediante el sistema web con un módulo de aplicaciones móviles (APPS) para el trámite de solicitud de permisos.

$H_0$ : Hipótesis nula  $\rightarrow$  En el caso en que los factores de éxito en cuanto agilidad en el trámite de permisos, conocimiento sobre el estado actual de una solicitud y conocimiento de la cantidad de días disponibles de vacación cuando no se usaba el sistema, sean iguales a los factores de éxito al utilizar el sistema.

$$H_0 : P_1 = P_2$$

$H_1$ : Hipótesis alterativa  $\rightarrow$  En caso de que los factores de éxito obtenidos al utilizar el sistema, sean diferentes a los factores por no utilizarlo.

$$H_1 : P_1 \neq P_2$$

En este caso, existen 2 posibilidades:

- Los factores de éxito por utilizar el sistema son mayor a los factores de éxito por no utilizarlo.

$$P_1 < P_2$$

- Los factores de éxito por utilizar son menores a los factores de éxito por no utilizarlo.

$$P_1 > P_2$$

### 3.9.6 Cálculo de estadístico T.

Mediante el cálculo del estadístico T, podemos determinar el grado de aceptación de la hipótesis, para ellos se utilizan las siguientes formulas:

$$P_n = \frac{X_n}{n}$$

Con n = número de factores tomados como beneficio (tabla No de la independiente)  
= 6

$$P_n = \frac{(P_1 * P_2) + (P_1 * P_2)}{n_1 + n_2}$$

$$g_t = n - 1$$

$$t = \frac{P_1 - P_2}{\sqrt{(\frac{P_1 * Q_1}{n_1}) + (\frac{P_2 * Q_2}{n_2})}}$$

Al reemplazar valores obtenemos:

$$P_1 = \frac{1}{6} \rightarrow P_1 =$$

$$P_2 = \frac{6}{6} \rightarrow P_2 =$$

Se puede observar claramente que  $P_2 > P_1$  lo que significa que la variable  $P_2$  es mejor que la variable  $P_1$ .

$Q_1$  y  $Q_2$  representan los porcentajes restantes de  $P_1$  y  $P_2$  respectivamente, que son calculados de la siguiente manera:

$$Q_1 = 1 - P_1 = 1 - \frac{1}{6} \rightarrow Q_1 = 0.8333$$

$$Q_2 = 1 - P_2 = 1 - \frac{6}{6} \rightarrow Q_2 = 0$$



Ahora podemos calcular el estadístico t de la siguiente manera:

$$t = \frac{P_1 - P_2}{\sqrt{\left(\frac{P_1 * Q_1}{n_1}\right) + \left(\frac{P_2 * Q_2}{n_2}\right)}}$$

$$t = \frac{0.1667 - 1}{\sqrt{\left(\frac{0.1667 * 0.8333}{6}\right) + \left(\frac{1 * 0}{6}\right)}}$$

$$t = -5.4772$$

Para obtener los rangos en la campana de Gauss, se utiliza un grado de error del 5% (debido a que es una muestra menor a 100) y se calculan los grados de libertad de la siguiente manera:

$$\alpha = 5\% = 0.05$$

$$\frac{\alpha}{2} = \frac{0.05}{2}$$

## Bibliografía

- Alvarez, M. A. (2002). *Visual Basic Script*.
- Andariel. (2013). *Sistema web, tiendas virtuales, catalogos web*.
- Baez, S. (2012). *Technology*.
- Bibbó, L. M. (2010). *Modelado de Sistemas Colaborativos*.
- Cabezas, L. (2004). *Manual Imprescindible de PHP5*. Madrid Anaya Multimedia.
- Campderrich Falgueras, B. (2003). *Ingenieria del software*.
- Camps, R., Casillas, A., Costal, D., Ginesta, M., Escofet, C., & Mora, O. (2005). *Base de Datos Software Libre*. Barcelona.
- Chaffey, D. (1998). *Groupware, Workflow and Intranets*.
- Eguiluz, J. (2003). *Introduccion a JavaScript*.
- et al, E. (1991). *Modelado de Sistemas Colaborativos*.
- Ferguson, J., Patterson, B., Beres, J., Boutquin, P., & Gupta, M. (2003). *La biblia del C#*. Madrid.
- Fernández Alarcón, V. (2006). *Desarrollo de Sistemas de Informacion Una metodologia basado en el modelado*.
- Fernando Alono, L. M. (2005). *Introduccion a la Ingenieria del Software Modelos de desarrollo de programas*.
- Garcia, J., & Rodríguez, J. (2000). *Aprende Java como si estuviera en primero*.
- Giardina, F. (2011). *ASP.NET Guia de Desarrollo de Aplicaciones y Paginas Web dinamicas*. Argentina.
- Griffths, A. (2010). *CodeIgniter 1.7*.
- Ian Sommerville, M. I. (2005). *Ingenieria del software*.
- Kelonye, M. (2014). *Mastering Ember.js*.
- Makarov, A. (2011). *Yii 1.1 Application Development Cookbook*.
- Mariño y Godoy. (2003). *Sistemas de Gestion de Informacion Universitarios Basados en Arquitectura Web Argentina: Congreso Argentino de Ciencias de la Computacion*.
- McCool, S. (2012). *Laravel Starter*.
- Naughton, P. (1996). *Manual de Java*. Madrid.
- Palenque, J. M. (2005). *Contabilidad y decisiones*.
- Quero Catalinas, E. (2003). *Sistemas operativos y lenguajes de programación*.
- Roemer, R. (2013). *Backbone.Js Testing*.

S. Pressman, R. (2010). *Ingeniería del Software Un enfoque práctico*.

SCS, C. (2008). *Servicios Web*.

Solís, C. (2015). *Manual del Guerrero: AngularJS*.

Universidad Tecnológica de la Mixteca. (2002). *Sistemas Colaborativos: Groupware & Workflow*.

Wesley, A., & Sommerville, I. (2004). *Software Engineering*.

Zapata, O. A. (2005). *Herramientas para elaborar tesis e investigaciones socioeducativas*.

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

## **ANEXOS**

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

**ANEXO “A”**  
**NOTA DE INGRESO Y SALIDA DE ALMACEN**

**COCHABAMBA, 2015**



**"SUÁREZ"**  
**CONSULTORIA & CONSTRUCCIÓN**

**NI :**

## NOTA DE INGRESO AL ALMACEN

PROYECTO/ALMACEN : \_\_\_\_\_

FECHA : \_\_\_\_\_

PROVEEDOR : \_\_\_\_\_ FACT/N.ENTREGA: \_\_\_\_\_

ORDEN DE COMPRA

Nº: \_\_\_\_\_ FECHA: \_\_\_\_\_

NOTA DE REMISION Nº \_\_\_\_\_ FECHA: \_\_\_\_\_

INGRESO POR COMPRA:

☐  
☐

INGRESO POR TRASPASO:

☐  
☐

INGRESO POR DEVOLUCION :

INGRESO POR INVENTARIO

(Tickear X = en la casilla correspondiente)

Observaciones :

NRO.	CODIGO	Fact.	DETALLE	CANT.	U/M	P.UNIT	TOTAL BS.
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

Bs.,

RECIBIDO POR :

ENTREGADO POR :

CONTABILIDAD

Nombre y Firma

Nombre y Firma

Nombre y  
Firma



**"SUAREZ"**

**CONSULTORIA & CONSTRUCCIÓN**

**NS**

## NOTA DE SALIDA DEL ALMACEN

FECHA : \_\_\_\_\_

MATERIALES :

☐  
☐

(Tickear X = en la casilla correspondiente)

EQUIPO/MAQ./HERRAM:

ALMACEN ORIGEN : \_\_\_\_\_

ALMACEN DESTINO: \_\_\_\_\_

N°	CÓDIGO	DETALLE	CANT.	U/M	P.UNIT	P.TOTAL	OBSERVACIONES
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							

AUTORIZADO POR	DESPACHADO POR ALMACENES:	TRANSPORTE POR :	RECEPCION POR :	CONTABILIDAD

Nombre y Firma      Nombre y Firma      Nombre y Firma      Nombre y Firma      Nombre y Firma

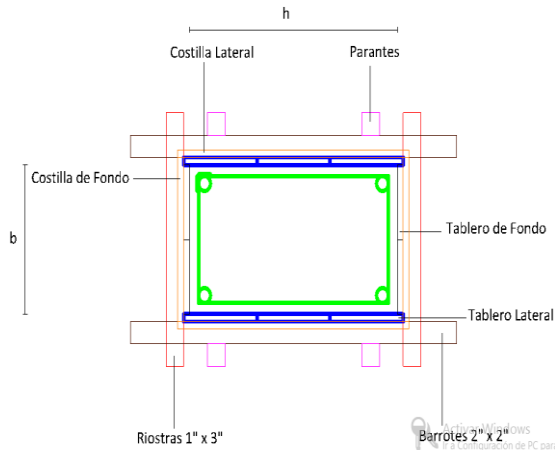
ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

**ANEXO “B”  
COMPUTOS METRICOS**

**COCHABAMBA, 2015**



# PLANILLA CALCULO DE ENCOFRADOS DE COLUMNAS



## Dimensiones de La Columna

H=	2,2	m
h=	35	cm
b=	20	cm

## Dimensiones

- Tablero de fondo = 1 Pulgada
- Tablero de lateral = 1 Pulgada
- Costillas de Fondo = 1 Pulgada \* 3 Pulgada Cada 60 Centimeros
- Costillas de Lateral = 1 Pulgada \* 3 Pulgada Cada 60 Centimeros
- Barrotes = 2 Pulgada \* 2 Pulgada Cada 50 Centimeros Longitud = 60 Centimeros
- Riostras = 1 Pulgada \* 3 Pulgada
- Flechas = 4 Bolillos \* 2,08 Metros de Altura Inclacion del bolillo = 45 Grados Diam(Pug)= 4
- Tabla(Bridas) = 1 Pulgada \* 4 Pulgada Longitud = 30 Centimeros
- Parantes = 2 Pulgada \* 2 Pulgada

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

**ANEXO “C”**  
**COSTOS DIRECTOS INVENTARIOS**

**COCHABAMBA, 2015**

# CONTROL DE INVENTARIO

FECHA	COMPRAS			VENTAS		
	Cantidad	Costo Unidad	Costo Total	Cantidad	Costo Unidad	Costo Total
01-mar						
05-mar	12.000	1.100	13.200.000			
08-mar			-	12.000	1.100	13.200.000
				2.000	1.000	2.000.000
12-mar	13.000	900	11.700.000			-
18-mar			-	13.000	900	11.700.000
				1.000	1.000	1.000.000
23-mar	8.000	1.200	9.600.000			-
26-mar	7.000	1.300	9.100.000			-
29-mar			-	7.000	1.300	9.100.000
				7.000	1.200	8.400.000
				1.000	1.200	1.200.000
				1.000	1.000	1.000.000
	43.600.000			44.000		37.000.000

Supuestos
Cantidad Vendida
44.000
Precio de Venta
6000
Ing Operacionales
100.000
Egre No Operac
100.000
Tasa Impositiva
35,00%

Estado de Resultados	
Método UEPS	
Ventas Brutas	264.000.000
- Costo de Ventas	37.000.000
Utilidad Bruta en Ventas	227.000.000
+ Ingresos No-Operacionales	100.000
- Gastos No-Operacionales	100.000
= Utilidad antes de Impuestos	227.000.000
- Impuestos	79.450.000
Utilidad	147.550.000

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

**ANEXO “D”  
ENTREVISTAS**

**COCHABAMBA, 2015**

## ENTREVISTA

1. ¿Cómo es que se administran los proyectos?

Lo administramos según a nuestras dos áreas que son la administrativa y el área técnica.

2. ¿Qué funciones tiene cada área en la constructora?

El área administrativa se encarga de todas las planillas del proyecto así como las más importantes de presupuestos, costos y algunos más, la parte técnica se encarga de supervisar las obras en el campo y maneja las planillas de cómputos métricos

3. ¿Al momento de administrar un proyecto cual es el problema?

El problema consiste en que tanto la parte técnica como la administrativa no tienen comunicación y eso ocasiona muchos errores al momento de administrar los proyectos

4. ¿Cuántos roles tiene la constructora en la parte técnica y parte administrativa?

Existen cerca de 7 a 8 roles de los cuales uno es el Gerente General y 3 roles de cada lado ósea de la parte Técnica y la parte Administrativa

5. ¿Cómo manejan la parte de inventarios?

Lo manejan manualmente, no existe un manejo en datos o documentos.

6. ¿Cómo se relacionan los ítems con los materiales?

Se relacionan en que un ítem tiene varios materiales, en otras palabras los ítems son como estándares que colocan las constructoras según el análisis económico que tienen acerca del precio de los materiales.

7. ¿Quién aprueba las solicitudes?

Todo pasa por manos del Gerente General.

ESCUELA MILITAR DE INGENIERÍA  
MCAL. ANTONIO JOSÉ DE SUCRE  
“BOLIVIA”

**ANEXO “E”**  
**TERMINOLOGIA DE SISTEMAS COLABORATIVOS**

**COCHABAMBA, 2015**

## TERMINOLOGIA

- Rol

Conjunto de propiedades conocimientos y responsabilidades que tendrá un usuario en un determinado momento. Permitirá entender cuál será el papel que el usuario tendrá en todo momento cuando interactúe con otros usuarios en el sistema.

- Usuario

Representa a un usuario dentro del sistema. Contará con un conjunto de propiedades que lo identificarán. Por ejemplo un nombre, un nickname, su imagen y otros datos que identifiquen al usuario.

- Objetos Colaborativos

Son los elementos creados por los propios usuarios que se pueden manipular en forma colaborativa gracias a las herramientas colaborativas que lo soportan. De tal manera que un usuario puede utilizar elementos creados por otros usuarios. Los objetos colaborativos más comunes en los ambientes groupware son documentos de texto, dibujos, páginas web o chat\_logs.

- Workspace

Es el lugar en el que la colaboración se lleva a cabo y define, en parte, el estilo de colaboración que se va a llevar a cabo. En este trabajo se rescata a los workspaces como los elementos que contextualizan la colaboración.

- Sesión

Una sesión es un periodo de interacción soportada por un sistema. En general un usuario ingresa en la sesión identificándose a través de un nombre de usuario y password y sale de la misma en forma explícita o simplemente cerrando la aplicación que maneja la sesión. La administración de la sesión entre otras cosas sirve para llevar un registro de las actividades de los usuarios mientras dura la

sesión y generalmente se registra el día y la hora en que los usuarios se conectan a la misma.

- Herramientas colaborativas

Son programas que, a diferencia de las aplicaciones monousuarios, completan ser utilizados por un grupo de personas. Naturalmente, el grupo manipula elementos que llamaremos objetos colaborativos que son los productos que se obtienen como resultado de la colaboración.

- Escenario Colaborativo

Es la integración de un conjunto de workspaces. A menudo, las aplicaciones groupware complejas necesitan más de un workspace, de una manera similar a una universidad virtual que estará compuesta por distintos workspaces con el aula, la biblioteca, etc. Los escenarios contienen asimismo los protocolos que estructuran el acceso y el uso de los diferentes workspaces por parte de los distintos roles existentes.

- Asociación

Permite asociar elementos del diseño. Con esta asociación vinculamos las herramientas que disponemos en un workspace, los roles que pueden participar de una sesión etc. Existen diversas asociaciones entre los elementos estos últimos elementos que se expresaran en el método a través de Asociaciones colaborativas.

- Contexto compartido

Es el repositorio donde se alojan los objetos colaborativos. Los usuarios tendrán acceso a este repositorio para crear, leer y modificar los objetos colaborativos y de acuerdo a la política de cada sistema colaborativo el contexto compartido puede ser único. En otros casos puede haber varios contextos compartidos, asociados a alguna componente del sistema.



- Telepuntero

Es el cursor del mouse de cada uno de los usuarios que está conectado a un objeto colaborativo y que puede ser movido por cada usuario. En algunos casos los cursores de los usuarios pueden ser visualizados todos al mismo tiempo. En otros casos, se ve solamente uno por vez, pero gracias al protocolo de colaboración los usuarios van cambiando el control sobre el cursor. Este recurso permite que un usuario señale alguna parte del objeto colaborativo en el transcurso de la colaboración.

- Protocolo

Un factor importante en el trabajo en grupo es el proceso social que se lleva a cabo. Sin gente interactuando, el sistema groupware está muerto. Los protocolos sirven para modelar, guiar y estructurar el proceso social que se lleva a cabo dentro del grupo. Los protocolos definen que herramientas y objetos colaborativos pueden ser utilizados por los distintos roles o usuarios. Un aspecto importante que define un protocolo es en qué momento puede participar cada usuario.

- Vista

Es una porción del contexto compartido que puede ser visualizada por un usuario. En algunos casos, los usuarios pueden visualizar todo el contexto compartido y en ese caso la vista es una representación del contexto compartido. En otros casos, por limitaciones de los dispositivos, la vista representa una porción de la información. La vista de cada usuario puede ser muy diferente.

- Acoplamiento

El acoplamiento es una medida que determina el grado de ensamble que tiene la misma componente en los puestos de trabajo de un sistema groupware. Las componentes pueden estar acopladas o no y el acoplamiento lo mencionaremos como un acoplamiento fuerte o débil. Se decimos que el sistema tiene la vista fuertemente acoplada, significa que los usuarios tienen la misma vista. Si tienen el workspace débilmente acoplado, estamos diciendo que los usuarios pueden estar

en distintos workspaces. El mismo concepto puede aplicarse a otros conceptos como el protocolo, la herramienta, el telepointer, etc. Esta medida determina en cierta forma la complejidad del sistema y de la colaboración.

- Awareness

Es la información que el sistema provee sobre el estado de la colaboración. En las reuniones presenciales estar al tanto de los otros es algo natural. Se puede percibir donde está ubicado cada uno, cuál es su estado, que actividad está desarrollando y con qué objeto. Por el contrario, mantener actualizada esta información en sistemas groupware es bastante difícil. Es por esto que los primeros sistemas groupware, que no mantenían esta información de manera eficiente, resultaban confusos e ineficientes comparados con el trabajo cara a cara. El awareness del workspace involucra mantener constantemente actualizada la información de los otros usuarios en relación al espacio compartido indicando al menos la identidad y la presencia de los usuarios.

- Avatar

Es la representación de un usuario dentro del sistema. Puede ser una pequeña imagen, un gráfico o un ícono. En algunos casos, como en los ambientes virtuales, el avatar se mueve dentro del ambiente y sirve para iniciar colaboraciones con ese usuario. El avatar puede servir como información básica de awareness y en muchos sistemas colaborativos indican la presencia y la ubicación de los usuarios dentro de workspaces.