

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA
GESTIÓN DE UNA EMPRESA DEDICADA A LA COMERCIALIZACIÓN DE
REPUESTOS PARA VEHÍCULOS AUTOMOTORES EN LA CIUDAD DE
CARACAS”**

REALIZADO POR:

Vanessa S, Cuzzi D

Mitchel K, Palomino W

Trabajo de Grado Presentado como Requisito Parcial para Optar al
Título de **INGENIERO DE SISTEMAS**

Barcelona, Julio de 2.010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA
GESTIÓN DE UNA EMPRESA DEDICADA A LA COMERCIALIZACIÓN DE
REPUESTOS PARA VEHÍCULOS AUTOMOTORES EN LA CIUDAD DE
CARACAS”**

JURADO CALIFICADOR:

Prof. Manuel Carrasquero
Asesor Académico

Prof. Rhonald Rodríguez
Jurado Principal

Prof. Victor Mujica
Jurado Principal

Barcelona, Julio de 2.010

RESOLUCIÓN

De acuerdo al Artículo número 41 del Reglamento de Trabajos de Grado:

“Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario, para su autorización”.

DEDICATORIA

Este trabajo va dedicado especialmente a mis padres Sofía Díaz De Cuzzi y Luigino Cuzzi Del Osa, ¡los mejores padres del mundo! (aunque suene trillada la frase), en vida me dieron todo lo mejor que pudieron, lamentablemente por mi rebeldía o quizás inmadurez no se los supe agradecer. Si hubiese sabido que la vida no me iba a dar tiempo para hacerlo créanme que hubiera aprovechado cada segundo a su lado para hacerlo, pero como dicen “uno no sabe lo que tiene hasta que lo pierde” nunca imagine que me harían tanta falta.

Mami cuando te fuiste llore desconsolada porque ya no estarías más a mi lado pero en ese momento no sabía la falta que me harías en el futuro, la falta que me haces para celebrar mis alegrías o para compartir mis tristezas, para darme esos consejos que solo una madre sabe dar. Hoy tu “chiquita” esta logrado una de sus metas ¡Graduarse! Y todo gracias a ti porque aunque no hayas estado a mi lado en estos últimos siete años me dejaste la mejor herencia, la educación y los principios que me inculcaste, recuerdo que siempre me decías que la educación era la única herencia que dejaban los padres y esa frase me la repetía cada vez que pensaba que no podía seguir adelante porque sé que allá en el cielo hoy estas orgullosa de mi.

Papi mi viejito sin vergüenza jejeje... no sabes cuánto te extraño!! Cuando mi mami se fue me dijiste que “se nos había ido el timón del barco” desde ese momento me aferre a ti como si fueras lo único que me quedaba, cuando enfermaste no podía entender ¿por qué Dios me hacia eso a mí? ahora después de varios años he podido entender que la “voluntad de Dios

nunca nos lleva hasta donde su Gracia no pueda protegernos” pero al igual que mi mami me haces falta en cada uno de los momentos importantes de mi vida. Hoy estoy a punto de lograr lo que tu tanto querías “verme graduada” lamentablemente no estás a mi lado para compartir esta felicidad pero sé que allá en el cielo estas orgulloso de tu hija.

Este triunfo se los dedico a los dos porque sin la educación que me dieron y todos los valores que me enseñaron nunca hubiese podido llegar donde estoy, son mi principal fuente de inspiración para seguir adelante y todos los triunfos que logre en la vida son para ustedes, por ustedes y de ustedes. Le doy gracias a Dios por darme los mejores padres de este mundo. ¡LOS AMO INMENSAMENTE!

P.D: No me alcanzarán las páginas para agradecerles y decirles todo lo que significan para mí.

Este logro también va dedicado a mis hermanos, Maritza Cuzzi, Jesmina Cuzzi y Edgardo Cuzzi por estar siempre a mi lado apoyándome y ayudándome a seguir adelante. Sé que siempre creyeron en mí y sabían que podía a lograr esta meta.

No puedo dejar de mencionar a unas personitas muy especiales, mis sobrinos, Nicol Vasellini, Edgarris Cuzzi, Luis Alejandro Cuzzi y Angel Villaroel y a mi primita Verónica Sanez,, mis niños bellos espero que este logro de su Tía los motive a ser unos profesionales del futuro LOS ADORO!!!!!!!

Vanessa Cuzzi

DEDICATORIA

Quiero dedicar este logro, primeramente a Dios que me guió durante todo este camino. Por haberme dado tantas oportunidades y permitirme llegar hasta donde estoy ahora y pidiéndole que me acompañe en los caminos que faltan por recorrer.

Le dedico también a mi mamá Amparo Weitze, que me ha apoyado en todas y cada una de las decisiones que he tomado. Con sus palabras de aliento, me ha enseñado que las cosas que más cuestan, son las que más se valoran. Cada vez que sentía que iba a caer, estaba ella a mi lado, con las palabras justas para seguir adelante. Por eso y muchísimas cosas más, te amo mamá!

A mi hermano Hans Palomino, que a su manera participó y fue pilar fundamental durante este maravilloso recorrido y espero poder retribuirle todo el esfuerzo que invirtió para ayudarme. Gracias Hans!

Por último y no menos importante, le dedico este logro a mi abuelita Úrsula Oelsner quién desde la Santa Gloria de Dios, me cuida y me sigue a donde quiera que yo vaya. Fuiste alegría y apoyo para todos nosotros y de esa manera te recuerdo abuela. Tú alegría y ocurrencia fue estímulo para seguir adelante y responder con una sonrisa a situaciones adversas. Te amo y te extrañó abuelita.

Mitchel Palomino.

AGRADECIMIENTO

Gracias a Dios por estar siempre a mi lado guiándome por el camino correcto y dándome fuerzas para superar cada una de sus pruebas.

A mis padres Luigino Cuzzi Del Osa y Sofia Díaz De Cuzzi, gracias por la educación que me dieron, por querer lo mejor para mí hasta el último momento de sus vidas, por todos los momentos compartidos, le doy gracias a Dios por haberme dado unos padres como ustedes. Siempre los recordare con una sonrisa en sus labios y sus ojos llenos de alegría. Son un ejemplo de constancia y de lucha a seguir para mí.

A dos primos que han sido como dos hermanos más para mí, Robert Sanéz y Emery Sanéz. Primito gracias por estar siempre a mi lado pendiente de mí en todo momento, cuidándome y protegiéndome TMQ primito. Primita gracias también porque en todo momento estas pendiente de mi, gracias por cuidarme cada vez que me dan unos de mis “achaques”, como los llamas, gracias por estar ahí hasta para curarme un dolor de uña jejeje... aunque no te lo diga muy seguido te ADORO y te extraño cada vez que no estás.

A mis tíos Mario Cuzzi y José Aponte y mi cuñado Nicolo Vasellini, gracias por creer siempre en mí, creer que soy capaz de lograr las metas que me proponga, por estar pendiente de mí en todo momento, a pesar de la distancia. Espero enorgullecerlos con este logro. LOS AMO, que Dios les regale mucha salud para poder tenerlos a mi lado por mucho tiempo.

A mis tías Juana Díaz, Mirna Díaz, Aura Díaz y Lidia Pietrangeli gracias por estar a mi lado en los momentos más difíciles, por cuidar a mis padres cuando ellos lo necesitaron, por quererme como una hija y desear siempre lo mejor para mí. Las quiero mucho!!!!!!

A mis hermanos Maritza Cuzzi, Jesmina Cuzzi y Edgardo Cuzzi. Mary gracias por estar siempre pendiente de mi, por aconsejarme, guiarme y regañarme cuando hace falta, por ser como una madre en estos últimos años y querer lo mejor para mí en todo momento. Jesmi gracias por estar ahí cada vez que lo necesito, por ayudarme a llevar la enfermedad de mi papá, nunca olvidare que estuviste a mi lado en todo momento preocupándote por mi y pendiente de que yo estudiara, gracias por todo porque tu al igual que Mary han sido como unas madres para mí. LAS AMO!!!! Edgardo mi hermanito bello con el que he peleado, llorado y reído, gracias simplemente por ser mi hermano, por estar siempre a mi lado, tenerme paciencia y quererme con mis defectos y virtudes TQM hermanito!

A mis sobrinos Nicol Vasellini, Edgarlis Cuzzi, Luis Alejandro Cuzzi y Angel Villaroel y a mi primita Verónica Sanez, gracias porque con su manera tan inocente de ver la vida me hacen recordar que no tenemos que ahogarnos en un vaso de agua, porque todos los problemas tienen solución, gracias por sacarme un sonrisa con cada una de sus ocurrencias. LOS AMO MIS NIÑOS BELLOS!!!

A mis primas Yanelis Sanez, Anais Salazar, Claudia Paraqueiomo, Concetta Cuzzi y Andreina Ferrer, mis queridísimas primas con las que he compartido hermosos momentos de mi vida, gracias por estar siempre pendiente de mi, por desearme lo mejor y por regalarme tantos momentos de risas y alegrías. Las quiero muuuuuucho!!!

A mi cuñada Saiduby Albornett gracias por estar a mi lado en estos últimos años, por escucharme y soportarme. Aunque nunca lo diga TQM cuñis!!!!

A mi novio bello, el mejor novio que existe! Armando Zabala Dios te puso en mi vida en el momento que más lo necesitaba, desde que llegaste empecé a sonreír de nuevo, gracias por estar siempre a mi lado para apoyarme y ayudarme cada vez que lo necesito, por todos esos días de estudios en los que me decías ¡si se puede, tu eres una muchacha muy inteligente! cada vez que pensaba en desistir, a tu lado he compartido momentos de alegrías y tristezas, he reído, llorado y peleado jejeje.. en fin mucho de los mejores momentos de mi vida han sido a tu lado. TE AMOOOOOOOOOOOOO!!!! Estas líneas se quedan cortas para expresar todo lo que siento por ti..

A mis suegritos el señor Armando Zabala y la Señora Inés Gutiérrez De Zabala, gracias por considerarme como un miembro más de su familia y tratarme como una hija, gracias por todos los momentos compartidos. Los quiero mucho!!!

A mis amigas de toda una vida Alejandra Girot, Fabiana Carias, María Elena Rodríguez y Vanessa Salazar "las mejores amigas del mundo" gracias porque siempre están a mi lado cada vez que las necesito, ustedes me enseñaron el verdadero sentido de la amistad, no las considero mis amigas si no mis hermanas, en todos estos años nuestra amistad a crecido y se ha fortalecido. !LAS ADORO!

A Wilfredo Sánchez y Rafael Guzmán mis queridos amigos gracias por estar pendientes de mi a pesar de la distancia. Los quiero mucho!!

A unas amigas de la universidad con las que empecé la carrera y he compartido muy buenos momentos Adrianny Torres y Daniela Ramos, han sido muchas las cosas que hemos vivido, días de estudios, risas y hasta tristezas. Todas esas cosas me han permitido darme cuenta que son unas grandes amigas, así que pueden estar orgullosas porque están en mi lista de mejores amigas jejeje... LAS QUIERO!!!

A todos esos amigos de la universidad con los que también he compartido buenos momentos, Nohely Boada, Rosio Noriega, Jenny Carreño, Zuleydy Amaiz, Rosimar Gómez, Rosmara Marín, Alexander Bidrogo, Carmen Paruta, Andreyana Camacaro, Gabriela Sosa, Leonel Goitia, Romili Briceño, Yolanda Rodríguez, Andres Ramos, Arisaiht Gómez, yocelyn Canelón, y John Suarez. Gracias!! Ustedes contribuyeron de una u otra forma a cumplir esta meta.

A mi compañero de tesis Mitchel Palomino, por aceptar compartir este reto conmigo, hacer amena la realización de este proyecto y trabajar juntos por cumplir esta meta. Muchas gracias!!!!

A nuestro asesor académico Prof. Mauel Carrasquero por guiarnos durante la realización de este proyecto, por sus prontas respuestas y correcciones y por todas sus enseñanzas. Muchas gracias fue un placer para mí tenerlo como asesor.

A la Distribuidora Sellini por abrirnos las puertas de su empresa y muy especialmente a la Sra. Maritza Cuzzi por brindarnos toda la información necesaria para culminar este proyecto. Muchas gracias!

Vanessa Cuzzi

AGRADECIMIENTO

Agradezco a Dios por esta maravillosa experiencia que fue la Universidad de Oriente y permitirme culminar esta etapa de mi vida.

Mi familia que estuvo apoyándome y evitándome caer cuando las situaciones no salían como esperaba. Sus palabras y esfuerzo fueron fuente de motivación para llegar hasta donde estoy ahora.

Agradezco también a mi compañera Vanessa Cuzzi por brindarme la oportunidad de participar en este gran proyecto, trabajando lado a lado y sacando esta tesis adelante. Siempre te recordaré y estaré eternamente agradecido por esta maravillosa experiencia.

Un cordial agradecimiento a nuestro Asesor Académico Ing. Manuel Carrasquero por la asesoría durante todo el proyecto. Agradezco también por haber tenido el privilegio de recibir sus clases en varias materias, entre las cuales, muchas de sus enseñanzas me sirvieron para desarrollar esta tesis. Por su dedicación y esfuerzo, tiene mi admiración y agradecimiento Ing. Manuel Carrasquero.

Agradezco a la Gerente General de la Empresa Distribuidora Sellini, la Sra. Maritza Cuzzi y a todo el personal de la empresa, por la valiosa colaboración que nos prestaron durante la investigación y desarrollo del proyecto.

Un agradecimiento muy especial a Armando Zabala, Daniela Benítez, Lizbeth Novoa, al Dr. César Salazar, al Dr. Kelvin Cárima, al Dr. Wilmer Luis Martínez, al Dr. Hendri Guzmán, al Lic. Omar Piñango, José Compañero, Rafael Barrios, quienes con su presencia e inclusive algunos ausentes, pero que con su palabra me incentivaron a investigar, a ir más allá en mi preparación para este proyecto, a no conformarme con lo que sabía y buscar nuevas técnicas y metodologías, donde el resultado es este maravilloso proyecto.

Muchísimas personas y nombres me vienen a la mente en este momento. Amigos que conseguí desde el 1er semestre de la carrera y que aún en el día de hoy me acompañan y me apoyan. Todos aportaron su granito de arena y gracias a ellos soy quién soy hoy.

Por lo pronto, mis mejores deseos para que culminen sus estudios y tengamos la dicha de trabajar juntos, lado a lado, hombro con hombro.

A los que ya salieron, espero tener la dicha de verlos en el camino del éxito, compartiendo sus alegrías y las mías por igual.

A mis vecinos, la familia Rivera quienes me ayudaron muchísimo con su consejo y asesoría. A la familia Torres por sus palabras de ánimo. Por todo ese apoyo, muchísimas gracias!

Mitchel Palomino.

RESUMEN

Esta investigación tiene como objetivo desarrollar un sistema de información denominado **“Sellini”**; que permita a la Empresa Distribuidora Sellini actualizar el registro de información de las actividades que realiza diariamente para lograr mayor efectividad y eficiencia y hacerlo en menos tiempo. La metodología empleada en el proyecto está basada en el Proceso Unificado de Desarrollo de Software (RUP), las herramientas usadas para el desarrollo del sistema fueron el Lenguaje de Programación Orientado a Objetos PHP y el Sistema Manejador de Base de Datos MySQL. El sistema **“Sellini”** fue realizado en cuatro fases: Inicio, Elaboración, Construcción y Transición. Se determinaron las necesidades de la empresa para la automatización de los procesos de: la elaboración de pedidos, registro de productos, registro de proveedores, registro de clientes y consultas sobre la información de los mismos. La investigación se fundamentó en las teorías de: Elmasri, R y Navathe, S. sobre sistemas de Base de Datos, Larman, C. sobre el Modelo de Lenguaje Unificada (UML), Pressman, R. sobre la Ingeniería de Software.

ÍNDICE

DEDICATORIA.....	IV
DEDICATORIA.....	VI
AGRADECIMIENTO.....	VII
AGRADECIMIENTO.....	XI
RESUMEN	XIII
ÍNDICE.....	XIV
ÍNDICE DE FIGURAS.....	XIX
ÍNDICE DE TABLAS.....	XXII
CAPÍTULO I.....	XXIII
EL PROBLEMA	XXIII
1.1 PLANTEAMIENTO DEL PROBLEMA	XXIII
1.2 OBJETIVOS	25
1.2.1 <i>Objetivo General</i>	25
1.2.2 <i>Objetivo Específico</i>	25
CAPÍTULO II	26
MARCO TEÓRICO REFERENCIAL	26
2.1 ANTECEDENTES DE LA INVESTIGACIÓN.....	26
2.2 BASES TEÓRICAS.....	28
2.2.1 <i>Sistemas</i>	28
2.2.1.1 <i>Concepto</i>	28
2.2.1.2 <i>Características de los sistemas</i>	29
2.2.1.3 <i>Análisis y diseño del sistema</i>	29
2.2.2 <i>Sistema de información (SI)</i>	29
2.2.2.1 <i>Concepto</i>	29
2.2.2.2 <i>Diseño del sistema de información</i>	30
2.2.2.3 <i>Actividades básicas de los sistemas de información</i>	30

2.2.2.4 Desarrollo del diseño del sistema	31
2.2.3 Programación Orientada a Objetos (P.O.O)	33
2.2.3.1 Concepto.....	33
2.2.3.2 Ventajas de la P.O.O	34
2.2.3.3 Características de los Lenguajes Programación Orientada a Objetos	35
2.2.4 Tecnologías Web de Intranet.....	37
2.2.4.1 Concepto.....	37
2.2.5 Protocolo HTTP	37
2.2.5.1 Concepto.....	37
2.2.6 Servidor Web Apache	38
2.2.6.1 Concepto.....	38
2.2.6.2 Ventajas de Servidor HTTP Apache.....	39
2.2.7 Aplicación Web.....	39
2.2.7.1 Concepto.....	39
2.2.7.2 Estructura de las Aplicaciones Web	39
2.2.8 Navegador Web	40
2.2.8.1 Concepto.....	40
2.2.9 Lenguaje HTML	41
2.2.9.1 Concepto.....	41
2.2.10 Hojas de estilo en cascada (CSS).....	42
2.2.10.1 Concepto.....	42
2.2.11 Técnica de Desarrollo Web Ajax.....	43
2.2.11.1 Concepto.....	43
2.2.12 Lenguaje de Programación PHP	44
2.2.12.1 Concepto.....	44
2.2.12.2 Usos del lenguaje de programación PHP	46
2.2.12.3 Ventajas del lenguaje de programación PHP	46
2.2.13 Base de Datos	47
2.2.13.1 Concepto.....	47
2.2.13.2 Propiedades implícitas de una Base de Datos	47
2.2.13.3 Ventajas en el uso de Bases de Datos	48
2.2.13.4 Arquitectura de una base de datos	49
2.2.13.5 Principales Bases de Datos Comerciales	49
2.2.13.6 Modelos de Bases de Datos	50
2.2.13.7 Requerimientos de las Bases de Datos.....	50
2.2.13.8 Diseño de Bases de Datos.....	50
2.2.14 Sistema Manejador de Base de Datos (SMBD).....	51

2.2.14.1 Concepto.....	51
2.2.14.2 Objetivos del Sistema Manejador de Datos	52
2.2.15 <i>Arquitectura Cliente – Servidor</i>	52
2.2.15.1 Concepto.....	52
2.2.15.2 Ventajas de la Arquitectura C/S.....	53
2.2.15.3 Desventajas de la Arquitectura C/S	54
2.2.16 <i>Lenguaje Estructurado de Consultas SQL</i>	54
2.2.16.1 Concepto.....	54
2.2.17 <i>Sistema Manejador de Base de Datos MySQL</i>	55
2.2.17.1 Concepto.....	55
2.2.18 <i>Ingeniería del Software</i>	55
2.2.18.1 Concepto.....	55
2.2.19 <i>Lenguaje Unificado de Modelado (UML)</i>	57
2.2.19.1 Concepto.....	57
2.2.19.2 Principales Diagramas del UML.....	57
2.2.20 <i>Proceso Unificado de Desarrollo de Software</i>	58
2.2.20.1 Concepto.....	58
2.2.20.2 Ciclo de vida del RUP	61
2.2.20.3 Flujos de Trabajo del Proceso Unificado.....	65
CAPÍTULO III.....	69
MARCO METODOLÓGICO	69
3.1 TIPOS DE INVESTIGACIÓN.....	69
3.1.1 <i>Nivel de la Investigación</i>	69
3.1.2 <i>Diseño de la Investigación</i>	70
3.2 TÉCNICAS A UTILIZAR	70
CAPÍTULO IV	73
RESULTADOS	73
4.1 FASE DE INICIO.....	73
4.1.1 <i>Descripción del Sistema Actual</i>	73
4.1.2 <i>Modelo del Dominio</i>	75
4.1.2.1 Glosario de Términos	77
4.1.3. <i>Estructura Organizativa</i>	78
4.1.3.1 Diagrama del Sistema y Ambiente ampliado.....	81
4.1.4 <i>Definición de la Misión-Visión</i>	82

4.1.5 <i>Análisis de la Problemática</i>	83
4.1.6 <i>Flujo de Trabajo de Requisitos</i>	84
4.1.6.1 Contexto del Sistema.....	85
4.1.6.2 Modelo de Caso de Uso.....	86
4.1.6.3 Descripción de los Casos de Uso del Sistema.....	90
4.1.7 <i>Flujo de Trabajo de Análisis</i>	105
4.1.7.1 Paquetes de Análisis del Sistema.....	105
4.1.7.2 Diagrama de Clase de Análisis	107
4.1.7.3 Diagrama de Colaboración	111
4.1.8 <i>Flujo de Trabajo de Diseño</i>	118
4.1.8.1 Arquitectura Candidata.....	118
4.1.8.2 Interfaz Inicio de Sesión.....	119
4.1.9 <i>Conclusión de la Fase de Inicio</i>	120
4.2 FASE DE ELABORACIÓN	121
4.2.1 <i>Flujo de Trabajo de Requisitos</i>	121
4.2.1.1 Diagrama de Caso de Uso General	122
4.2.2 <i>Flujo de Trabajo de Análisis</i>	128
4.2.2.1 Diagrama de Clase de Análisis	130
4.2.2.2 Diagrama de Colaboración	132
4.2.3 <i>Flujo de Trabajo de Diseño</i>	135
4.2.3.1 Diseño de la Arquitectura.....	135
4.2.3.2 Diagrama de Clase de Diseño.....	136
4.2.3.3 Diseño de la Base de Datos	140
4.2.4 <i>Implementación</i>	145
4.2.4.1 Identificación de Componentes de la Arquitectura.....	146
4.2.4.2 Implementación de la Arquitectura.....	147
4.2.5 <i>Conclusión de la Fase de Elaboración</i>	148
4.3 FASE DE CONSTRUCCIÓN	149
4.3.1 <i>Flujo de Trabajo: Implementación</i>	149
4.3.2 <i>Flujo de Trabajo: Pruebas</i>	168
4.3.2.1 Pruebas de Integración.....	168
4.3.2.2 Pruebas de Caja Negra.....	169
4.3.3 Conclusión de la Fase de Construcción	173
4.4 FASE DE TRANSICIÓN	173
4.4.1 <i>Evaluación de la Fase de Transición</i>	174
4.4.2 <i>Flujo de Trabajo Pruebas</i>	175

4.4.2.1 Aplicación de las Pruebas.....	175
4.4.3 <i>Conclusión de la Fase de Transición</i>	175
CONCLUSIÓN	177
RECOMENDACIONES.....	179
BIBLIOGRAFÍA	180
ANEXOS.....	183

ÍNDICE DE FIGURAS

FIGURA 2.1: ESTRUCTURA GENERAL DE UNA APLICACIÓN WEB	40
FIGURA 2.2. EJECUCIÓN DE UN SCRIPT PHP.....	45
FIGURA 2.3 CICLO DE VIDA DEL RUP.....	62
FIGURA 2.4: FLUJOS DE TRABAJO: REQUISITOS, ANÁLISIS, DISEÑO, IMPLEMENTACIÓN Y PRUEBA / FASES: INICIO, ELABORACIÓN, CONSTRUCCIÓN Y TRANSICIÓN	66
FIGURA 4.1: MODELO DE DOMINIO	77
FIGURA 4.2: ESTRUCTURA ORGANIZATIVA DE LA EMPRESA DISTRIBUIDORA SELLINI	81
FIGURA 4.3: DIAGRAMA DEL SISTEMA Y AMBIENTE AMPLIADO	82
FIGURA 4.4: DIAGRAMA DE CASO DE USO GENERAL DEL SISTEMA.....	89
FIGURA 4.5: DIAGRAMA DE PAQUETE DE ANÁLISIS DEL SISTEMA.....	107
FIGURA 4.6: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO GESTIONAR SESIÓN.....	108
FIGURA 4.7: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO GESTIONAR PROVEEDOR	109
FIGURA 4.8: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO GESTIONAR CLIENTE.....	109
FIGURA 4.9: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO GESTIONAR PRODUCTO.....	110
FIGURA 4.10: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO PROCESAR VENTAS.....	110
FIGURA 4.11: DIAGRAMA DE CLASES DE ANÁLISIS PARA EL CASO DE USO REALIZAR CONSULTAS Y REPORTES	111

FIGURA 4.12: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR SESIÓN	112
FIGURA 4.13: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR PROVEEDORES.....	113
FIGURA 4.14: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR CLIENTES.....	114
FIGURA 4.15: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR PRODUCTOS.....	115
FIGURA 4.16: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO PROCESAR VENTAS.....	116
FIGURA 4.17: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO REALIZAR CONSULTAS	117
FIGURA 4.18: ARQUITECTURA CANDIDATA DEL SISTEMA	119
FIGURA 4.19: PROTOTIPO DE INTERFAZ DE INICIO DE SESIÓN.....	120
FIGURA 4.20: CASO DE USO ADMINISTRAR BASE DE DATOS	124
FIGURA 4.21: CASO DE USO GESTIONAR USUARIO.....	128
FIGURA 4.22: CASO DE USO GENERAL DEL SISTEMA	129
FIGURA 4.23: DIAGRAMA DE CLASE DE ANÁLISIS PARA EL CASO SE USO ADMINISTRAR BASE DE DATOS	131
FIGURA 4.24: DIAGRAMA DE CLASE DE ANÁLISIS PARA EL CASO DE USO GESTIONAR USUARIO	131
FIGURA 4.25: DIAGRAMA DE COLABORACIÓN DEL CASO DE USO ADMINISTRAR BASE DE DATOS	133
FIGURA 4.26: DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR USUARIO	134
FIGURA 4.27: DISEÑO DE LA ARQUITECTURA DEL SISTEMA	136
FIGURA 4.28: DIAGRAMA GENERAL DE LA CLASE DE DISEÑO	139
FIGURA 4.29: DIAGRAMA DE LA CLASE DE DISEÑO, VISTA VENDEDOR.....	140

FIGURA 4.30: RELACIÓN DE TABLAS DE LA BASE DE DATOS	141
FIGURA 4.31: TABLA ACCESO DE USUARIOS	142
FIGURA 4.32: TABLA USUARIOS	142
FIGURA 4.33: TABLA PRODUCTOS.....	143
FIGURA 4.34: TABLA PROVEEDORES	143
FIGURA 4.35: TABLA CATEGORÍAS.....	144
FIGURA 4.36: TABLA DEPÓSITO	144
FIGURA 4.37: TABLA PEDIDOS	144
FIGURA 4.38: TABLA DETALLES DE PEDIDOS	145
FIGURA 4.39: TABLA CLIENTES.....	145
FIGURA 4.40: DIAGRAMA DE DESPLIEGUE DEL SISTEMA.....	147
FIGURA 4.41: PANTALLA DE INICIO DE SESIÓN.....	148
FIGURA 4.42: PANTALLA DE INICIO DE SESIÓN.....	150
FIGURA 4.43: PANTALLA DE CREAR PEDIDO	154
FIGURA 4.44: REPORTE PDF DEL PEDIDO	161
FIGURA 4.45: HOJA DE EXCEL MANEJADA POR LA EMPRESA PARA ALMACENAR LA INFORMACIÓN DE LOS PRODUCTOS.....	183
FIGURA 4.46 FICHA PERSONALIZADA DONDE SE ALMACENAN LA INFORMACIÓN DE LOS CLIENTES Y PROVEEDORES (PRIMERA CARA)	184
FIGURA 4.47 FICHA PERSONALIZADA DONDE SE ALMACENAN LA INFORMACIÓN DE LOS CLIENTES Y PROVEEDORES (SEGUNDA CARA).....	185

ÍNDICE DE TABLAS

TABLA 4.1: LISTA DE RIESGOS DEL SISTEMA	87
TABLA 4.2: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR SESIÓN	113
TABLA 4.3: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR PROVEEDORES	114
TABLA 4.4: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR CLIENTES	115
TABLA 4.5: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR PRODUCTOS.....	116
TABLA 4.6: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO PROCESAR VENTA.....	117
TABLA 4.7: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO REALIZAR CONSULTAS	118
TABLA 4.8: LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO ADMINISTRAR BASE DE DATOS.....	133
TABLA 4.9 LEYENDA DEL DIAGRAMA DE COLABORACIÓN PARA EL CASO DE USO GESTIONAR USUARIO	134
TABLA 4.10: CLASE DE EQUIVALENCIA PARA LA OPERACIÓN: CREAR CLIENTE.....	170
TABLA 4.11: CLASE DE EQUIVALENCIA PARA LA OPERACIÓN: MODIFICAR CLIENTE.....	171
TABLA 4.12: CLASE DE EQUIVALENCIA PARA LA OPERACIÓN: CREAR PRODUCTO.....	172
TABLA 4.13 SE MUESTRA LA CLASE DE EQUIVALENCIA PARA LA OPERACIÓN CREAR PEDIDO	173

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del Problema

Es del conocimiento popular que los filtros juegan un papel muy importante en la vida útil de los vehículos. Entiéndase por vehículos todo lo relacionado con los medios de transporte privado o público y las distintas maquinarias necesarias en las construcciones civiles, agricultura y pesca. Existe una gran variedad de filtros para los vehículos, estos son: Filtros de Aceite, filtros de Agua, filtros de Combustibles, (gasolina, diesel gas), Filtros Separadores y Filtros Hidráulicos.

Considerando la gran demanda en el mercado de los filtros, el Sr. Nicolo Del Osa, decidió en 1990 crear una pequeña empresa dedicada a la venta y distribución de filtros, la cual lleva el nombre de DISTRIBUIDORA SELLINI, C.A., ubicada en la ciudad de Caracas. En sus inicios esta empresa se dedicaba sólo a la venta de filtros para vehículos livianos. Con el pasar de los años fue ampliando el ramo de comercialización y se especializó en filtros para las maquinarias pesadas como: tractores, remolques, martillos eléctricos, compresores, barcos, plantas eléctricas, entre otros.

Por lo descrito anteriormente, la Empresa almacena un registro de datos que permite identificar el producto; de allí que cada filtro manejado por esta Empresa posee un código único, asociado a su fabricante, que lo

identifica para almacenarlo en una posición física; estos datos son registrados mediante la herramienta de Microsoft Office (Excel), generando duplicidad de la información y pérdida de tiempo al momento de realizar consultas o reportes de interés para la empresa.

Distribuidora Sellini almacena el historial del consumo de sus clientes y las compras a proveedores en fichas personalizadas, las cuales son llenadas manualmente con toda la información correspondiente, esto ocasiona retrasos en la actualización de la información y principalmente, obstaculiza el intercambio de información para la obtención de las diferentes estadísticas de proveedores y clientes.

Por consiguiente, representa un retraso de información que requiere ser abordado con herramientas de actualización y vanguardia acorde con los adelantos de los sistemas de información.

Por las razones expresadas en el párrafo anterior, este proyecto tuvo como propósito desarrollar un sistema de información que permita el cumplimiento de las actividades de manera más eficiente y en un menor tiempo, a través de interfaces fáciles de manejar para los usuarios. Para alcanzar el objetivo final se utilizaron los lineamientos del proceso unificado de desarrollo de software (RUP); así como el Lenguaje de Programación Orientado a Objetos PHP y el Sistema Manejador de Base de Datos MySQL.

El alcance del trabajo estuvo enmarcado en el cumplimiento de las cuatro (4) fases del Proceso Unificado de Desarrollo (RUP), las cuales son: Fase De Inicio, Elaboración, Construcción y Transición. Es potestad de la Empresa la realización del manual de usuario y su posterior implementación.

La originalidad del proyecto radicó en el hecho de que fue la primera vez que en esta Empresa se le presenta la propuesta para desarrollar un sistema de información, utilizando los principios del Análisis y Diseños de Sistemas, así como también la Orientación a Objetos; para su logro se contó con la participación de estudiantes de la carrera de la Ingeniería de Sistemas.

El diseño de este proyecto estuvo orientado a atender los siguientes requerimientos: Registro de Proveedores, Clientes, Productos y Pedidos, de la Empresa Distribuidora Sellini, lo que permite optimizar el trabajo de los usuarios y hacer más efectiva la información, atención y manejo de datos para la operatividad eficaz de la Empresa.

1.2 Objetivos

1.2.1 Objetivo General

- ❖ Desarrollar un sistema de información para la gestión de una empresa dedicada a la comercialización de repuestos para vehículos automotores en la ciudad de Caracas.

1.2.2 Objetivo Específico

- ✓ Describir la situación actual del sistema bajo estudio.
- ✓ Definir los requerimientos del sistema propuesto, considerando las necesidades y carencias de sus usuarios.
- ✓ Diseñar la estructura del software.
- ✓ Diseñar la base de datos del sistema de información.
- ✓ Diseñar la interfaz para la comunicación entre el usuario y el sistema.
- ✓ Codificar los diferentes módulos del sistema.

CAPÍTULO II

MARCO TEÓRICO REFERENCIAL

2.1 Antecedentes de la Investigación

En la Empresa Sellini es la primera vez que se desarrolla un Sistema de Información utilizando un lenguaje UML, sin embargo en otras empresas se han realizado proyectos con esta metodología, los cuales pueden contribuir con esta investigación. A continuación se nombran algunos de estos trabajos.

Ortiz (2009), realizó un trabajo de grado titulado “Diseño de un sistema de información que permita la evaluación, cuantificación e identificación de las fallas de las bombas ubicadas en la superintendencia de movimientos de crudos de la refinería Puerto la Cruz”. En este proyecto se empleó el Lenguaje Unificado Modelado, el cual se basa en la elaboración de un conjunto de diagramas con el fin de establecer la estructura del software del proyecto. Para el diseño de base de datos se utilizó el modelo relacional el cual permitió establecer las relaciones necesarias para garantizar la menor cantidad posible de redundancia de datos y a su vez mantener la integridad de la información que se enlace.

Pino (2009), realizó un trabajo de grado titulado “Diseño de un sistema de información para el control, evaluación y estimación de las horas hombre invertidas en el proceso de desarrollo de software”. Durante esta investigación se diseñó un sistema Web para una empresa desarrolladora de software; este permite llevar un control completo de las actividades de

producción. A través de él se registran todos los proyectos activos de la empresa. Adicionalmente, la información de control recopilada es usada para estimar costos en el futuro de forma confiable. Para realizar la labor de diseño se conceptualizaron nuevas metodologías de diagramación de entidades y cálculos de costos, así como un indicador de gestión apropiado y veraz orientado a la evaluación justa del trabajo.

Borjas (2007), realizó un trabajo de grado titulado “Diseño de un sistema de información para el registro de entrada y salida de vehículos de transporte del instituto de deportes del estado Anzoátegui”. En el departamento de Transporte del Instituto de Deportes del Estado Anzoátegui, se han podido comprobar algunas deficiencias relacionadas con el proceso de registro de viajes de los vehículos de transporte, lo cual genera una labor poco eficiente. Por esta razón, se planteó el diseño de un sistema de información transaccional que acelere los procedimientos cotidianos, de forma que se reduzca el tiempo. En tal sentido, para el modelado de este sistema se empleó el Lenguaje Unificado de Modelado, el cual se basa en la elaboración de un conjunto de diagramas con el fin de establecer la estructura de del software del proyecto.

Olivier y El Nemer (2007), realizaron un trabajo de grado titulado “Desarrollo de un software para la automatización del flujo de información asociado a los clientes y proveedores de un local comercial”. El local comercial Novedades Olga realiza todas sus actividades administrativas de forma manual, trayendo como consecuencia retrasos en el procesamiento de información. El desarrollo de este trabajo permitió la automatización de los procesos administrativos gestionados por el mencionado local, tales como: Las Ventas y Cuentas por Cobrar, realización de Compras, Inventario, entre otras. Se siguió la metodología del proceso Unifico de Desarrollo de Software

(Jacobson, Booch y Rumbaugh), debido a su eficiencia en desarrollo de software aplicando las fases de inicio, elaboración, construcción y transición.

Álvarez y Márquez (2004), realizaron un trabajo de grado titulado “Diseño de un Plan Estratégico para una Empresa de Venta y Servicios de Equipos de Computación en la Zona de Lechería”. En el presente trabajo se realizó el diseño de un plan estratégico para la empresa Micromax Oriente C.A, ubicada en la avenida Intercomunal, frente al elevado de Lechería. Esta empresa presenta necesidades de mejorar su funcionamiento, pues en la actualidad muestra serios problemas, específicamente en el área de gerencias y en el resto de sus departamentos. Este estudio está enfocado en la aplicación de la teoría estratégica, herramienta que permitió crear una base para la comprensión de los medios internos y externos de la empresa objetos de estudio. Se desarrolló la misión y los objetivos de la organización. Se diseñaron planes de acción para establecer cómo y con qué recursos se implantaran las estrategias.

2.2 Bases Teóricas

2.2.1 Sistemas

2.2.1.1 Concepto

Conjuntos de elementos dinámicamente relacionados formando una actividad para alcanzar un objetivo operando sobre datos/energía/materia para proveer información/energía/materia. (Cohen y Asín, 2000).

2.2.1.2 Características de los sistemas

- ❖ **Estabilidad:** Calidad por lo que el sistema funciona eficazmente.
- ❖ **Adaptabilidad:** Calidad que le permite evolucionar dinámicamente en su entorno.
- ❖ **Eficiencia:** Calidad que permite al sistema alcanzar su objetivo con economía de medios.
- ❖ **Sinergia:** La capacidad de actuación del sistema es superior a la de sus componentes individuales.

2.2.1.3 Análisis y diseño del sistema

El análisis de sistemas, es el proceso de clasificación e interpretación de hechos, diagnósticos de problemas y empleo de la información para recomendar mejoras al sistema. El análisis especifica qué es lo que el sistema debe hacer.

El diseño del sistema, es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. Es como los planos de un edificio: especifica todas las características del producto terminado; es decir, establece como alcanzar el objetivo.

2.2.2 Sistema de información (SI)

2.2.2.1 Concepto

Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. En un sentido amplio, un

sistema de información no necesariamente incluye equipo electrónico (hardware). Sin embargo, en la práctica se utiliza como sinónimo de Sistema de Información computarizado. (Cohen y Asín, 2000)

2.2.2.2 Diseño del sistema de información

El diseño de un Sistema de Información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la de desarrollo de software a la que denominan diseño físico.

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida.

El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas, y símbolos especiales). La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software. (Senn, 1992)

2.2.2.3 Actividades básicas de los sistemas de información

El sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información:

- ❖ **Entrada de información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas.
- ❖ **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior.
- ❖ **Procesamiento de información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.
- ❖ **Salida de información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes.

2.2.2.4 Desarrollo del diseño del sistema

Es un proceso que consta de las siguientes actividades:

- ❖ **Investigación preliminar**

La solicitud para recibir ayuda de un Sistema de Información puede originarse por varias razones; sin importar cuales sean éstas, el proceso se inicia siempre con la petición de una persona (administrador, empleado, ó

especialista en sistemas). Cuando se formula la solicitud, comienza la primera actividad: la investigación preliminar. Esta tiene tres partes: Aclaratoria de la Solicitud, Estudio de Factibilidad y Aprobación de la Solicitud.

- **Aclaración de la solicitud:** Muchas solicitudes que provienen de empleados y usuarios no están formulados de manera clara. Por consiguiente antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea. Si el solicitante pide ayuda sin saber qué es lo que está mal o donde se encuentra el problema, la aclaración del mismo se vuelve más difícil. En cualquier caso, antes de seguir adelante, la solicitud del proyecto debe estar claramente planteada.

- **Estudio de factibilidad:** El sistema solicitado debe ser factible en tres aspectos:
 - ✓ **Factibilidad técnica:** El trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible?, si se necesita una nueva tecnología ¿cuál es la posibilidad de desarrollarla?

 - ✓ **Factibilidad económica:** El crear el sistema, ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿los costos asociados con la decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto?

- ✓ **Factibilidad operacional:** Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios?.

- **Aprobación de la solicitud:** No todos los proyectos solicitados son deseables o factibles. Los que son deben incorporarse a los planes. Muchas organizaciones desarrollan sus planes para Sistemas de Información con el mismo cuidado con el que planifican nuevos productos y programas de fabricación o la expansión de las instalaciones. Después de aprobar la solicitud del proyecto se estima su costo, el tiempo necesario para terminarlo y las necesidades de personal; con esta información se determina dónde ubicarlo dentro de la lista existente de proyectos.

2.2.3 Programación Orientada a Objetos (P.O.O)

2.2.3.1 Concepto

La Programación Orientada a Objetos (P.O.O, por sus siglas en inglés, *Object Oriented Programming*) es una forma especial de programar, promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software, ofrece una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas. (Schach, 2005)

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

2.2.3.2 Ventajas de la P.O.O

- **Uniformidad:** Ya que la representación de los objetos lleva o implica tanto el análisis como el diseño y la codificación de los mismos.
- **Comprensión:** Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa trata.
- **Flexibilidad:** Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos datos aparezcan.
- **Estabilidad:** Dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia, permite aislar las partes del programa que permanecen inalterables en el tiempo.
- **Reusabilidad:** La noción de objeto permite que programas que traten las mismas estructuras de información reutilicen las definiciones de objetos empleadas en otros programas e incluso los procedimientos

que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos donde estos están relacionados de una manera particular.

2.2.3.3 Características de los Lenguajes Programación Orientada a Objetos

Los lenguajes de programación orientada a objetos (como C++, C# y Java) se caracterizan por tres conceptos clave: Encapsulación, Herencia y Polimorfismo, que son compatibles con este aspecto natural de identificación y clasificación de objetos.

✓ Encapsulación

La encapsulación facilita la comprensión de los grandes programas; la ocultación de datos les hace más eficaces. Los objetos pueden interactuar con otros objetos sólo a través de los atributos y métodos del objeto que se muestran públicamente. Cuantos más atributos y métodos se muestren públicamente, más difícil será modificar la clase sin que ello afecte al código que utiliza la clase. Una variable oculta se podría cambiar de un long a un double, sin que ello afecte al código que utilicen los objetos creados (instanciados) de esa clase. El programador solo se debería preocupar por los métodos en la clase que han tenido acceso a esa clase, en lugar de por todos los sitios en el programa que un objeto ha instanciado desde los que se puede llamar a la clase.

✓ Herencia

Proporciona dos ventajas evidentes a los programadores. La primera, y más importante, es que permite crear jerarquías que expresen las relaciones entre los diferentes tipos.

La segunda ventaja es que las clases pueden heredar características más generales de las clases superiores dentro de la jerarquía. En lugar de desarrollar nuevas clases a partir de cero, las clases nuevas podrán heredar las funciones de las clases existentes y, a continuación, modificar o ampliar esta función. La clase principal desde la que la nueva clase hereda las propiedades se conoce como la clase básica y la nueva clase se denomina la clase derivada.

✓ Polimorfismo

Polimorfismo significa, fundamentalmente, que las clases pueden tener el mismo comportamiento, pero implementarse de distintas maneras. Esto resulta muy útil en términos de programación, ya que permite trabajar con tipos de objetos genéricos cuando lo que interesa no es cómo implementa cada clase las funciones.

✓ Clases y Objetos

Como su propio nombre lo indica, la programación orientada a objetos está relacionada con objetos. Un objeto se compone de datos que describen al objeto y las operaciones que se pueden realizar en el objeto. No obstante, cuando se crea un programa, en realidad se declaran y definen clases, no objetos.

Una clase es un tipo definido por el usuario, encapsula tanto los datos como los métodos que funcionen en esos datos. Una clase es algo muy parecido a una plantilla, que se utiliza para crear (instanciar) objetos.

2.2.4 Tecnologías Web de Intranet

2.2.4.1 Concepto

Red diseñada para el procesamiento de información dentro de una compañía u organización. Entre sus usos se incluyen servicios tales como distribución de documentos, distribución de software, acceso a bases de datos y aprendizaje. Las intranets deben su nombre a que en ellas se utilizan a menudo aplicaciones asociadas a Internet, tales como páginas Web, sitios FTP, correo electrónico, grupos de noticias y listas de distribución, a las cuales únicamente se pueden tener acceso a los terminales de la propia compañía u organización.

2.2.5 Protocolo HTTP

2.2.5.1 Concepto

El protocolo HTTP (*Hypertext Transfer Protocol*) es el protocolo base de la W.W.W. Se trata de un protocolo simple, orientado a conexión y sin estado.

La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, *Transport Control Protocol*) de modo conectado, un protocolo que establece un canal de

comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente.

El protocolo no mantiene estado; es decir, cada transferencia de datos es una conexión independiente de la anterior, sin relación alguna entre ellas, hasta el punto de que para transferir una página Web tenemos que enviar el código HTML del texto, así como las imágenes que la componen, pues en la especificación inicial de HTTP, la 1.0, se abrían y usaban tantas conexiones como componentes tenía la página, transfiriéndose por cada conexión un componente (el texto de la página o cada una de las imágenes).

2.2.6 Servidor Web Apache

2.2.6.1 Concepto

El servidor web Apache es un esfuerzo por desarrollar y mantener un servidor HTTP de código abierto para sistemas operativos modernos. El objetivo del proyecto es proveer un servidor web seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

El servidor web Apache es un proyecto de la fundación de software Apache (*Apache Software Foundation*) y ha sido el servidor web más popular en Internet desde Abril de 1996.

2.2.6.2 Ventajas de Servidor HTTP Apache

- Modular.
- Open source.
- Multi-plataforma.
- Extensible.
- Popular (facilidad para conseguir ayuda/suporte).
- Gratuito.

2.2.7 Aplicación Web

2.2.7.1 Concepto

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web, a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje (HTML, JavaScript, Java, etc.) soportado por los navegadores Web en la que se confía la ejecución al navegador. (Pressman, 2002)

2.2.7.2 Estructura de las Aplicaciones Web

Una aplicación web está normalmente estructurada como una aplicación de tres (3) capas. En su forma más común, el navegador web ofrece la primera capa y un motor de ejecución de programas capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, entre muchos otros).

El servidor de aplicación es el requerido para la ejecución de la aplicación. Por último, una base de datos instalado en los servidores físicos constituye el último bloque del proceso, ver **Figura 2.1**.

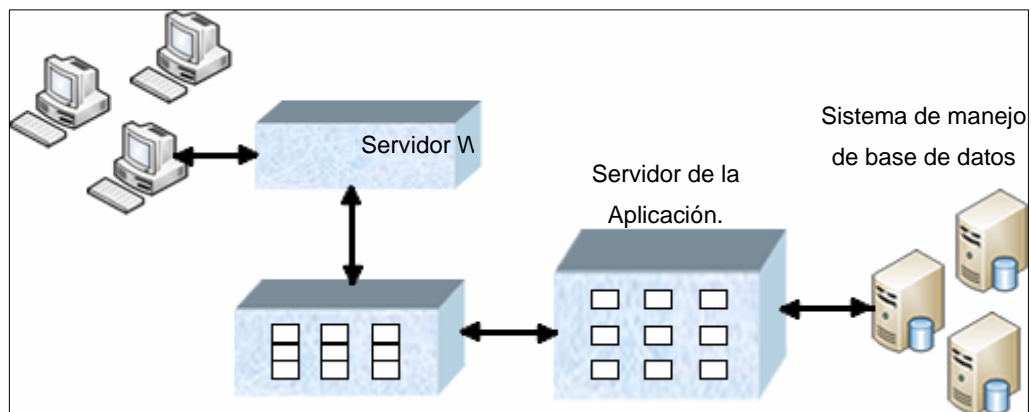


Figura 2.1: Estructura General de una Aplicación Web

Fuente: Microsoft Corporation, 2001

El navegador web manda peticiones al servidor de la aplicación, este ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

2.2.8 Navegador Web

2.2.8.1 Concepto

Un navegador web, es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet o localmente a través de una Intranet. Esta red de documentos es denominada (WWW, *World Wide Web*) o Telaraña Mundial. Los navegadores actuales permiten

mostrar y/o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor Web). Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen. (Pressman, 2002)

2.2.9 Lenguaje HTML

2.2.9.1 Concepto

El lenguaje HTML (*HyperText Markup Language*), se utiliza para crear documentos que muestren una estructura de hipertexto. Un documento de hipertexto es aquel que contiene información cruzada con otros documentos, lo cual permite pasar de un documento al referenciado desde la misma aplicación con la que lo estamos visualizando. HTML permite, además, crear documentos de tipo multimedia, es decir, que contengan información más allá de la simplemente textual.

El lenguaje HTML actualmente se encuentra en la versión 5 y empieza a proporcionar funcionalidades más avanzadas para crear páginas más ricas

en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (*Extensible Hypertext Markup Language*) que se suele definir como una versión XML validable de HTML, proporcionando un esquema XML contra el que validar el documento para comprobar si está bien formado.

2.2.10 Hojas de estilo en cascada (CSS)

2.2.10.1 Concepto

CSS (*Cascading Style Sheets*) es un lenguaje de hojas de estilo. Los estilos, añadidos a HTML en su versión 4.0, se almacenan en archivos CSS y definen como mostrar elementos HTML.

HTML nunca ha sido concebido para contener etiquetas de formato de un documento, sino para definir su contenido. Cuando las etiquetas de atributos de fuente y color fueron añadidos a HTML 3.2, el proceso de desarrollo de aplicaciones web se convirtió en una tarea difícil para los desarrolladores, ya que el desarrollo de web sites extensos, donde la información de fuentes y colores debe ser añadida a cada página por separado, se convierte en un largo y caro proceso. Para resolver este problema, la w3c (*World Wide Consortium*) creó el lenguaje CSS. En HTML 4.0, todo el código de formato de datos puede ser removido del documento HTML y almacenado en un archivo CSS separado. Actualmente, todos los navegadores web soportan CSS.

2.2.11 Técnica de Desarrollo Web Ajax

2.2.11.1 Concepto

El término AJAX es un acrónimo de Asynchronous JavaScript + XML. Se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que conforman a AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XML, XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores. En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón,

seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

2.2.12 Lenguaje de Programación PHP

2.2.12.1 Concepto

Es un lenguaje de programación que permite, entre otras cosas, la generación dinámica de contenidos en un servidor web. Entre sus principales características, se pueden destacar su potencia, alto rendimiento y su facilidad de aprendizaje. PHP es una eficaz herramienta de desarrollo para los programadores web, ya que proporciona elementos que permiten generar de manera rápida y sencilla sitios web dinámicos. (Alonso, 2006)

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Este proceso se ve reflejado en la **Figura 2.2**.



Figura 2.2. Ejecución de un script PHP

Fuente: Alonso, S. 2006

Utilizando PHP también es posible generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl". Su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, sin embargo existen también compiladores comerciales.

2.2.12.2 Usos del lenguaje de programación PHP

Los principales usos del PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (*GIMPTool Kit*), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS X. (Alonso, 2006)

2.2.12.3 Ventajas del lenguaje de programación PHP

- La principal ventaja se basa en ser un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.

- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Posee una muy buena documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos. (Alonso, 2006)

2.2.13 Base de Datos

2.2.13.1 Concepto

Es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito.

2.2.13.2 Propiedades implícitas de una Base de Datos

- Una Base de Datos representa algún aspecto del mundo real, en ocasiones llamado mini-mundo o universo de discurso. Las modificaciones del mini-mundo se reflejan en la base de datos.
- Una Base de Datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.

- Toda Base de Datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios. (Elmasri y Navathe, 2000)

2.2.13.3 Ventajas en el uso de Bases de Datos

La utilización de bases de datos como plataforma para el desarrollo de sistemas de aplicación en las organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentarán a continuación:

- Globalización de la información: Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Eliminación de información inconsistente: Si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo.
- Permite compartir información.
- Permite mantener la integridad en la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: El concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo

de sistemas de bases de datos. La independencia de datos implica un divorcio entre programas y datos.

2.2.13.4 Arquitectura de una base de datos

- **Nivel Físico:** Es el nivel real de los datos almacenados. Es decir cómo se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar cualificadas para ello.
- **Nivel Conceptual:** Es el correspondiente a una visión de la base de datos desde el punto de visto del mundo real. Es decir, se trata con la entidad u objeto representado, sin importar como está representado o almacenado.
- **Nivel Visión:** Son partes del esquema conceptual. El nivel conceptual presenta toda la Base de Datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas.

2.2.13.5 Principales Bases de Datos Comerciales

Entre las principales bases de datos comerciales tenemos:

- Base de Datos Jerárquica.
- Base de Datos de Red.
- Base de Datos Relacional.
- Base de Datos Orientada a Objetos. (Post, 2006)

2.2.13.6 Modelos de Bases de Datos

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

Los modelos de datos no son cosas físicas: Son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

2.2.13.7 Requerimientos de las Bases de Datos

El análisis de requerimientos para una Base de Datos incorpora las mismas tareas que el análisis de requerimientos del software. Es necesario un contacto estrecho con el cliente; es esencial la identificación de las funciones e interfaces; se requiere la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos".

2.2.13.8 Diseño de Bases de Datos

Para almacenar los datos de la aplicación, es necesario también diseñar la estructura de la base de datos que va a contener esos datos. El diseño de la Base de Datos es importante para optimizar el tamaño ocupado por los datos y la agilidad de consulta de los mismos.

El diseño de la Base de Datos es fundamental para obtener cualidades como la Integridad de los datos, la seguridad, el tiempo de respuesta, la

conurrencia. Cualidades que deben ser mantenidas mediante la evaluación y el análisis, una vez que la base de datos entra en funcionamiento.

Un buen diseño inicial es el pilar básico de una Base de Datos eficiente. Las optimizaciones en el rendimiento son menos costosas de implementar si se plantean en la fase de diseño que no si se plantean en fases posteriores

El diseño de Bases de Datos se compone de dos fases altamente diferenciadas: Diseño Lógico y Diseño Físico.

- **Diseño Lógico:** Consiste en analizar los requerimientos de la empresa y cuáles van a ser los componentes de la Base de Datos, como por ejemplo las tablas y las restricciones. El Diseño Lógico no tiene en cuenta dónde o cómo van a estar almacenados físicamente los datos.

- **Diseño Físico:** Consiste en implementar el Diseño Lógico sobre los recursos físicos que disponemos, aprovechando las capacidades de nuestro sistema software/hardware. Y de esta forma conseguir un acceso eficiente a los datos.

2.2.14 Sistema Manejador de Base de Datos (SMBD)

2.2.14.1 Concepto

Un SMBD consiste de una Base de Datos y un conjunto de aplicaciones (programas) para tener acceso a ellos. Comúnmente, la Base de Datos contiene información interrelacionada y referente a una misma entidad o empresa.

El objetivo primordial de un SMBD es crear un ambiente en el que sea posible almacenar y recuperar información en forma eficiente y conveniente.

Otro modelo que se utiliza comúnmente para manipular una base de datos es el llamado Sistema de Procesamiento de Archivos, el cual consta de un conjunto de programas que permiten el acceso a la base de datos, pero no optimizan los métodos utilizados en la base de datos. (Post, 2006)

2.2.14.2 Objetivos del Sistema Manejador de Datos

Existen distintos objetivos que deben cumplir los SMBD:

- Abstracción de la Información.
- Independencia.
- Consistencia.
- Seguridad.
- Integridad.
- Respaldo.
- Control de la Concurrencia.
- Manejo de Transacciones.
- Tiempo de Respuesta. (Post, 2006)

2.2.15 Arquitectura Cliente – Servidor

2.2.15.1 Concepto

Se usa para caracterizar un SGBD cuando la aplicación se ejecuta físicamente en una máquina, llamada cliente, y otra, el servidor, se encarga

del almacenamiento y el acceso a los datos. Los proveedores ofrecen diversas combinaciones de clientes y servidores; por ejemplo, un servidor para varios clientes. (Elmasri y Navathe, 2000)

2.2.15.2 Ventajas de la Arquitectura C/S

- **Centralización del control:** Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- **Escalabilidad:** Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente).
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

2.2.15.3 Desventajas de la Arquitectura C/S

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).
- Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

2.2.16 Lenguaje Estructurado de Consultas SQL

2.2.16.1 Concepto

El lenguaje de consulta estructurado es un sub-lenguaje de Base de Datos utilizado para la consulta, actualización y administración de bases de

datos relacionales, el estándar de facto para los productos de bases de datos. (Elmasri y Navathe, 2000)

2.2.17 Sistema Manejador de Base de Datos MySQL

2.2.17.1 Concepto

Es un Sistema Manejador de Bases de Datos Relacional de Código Abierto, que es gratis para muchos usos. MySQL al principio enfrentó oposición debido a su falta de apoyo a construcciones básicas de SQL tales como consultas anidadas y claves externas. Últimamente, sin embargo, MySQL encontró una base de usuarios entusiastas por sus términos de licencia tan liberal, su ejecución tan vivaz y facilidad de uso. También fue ayudado en parte por una amplia variedad de tecnologías tales como PHP, Python, Java, Perl y el hecho que tenía módulos estables y muy bien documentados. (Elmasri y Navathe, 2000)

2.2.18 Ingeniería del Software

2.2.18.1 Concepto

Es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales. Es la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software.

La Ingeniería del Software es una tecnología multicapa, las cuales son: Un Enfoque de Calidad, Proceso, Métodos y Herramientas.

Cualquier enfoque de ingeniería (incluida la Ingeniería del Software) debe apoyarse sobre un compromiso de organización de calidad.

El fundamento de la Ingeniería del Software es la capa de proceso. El proceso de Ingeniería del Software es la unión que mantienen juntas las capas de tecnología y permite un desarrollo racional y oportuno de la ingeniería del software.

Los métodos de la Ingeniería del Software indican “cómo” construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen Análisis de Requisitos, Diseño, Construcción de Programas, Pruebas y Mantenimiento. Los métodos de la Ingeniería del Software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Las herramientas de la Ingeniería del Software proporcionan un enfoque automático o semiautomático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado ingeniería del software asistida por computadora. (Pressman, 2002)

2.2.19 Lenguaje Unificado de Modelado (UML)

2.2.19.1 Concepto

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El UML es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos en principio, en un lenguaje y en un proceso para modelar. El lenguaje de modelado es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño. (Fowler y Scott, 1997)

2.2.19.2 Principales Diagramas del UML

Entre los principales diagramas se encuentran:

- Diagrama de Casos de uso.
- Diagramas de Interacción.
- Diagramas de Paquetes.
- Diagramas de Estados.

- Diagramas de Actividades.
- Diagramas de Emplazamiento. (Fowler y Scott, 1997)

2.2.20 Proceso Unificado de Desarrollo de Software

2.2.20.1 Concepto

El Proceso Racional Unificado (*Rational Unified Process*), habitualmente resumido como RUP es un proceso de desarrollo de software y, junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El Proceso Unificado posee tres paradigmas que lo identifican y particularizan de forma única, estos son:

- Dirigido por Casos de Uso.
- Centrado en la Arquitectura.
- Iterativo e Incremental. (Schach, 2005)

❖ Proceso Unificado Dirigido por Casos de Uso

Cuando un usuario utiliza el sistema, se lleva a cabo una interacción que consiste en una secuencia de acciones (tanto por parte del usuario como del sistema) que proporcionan al usuario un resultado de valor importante

para él. Un caso de uso es precisamente una interacción de ese tipo, que deriva en que el sistema debe incluir un fragmento de funcionalidad para proporcionar al usuario el resultado de valor esperado.

Por tanto, los casos de uso representan los requisitos funcionales del sistema, es decir, las necesidades de los usuarios y de la organización en donde se desplegará el sistema. Dicho con otras palabras, el modelo de casos de uso responde a la pregunta ¿qué debe hacer el sistema para satisfacer a sus usuarios?

De nada sirve desarrollar un sistema muy avanzado tecnológicamente pero que no cumple con las expectativas de los usuarios; un producto software se crea para dar servicio a sus usuarios. Durante todo el desarrollo hay que tener en cuenta los casos de uso, esto es, los casos de uso guían todo el proceso de desarrollo. La forma en que esto se lleva a cabo es desarrollando el sistema incrementalmente, añadiendo poco a poco fragmentos de funcionalidad para realizar un conjunto de casos de uso, y demostrando la validez del sistema, comprobando que la funcionalidad desarrollada realmente realiza esos casos de uso. En definitiva, los casos de uso proporcionan el hilo conductor para avanzar en el desarrollo del software.

La captura de requisitos tiene dos objetivos: Encontrar los verdaderos requisitos y representarlos de un modo adecuado para los usuarios, clientes y desarrolladores. Los verdaderos requisitos son aquellos que cuando se implementen añadirán el valor esperado para los usuarios. Normalmente un sistema tiene muchos usuarios. Cada tipo de usuarios se representa por un actor. Los actores utilizan el sistema interactuando con los casos de usos. El modelo de casos de usos está compuesto por todos los actores y casos de uso de un sistema. (Schach, 2005)

❖ **Proceso Unificado Centrado en la Arquitectura**

La arquitectura del software sirve para poder contemplar el futuro sistema desde varios puntos de vista antes de que se construya y durante la construcción. El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema, es decir, especifica la forma, estructura y comportamiento del sistema desde diversos puntos de vista, todos ellos a un nivel de detalle que permita tener una idea global clara sin perderse en detalles insignificantes (que, precisamente, no influyen en la arquitectura del sistema).

De esta forma quedan cubiertos los dos aspectos fundamentales del sistema: ¿Qué hace el sistema para cada usuario? (casos de uso); ¿Cómo es el sistema y cómo funciona para realizar esos casos de uso? (arquitectura de software). Ambas especificaciones se complementan y evolucionan en paralelo. Por un lado, los casos de uso deben encajar en la arquitectura del sistema cuando se lleven a cabo (adaptar los casos de uso a lo que la arquitectura permite); por otro lado, la arquitectura debe permitir la realización de todos los casos de uso (adaptar la arquitectura a lo requerido por los casos de uso).

En la práctica, la arquitectura se diseña para permitir realizar los casos de uso, pero pueden existir limitaciones externas que obliguen a redefinir el modelo de casos de uso, dando lugar a un ciclo de realimentación. Pero existen otros parámetros que influyen en la arquitectura: la plataforma y tecnología de destino, componentes reutilizables disponibles, sistemas heredados y requisitos no funcionales. (Schach, 2005)

❖ **Proceso Unificado Iterativo e Incremental**

Propone una descomposición incremental del problema a través de refinamientos sucesivos y una producción incremental de la solución, a través de la realización de varios ciclos. Esta filosofía es lógica cuando se aplica a sistemas grandes ya que “no se puede abarcar todo a la vez”.

El Proceso Unificado tiene 4 fases o incrementos y en cada uno se consideran distintos flujos de trabajo o modelos que suponen mayor o menor número de horas de trabajo dependiendo de la fase incremental en la que se encuentre el desarrollo. Cada incremento consta de todos los pasos de un ciclo de vida completo, que se repiten (iteración) hasta obtener el desarrollo exacto del sistema. Para ello se hacen diagramas cada vez más precisos: el proceso es repetitivo iterativo y cada vez se amplía y detalla más incremental. (Schach, 2005)

2.2.20.2 Ciclo de vida del RUP

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

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

Cada ciclo produce una nueva versión del sistema, y cada versión es un producto preparado para su entrega; consta de un cuerpo de código fuente

incluido en componentes que puede compilarse y ejecutarse, además de manuales y otros productos asociados. Sin embargo, el producto terminado no sólo debe ajustarse a las necesidades de los usuarios, sino también a las de todos los interesados, es decir, toda la gente que trabajará con el producto. El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye el modelo de la arquitectura y el modelo visual (artefactos modelados con el UML). (Schach, 2005)

En la **Figura 2.3** se muestra como varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

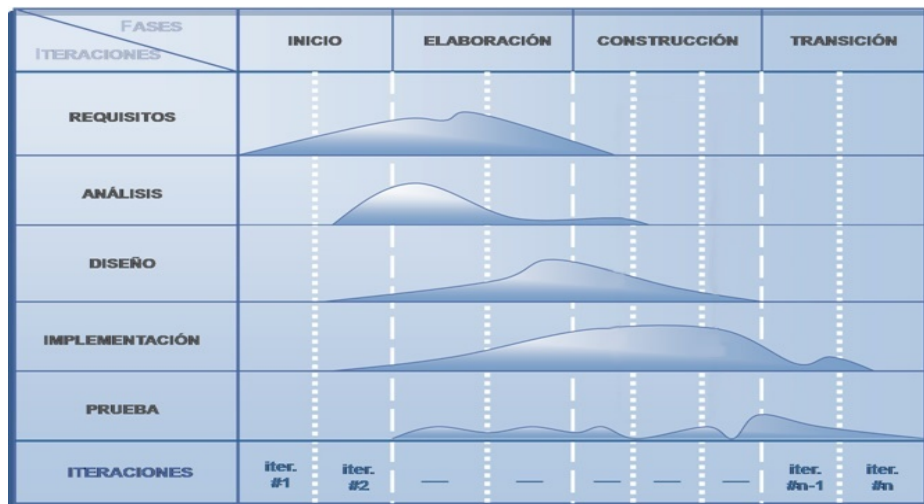


Figura 2.3 Ciclo de vida del RUP

Fuente: Schach, 2005

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases (inicio, elaboración, construcción y transición). Cada fase luego se descompone en iteraciones con sus respectivos incrementos, y concluyen con la disponibilidad de un conjunto de modelos desarrollados hasta conseguir un

estado predefinido, llamado hito, que permite la toma de decisiones importantes para así continuar la fase siguiente, permite además controlar el progreso del trabajo según pasa por cada fase. A continuación se describe brevemente cada fase. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ Fase de Inicio

El objetivo general de esta fase es poner en marcha el proyecto de desarrollo, pero antes se debe desarrollar el análisis del negocio lo cual lleva a delimitar el alcance (ámbito) del sistema, el ámbito es necesario para comprender lo que debe cubrir la arquitectura y así ubicar y enfrentar tempranamente los riesgos dentro de los límites del sistema.

En esta fase se identifican los requisitos funcionales del sistema mediante un modelo de casos de uso que comprende los casos de uso más críticos, además se identifican y priorizan los riesgos más importantes. Adicionalmente se planifica en detalle la fase de elaboración. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ Fase de Elaboración

Para esta fase se recopilan la mayor parte de los requisitos funcionales, aún pendientes, mediante los casos de uso y se diseña la arquitectura del sistema. Los objetivos fundamentales son:

- Recopilar la mayor parte de los requisitos definiéndolos como casos de uso.
- Establecer una arquitectura sólida para guiar las fases posteriores.

- Continuar la observación y control de los riesgos críticos.
- Completar el plan de proyecto.

Para su cumplimiento, es necesario tomar las decisiones considerando la comprensión global del sistema: ámbito, requisitos funcionales y no funcionales. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ **Fase de Construcción**

El objetivo principal de esta fase es culminar y entregar un producto software en su versión operativa inicial conocida como “Versión Beta”. El producto debería poseer la calidad adecuada y la seguridad de cumplir los requisitos para así emprender su aplicación en la comunidad de usuarios. Para alcanzar todo lo anterior se deben detallar todos los casos de uso y escenarios restantes, modificar si es pertinente la descripción de la arquitectura, dejar cerrados los modelos de análisis, diseño e implementación continuados en los flujos de trabajo de las iteraciones adicionales. Además integrar los subsistemas y someterlos a prueba y hacer lo propio con todo el sistema.

Al finalizar esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado en desarrollar. Y es aquí donde se establece y aclara la expectativa acerca de sí el producto cubre suficientemente las necesidades de los usuarios, como para hacer entrega de la versión inicial. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ Fase de Transición

Esta fase tiene como objetivos fundamentales:

- Cumplir con todos los requisitos establecidos en fases anteriores, hasta satisfacer a todos los usuarios.
- Gestionar y preparar todos los aspectos de implantación relativos a la operación del sistema en el contexto del usuario.
- La aplicación de pruebas de aceptación, que no son más, que la corrección de los posibles defectos acusados por los usuarios en la versión beta.

La fase de transición conlleva actividades como: la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos encontrados posterior a la entrega del software. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

2.2.20.3 Flujos de Trabajo del Proceso Unificado

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones se muestra en la **Figura 2.4**.

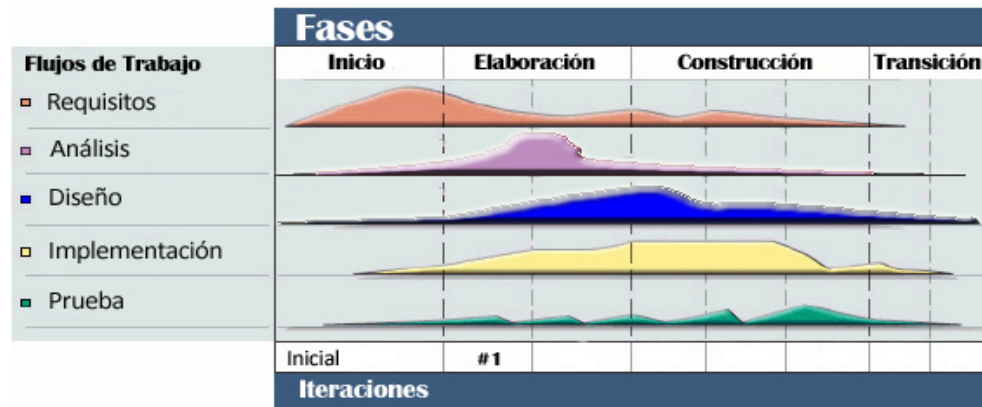


Figura 2.4: Flujos de Trabajo: Requisitos, Análisis, Diseño, Implementación y Prueba / Fases: Inicio, Elaboración, Construcción y Transición

Fuente: IBM. 2009

❖ Requisitos

Para la captura de requisitos se debe crear un modelo que represente el sistema a desarrollar, este modelo contiene los requisitos funcionales, es decir, los casos de uso, entonces se habla de Modelo de Casos de Uso.

El objetivo fundamental de este flujo de trabajo es identificar, documentar y detallar los casos de uso representativos e identificar igualmente a los actores. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ Análisis

El modelo de análisis, tiene dos propósitos fundamentales: refinar los casos de uso con más detalle y establecer la asignación inicial de funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento. También se debe señalar que el modelo de análisis hace abstracciones y evita resolver algunos problemas y tratar algunos requisitos,

por tanto se posponen al diseño y la implementación. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ **Diseño**

Es en este flujo en donde se moldea el sistema y se establece su forma, es decir, su arquitectura, ello con la intención de dar soporte a sus requisitos funcionales y no funcionales. Es importante señalar además, que el diseño se vale del modelo de análisis obtenido en el flujo de igual nombre, el cual proporciona una comprensión detallada de los requisitos, esto le es esencial para los propósitos establecidos por el diseño.

El diseño también se plantea la misión de establecer adecuadas entradas y correctos puntos de partida para el flujo de implementación mediante la captura de requisitos o subsistemas individuales, interfaces y clases. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ **Implementación**

El objetivo principal de este flujo de trabajo es desarrollar la arquitectura y sistema como un todo. Aquí se implementa el sistema en componentes como código fuente, y ejecutables entre otros. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

❖ **Pruebas**

Las pruebas verifican lo que resultó de la implementación y prueban las construcciones internas así como las versiones finales del sistema a entregarse.

Se planifican las pruebas de integración en cada iteración y las pruebas de sistema al final de cada iteración. Igualmente se diseñan e implementan los casos de prueba que indican que probar y se crean los procedimientos de prueba que señalan como hacer las pruebas. Finalmente se ejecutan las diferentes pruebas y se tratan los resultados de las construcciones, en caso de defectos se prueban nuevamente o son devueltas a un flujo de trabajo anterior. (Rumbaugh, J., Jacobson, I. y Booch, G., 2000)

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Tipos de Investigación

Existen muchos modelos y formas de clasificarlos. Sin embargo, lo importante es precisar los criterios de clasificación.

En este sentido se identifican:

- ✓ Tipos de investigación según el Nivel.
- ✓ Tipos de investigación según el Diseño.
- ✓ Tipos de investigación según el Propósito.

Sin embargo, independientemente de su clasificación, todos son tipos de investigación, y al no ser excluyentes, un estudio puede ubicarse en más de una clase.

3.1.1 Nivel de la Investigación

El nivel de investigación se refiere al grado de profundidad con que se aborda un fenómeno u objeto de estudio.

El nivel de investigación de este proyecto es del tipo **Descriptivo**.

La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o

comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos se refiere.

3.1.2 Diseño de la Investigación

El diseño de la investigación es la estrategia general que adopta el investigador para responder al problema planteado. En atención al diseño la investigación se clasifica en:

- ✓ Documental.
- ✓ De Campo.
- ✓ Experimental.

El diseño de investigación de este proyecto es de **Campo**.

La investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene información pero no altera las condiciones existentes.

3.2 Técnicas a Utilizar

Se entenderá por técnica, el procedimiento o forma particular de obtener datos o información.

Como el diseño de esta investigación es de **Campo**, las técnicas a utilizar serán las siguientes:

- ✓ **La Observación:** Es una técnica que consiste en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho, fenómeno o situación que se produzca en la naturaleza o en la sociedad, en función de unos objetivos de investigación preestablecidos.

La observación puede ser estructurada o no estructurada:

- ✓ **Estructurada:** Es aquella que además de realizarse en correspondencia con unos objetivos, utiliza guía diseñada previamente, en la que se especifican los elementos que serán observados.
- ✓ **No Estructurada:** Es la que se efectúa en función de un objeto, pero sin guía prediseñada que especifique cada uno de los aspectos que deben ser observados.
- ❖ **La Encuesta:** Se define como una técnica que pretende obtener información que suministra un grupo o muestra de sujetos acerca de si mismos, o en relación con un tema en particular.

La encuesta puede ser oral o escrita:

- ✓ **Encuesta Oral:** Esta se fundamenta en un interrogatorio “cara a cara” o por vía telefónica, en el cual el encuestador pregunta y el encuestado responde. Se realizan pocas y breves preguntas porque su duración es bastante corta.

- ✓ **Encuesta Escrita:** Se realiza a través de un cuestionario auto-administrado, el cual como su nombre lo indica, siempre es respondido de forma escrita por el encuestado.

- ❖ **La Entrevista:** Más que un simple interrogatorio, es una técnica basada en un dialogo o conversación “cara a cara”, entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida.

La entrevista se clasifica en:

- ✓ **Entrevista Estructurada o Formal:** Es la que se realiza a partir de una guía prediseñada que contiene preguntas que serán formuladas al entrevistado.

- ✓ **Entrevista no Estructurada o Informal:** En esta modalidad no se dispone de una guía de preguntas elaboradas previamente. Sin embargo, se orienta por unos objetivos preestablecidos, lo que permite definir el tema de la entrevista.

- ✓ **Entrevista Semi-estructurada:** Aun cuando existe una guía de preguntas, el entrevistador puede realizar otras no contempladas inicialmente. Esto se debe a que una respuesta puede dar origen a una pregunta adicional o extraordinaria.

CAPÍTULO IV

RESULTADOS

4.1 Fase de Inicio

La fase de inicio es la más importante del Proceso Unificado, en ella se establece una vista general de los objetivos del proyecto, determina si es viable y decide si se continuará el proyecto.

El objetivo de esta fase es conocer y analizar a fondo las actividades y procesos llevados a cabo en la Empresa Distribuidora Sellini, a través la comunicación entre los investigadores y los usuarios.

4.1.1 Descripción del Sistema Actual

La Empresa Distribuidora Sellini dedicada a la venta de filtros para todo tipo de maquinaria automotriz, realiza diariamente dos procesos, para la compra y venta de filtros, los cuales son:

- a) Compra de mercancía.
- b) Venta de mercancía.

A continuación se describen ambos procesos:

a) Compra de mercancía: El Gerente de Ventas, Vendedores y Almacenista, conjuntamente, realizan la reposición y adquisición de nuevos productos. El Gerente de Ventas elabora la Requisición en original y copia,

que contemplan los siguientes descriptores: cantidad, código y descripción del producto, el original es enviado al Gerente General quien elabora, a su vez, una solicitud de Cotización con los mismos descriptores: cantidad código y descripción del producto. Esta cotización debe hacerse; en original y tres (3) copias las cuales son enviadas a tres (3) posibles Proveedores. Una vez recibidas las Cotizaciones, el Gerente de Ventas elabora la Orden de Compra que contempla: Cantidad código, descripción y nombre de la empresa a la cual está dirigida, en original y tres (3) copias, para ser enviadas al Presidente de la Empresa para su aprobación. Esta Orden de Compra debe ir acompañada por las Cotizaciones de los Proveedores.

Una vez aprobada la Orden de Compra se envía al Proveedor la Orden original, las tres (3) copias son repartidas de la siguiente manera: al Gerente Ventas, al Almacenista y la otra copia va al archivo.

La mercancía es recibida por el Almacenista quien se encarga de verificar que la Orden de Compra original, que acompaña a la Factura, coincida con la copia archivada, verifica la mercancía, sella y firma original y copia de la Factura en aceptación de la misma. En el caso de compra a crédito el Gerente Administrativo indica la fecha de pago y recibe la copia de la Factura, en caso de compra de contado el Gerente Administrativo entrega el cheque y recibe la Factura original.

b) Venta de mercancía: Distribuidora Sellini vende su mercancía al Mayor y al Detal, a continuación se describen los procesos de venta:

- ✓ **Venta de mercancía al mayor:** Los Clientes envían una solicitud de Cotización, la cual es recibida por el Gerente General y enviada a almacén para verificar su existencia. Una vez comprobada su

existencia, se da respuesta a la solicitud de Cotización. De ser aceptada por el Cliente éste emite la Orden de Compra en original. El Gerente de Ventas elabora la Factura en original y dos (2) copias, el original y la copia 1 destinadas al Cliente y la copia 2 es archivada por el Gerente Administrativo. Se genera y despacha el pedido junto con la Orden de Compra original y la Factura en original y copia. El Cliente indica las condiciones de pago, de ser a crédito se establece la fecha de pago y se entrega la copia de la Factura; la original es retenida por el Gerente Administrativo de la Empresa Distribuidora Sellini hasta que sea cancelada la deuda total.

Si el pago es de contado se recibe el cheque y se entrega original de la Factura al Cliente y la copia 1 es desechada.

- ✓ **Venta de mercancía al detal:** Los Clientes se dirigen a la empresa y realizan sus pedidos, se verifica en almacén la existencia y se despacha la mercancía, el Cajero procesa la mercancía, efectúa el cobro y genera Factura en original y copia. El pago se ejecuta de contado y se le entrega al Cliente la Factura original. La copia es archivada.

4.1.2 Modelo del Dominio

Un Modelo de Dominio es un diagrama de la disciplina de análisis, construido con las reglas de UML, representado como uno o más diagramas de clases y que contiene no conceptos propios de un sistema de software, sino de la propia realidad física.

Los modelos de dominio pueden utilizarse para capturar y expresar los conocimientos obtenidos en un área bajo análisis como paso previo al diseño de un sistema. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sistema al cual se va a servir.

Es posible capturar un mayor grado de detalle en estos modelos de dominio, corresponde al analista decidir cuánto detalle va a ser necesario y hasta dónde llegar a modelar. El objetivo es capturar lo necesario para comprender donde va a funcionar el sistema que se está diseñando y esto demanda una cantidad distinta de detalles cada vez.

A continuación se describe el proceso manual del Modelo de Dominio.

A través de esta descripción se pretende lograr una mayor comprensión del diagrama explicando paso a paso el proceso, el cual inicia de la siguiente manera: Uno o muchos Clientes elaboran una o muchas Órdenes de Compras y se envían a un Gerente de Ventas, el cual solicita pedido a un Almacenista, éste despacha uno o muchos pedidos, uno o muchos pedidos se envían a uno o muchos Clientes, un Cliente paga una o muchas Facturas y envían a un Gerente Administrativo.

Un Gerente de Ventas elabora una o muchas Órdenes de Compras, envía a uno o muchos Proveedores, uno o muchos Proveedores generan uno o muchos pedidos, envían uno o muchos pedidos a un Almacenista, éste reporta a un Gerente de Ventas, un Gerente de Ventas envía una o muchas Facturas a un Gerente Administrativo, éste paga a uno o muchos Proveedores.

En la **Figura 4.1** se muestra el Modelo de Dominio del sistema bajo estudio.

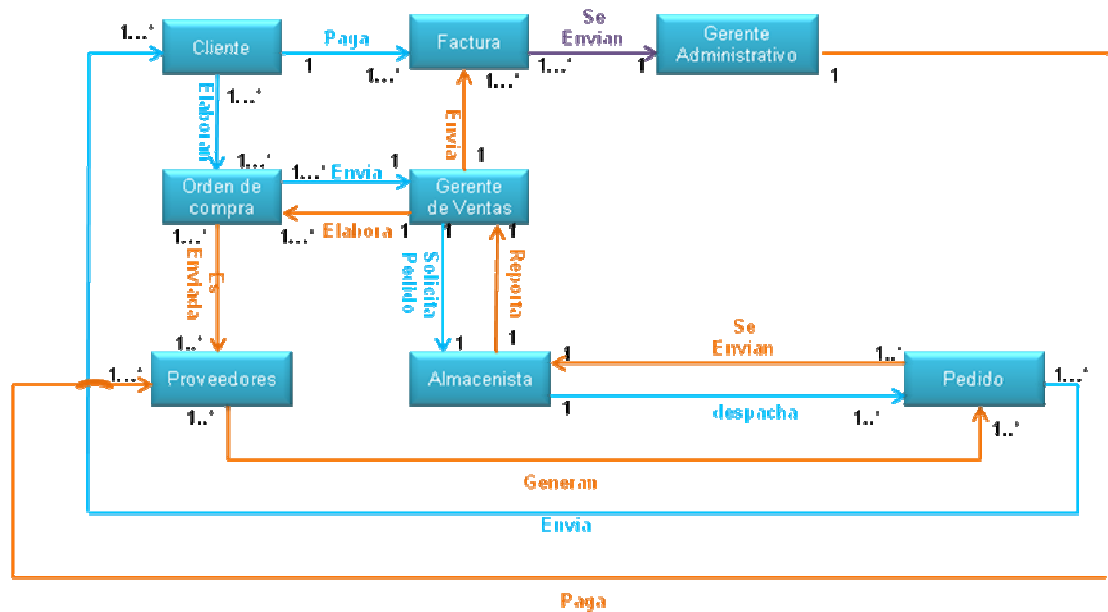


Figura 4.1: Modelo De Dominio
Fuente: Elaboración propia

4.1.2.1 Glosario de Términos

Para un mayor entendimiento del modelo de dominio se describirán cada uno de los términos utilizados en la **Figura 4.1**.

- ✓ **Gerente de Ventas:** Es la persona encargada supervisar a los Vendedores, coordina los despachos y la recepción de la mercancía.
- ✓ **Gerente Administrativo:** Se encarga de controlar la parte administrativa de la empresa, cuenta por cobrar, cuentas por pagar, banco, nómina e inventario.

- ✓ **Almacenista:** Persona responsable del inventario de la mercancía para la venta, es quien recibe y entrega los productos.
- ✓ **Cliente:** Es la persona que realiza pedidos y paga facturas a la empresa.
- ✓ **Proveedor:** Es la persona que surte a la empresa con mercancía, a la cual se le pagan las facturas emitidas.
- ✓ **Orden de compra:** En esta se especifica la cantidad, el código y la descripción del producto.
- ✓ **Pedido:** Incluye los productos que vende o compra la empresa, generados luego de una orden de compra.
- ✓ **Factura:** Es un documento administrativo que refleja toda la información de una operación de compraventa. La información fundamental que aparece en una factura debe reflejar la entrega de un producto, junto a la fecha de pago, además de indicar la cantidad a pagar.

4.1.3. Estructura Organizativa

La estructura organizativa bajo la cual se rige la Empresa Distribuidora Sellini se muestra y se describe a continuación.

- ✓ **Presidente:** Es la persona con mayor autoridad dentro de la empresa encargada de dirigir el funcionamiento de la misma, tiene la última palabra a la hora de tomar una decisión en cualquier reunión, sesión de trabajo o asamblea.
- ✓ **Gerente General:** Se encarga de designar todas las posiciones gerenciales, realiza evaluaciones periódicas acerca del cumplimiento de las funciones de los diferentes departamentos, coordina con la oficina administrativa para asegurar que los registros y sus análisis se

están llevando correctamente. Éste le rinde cuenta al presidente de la empresa.

- ✓ **Gerente Administrativo:** Se encarga de controlar la parte administrativa de la empresa, cuentas por cobrar, cuentas por pagar, banco, nómina e inventario.
- ✓ **Gerente de Ventas:** Supervisa a los vendedores, realiza las rutas de visitas a los clientes, coordina los despachos y recepción de mercancía, verifica el crédito rotativo de las ventas por cada cliente.
- ✓ **Secretaria:** Sus funciones principales están relacionadas con el trabajo de oficina, como son:
 - Tramitar la entrada y salida de la correspondencia.
 - Recibir documentos.
 - Atender llamadas telefónicas.
 - Canalizar las entrevistas de su superior.
 - Archivar los documentos.
 - Realizar los cálculos elementales.
 - Estar al día de la tramitación de expedientes.
 - Tener actualizada la agenda, tanto telefónica como de direcciones y de reuniones.
 - Dominar la herramienta Microsoft Office (Excel).
- ✓ **Contador Público:** Es un profesional externo a la empresa, dedicado a aplicar, manejar e interpretar la contabilidad de la empresa, con la finalidad de producir informes para la gerencia, que a su vez, sirvan a la toma de decisiones. Lleva los libros o registros contables, registros de bienes y derechos. Dentro los informes del Contador se encuentran los estados contables o estados financieros y las rendiciones de cuentas o balances de ejecuciones presupuestarios, utilizados internamente en la empresa.

- ✓ **Comisario:** Es la persona que asesora a la empresa. Se encargada de revisar y aprobar los estados financieros generados por el Contador de la empresa.
- ✓ **Vendedor:** Es la persona a quien se le encomienda la venta de los productos que posee la empresa.

Algunas de las tareas que realiza un vendedor son:

- Investigación del mercado para detectar nuevos clientes.
 - Atención de reclamos.
 - Cobranza de sus clientes.
 - Oferta de mercancía para la venta.
 - Sugiere la compra de nuevos productos requeridos por la cartera de clientes.
- ✓ **Chofer:** Se encarga de entregar la mercancía, de recibir la factura y el pago en caso de que lo se haya realizado una venta de contado. Retira la mercancía de los Proveedores, así como también realiza pagos de servicios, diligencias bancarias, retiro de los pagos en las ventas a crédito y otras diligencias necesarias para la empresa.
 - ✓ **Almacenista:** Persona responsable del inventario de la mercancía para la venta, es quien recibe y entrega los productos. Debe mantener el depósito ordenado, respetando los lineamientos de almacenamiento, tales como: la ubicación física, clasificación y ordenamiento.
 - ✓ **Cajera:** Entre las funciones que cumple la cajera están:
 - Recibir pagos.
 - Conformar cheques con el banco emisor. vía telefónica.
 - Cancelar las facturas.
 - Realizar entrada a caja mediante el recibo de ingreso y Concilia caja.

- Elaborar depósitos bancarios por día de la cobranza y anexarlo a la conciliación de caja.

En la **Figura 4.2** se muestra la Estructura Organizativa de la Empresa Distribuidora Sellini.

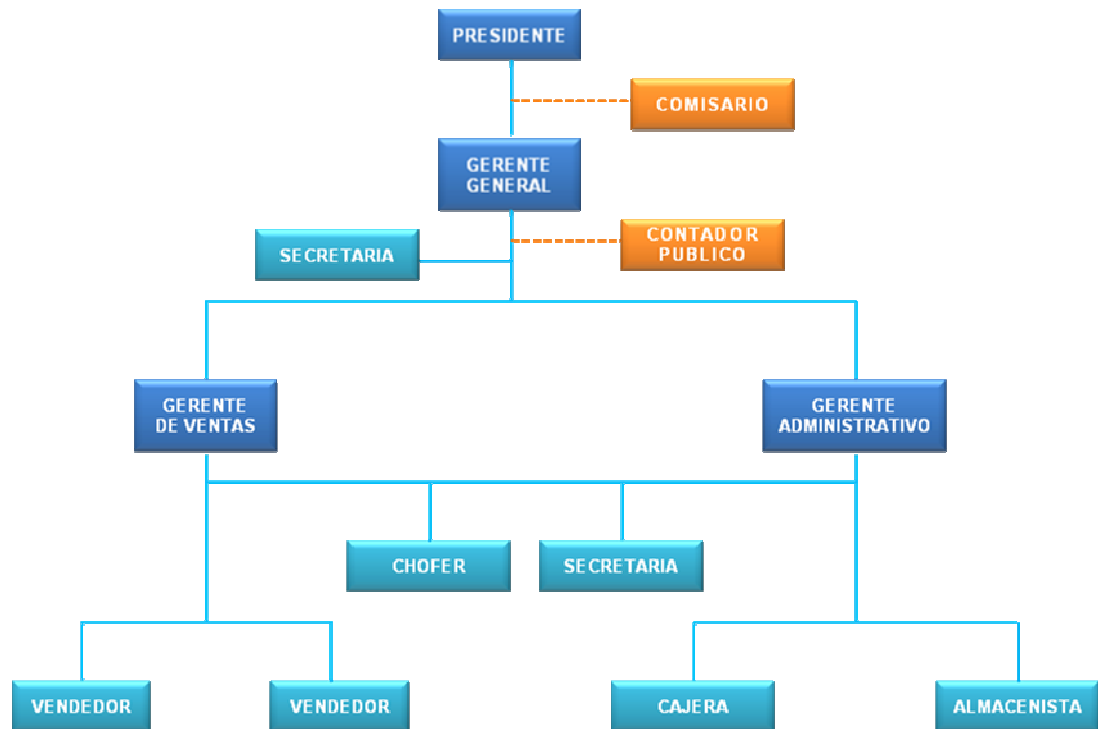


Figura 4.2: Estructura Organizativa de la Empresa Distribuidora Sellini
Fuente: Empresa Distribuidora Sellini

4.1.3.1 Diagrama del Sistema y Ambiente ampliado

La Empresa Distribuidora Sellini está conformada por un Departamento llamado Dirección General en el cual intervienen diversos entes privados y compradores independientes que permiten que la Empresa avance satisfactoriamente en el cumplimiento de sus actividades diarias.

La Gerencia está conformada por la Gerencia General, Gerencia Administrativa y la Gerencia de Ventas, las cuales mantienen relaciones entre si.

En la **Figura 4.3** se muestra de forma gráfica la relación que existe entre cada una de las áreas que conforman la Empresa Distribuidora Sellini, así como la comunicación interdepartamental de las mismas y con el entorno que la rodea, lo que permite en gran medida una mayor comprensión de su estructura y funcionamiento.



Figura 4.3: Diagrama del Sistema y Ambiente ampliado
Fuente: Elaboración Propia

4.1.4 Definición de la Misión-Visión

En este punto se describe la Misión y Visión que posee la Empresa Distribuidora Sellini.

❖ **Misión**

“Distribuidora Sellini es una empresa perteneciente al sector metalmeccánico a nivel nacional, dedicada a la venta y distribución de elementos filtrantes para todo tipo de maquinaria automotriz, ofreciendo la mayor variedad en filtros de la más alta calidad a precios justos.”

❖ **Visión**

“Aspirando a ser la empresa líder en comercialización de filtros para motores de combustión interna.”

4.1.5 Análisis de la Problemática

Para determinar los diversos factores que ocasionan problemas en la empresa, fue necesario realizar varias visitas y entrevistas no estructuradas al personal que labora en la misma. en la actualidad, Distribuidora Sellini carece de herramientas que le permita llevar un control óptimo de la información que maneja; debido a esto; es necesario hacer énfasis en los siguientes aspectos para diseñar así el nuevo sistema que permitirá mejorar esta problemática.

Esta afirmación es el resultado de la observación directa hecha a la Empresa por los investigadores; en la misma pudieron ser observados los inconvenientes que se profundizan a continuación:

- ✓ No se generan reportes automáticos y concisos de la información requerida.
- ✓ Los datos de los Clientes y Proveedores son almacenados en fichas personalizadas las cuales son llenadas manualmente con toda la información correspondiente.
- ✓ Cada filtro manejado por esta empresa posee un código único, asociado a su fabricante, que permite identificarlo y almacenarlo en una posición física, estos datos son registrados con la herramienta Microsoft Office (Excel), generando duplicidad y pérdida de tiempo al momento de realizar consultas o reportes de interés para la Empresa.
- ✓ Presenta problemas por retrasos a la hora de elaborar las facturas.
- ✓ Hay retardos en la ejecución manual de los registros.

4.1.6 Flujo de Trabajo de Requisitos

Muchos proyectos de desarrollo de sistemas, tienen problemas con su calidad o lo que es peor, no llegan a concluir; fracasando por diversas causas dentro de las que juega un papel fundamental una mala gestión de requisitos ya que, al ser uno de los primeros y fundamentales flujos de trabajo que se llevan a cabo a la hora de construir un software, los errores de comprensión cometidos en esta etapa inicial de los proyectos son los más costosos de resolver.

Un requisito es una característica, condición funcional o capacidad, que debe tener un producto o servicio para satisfacer las necesidades de un determinado cliente.

4.1.6.1 Contexto del Sistema

Se habla del término contexto porque lo que interesa, es conocer las acciones que ejecutan los actores con relación a los objetos de gestión que modelarán en el sistema.

Para comprender el contexto en el cual se desarrolló el sistema bajo estudio fue necesario realizar entrevistas no estructuradas, al personal que labora en la empresa (usuarios del sistema) y visitas a los diferentes departamentos de la Empresa Distribuidora Sellini para de esta manera, conocer y entender los procesos llevados a cabo diariamente dentro de la mencionada Empresa.

❖ Requisitos Esenciales del Sistema

- ✓ Automatizar el proceso de facturación de la Empresa Distribuidora Sellini.
- ✓ Guardar en la Base de Datos toda la información referente a los Clientes y Proveedores, para así poder realizar reportes automatizados de cualquier índole, reduciendo y eliminando el almacenamiento de información en archivos.
- ✓ Establecer claves de acceso al personal asociado, para mantener la seguridad del sistema.

❖ Requisitos Funcionales

En los requisitos funcionales se concentran aquellos que describen los servicios (funciones) que se esperan del sistema y que tengan que ver con algún proceso propiamente de gestión (crear, modificar, eliminar, consultar).

- ✓ Registrar, consultar, modificar y eliminar usuarios.
- ✓ Registrar, consultar, modificar y eliminar datos de los proveedores.
- ✓ Registrar, consultar, modificar y eliminar datos de los clientes.
- ✓ Registrar, consultar, modificar y eliminar productos.
- ✓ Realizar el seguimiento de la mercancía que entra y sale de la empresa.
- ✓ Realizar consultas de interés para los actores del sistema.
- ✓ Generar reportes de la información que pueda interesar a los usuarios.

❖ **Requisitos no Funcionales**

- ✓ La estructura y diseño del sistema es escalable lo que significa que debe adaptarse fácilmente a cualquier cambio o mejora.
- ✓ Posee una interfaz visual, fácil y cómoda que permite la interacción entre los usuarios y el sistema.

❖ **Identificación de los Riesgos**

Toda actividad conlleva un riesgo, ya que la actividad exenta de ello representa inmovilidad total. Es necesario conocer los posibles riesgos del sistema para evitar o controlar los errores en el futuro.

En la **Tabla 4.1** se muestra la lista de riesgos a ser considerada en el sistema.

4.1.6.2 Modelo de Caso de Uso

Un modelo de caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema. Cada caso de uso proporciona uno o más

escenarios que indican cómo debería interactuar el sistema con el usuario para conseguir un objetivo específico. Lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Para obtener un modelo de caso de uso acorde con los requerimientos, es necesario realizar la identificación de los actores del sistema y de los casos de uso.

En la **Figura 4.4** se muestra el diagrama de caso de uso general del sistema.

Tabla 4.1: Lista de Riesgos del Sistema

Descripción	Nivel	Impacto	Responsable	Ejecución
No se establece comunicación entre el Sistema y la Base de Datos propuesta	Muy Crítico	Software	Desarrolladores	Verificar la Base de Datos.
No lograr una interfaz adecuada para las necesidades del Sistema.	Crítico	Software	Desarrolladores	Profundizar el análisis del flujo de actividades con apoyo del futuro usuario
Rendimiento del equipo en el que está instalada la Base de Datos	Crítico	Software y/o Hardware	Distribuidora Sellini	Revisión y diagnóstico del equipo.
Rendimiento de las conexiones	Crítico	Hardware	Distribuidora Sellini	Revisar cableado.

Fuente: Elaboración propia

❖ **Identificación de los Actores**

Es importante realizar la identificación de los actores que utilizarán el sistema “**Sellini**” ya que ello permite obtener una visión del entorno del sistema. A continuación se describe cada uno de los actores o usuarios del sistema:

- ✓ **Gerente General:** Se encarga de designar todas las posiciones gerenciales, realiza evaluaciones periódicas acerca del cumplimiento de las funciones en los diferentes departamentos; éste será el Administrador del Sistema. Tiene la potestad de Gestionar Sesión, Proveedores, Clientes y Productos así como también de realizar las consultas. Rinde cuentas al Presidente de la empresa.
- ✓ **Gerente de Ventas:** Supervisa a los vendedores, coordina los despachos y compra de mercancía, verifica el crédito rotativo de las ventas por cada Cliente. Tienen acceso a los casos de uso Gestionar Sesión, Gestionar Clientes, y realizar Consultas.
- ✓ **Gerente Administrativo:** Se encarga de controlar la parte administrativa de la empresa, cuentas por cobrar, cuentas por pagar, banco, nómina e inventario. Éste puede Gestionar Sesión, Proveedores y Productos así como también realiza Consultas.
- ✓ **Vendedor:** Se encarga de procesar los pedidos que los clientes realizan a la empresa. Tiene acceso a la Gestión de Sesión, realizar Consultas y a procesar ventas.
- ✓ **Personal Administrativo:** Los actores denominados Personal Administrativo son:

- **Secretaria:** Lleva el orden de las actividades que deben realizar sus superiores. Ésta tiene acceso a la Gestión de Sesión, y a la realización de Consultas.
- **Presidente:** Es la persona con mayor autoridad dentro de la empresa encargada de dirigir el funcionamiento de la misma, éste solo realiza operaciones básicas como Gestionar Sesión y procesar Consultas.

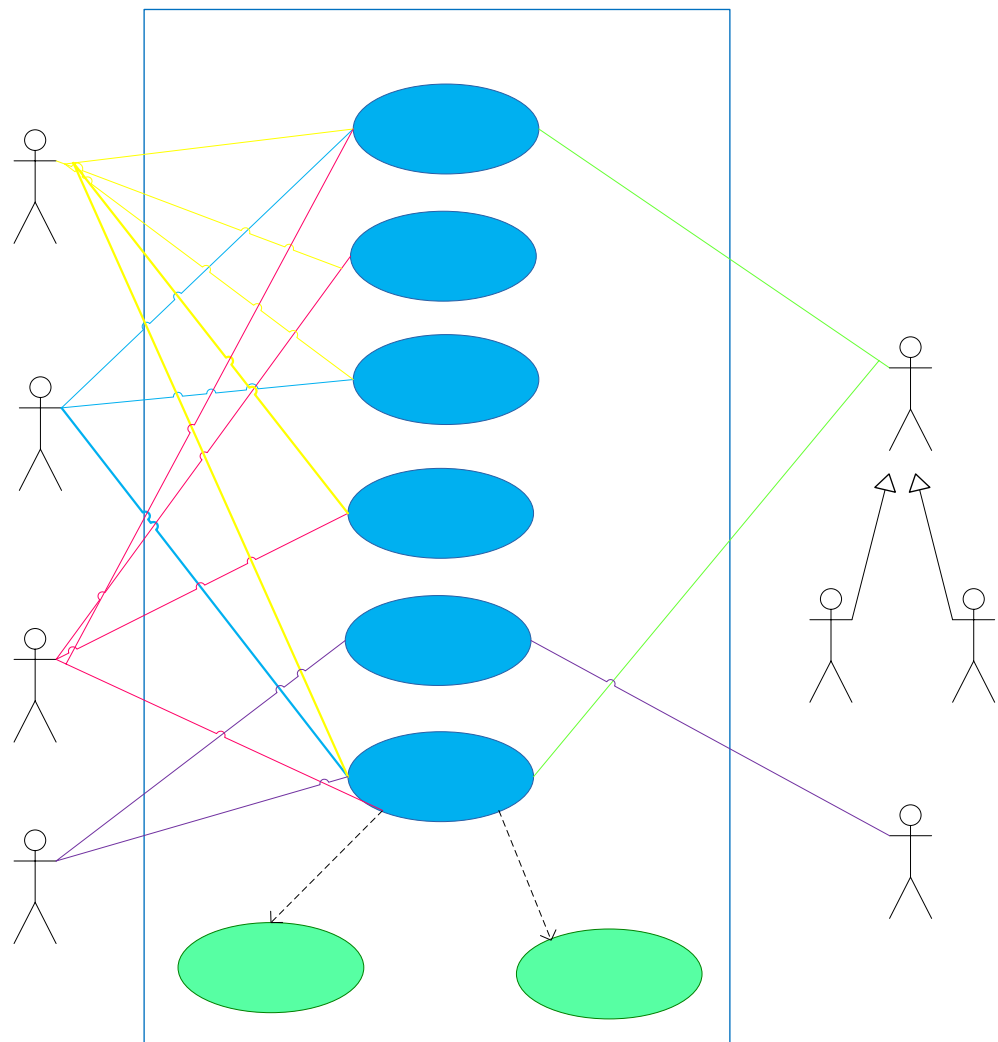


Figura 4.4: Diagrama de Caso de Uso General del Sistema

Fuente: Elaboración Propia

Gerente General

4.1.6.3 Descripción de los Casos de Uso del Sistema

El modelo de casos de uso muestra los principales procesos que ejecuta el sistema de “**Sellini**”. A continuación se realiza la explicación de cada uno de ellos a fin de comprender mejor el funcionamiento del sistema.

- ❖ **Gestionar Sesión:** Este caso de uso permite a los actores entrar al sistema para realizar las operaciones que necesite.

- **Pre-condición:** El actor debe ingresar al menú Inicio de Sesión donde ingresará su nombre y clave de usuario, de esta manera el sistema validará su entrada y le dará acceso dentro de los roles que le corresponden, de no pertenecer a la Base de Datos del Sistema le negará la entrada al mismo.

- ✓ **Flujo de Eventos:**
 - Flujo Principal:
 - 1 El actor invoca al caso de uso Gestionar Sesión.
 - 2 El sistema despliega una interfaz como campos de datos a rellenar.
 - 3 El actor rellena los campos.
 - 4 El sistema verifica los datos ingresados.
 - 5 El actor entra al sistema
 - 6 Finaliza el caso de uso.

 - Flujo Alternativo:
 - Paso 2, el usuario selecciona cancelar.
 - Paso 3, el usuario selecciona cancelar.

Paso 4, el sistema no puede verificar los datos debido a que los campos se encuentran vacíos, el Usuario no se encuentra registrado, los datos ingresados no coinciden entre ellos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

❖ **Gestionar Proveedores:** Este caso de uso permite tanto al Gerente General como al Gerente Administrativo realizar operaciones como Crear, Modificar o Eliminar Proveedores, las cuales se describen a continuación:

➤ **Crear Proveedor:** Permite tanto al Gerente General como al Gerente Administrativo Crear nuevos Proveedores en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Proveedores y seleccionar la opción Crear.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Proveedor.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Crear.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega una interfaz con campos para ser rellenados con los datos del Proveedor.
- 5 El actor ingresa los datos del Proveedor.
- 6 El actor solicita Crear Proveedor.
- 7 El sistema registra los datos.
- 8 Finaliza el caso de uso.

- **Flujo Alternativo:**

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 5, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que la opción seleccionada se encuentra vacía, ya existe o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Modificar Proveedor:** Permite, tanto al Gerente General como al Gerente Administrativo, modificar los proveedores existentes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Proveedores y seleccionar la opción Modificar.

✓ **Flujo de Eventos:**

- **Flujo Principal:**

1 El actor invoca al caso de Gestionar Proveedores.

2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Modificar.

3 El actor selecciona el botón deseado.

4 El sistema despliega un campo de búsqueda donde se introducirá el nombre del Proveedor deseado.

5 El sistema despliega un formulario con los datos del Proveedor.

6 El actor modifica los datos necesarios del Proveedor.

7 El actor solicita Modificar Proveedor.

8 El sistema registra los datos.

9 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Proveedor no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Eliminar Proveedor:** Permite, tanto al Gerente General como al Gerente Administrativo, eliminar los proveedores existentes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Proveedores y seleccionar la opción Eliminar.

✓ **Flujo de Eventos:**

- Flujo Principal:

1 El actor invoca al caso de Gestionar Proveedores.

2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Eliminar.

3 El actor selecciona el botón deseado.

- 4 El sistema despliega un campo de búsqueda donde se introducirá el nombre del Proveedor deseado.
- 5 El sistema despliega un formulario con los datos del Proveedor y un botón con la opción Eliminar Proveedor.
- 6 El actor selecciona la opción de Eliminar Proveedor.
- 7 El sistema elimina el Proveedor.
- 8 Finaliza el caso de uso.

- Flujo Alterno:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Proveedor no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede eliminar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

❖ **Gestionar Clientes:** Permite al Gerente General y al Gerente de Ventas realizar operaciones como, Crear, Modificar o Eliminar Clientes. Estas operaciones se describen a continuación:

➤ **Crear Cliente:** Permite, tanto al Gerente General como al Gerente de Ventas, crear nuevos clientes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Clientes y seleccionar la opción Crear.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Cliente.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Crear.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega una interfaz con campos para ser rellenos con los datos del Cliente.
- 5 El actor ingresa los datos del Cliente.
- 6 El actor solicita Crear Cliente.
- 7 El sistema registra los datos.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 5, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que la opción seleccionada se encuentra vacía, ya existe o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Modificar Cliente:** Permite, tanto al Gerente General como al Gerente de Ventas, modificar los clientes existentes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Clientes y seleccionar la opción Modificar.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Clientes.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Modificar.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá la nombre del Cliente deseado.
- 5 El sistema despliega un formulario con los datos del Cliente.
- 6 El actor modifica los datos necesarios del Cliente.
- 7 El actor solicita Modificar Cliente.
- 8 El sistema registra los datos.
- 9 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Cliente no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso7, el sistema no puede almacenar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- **Eliminar Cliente:** Permite, tanto al Gerente General como al Gerente de Ventas, eliminar los clientes existentes en el sistema.
- ✓ **Pre-condición:** El actor debe ingresar al menú Clientes y seleccionar la opción Eliminar.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Clientes.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Eliminar.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá la nombre del Cliente deseado.
- 5 El sistema despliega un formulario con los datos del Cliente y un botón con la opción Eliminar Cliente
- 6 El actor selecciona la opción de Eliminar Cliente.
- 7 El sistema elimina el Cliente.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Cliente no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso7, el sistema no puede eliminar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- ❖ **Gestionar Productos:** Permite, al Gerente General y al Gerente Administrativo, realizar operaciones como: Crear, Modificar o Eliminar productos. Estas operaciones se describen a continuación:
 - **Crear Producto:** Permite, tanto al Gerente General como al Gerente Administrativo, crear productos en el sistema.
 - ✓ **Pre-condición:** El actor debe ingresar al menú Productos y seleccionar la opción Crear.
 - ✓ **Flujo de Eventos:**
 - Flujo Principal:
 - 1 El actor invoca al caso de Gestionar Producto.
 - 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Crear.
 - 3 El actor selecciona el botón deseado.
 - 4 El sistema despliega una interfaz con campos para ser rellenados con los datos del producto.
 - 5 El actor ingresa los datos del Producto.
 - 6 El actor solicita Crear Producto.

- 7 El sistema registra los datos.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 5, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que la opción seleccionada se encuentra vacía, ya existe o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Modificar Producto:** Permite, tanto al Gerente General como al Gerente Administrativo, modificar los productos existentes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Productos y seleccionar la opción Modificar.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Productos.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Modificar.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá el código del Producto deseado.
- 5 El sistema despliega un formulario con los datos del Producto.

- 6 El actor modifica los datos necesarios del producto.
- 7 El actor solicita Modificar Producto.
- 8 El sistema registra los datos.
- 9 Finaliza el caso de uso.

- Flujo Alterno:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Producto no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Eliminar Producto:** Permite, tanto al Gerente General como al Gerente Administrativo, eliminar los productos existentes en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú Productos y seleccionar la opción Eliminar.

✓ **Flujo de Eventos:**

- Flujo Principal:

1 El actor invoca al caso de Gestionar Productos.

- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Eliminar.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá el código del Producto deseado.
- 5 El sistema despliega un formulario con los datos del producto y un botón con la opción Eliminar Producto.
- 6 El actor selecciona la opción de Eliminar Producto.
- 7 El sistema elimina el Producto.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Producto no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede eliminar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

❖ **Procesar Venta:** Permite al Vendedor procesar los Pedidos hechos por los Clientes, estos pedidos pueden ser Creados y Eliminados.

➤ **Crear Pedido:** Permite al Vendedor crear pedidos en el sistema.

- ✓ **Pre-condición:** El actor debe ingresar al menú Pedidos y seleccionar la opción Crear.

- ✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de uso Procesar Venta.
- 2 El sistema despliega un menú con diferentes opciones a seleccionar, entre las cuales se encuentra Crear.
- 3 El actor selecciona la opción Crear.
- 4 Se muestra la interfaz seleccionada y se rellenan los campos con toda la información del Pedido.
- 5 El sistema registra el pedido.
- 6 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el usuario selecciona cancelar.

Paso 3, el usuario selecciona cancelar.

Paso 4, el usuario selecciona cancelar.

Paso 5, el sistema no puede almacenar los datos debido a que alguno se encuentra vacío o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- **Eliminar Pedido:** Permite al Vendedor eliminar los pedidos existentes en el sistema.

- ✓ **Pre-condición:** El actor debe ingresar al menú Pedidos y seleccionar la opción Eliminar.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Procesar Venta.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Eliminar.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá el nombre del cliente deseado.
- 5 El sistema despliega un formulario con todos los pedidos del Cliente, con la opción de Eliminar al lado de cada pedido.
- 6 El actor selecciona la opción de Eliminar Pedido.
- 7 El sistema elimina el Pedido.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el Cliente no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede eliminar los datos debido a que no ha seleccionado ninguna opción o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- ❖ **Realizar Consultas:** Permite a todos los actores registrados en el sistema realizar consultas de interés relacionada con los Usuarios, Productos, Proveedores, Clientes y Stock mínimo.

- **Pre-condición:** El actor debe ingresar al menú consultas donde podrá consultar el listado de los Productos, Proveedores, Clientes, Usuarios y Depósito. Este caso de usos está disponible para todos los actores del sistema. Es importante mencionar que cada actor tiene acceso solo a las consultas permitidas. El Gerente General puede consultar Usuario, Productos, Proveedores, Clientes y Depósito. Mientras que el resto de los actores solo podrán consultar Productos, Proveedores, Clientes y Depósito.

- ✓ **Flujo de Eventos:**
 - Flujo Principal:
 - 1 El actor invoca al caso de uso Realizar Consultas.
 - 2 El sistema despliega un menú con diferentes opciones a seleccionar, las cuales son: Productos, Proveedores y Clientes.
 - 3 El actor selecciona la opción que desea.
 - 4 Se muestra la interfaz seleccionada donde se mostrará un listado de la información escogida.
 - 5 El sistema busca la información.
 - 6 Finaliza el caso de uso.

 - Flujo Alterno:

Paso 2, el usuario selecciona cancelar.

Paso 3, el usuario selecciona cancelar.

Paso 4, el usuario selecciona cancelar.

Paso5, el sistema no puede buscar la información debido a que se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

4.1.7 Flujo de Trabajo de Análisis

Permite analizar detalladamente, los datos conseguidos en la captura de requisitos con la finalidad de refinándolos y estructurándolos, para así obtener una comprensión más precisa de los mismos, que ayuden a estructurar completamente el sistema.

4.1.7.1 Paquetes de Análisis del Sistema

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas; evidenciando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de los mismos y minimizar su acoplamiento externo entre los paquetes. Cada paquete puede asignarse a un individuo o a un equipo y las dependencias entre ellos, pueden indicar el orden de desarrollo requerido.

Los usos más comunes para los diagramas de paquete son para organizar diagramas de casos de uso y diagramas de clase.

Los paquetes que contienen la funcionalidad del sistema, y que representa los casos de uso y sus respectivas realizaciones en análisis son los paquetes de Gestión de Sesión, Realización de Consultas, Procesar Ventas y Gestión de Información. También está contenido el paquete de Base de Datos que representa la Base de Datos del sistema y la funcionalidad necesaria para la conexión y el tratamiento de los datos, el resto de los paquetes tienen dependencia de éste ya que todos necesitan de los datos para poder ejecutar sus procesos.

En la **Figura 4.5** se muestra el diagrama de paquete de análisis del sistema.

❖ **Especificación de los Grupos Formalmente Descritos**

Es importante conocer el nivel de acceso y la jerarquía que tiene cada usuario dentro del sistema, para de esta manera definir cuál es la información a la que pueden acceder.

Los actores con menor nivel dentro del sistema son el Presidente y la Secretaria, los cuales sólo pueden realizar las operaciones básicas, el Gerente de Ventas y el Gerente Administrativo poseen los privilegios de los anteriores y adicionalmente, pueden gestionar algunos grupos a los cuales tienen acceso, luego está el Vendedor el cual puede realizar operaciones básicas y procesar ventas; finalmente, se encuentra el Gerente General el cual tiene el mayor nivel dentro del sistema, éste tiene acceso a la mayoría de los paquetes del sistema (el único paquete al que no accede es al de procesar ventas).

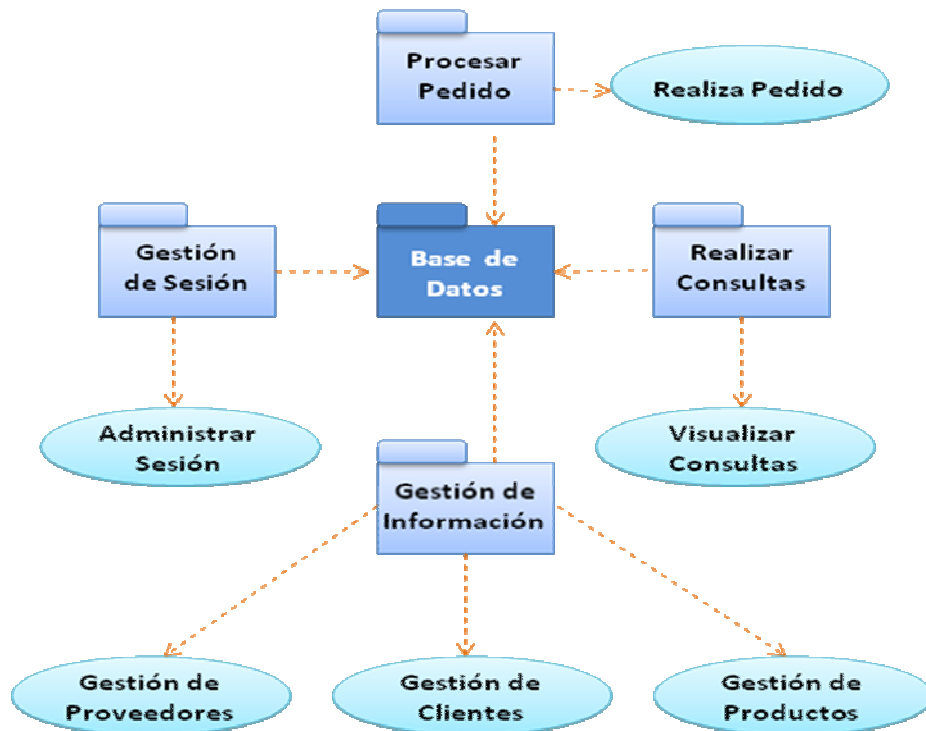


Figura 4.5: Diagrama de Paquete de Análisis del Sistema
Fuente: Elaboración Propia

4.1.7.2 Diagrama de Clase de Análisis

Luego de identificar y analizar el diagrama de casos de uso general del sistema, se procede a realizar los diagramas de clases de análisis con la información que se maneja del sistema.

- ✓ **Clase de Interfaz:** Se utiliza para modelar la interfaz entre el sistema y sus actores.
- ✓ **Clase de Entidad:** Se utilizan para modelar información que posee una larga vida y que a menudo es persistente.

- ✓ **Clase de Control:** Representan la coordinación, secuencia, transacciones y control, de los objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.

❖ Descripción del Caso de Análisis

- ✓ **Gestionar Sesión:** El caso de uso Gestionar Sesión, representa la interfaz principal, ésta le indicará al usuario que debe ingresar su Nombre y Clave para permitirle el acceso al sistema, el Gestor Usuario se encarga de validar los datos ingresados y enviarlos a la unidad correspondiente. El usuario podrá tener acceso sólo a las interfaces que le correspondan. Este diagrama se muestra en la [figura 4.6](#).

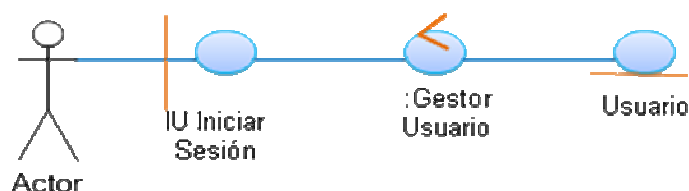


Figura 4.6: Diagrama de Clases de Análisis para el Caso de Uso Gestionar Sesión

Fuente: Elaboración Propia

- ✓ **Gestionar Proveedores:** Este caso de uso despliega una interfaz llamada Proveedor, que le permitirá a los actores realizar las distintas acciones que este caso contiene, como, Crear, Modificar, Eliminar; el Gestor Proveedor se encarga de procesar las operaciones y almacenarlos en la entidad correspondiente. Este diagrama se muestra en la **Figura 4.7**.

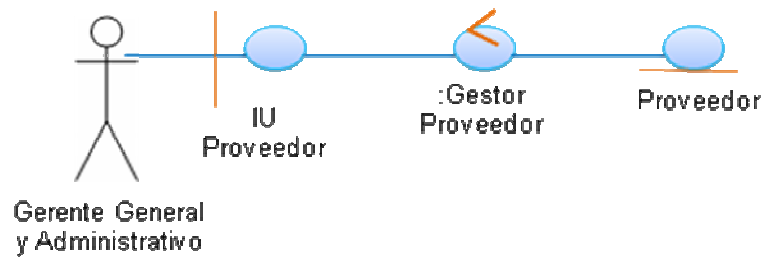


Figura 4.7: Diagrama de Clases de Análisis para el Caso de Uso Gestionar Proveedor

Fuente: Elaboración Propia

- ✓ **Gestionar Cliente:** Al igual que el caso de uso Gestionar Proveedor, este caso de uso desplegará una interfaz con diferentes botones de selección que indicarán las acciones que puede realizar el usuario. El Gestor Cliente se encargará de procesar las operaciones y almacenarlos en la unidad correspondiente. Este diagrama se muestra en la **figura 4.8**.

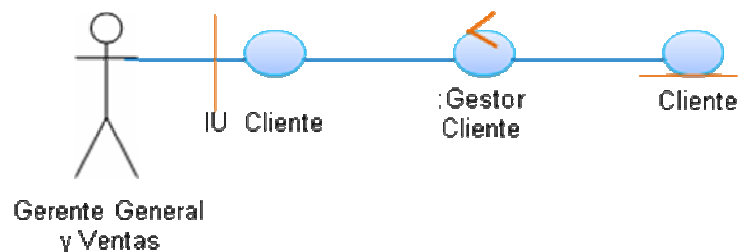


Figura 4.8: Diagrama de Clases de Análisis para el Caso de Uso Gestionar Cliente

Fuente: Elaboración Propia

- ✓ **Gestionar Productos:** Este caso de uso despliega una interfaz llamada Productos, que le permitirá a los actores realizar las distintas acciones que este caso contiene, como, Crear, Modificar, Eliminar; el Gestor Producto se encarga de procesar las operaciones y

almacenarlos en la entidad correspondiente. Este diagrama se muestra en la **Figura 4.9**.

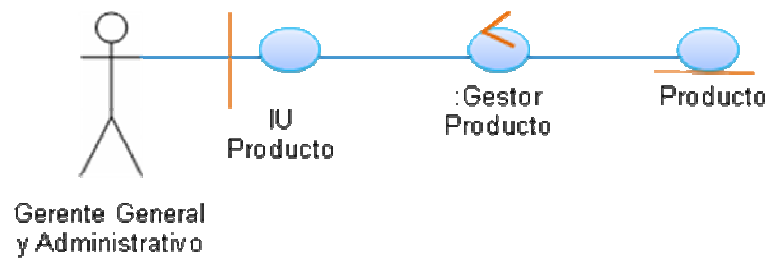


Figura 4.9: Diagrama de Clases de Análisis para el Caso de Uso Gestionar Producto

Fuente: Elaboración Propia

- ✓ **Procesar Venta:** Este caso de uso despliega una interfaz llamada Ventas, que le permitirá a los actores realizar las distintas acciones que este caso contiene, como, Crear o Eliminar. El Gestor Ventas se encargará de procesar la operación y almacenarlos en la unidad correspondiente. Este diagrama se muestra en la **Figura 4.10**.

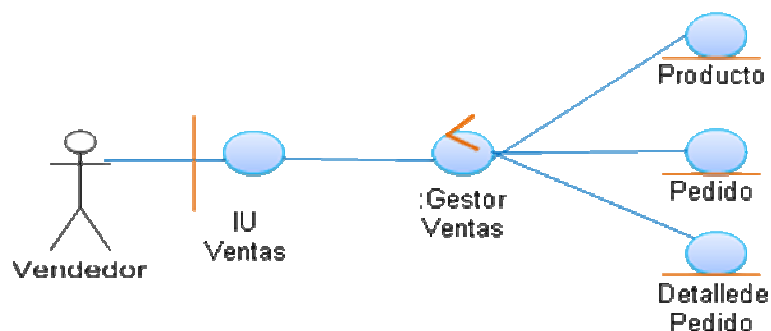


Figura 4.10: Diagrama de Clases de Análisis para el Caso de Uso Procesar Ventas

Fuente: Elaboración Propia

- ✓ **Realizar Consultas:** Esta interfaz muestra diferentes opciones de selección que pueden ser de interés para el usuario, el gestor

consultas se encarga de procesar la operación seleccionada. Es importante mencionar que cada actor tiene acceso solo a las consultas permitidas. El Gerente General puede consultar Usuarios, Productos, Proveedores y Clientes. El Gerente de Ventas y el Vendedor pueden consultar Productos, Proveedores, Clientes y Depósito. El Gerente Administrativo y el Personal Administrativo pueden consultar Productos, Proveedores y Clientes. Este diagrama se muestra en la **Figura 4.11**.

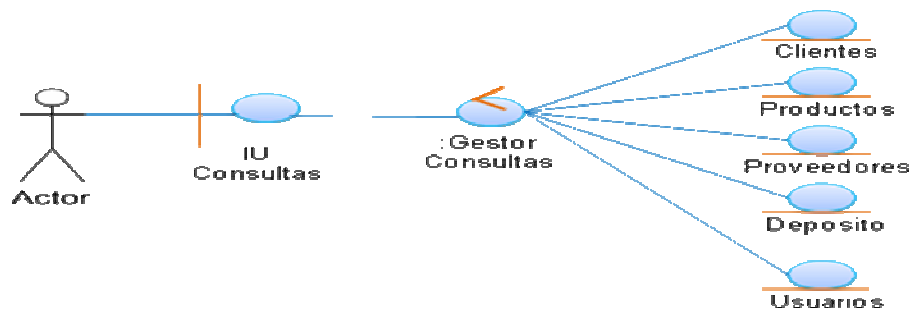


Figura 4.11: Diagrama de Clases de Análisis para el Caso de Uso Realizar Consultas y Reportes

Fuente: Elaboración Propia

4.1.7.3 Diagrama de Colaboración

Un diagrama de colaboración muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran explícitamente las relaciones de los roles. Por otra parte, los diagramas de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

El diagrama de colaboración muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común. Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Esta implementación es llamada "enlace".

❖ Descripción del Diagrama de Colaboración

- ✓ **Gestionar Sesión:** Este caso de uso comienza cuando el actor selecciona la interfaz principal de Iniciar Sesión e ingresa los datos que son solicitados (1) el gestor Usuario se encarga de procesar y validar los datos ingresados para direccionarlos a la página que le corresponde (2), los datos ingresados son almacenados en la entidad respectiva; (3) esta entidad retorna el estado del registro (4), se transfiere a la interfaz (5) y finalmente, el usuario puede acceder al sistema (6). Esta leyenda se muestra en la **Tabla 4.2**.

En la **Figura 4.12** se muestra el Diagrama de Colaboración para el Caso de Uso Gestionar Sesión.

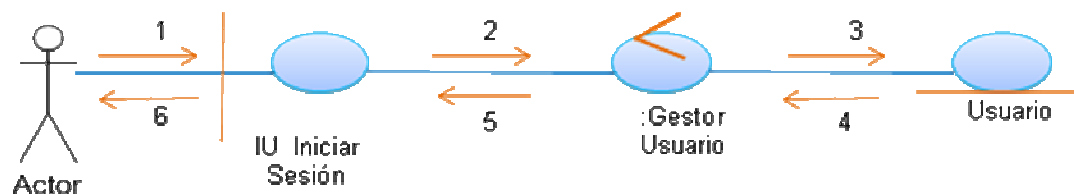


Figura 4.12: Diagrama de Colaboración para el Caso de Uso Gestionar Sesión
Fuente: Elaboración Propia

Tabla 4.2: Leyenda del Diagrama de Colaboración para el Caso de Uso Gestionar Sesión

Caso de Uso	Pasos					
Gestionar Sesión	(1) El Actor selecciona la interfaz Administrar Usuario.	(2) validar y transferir los datos.	(3) almacenar los datos.	(4) devolver al estado de registro.	(5) transferir mensaje de registro.	(6) mostrar resultados de conclusión satisfactoria o de error.

Fuente: Elaboración Propia

- ✓ **Gestionar Proveedor:** Una vez que el usuario haya ingresado al sistema, podrá seleccionar esta interfaz la cual posee las acciones de: crear, modificar o eliminar algunos datos y proveedores (1), dichos datos serán validados y transferidos por el gestor Proveedor (2) y almacenados en la entidad respectiva (3), esta entidad retorna el estado del registro (4), transfiere a la interfaz (5) y muestra los resultados de conclusión satisfactoria o de error (6). Esta leyenda se muestra en la **Tabla 4.3**.

En la **Figura 4.13** se muestra el Diagrama de Colaboración para el Caso de Uso Gestionar Proveedores.

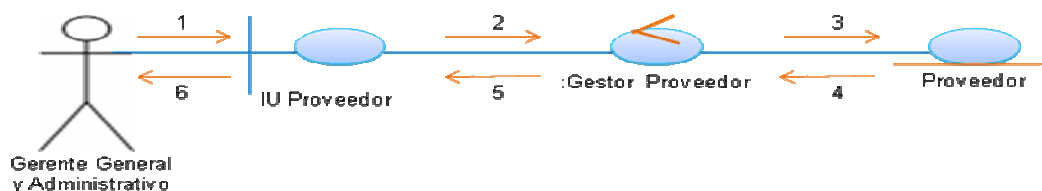


Figura 4.13: Diagrama de Colaboración para el Caso de Uso Gestionar Proveedores

Fuente: Elaboración Propia

Tabla 4.3: Leyenda del Diagrama de Colaboración para el Caso de Uso Gestionar Proveedores

Caso de Uso	Pasos					
Gestionar Proveedor	(1) el actor selecciona la interfaz Proveedor	(2) validar y transferir los datos	(3) almacenar los datos.	(4) devolver al estado de registro.	(5) transferir mensaje de registro.	(6) mostrar resultados de conclusión satisfactoria o de error.

Fuente: Elaboración Propia

- ✓ **Gestionar Cliente:** Al seleccionar esta interfaz el actor podrá crear, modificar o eliminar algunos datos y clientes (1), los datos serán validados y transferidos por el gestor Cliente (2) y almacenados en la entidad respectiva (3), esta entidad retorna el estado del registro (4), transfiere a la interfaz (5) y muestra los resultados de conclusión satisfactoria o de error (6). Esta leyenda se muestra en la **Tabla 4.4**.

En la **Figura 4.14** se muestra el Diagrama de Colaboración para el Caso de Uso Gestionar Clientes.

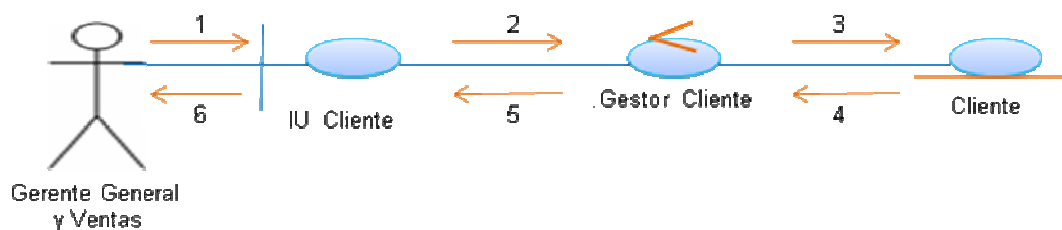


Figura 4.14: Diagrama de Colaboración para el Caso de Uso Gestionar Clientes
Fuente: Elaboración Propia

Tabla 4.4: Leyenda del Diagrama de Colaboración para el Caso de Uso Gestionar Clientes

Caso de Uso	Pasos					
	Gestionar Cliente	(1) el actor selecciona la interfaz Cliente	(2) validar y transferir los datos	(3) almacenar los datos.	(4) devolver al estado de registro.	(5) transferir mensaje de registro.

Fuente: Elaboración Propia

- ✓ **Gestionar Productos:** Al seleccionar esta interfaz el actor podrá crear, modificar o eliminar algunos datos y productos (1), los datos serán validados y transferidos por el gestor Cliente (2) y almacenados en la entidad respectiva (3), esta entidad retorna el estado del registro (4), transfiere a la interfaz (5) y muestra los resultados de conclusión satisfactoria o de error (6). Esta leyenda se muestra en la **Tabla 4.5**.

En la **Figura 4.15** se muestra el Diagrama de Colaboración para el Caso de Uso Gestionar Productos.

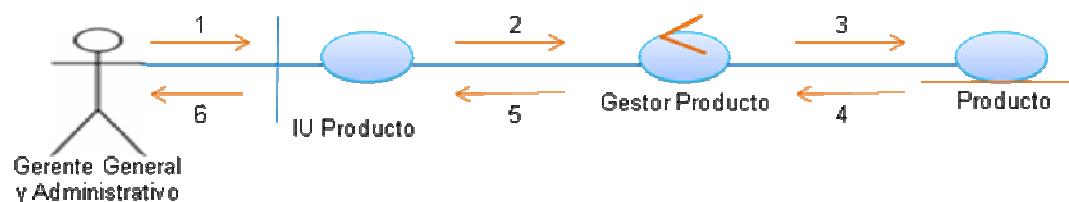


Figura 4.15: Diagrama de Colaboración para el Caso de Uso Gestionar Productos

Fuente: Elaboración Propia

Tabla 4.5: Leyenda del Diagrama de Colaboración para el Caso de Uso Gestionar Productos

Caso de Uso	Pasos					
	Gestionar Productos	(1) el actor selecciona la interfaz Producto	(2) validar y transferir los datos	(3) almacenar los datos.	(4) devolver al estado de registro.	(5) transferir mensaje de registro.

Fuente: Elaboración Propia

- ✓ **Procesar Ventas:** Al seleccionar esta interfaz el actor indica el producto (1), el producto es procesado (2) y buscado en la unidad producto (3), se visualiza la existencia (4) e indica la cantidad deseada (5), se procesa el artículo (6), registra el Ítem (7), luego se procesa el pedido (8) y se procesa el registro de pedido (9), finalmente, se realiza el registro real (10). Esta leyenda se muestra en la **Tabla 4.6**.

En la **Figura 4.16** se muestra el Diagrama de Colaboración para el Caso de Uso Procesar Ventas.

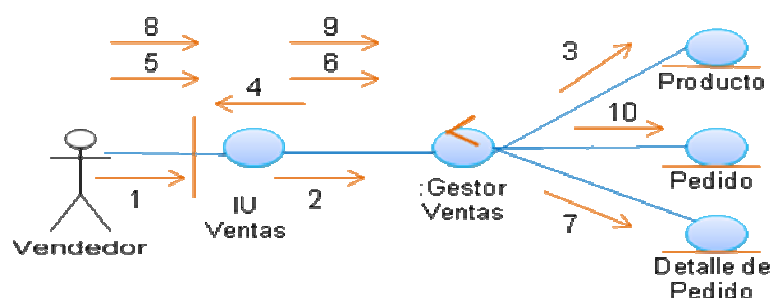


Figura 4.16: Diagrama de Colaboración para el Caso de Uso Procesar Ventas
Fuente: Elaboración Propia

Tabla 4.6: Leyenda del Diagrama de Colaboración para el Caso de Uso Procesar Venta

Caso de Uso	Pasos					
Procesar Venta	(1) Indicar Producto	(2) Procesar Producto	(3) Buscar Productos.	(4) Visualizar existencia.	(5) Indicar Cantidad	(6) Procesar Artículos
	(7) Registrar Ítem	(8) Procesar Pedido	(9) Procesar Registro de Pedido	(10) Registro Real		

Fuente: Elaboración Propia

- ✓ **Realizar Consultas:** En esta interfaz el actor podrá realizar consultas de interés (1), las consultas serán validadas y transferidas por el Gestor Consultas (2) y buscadas en la entidad correspondiente (3), esta entidad retorna el estado del registro (4), transfiere a la interfaz (5) y muestra los resultados (6). Esta leyenda se muestra en la **Tabla 4.7**.

En la **figura 4.17** se muestra el Diagrama de Colaboración para el Caso de Uso Realizar Consultas.

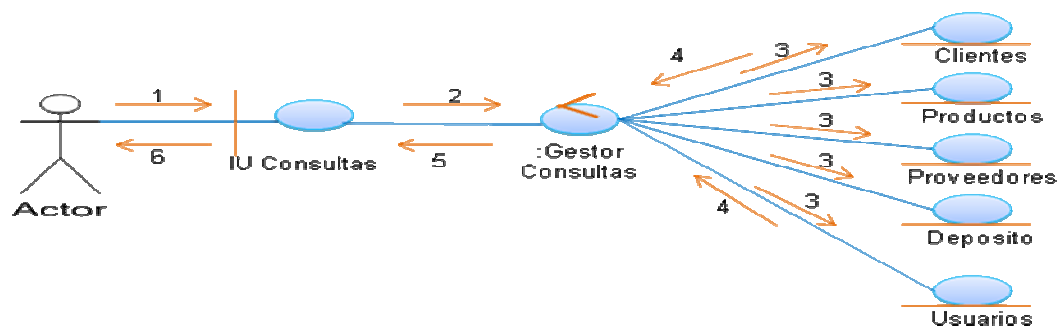


Figura 4.17: Diagrama de Colaboración para el Caso de Uso Realizar Consultas

Fuente: Elaboración Propia

Tabla 4.7: Leyenda del Diagrama de Colaboración para el Caso de Uso Realizar Consultas

Caso de Uso	Pasos					
Realizar Consultas	(1) el actor selecciona la interfaz Consultas	(2) validar y transferir los datos	(3) buscar los datos	(4) devolver al estado de registro.	(5) transferir mensaje de registro.	(6) mostrar resultados o un mensaje de error.

Fuente: Elaboración Propia

4.1.8 Flujo de Trabajo de Diseño

En el flujo de trabajo de diseño se hace un refinamiento del análisis, basándose en los requisitos adicionales, con la finalidad de crear la estructura del sistema.

4.1.8.1 Arquitectura Candidata

Del análisis de los requisitos funcionales del sistema, se definió el software a desarrollar como una aplicación Web, lo que llevó a proponer la arquitectura que se muestra en la **Figura 4.18** en ésta se visualiza la plataforma donde funcionará el sistema.

La arquitectura se planteó en base a la conceptualización de un sistema distribuido. Que requerirá la implementación de una intranet en la empresa. Las Máquinas Cliente comprenden los componentes relacionados con la captura y validación de datos, y estarán ubicadas en las distintas salas de trabajo.

En el Servidor se encuentran todos los componentes que toman decisión sobre el flujo de eventos generado por las solicitudes que los usuarios envían al sistema. Estas solicitudes serán las tareas que el sistema deba realizar de manera automática o no. Tareas como consultas y actualizaciones. Adicionalmente, provee los servicios necesarios para establecer la conexión con la base de datos del sistema y con privilegios a los usuarios que intentan acceder a la información.

4.1.8.2 Interfaz Inicio de Sesión

La interfaz de inicio de sesión, mostrada en la **Figura 4.19** es la primera a la que tiene acceso cualquiera de los usuarios de **“Sellini”**.

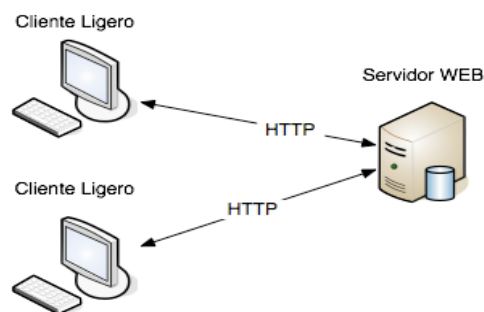
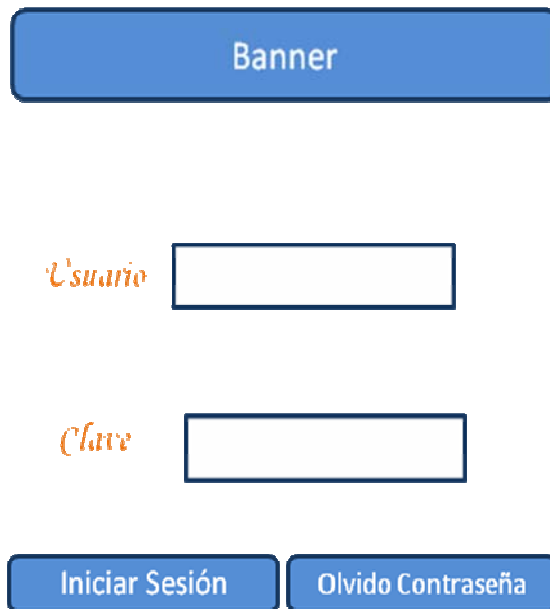


Figura 4.18: Arquitectura Candidata del Sistema

Fuente: Elaboración Propia

Se puede observar un área asignada para el banner que tendrá el logo de la Empresa Distribuidora Sellini, un área para el ingreso de los datos de identificación del usuario, un botón iniciar sesión y otro para el caso de olvidar contraseña.



The image shows a login interface prototype. At the top is a blue rounded rectangle labeled "Banner". Below it are two input fields: the first is labeled "Usuario" and the second is labeled "Clave", both in orange italicized text. At the bottom are two blue rounded buttons: "Iniciar Sesión" and "Olvido Contraseña".

Figura 4.19: Prototipo de Interfaz de Inicio de Sesión
Fuente: Elaboración Propia

4.1.9 Conclusión de la Fase de Inicio

En esta fase se conocieron las necesidades de la empresa y de esta manera se estableció el modelo de casos de uso de este sistema, el cual fue bastante extenso, a pesar de ser la fase de inicio, debido a la complejidad del sistema. Se definieron los actores, se relacionaron con los respectivos casos de uso, y la función que cumple cada uno en el sistema. Posteriormente, trabajamos con el flujo de análisis usando como entrada los objetos obtenidos en la captura de requisitos. Se obtuvo un esbozo del sistema usando clases de análisis y realizaciones de casos de uso, que guiaron al siguiente paso, en el flujo de diseño, a definir la arquitectura del sistema.

4.2 Fase de Elaboración

La Fase de Elaboración construye la línea base de la Arquitectura del Sistema. Los principales objetivos para esta fase son:

- ✓ Desarrollar requisitos pendientes, formulando los requisitos funcionales como casos de uso.
- ✓ Establecer una base de la arquitectura sólida -la línea base de la arquitectura- para guiar el trabajo durante las fases de construcción y transición así como en las posteriores generaciones del sistema.
- ✓ Continuar la observación y control de los riesgos críticos que aún queden e identificar riesgos significativos hasta el punto de que se pueda estimar su impacto en el análisis negocio y en particular en la apuesta económica.
- ✓ Completar los detalles del plan del proyecto.

4.2.1 Flujo de Trabajo de Requisitos

En este Flujo de Trabajo se recolectan los requisitos que no fueron identificados en la fase previa. A continuación se describen los nuevos requisitos identificados:

- ✓ Para disminuir los riesgos de pérdida de información se deberá realizar respaldos de la información contenida en la Base de Datos.
- ✓ En caso de que la Base de Datos presente algún problema se contará con los respaldos realizados previamente para recuperar la información.
- ✓ Si alguno de los datos del usuario está mal registrado éste se puede modificar. Si un usuario se retira de la empresa éste puede ser eliminado del sistema.

4.2.1.1 Diagrama de Caso de Uso General

En esta segunda etapa de captura de requisitos se incorporarán dos (2) nuevos casos de uso al sistema, los cuales son:

- ✓ Administrar Base de Datos.
- ✓ Gestionar Usuario.

❖ Descripción del caso de uso Administrar Base de Datos

Este caso de uso permitirá al Gerente General Crear Respaldos de la Base de Datos y Restaurar la Base de Datos.

- **Crear Respaldo:** La información existente en la Base de Datos será guardada en Unidades de Almacenamiento Externo. De esta manera, el sistema contará con un respaldo actualizado mensualmente, lo cual disminuirá los riesgos de perder la información del sistema Sellini.
- ✓ **Pre-condición:** El actor debe ingresar al menú Administrar Base de Datos en el que se encontrará la opción Crear Respaldo que estará disponible sólo para el actor Gerente General.
- ✓ **Flujo de Eventos:**
 - Flujo Principal:
 - 1 El actor invoca al caso de uso Administrar Base de Datos.
 - 2 El sistema despliega los campos de opciones a seleccionar donde podrá respaldar una o todas las tablas a la vez.
 - 3 El actor selecciona la opción de su preferencia.
 - 4 El sistema almacena las tablas seleccionadas.
 - 5 Finaliza el caso de uso.

- **Flujo Alternativo:**

Paso 2, el usuario selecciona cancelar.

Paso 3, el usuario selecciona cancelar.

Paso 4, el sistema no puede almacenar los datos debido a que la opción seleccionada se encuentra vacía o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

➤ **Restaurar la Base de Datos:** Este caso de uso permite al actor Restaurar la información de la Base de Datos a través de los respaldos que han sido realizados por el Administrador de la Base de Datos.

✓ **Pre-condición:** El actor debe ingresar al menú Administrar Base de Datos en el que se encontrará la opción Restaurar, que estará disponible sólo para el actor Gerente General.

✓ **Flujo de Eventos:**

- **Flujo Principal:**

1. El actor invoca al caso de uso Administrar Base de Datos.
2. El sistema despliega los campos de opciones a seleccionar donde podrá escoger la tabla o las tablas que desee Restaurar.
3. El actor selecciona el campo deseado.
4. El sistema almacena los datos de los campos seleccionados en el Sistema Manejador de Base de Datos (SMBD).
5. Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el usuario selecciona cancelar.

Paso 3, el usuario selecciona cancelar.

Paso 4, el sistema no puede almacenar los datos debido a que no ha seleccionado ninguna opción o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

En la **Figura 4.20** se muestra el Caso de Uso Administrar Base de Datos.

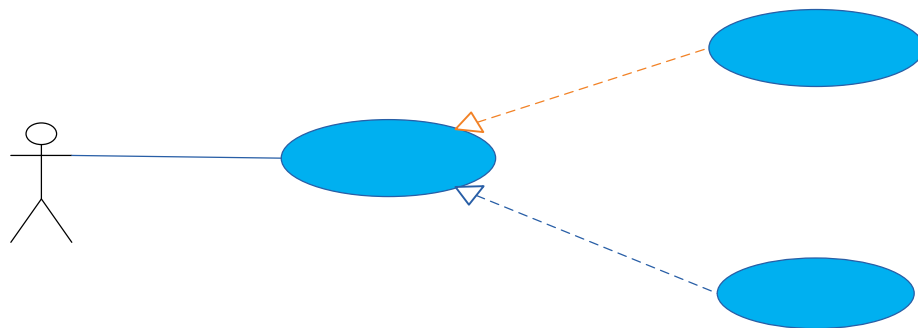


Figura 4.20: Caso de Uso Administrar Base de Datos

Fuente: Elaboración Propia

❖ Descripción del Caso de Uso Gestionar Usuario

Este caso de Uso tendrá las opciones básicas de gestión, las cuales permitirán al Gerente General crear, eliminar o modificar los usuarios que interactúan con el sistema.

- **Crear Usuario:** Permite al Gerente General Crear nuevos usuarios en el sistema.

- ✓ **Pre-condición:** El actor debe ingresar al menú usuario y seleccionar la opción Crear Usuario.

- ✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Usuario.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de crear usuario.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega una interfaz con campos para ser rellenos con los datos del usuario.
- 5 El actor ingresa los datos del usuario.
- 6 El actor solicita guardar los datos.
- 7 El sistema registra los datos.
- 8 Finaliza el caso de uso.

- Flujo Alternativo:

- Paso 2, el actor selecciona cancelar.

- Paso 3, el actor selecciona cancelar.

- Paso 5, el actor selecciona cancelar.

- Paso 6, el actor selecciona cancelar.

- Paso 7, el sistema no puede almacenar los datos debido a que la opción seleccionada se encuentra vacía, ya existe o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- **Modificar Usuario:** Permite al Gerente General Modificar los usuarios existentes en el sistema en el sistema.

✓ **Pre-condición:** El actor debe ingresar al menú usuario y seleccionar la opción Modificar Usuario.

✓ **Flujo de Eventos:**

- Flujo Principal:

- 1 El actor invoca al caso de Gestionar Usuario.
- 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de modificar usuario.
- 3 El actor selecciona el botón deseado.
- 4 El sistema despliega un campo de búsqueda donde se introducirá la cédula del usuario deseado.
- 5 El sistema despliega un formulario con los datos del usuario.
- 6 El actor modifica los datos necesarios del usuario.
- 7 El actor solicita Modificar Usuario.
- 8 El sistema registra los datos.
- 9 Finaliza el caso de uso.

- Flujo Alternativo:

Paso 2, el actor selecciona cancelar.

Paso 3, el actor selecciona cancelar.

Paso 4, el actor selecciona cancelar.

Paso 5, el sistema no puede mostrar los datos debido a que el usuario no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede almacenar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

- **Eliminar Usuario:** Permite al Gerente General eliminar los usuarios existentes en el sistema.

- ✓ **Pre-condición:** El actor debe ingresar al menú Usuarios y seleccionar la opción Eliminar Usuario.

- ✓ **Flujo de Eventos:**
 - Flujo Principal:
 - 1 El actor invoca al caso de Gestionar Usuario.
 - 2 El sistema despliega una interfaz con botones a seleccionar entre los cuales se encuentra el de Eliminar Usuario.
 - 3 El actor selecciona el botón deseado.
 - 4 El sistema despliega un campo de búsqueda donde se introducirá la cédula del usuario deseado.
 - 5 El sistema despliega un formulario con los datos del usuario y un botón con la opción Eliminar Usuario.
 - 6 El actor selecciona la opción de Eliminar Usuario.
 - 7 El sistema elimina el usuario.
 - 8 Finaliza el caso de uso.

 - Flujo Alternativo:
 - Paso 2, el actor selecciona cancelar.
 - Paso 3, el actor selecciona cancelar.
 - Paso 4, el actor selecciona cancelar.
 - Paso 5, el sistema no puede mostrar los datos debido a que el usuario no está registrado o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

Paso 6, el actor selecciona cancelar.

Paso 7, el sistema no puede eliminar los datos debido a que los campos se encuentran vacíos o se presenta un error de conexión. Se muestra un mensaje de error indicando el problema.

En la **Figura 4.21** se muestra el Caso de Uso Gestionar Sesión.

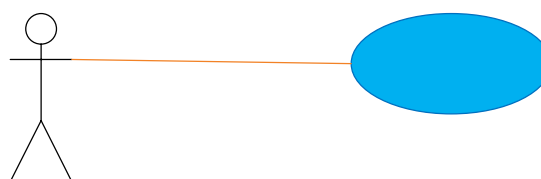


Figura 4.21: Caso de Uso Gestionar Usuario

Fuente: Elaboración Propia

Después de definir los dos nuevos casos de usos encontrados en esta fase se incorporarán al caso de uso General presentado anteriormente en la **Figura 4.3** de la fase de inicio.

En la **Figura 4.22** se muestra en diagrama General de Caso de Uso definitivo del sistema **“Sellini”**.

4.2.2 Flujo de Trabajo de Análisis

En la Fase de Inicio se desarrolló el análisis de la arquitectura sólo hasta el punto de determinar que había una arquitectura del sistema factible. Ahora, en esta fase se extiende el análisis hasta el punto de que pueda servir como base a la línea principal de la arquitectura ejecutable.

Gerente General

En la Fase de Elaboración se presentan los diagramas de clases de análisis y colaboración para el nuevo caso de uso Administrar Base de Datos.

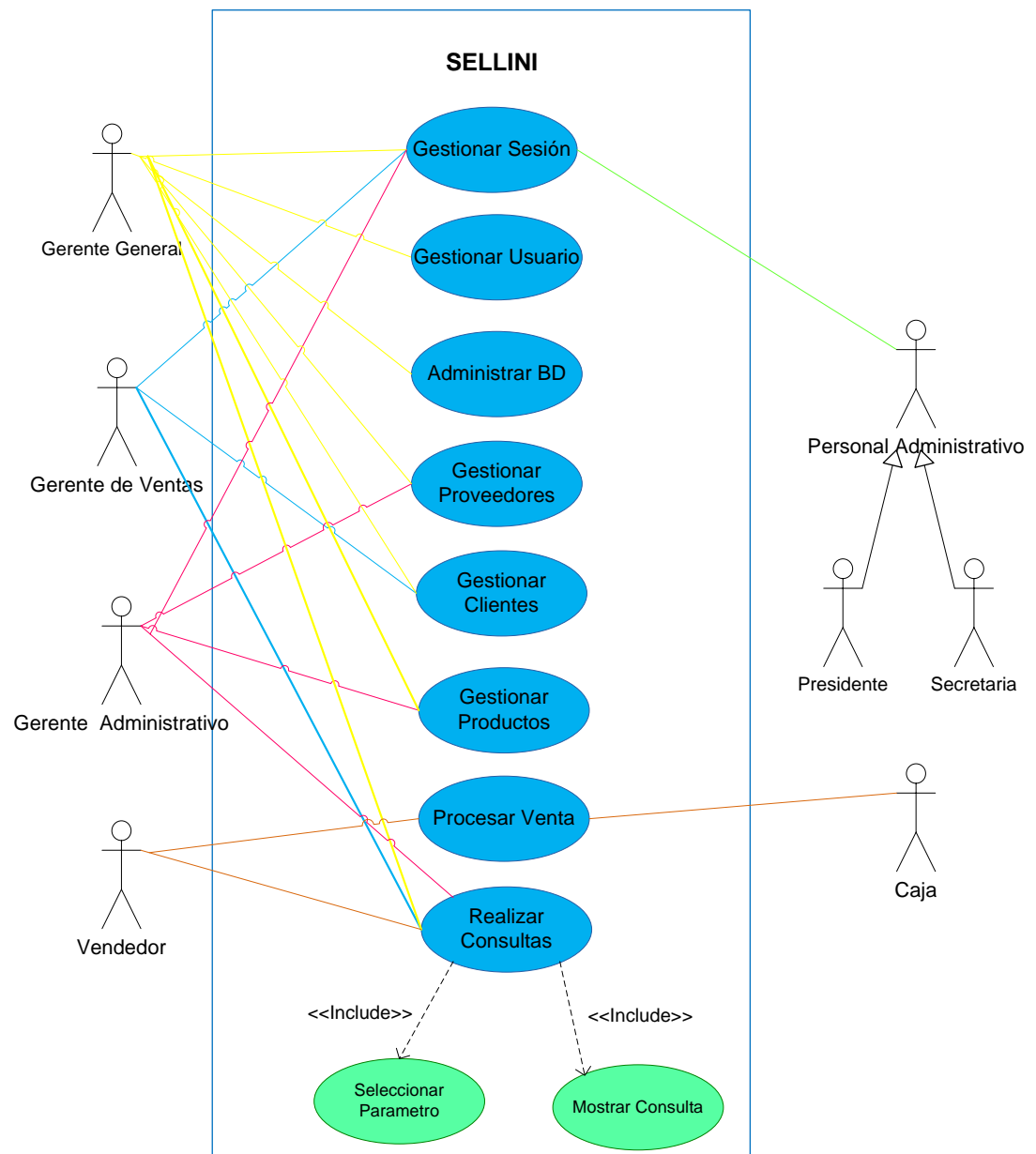


Figura 4.22: Caso de Uso General del Sistema
Fuente: Elaboración Propia

4.2.2.1 Diagrama de Clase de Análisis

Una vez identificado y descrito el caso de uso Administrar Base de Datos, se procede a realizar el Diagrama de Clase de Análisis mediante una abstracción de una o varias clases básicas, las cuales son de interfaz, de control o de entidad.

➤ Descripción de Diagrama de Clase de Análisis para el Caso de Uso Administrar Base de Datos

Este caso de uso despliega una interfaz principal llamada Base de Datos que le permitirá al usuario desplazarse para realizar las distintas acciones que contempla el caso de uso, la Interfaz Respaldo mostrará al usuario varias opciones de selección donde se especifican las tablas que se desean respaldar. La Interfaz Restaurar mostrará al usuario un botón que le permitirá examinar los archivos para seleccionar las tablas que desea restaurar. Se denotan las clases de control, Gestor Administrar Base de Datos que se encarga de manejar de forma general los procesos relacionados con la Administración de la Base de Datos, el Gestor Respaldo se encarga de buscar en la entidad "Información" las tablas de la base de datos que han sido seleccionadas para ser respaldadas, el gestor Restaurar se encarga de guardar en la entidad "Información", las tablas que han sido restauradas.

Este caso de uso es manejado únicamente por el actor Gerente General, quien es designado por la empresa como el Administrador del Sistema.

En la **Figura 4.23** se muestra el Diagrama de Clase de Análisis para el Caso se Uso Administrar Base de Datos.

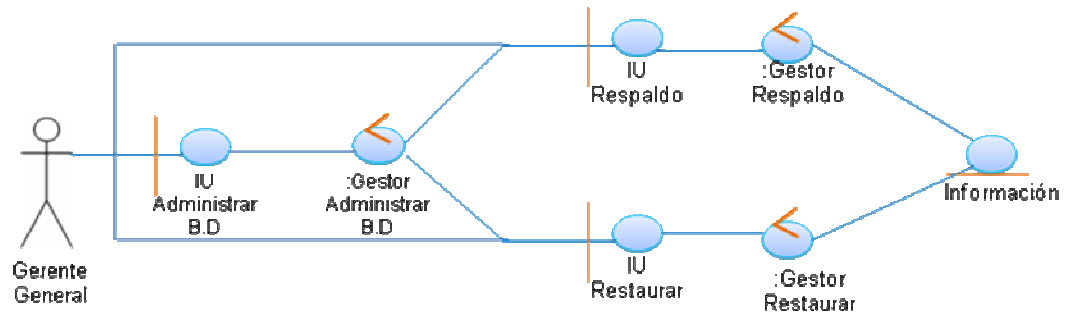


Figura 4.23: Diagrama de Clase de Análisis para el Caso se Uso Administrar Base de Datos

Fuente: Elaboración Propia

➤ **Descripción de Diagrama de Clase de Análisis para el Caso de Uso Gestionar Usuario**

Este caso de uso despliega una interfaz principal llamada Usuario, que le permitirá al actor realizar las distintas acciones que este caso contiene, como: Crear, Modificar, Eliminar. El Gestor Usuarios se encarga de procesar los datos ingresados y almacenarlos en la entidad correspondiente. Este diagrama se muestra en la **Figura 4.24**.

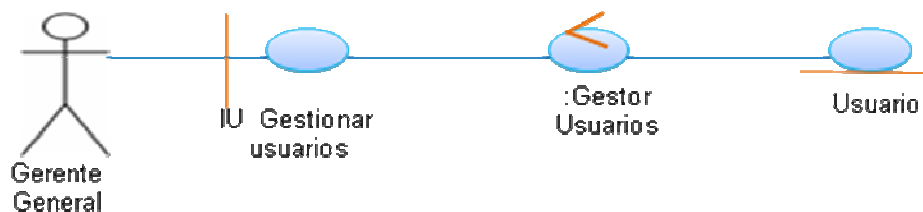


Figura 4.24: Diagrama de Clase de Análisis para el Caso de Uso Gestionar Usuario

Fuente: Elaboración Propia

4.2.2.2 Diagrama de Colaboración

En los diagramas de colaboración, la idea es ver los objetos en extenso, donde las interacciones muestran los flujos y los mensajes entre ellos; estos diagramas tienen como objetivo fundamental identificar los requisitos y responsabilidades sobre los objetos.

➤ **Descripción del Diagrama de Colaboración para el Caso de Uso Administrar Base de Datos.**

Este caso de uso comienza cuando el actor selecciona la interfaz principal Administrar Base de Datos (1) y el gestor Administrar Base de Datos se encarga de procesar su elección (2). En caso de seleccionar la opción Respaldo se activa la interfaz Respaldo (3), donde el actor procederá a seleccionar los datos que desea respaldar (4) que serán validados y transferidos por el Gestor de Respaldo (5) y almacenados en la entidad Información (6); esta entidad retorna el estado del registro (7) y se muestra al usuario un estado de conclusión satisfactoria (8).

En caso de seleccionar la opción Restaurar se activa la interfaz Restaurar (9), donde el actor procederá a seleccionar los datos que desea recuperar (10) que serán validados y transferidos por el gestor de Restaurar (11) y buscados en la entidad Información (12). Esta entidad retorna el estado del registro (13) y se muestra al usuario un estado de conclusión satisfactoria (14). Esta leyenda se muestra en la **Tabla 4.8**.

En la **Figura 4.25** se muestra el Diagrama de Colaboración del Caso de Uso Administrar Base de Datos.

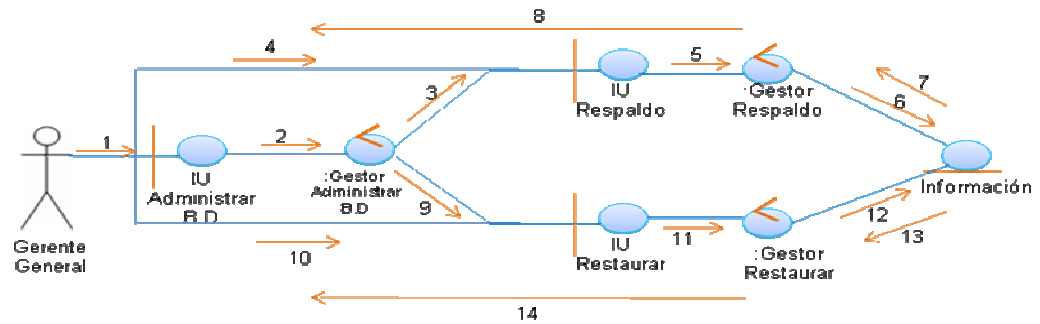


Figura 4.25: Diagrama de Colaboración del Caso de Uso Administrar Base de Datos

Fuente: Elaboración Propia

Tabla 4.8: Leyenda del Diagrama de Colaboración para el Caso de Uso Administrar Base de Datos

CASO DE USO	PASOS					
Administrar Base de Datos	(1) Seleccionar operación.	(2) Procesar selección realizada.	(3) Desplegar interfaz Respaldo.	(4) Seleccionar los datos que desea respaldar.	(5) Validar y transferir datos.	(6) Buscar datos.
	(7) Devolver el estado del registro.	(8) Mostrar mensaje conclusión satisfactoria.	(9) Desplegar interfaz Restaurar.	(10) Seleccionar los datos que desea recuperar.	(11) Validar y transferir datos.	(12) Almacenar datos.
	(13) Devolver el estado del registro.	(14) Mostrar mensaje conclusión satisfactoria.				

Fuente: Elaboración Propia

➤ **Descripción del Diagrama de Colaboración para el Caso de Uso Gestionar Usuario.**

Este caso de uso comienza cuando el actor selecciona la interfaz Gestionar Usuarios (1), el Gestor Usuarios se encarga de validar los cambios realizados (2), los cuales son almacenados en la entidad respectiva (3). Esta entidad retorna el estado del registro (4) y muestra un mensaje de conclusión satisfactoria (5). Esta leyenda se muestra en la **Tabla 4.9**.

En la **Figura 4.26** se muestra el Diagrama de Colaboración para el Caso de Uso Gestionar Usuario.

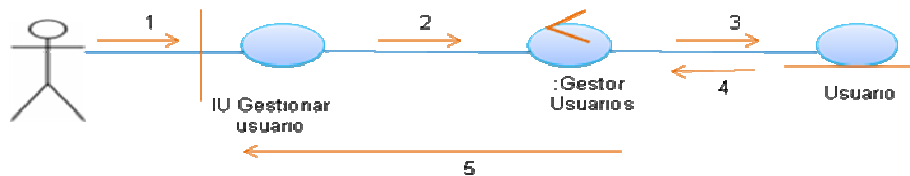
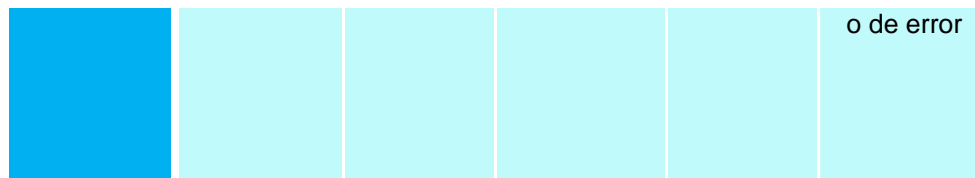


Figura 4.26: Diagrama de Colaboración para el Caso de Uso Gestionar Usuario
Fuente: Elaboración Propia

Tabla 4.9 leyenda del Diagrama de Colaboración para el Caso de Uso Gestionar Usuario

CASO DE USO	PASOS				
Administrar Base de Datos	(1) Seleccionar operación.	(2) Validar y Transferir los datos.	(3) Almacenar los datos	(4) Devolver el estado del registro.	(5)Mostrar resultados de conclusión satisfactoria



Fuente: Elaboración Propia

4.2.3 Flujo de Trabajo de Diseño

En el flujo de trabajo anterior, se describieron de forma clara los requisitos funcionales del sistema, ahora en este flujo se procederá al diseño de todos ellos, tomando en cuenta los requisitos adicionales para así desarrollar la arquitectura del sistema. Se mostrará un diseño de esta arquitectura, el modelado y diseño de la base de datos y el diseño de las páginas que conforman la aplicación.

4.2.3.1 Diseño de la Arquitectura

La arquitectura se divide en tres (3) capas para su mejor comprensión.

- ✓ La Capa General de la Aplicación que contempla los paquetes: Gestión de Sesión, Gestión de Información, y Realización de Consultas, Administración de la Base de Datos, Gestión de Usuario, Procesamiento de Pedidos. Estos paquetes se interconectan entre sí y son enlazados con el paquete Navegador Web en la Capa Intermedia.
- ✓ La Capa Intermedia, que comprende el comportamiento lógico del sistema. En esta capa se cuenta con el Wamp Server, el cual contiene las herramientas: PHP que ejecuta los scripts que contienen la funcionalidad del sistema, MySQL que contiene la base de datos del sistema y la funcionalidad para establecer conexión con ésta y manipularla, y por último, el Servidor Apache para el manejo estático y

dinámico del entorno. En este nivel también se encuentra el Navegador Web, el Lenguaje JavaScript, Lenguaje de Hojas de Estilo en Cascada (CSS) y el Sistema Manejador de Base de Datos (SMBD).

- ✓ La Capa de Software del Sistema, que comprende el Sistema Operativo, Windows 2000/XP/Vista/7, donde se ejecuta la aplicación y el protocolo HTTP. En la **Figura 4.27** se puede observar el Diseño de la Arquitectura del Sistema.

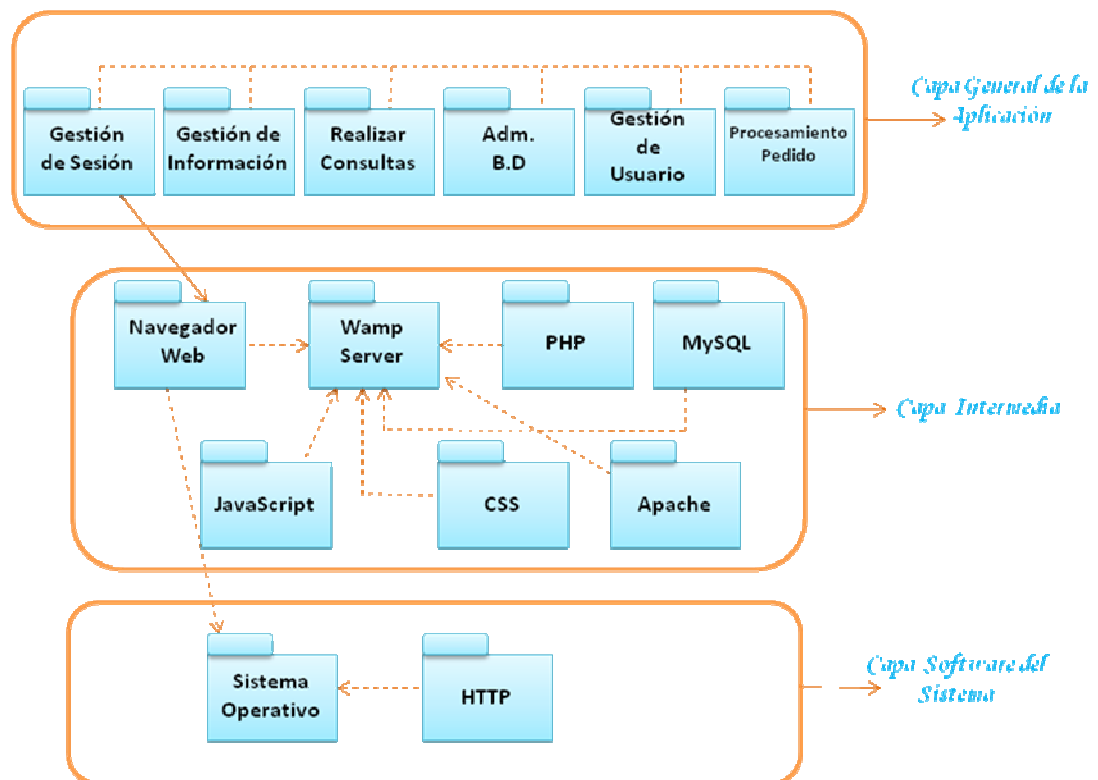


Figura 4.27: Diseño de la Arquitectura del Sistema
Fuente: Elaboración Propia

4.2.3.2 Diagrama de Clase de Diseño

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas donde se crea el diseño conceptual de la información que se manejará en el sistema y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

❖ Diagrama General de la Clase de Diseño

En la **Figura 4.28** se muestra el Diagrama General de la Clase de Diseño, la interfaz principal posee varias operaciones las cuales son: ***usuarios()***, ***productos()***; ***clientes()***, ***proveedores()***, ***consultas()***, y ***baseDeDatos()***, es importante destacar que el acceso a estos métodos depende del nivel que tenga el usuario en el sistema. La Interfaz Principal tiene agregada varias interfaces las cuales son: ***Gestionar Usuarios***, ***Gestionar Productos***, ***Gestionar Clientes***, ***Gestionar Proveedores***, ***Realizar Consultas***, ***Administrar Base de Datos***.

- La interfaz Gestionar Usuario realiza las operaciones de: ***crearUsuario()***, ***modificarUsuario()***, ***eliminarUsuario()***. Esta interfaz tiene una relación con la tabla Usuario, la cual posee entre sus atributos, nombre, apellido, cédula, sexo, fechaNacimiento, teléfono y entre sus funciones posee, ***idUsuario:(Varchar)***, ***nombre:(char)***, ***apellido:(char)***, ***sexo:(char)***, ***fechaNaci:(Varchar)***, ***teléfono:(Varchar)***.
- La interfaz Gestionar Producto realiza las operaciones de: ***crearProducto()***, ***modificarProducto()***, ***eliminarProducto()***. Esta interfaz tiene una relación con la tabla Producto, la cual posee entre sus atributos, código, tipo, marca, descripción, cantidad, stockMaxy entre

sus funciones están, idProducto:(Varchar), tipo:(char), marca:(Varchar), descripción:(Varchar), cantida:(INT), stockMax:(INT).

- La interfaz Gestionar Cliente realiza las operaciones de: crearCliente(), modificarCliente(), eliminarCliente(). Esta interfaz tiene una relación con la tabla Cliente, la cual posee entre sus atributos, rif, nombreEmpresa, nombreContacto, estado, ciudad, teléfono y entre sus funciones posee: idCliente:(Varchar), rif:Varchar, nombreContac:(char), estado:(char), ciudad:(char), telefono:(Varchar).
- La interfaz Gestionar Proveedor realiza las operaciones de: crearProveedor(), modificarProveedor(), eliminarProveedor(). Esta interfaz tiene una relación con la tabla Proveedor, la cual posee entre sus atributos, rif, nombreEmpresa, nombreContacto, estado, ciudad, teléfono y entre sus funciones posee, idCliente:(Varchar), rif:(Varchar), nombreContac:(char), estado:(char), ciudad:(char) , telefono:(Varchar).
- La interfaz Gestionar Proveedor realiza las operaciones de: consultarUsuario(), consultarProducto(), consultarProveedor(), consultarCliente(), consultarStockMin().
- La interfaz Gestionar Proveedor realiza las operaciones de: respaldarBaseDatos(), restaurarBaseDatos().

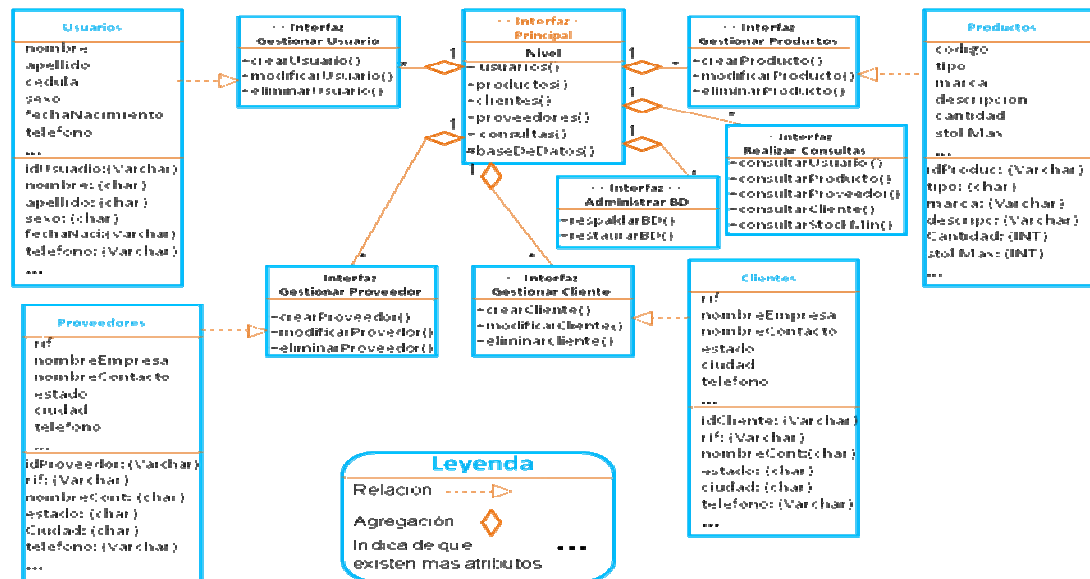


Figura 4.28: Diagrama General de la Clase de Diseño
Fuente: Elaboración Propia

❖ Diagrama de Clase de Diseño para el Vendedor

En la **Figura 4.29** se muestra el Diagrama General de la Clase de Diseño, la interfaz principal posee varias operaciones las cuales son: **usuarios()**, **productos()**; **clientes()**, **proveedores()**, **consultas()**, y **baseDeDatos()**, es importante destacar que el acceso a estos métodos depende del nivel que tenga el usuario en el sistema. La Interfaz Principal tiene agregada varias interfaces las cuales son: **Gestionar Usuarios**, **Gestionar Productos**, **Gestionar Clientes**, **Gestionar Proveedores**, **Realizar Consultas**, **Administrar Base de Datos**.

- La interfaz Gestionar Usuario realiza las operaciones de: crearUsuario(), modificarUsuario(), eliminarUsuario(). Esta interfaz tiene una relación con la tabla Usuario, la cual posee entre sus atributos, nombre, apellido, cédula, sexo, fechaNacimiento, teléfono y

entre sus funciones tiene: idUsuario:(Varchar), nombre:(char), apellido:(char), sexo:(char), fechaNaci:(Varchar), teléfono:(Varchar).

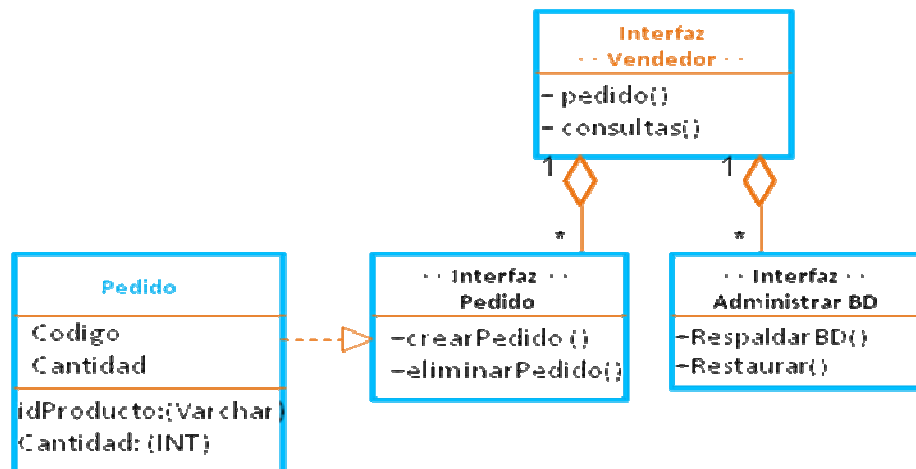


Figura 4.29: Diagrama de la Clase de Diseño, Vista Vendedor
Fuente: Elaboración Propia

4.2.3.3 Diseño de la Base de Datos

La base de datos constituye la fuente central de datos para todo sistema, es por esto que la integridad y fiabilidad de la misma es de vital importancia para la correcta funcionalidad de las aplicaciones asociadas a ésta.

Una base de datos relacional es un conjunto de dos o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común. En ambos casos posee las mismas características como por ejemplo, el nombre del campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave.

➤ Modelo de Datos

El modelo de datos define la estructura de la base de datos para la empresa Distribuidora Sellini.

➤ Estructura de la Base de Datos

Partiendo del modelo de datos y una vez normalizada las entidades, se implementó el diseño de la base de datos del sistema **“Sellini”**, a través del sistema gestor de base de datos MySQL. El conjunto de tablas que conforman la base de datos del sistema, serán mostradas y descritas a continuación.

En la **Figura 4.30** se muestran los atributos de cada entidad así como su relación con otras entidades.

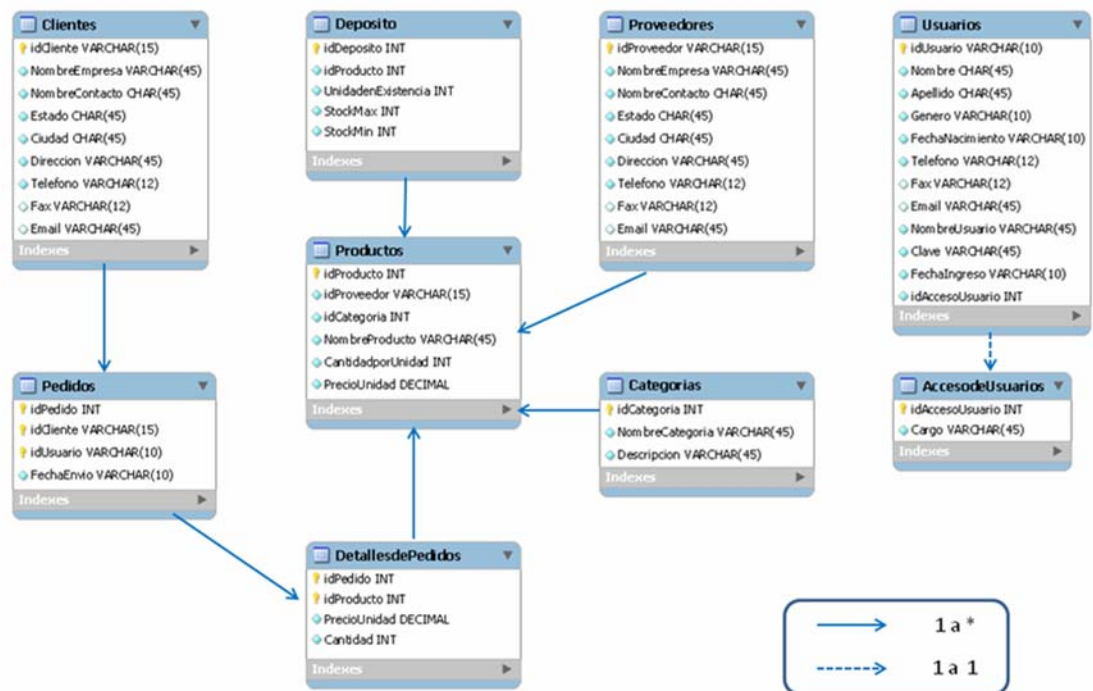


Figura 4.30: Relación de tablas de la base de datos
Fuente: Elaboración Propia

- ✓ **Tabla Acceso de Usuarios:** Contiene el cargo y el tipo de acceso que tendrá el usuario al ingresar al sistema, estos datos se usan para lograr que el usuario entre solo a los módulos asignados. Se muestra en la **Figura 4.31**.

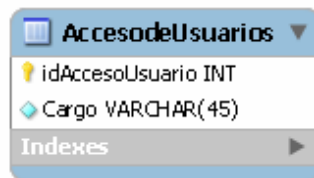


Figura 4.31: Tabla Acceso de Usuarios
Fuente: Elaboración Propia

- ✓ **Tabla Usuarios:** Contiene los datos de todos los usuarios registrados en el sistema los cuales tendrán acceso al sistema **“Sellini”**. Ésta se muestra en la **Figura 4.32**.

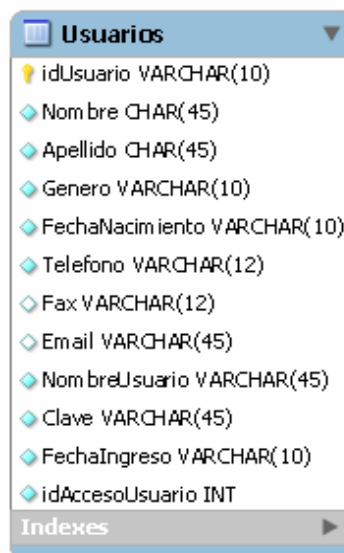


Figura 4.32: Tabla Usuarios
Fuente: Elaboración Propia

- ✓ **Tabla Productos:** Contiene toda la información sobre los productos que la empresa posee. Se muestra en la **Figura 4.33**.

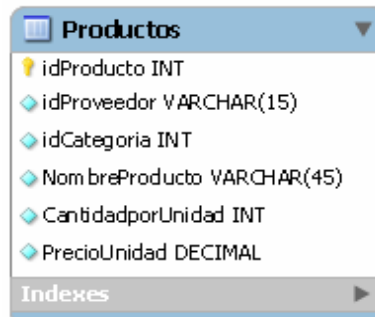


Figura 4.33: Tabla Productos
Fuente: Elaboración Propia

- ✓ **Tabla Proveedores:** Contiene todos los datos que la empresa necesita saber sobre los proveedores. Se muestra en la **Figura 4.34**.

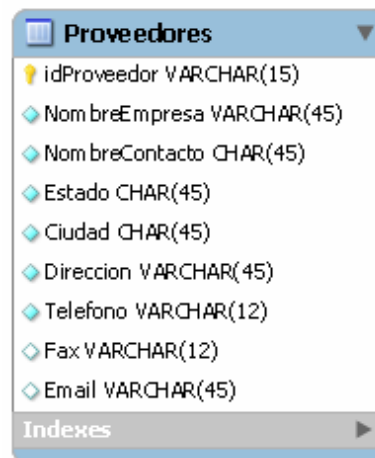


Figura 4.34: Tabla Proveedores
Fuente: Elaboración Propia

- ✓ **Tabla Categorías:** Permite conocer el tipo de filtro y la descripción del mismo. Se muestra en la **Figura 4.35**.



Figura 4.35: Tabla Categorías
Fuente: Elaboración Propia

- ✓ **Tabla Depósito:** Esta tabla contiene información que permitirá a los usuarios del sistema conocer la ubicación de los productos en el depósito, su Stock mínimo y stock máximo. Se muestra en la **Figura 4.36**.



Figura 4.36: Tabla Depósito
Fuente: Elaboración Propia

- ✓ **Tabla Pedidos:** Contiene datos referentes al número del pedido, el cliente que realiza el pedido y el vendedor que lo factura. Se muestra en la **Figura 4.37**.

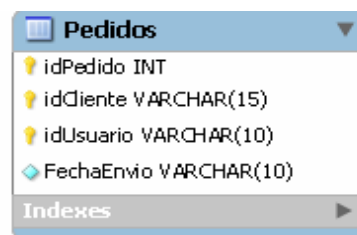
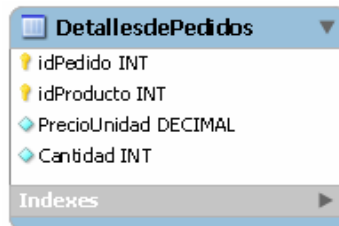


Figura 4.37: Tabla Pedidos
Fuente: Elaboración Propia

- ✓ **Tabla Detalles de Pedidos:** Permite conocer el precio por unidad y la cantidad que el cliente está adquiriendo. Se muestra en la **Figura 4.38**.

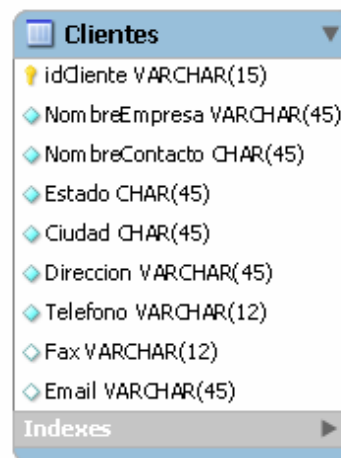


DetallesdePedidos	
idPedido	INT
idProducto	INT
PrecioUnidad	DECIMAL
Cantidad	INT

Indexes

Figura 4.38: Tabla Detalles de Pedidos
Fuente: Elaboración Propia

- ✓ **Tabla Clientes:** Contiene los datos de todos los clientes de la empresa. Se muestra en la **Figura 4.39**.



Clientes	
idCliente	VARCHAR(15)
NombreEmpresa	VARCHAR(45)
NombreContacto	CHAR(45)
Estado	CHAR(45)
Ciudad	CHAR(45)
Direccion	VARCHAR(45)
Telefono	VARCHAR(12)
Fax	VARCHAR(12)
Email	VARCHAR(45)

Indexes

Figura 4.39: Tabla Clientes
Fuente: Elaboración Propia

4.2.4 Implementación

En este flujo, se toma como base los resultados del diseño para implementar el sistema en términos de componentes, dónde se efectúa la primera versión del modelo de implementación, que describe la realización física de los elementos del modelo de diseño.

4.2.4.1 Identificación de Componentes de la Arquitectura

Un diagrama de despliegue muestra sobre cuál componente de hardware se instala (o despliega) cada componente de software. También muestra los enlaces de comunicación entre los componentes de hardware. Este diagrama expone las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre estos nodos.

Para poder implementar la arquitectura de este sistema, se identificaron los componentes mostrados en la **Figura 4.40**, en ella se muestra el servidor Web Apache 2.2.11 que permite la distribución de las páginas, el motor de PHP 5.3 para ejecutar los scripts que contienen la funcionalidad del sistema y el sistema manejador de base de datos (DBMS) MySQL5.1.36 que contiene la base de datos y provee la conexión entre ésta y la administración del manejo de los datos.

Adicionalmente, en la misma figura, se observa el cliente provisto de navegador Web encargado de generar el documento HTML mediante el cual los usuarios podrán visualizar e interactuar con la aplicación. El cliente y el servidor se conectan a través de la Intranet mediante el protocolo de transmisión http.

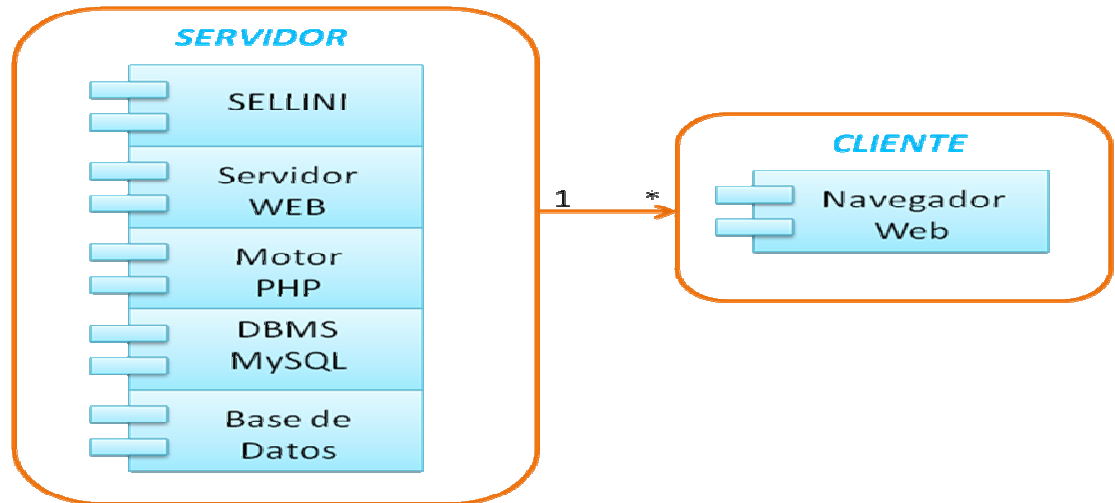


Figura 4.40: Diagrama de Despliegue del Sistema
Fuente: Elaboración Propia

4.2.4.2 Implementación de la Arquitectura

En esta fase se implementó la arquitectura a través de la construcción de un prototipo de la página principal de Inicio de Sesión denominada Index, la cual conforma parte del caso de uso Gestionar Sesión, también se ejecutan los componentes que conforman esta arquitectura.

Esta página fue realizada utilizando lenguaje de scripts PHP, mostrando la interfaz de Inicio de Sesión y conteniendo las instrucciones necesarias para conectar y consultar la Base de Datos. En la **Figura 4.41** se muestra la pantalla de inicio de sesión.

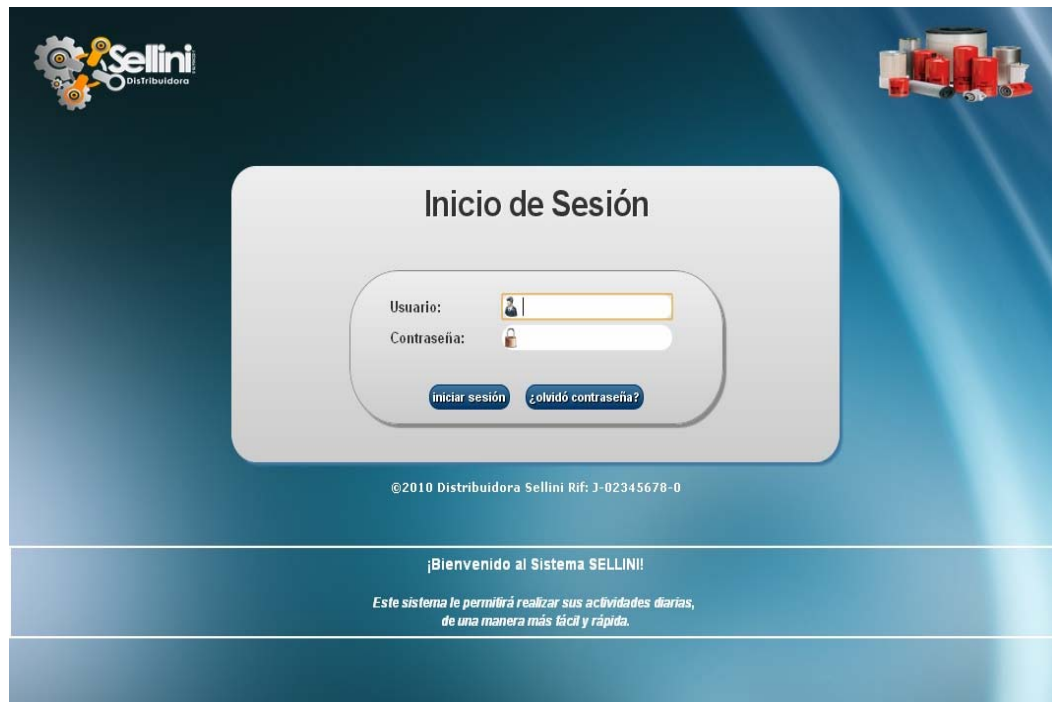


Figura 4.41: Pantalla de Inicio de Sesión
Fuente: Elaboración Propia

4.2.5 Conclusión de la Fase de Elaboración

La Fase de Elaboración mostró la existencia de requisitos que no fueron abordados en la fase anterior y que fueron cubiertos con la creación de dos (2) casos de uso los cuales son: “Administrar Base de Datos” y “Gestionar Usuario”, a los cuales se le aplicaron el flujo de trabajo de análisis igual que en la fase anterior.

Con la finalización de esta fase se obtienen todos los casos de usos que intervendrán en el sistema. Lo que permite describir de la arquitectura, que consistió: en la identificación de los subsistemas, el diseño de clases, el diseño de las tablas pertenecientes a la base de datos y la realización del diagrama de despliegue del sistema, donde se muestra la distribución física

del sistema en términos de funcionalidad, completando así, la línea base de la arquitectura.

4.3 Fase de Construcción

La finalidad de esta fase es alcanzar la capacidad operacional del sistema, para ello los diversos modelos se desarrollan hasta completarse. La arquitectura no debe cambiar significativamente, ya que la mayor parte se definió en la Fase de Elaboración. Los componentes, características y requisitos, deben ser implementados, integrados y probados en su totalidad, obteniendo así, una versión aceptable y de calidad del producto.

La Fase Construcción es donde se dedica mayor tiempo y esfuerzo. Es la línea base de la arquitectura, crece hasta convertirse en el sistema completo.

4.3.1 Flujo de Trabajo: Implementación

En esta etapa se describen las interfaces de Inicio de Sesión y Realizar Pedido, así como también el reporte en PDF del pedido. Para ello se seguirá el siguiente esquema.

- ✓ Breve explicación de la interfaz (en qué consiste).
- ✓ Identificación de los datos a manejar por el área. (datos que debe introducir el usuario durante su visita).
- ✓ Imagen referencial de la pantalla del sistema.
- ✓ Código Fuente.

❖ Inicio de Sesión

Esta Interfaz permitirá a los usuarios ingresar al sistema, al introducir el nombre y clave de usuario el sistema validará los datos y direccionará a cada usuario a la interfaz que le corresponde. En la **Figura 4.42** se muestra la pantalla de Inicio de Sesión.

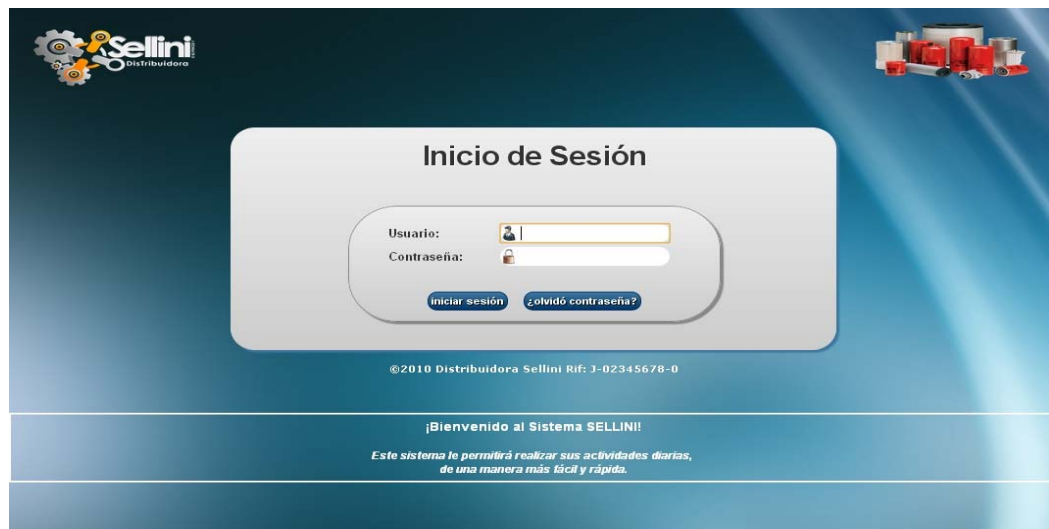


Figura 4.42: Pantalla de Inicio de Sesión
Fuente: Elaboración Propia

✓ CÓDIGO FUENTE

```
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>...: Inicio de Sesi&ocute;n ...</title>

    <link rel='stylesheet' type='text/css' href='css/style.css' />
    <script language="javascript">
      <!-- función limpiar formulario y enfocar en el 1er campo de texto -->
      function limpiar() {
```

```

        document.myform.reset();
        document.myform.user.focus();
    }
    <!-- Función olvido contraseña -->
    function Olvido () {
        alert ("¡Lo sentimos. Debe ponerse en contacto con el Administrador del
Sistema!");
    }
</script>
</head>
<body onLoad="limpiar();">
<div id="logo">
<imgsrc="image/logui_blanco.png"    width="170px"    height="70px"
border="0">
</div>
<div id="filtro">

</div>

<div id="page-wrap" style=" width:600px; clear:both;">
<header>
    <divalign="center">
<h1>Inicio de Sesi&oacute;n</h1>
</div>
</header>
<div id="form2">
<form    name="myform"    id="myform"    action="validar_sesion.php"
method="post">

```



```

</form>
</div>
</div>
<footer>
&copy;2010 Distribuidora Sellini Rif: J-02345678-0
</footer>
<center id="intro">
<h3 style="margin-top:5px">
¡Bienvenido al Sistema SELLINI!
</h3><br />
<p id="intro_texto">Este sistema le permitir&acute; realizar sus
actividades diarias,
<br />de una manera m&acute;sf&acute;cil y r&acute;pida. </p>
</center>
</body>
</html>

```

❖ Realizar Pedido

Esta interfaz permitirá al actor Vendedor elaborar todos los pedidos que realiza la Empresa diariamente de una manera más fácil y rápida, seleccionando en nombre del cliente, el código del producto e ingresando la cantidad deseada por cada producto. Esta interfaz se muestra en la **Figura 4.43**.

Nombre del Vendedor: Hans Palomino

Nombre del Cliente: Rivera Company

Código de Producto: Seleccione Cantidad:

Pedido N°:55 Cliente: Rivera Company

Código	Marca	Tipo	Descripción	Precio	Cantidad	Eliminar
2301	Wix	Filtro Aceite	Toyota	124.00	9	Eliminar
6123	Nke	Filtro Agua	Ford	25.00	5	Eliminar
8946	Fleetguard	2	Toyota	721.00	2	Eliminar

Figura 4.43: Pantalla de Crear Pedido
Fuente: Elaboración Propia

✓ CÓDIGO FUENTE

```
<!DOCTYPE html>
```

```
<?php
```

```
    session_start();
```

```
    require_once("../myconexion.php");
```

```
    if(isset($_COOKIE['usNivel'])==05) {
```

```
    ?>
```

```
<html lang="es">
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
<title>...: Vendedor | Crear Pedido :...</title>
```

```
    <link rel='stylesheet' type='text/css' href='../css/style.css' />
```

```

<script type="text/javascript" src="../js/jquery-latest.js"> </script>
<script type="text/javascript" src="../js/jquery.validate.js"> </script>
<script language="javascript" type="text/javascript" charset="utf-
8">
<!------- Función para el Frame ----->
    function calcHeight()
    {
        //find the height of the internal page
        Var
        the_height=document.getElementById('frame').contentWindow.document.bod
y.scrollHeight;
        //change the height of the iframe
        document.getElementById('frame').height=the_height;
    }
<!------- Función limpiar formulario y enfocar en el 1er campo de texto ----->
    function limpiar()
    {
        document.buscando.reset();
        document.buscando.cliente.focus();
    }
<!-------Función validar submit ----->
    function enviar (){
        document.buscando.submit();
    }
</script>

<!------- Script Botones de Menú ----->

```

```

<script language="javascript" type="text/javascript">
    function MM_preloadImages() { //v3.0
var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
    for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0)
        { d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}
    }
    function MM_swapImgRestore() { //v3.0
var i,x,a=document.MM_sr;
    for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
    }

    function MM_findObj(n, d) { //v4.01
var p,i,x;
    if(!d) d=document;
    if((p=n.indexOf("?"))>0&&parent.frames.length) {
d=parent.frames[n.substring(p+1)].document;
    n=n.substring(0,p);
    }
    if(!(x=d[n])&&d.all)
        x=d.all[n];
    for (i=0;!x&&i<d.forms.length;i++)
        x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++)
        x=MM_findObj(n,d.layers[i].document);
    if(!x && d.getElementById)
        x=d.getElementById(n); return x;
    }
}

```

```

        function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments;
        document.MM_sr=new Array;
for(i=0;i<(a.length-2);i+=3)
    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x;
        if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
    }
</script>
</head>
<body onLoad="limpiar()">
<div id="page-wrap">
<header>
    <div class="banner">
        
    </div>
        <nav>
            <ul class="groupV">
<li><a class="actual" href="Vcrear_pedido.php">Pedidos</a></li>
                <li><a href="Vconsultas.php">Consultas</a></li>
                <li><a href=" ../cerrar_sesion.php">Cerrar
Sesi&oacute;n</a></li>
            </ul>
        </nav>
    </header>
    <div id="menu">
        <ul>
<li>

```

```

    <a href="Vcrear_pedido.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Crear','../boton/Crear-over.png',1)"></a>

```

```

    </li>

```

```

    <li>

```

```

    <a href="Veliminar_pedido.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Eliminar','../boton/Eliminar-
over.png',1)"></a>

```

```

    </li>

```

```

    </ul>

```

```

</div>

```

```

    <section id="main-content3">

```

```

    <div id="guts" style="width:700px;">

```

```

    <div id="formPedido">

```

```

    <form action="formPedido.php" target="frame" method="post"
enctype="application/x-www-form-urlencoded" name="buscando"
id="buscando">

```

```

    <fieldset id="personal">

```

```

        <legend id="legende">B&uacute;squeda</legend>

```

```

    <br>

```

```

        <div id="buscar">

```

```

        <h3> Nombre del Vendedor: <?php echo
'<b>'.$_COOKIE['usNombre']. " ".$_COOKIE['usApellido'].'</b>'; ?> </h3>

```

```

        <br />

```

```

        &nbsp;

```

```

        Nombre del Cliente:

```

```

        <select id="cliente" name="cliente" class="required">

```

```

<option value="0">Seleccione Cliente</option>

<?php
    $sql = mysql_query("SELECT * FROM clientes ORDER BY
NombreEmpresa ASC");
        while( $row = mysql_fetch_array( $sql )) {
?>
        <option value="<?php echo $row['idCliente'];?>"><?php
echo $row['NombreEmpresa'];?></option>
        <?php
        }
        ?>
        </select>
        &nbsp;
        <input type="image" src="../images/superlupa.png"
class="lupa" value="Aceptar" name="aceptar" />
        <!-- Campo oculto que envía el id del vendedor -->
        <input type="hidden" name="idVendedor" value="<?php
echo $_COOKIE['usId']; ?>" />
        </div>
        </fieldset>
    </form>
    <iframe id="frame" name="frame" src="formPedido.php"
width="600px" onLoad="calcHeight();" frameborder="0" scrolling="no" />
    </div>
</div>
</section>
</div>
<footer>

```

©2010 Distribuidora Sellini Rif: J-02345678-0

```
</footer>
</body>
</html>
<?php
}
else {
?>
<script language="javascript" type="text/javascript" charset="utf-8">
alert ("!Lo sentimos, debes iniciar sesi3n para ver este contenido!");
    window.location="../index.php";
</script>
<?php
}
?>
```

❖ Reporte del Pedido

El Vendedor debe imprimir cada reporte y enviarlo a Caja para que el cliente pueda cancelar el pedido. Este reporte se muestra en la **Figura 4.44**.



Distribuidora Sellini

Rif: J-02345678-0

Av. Romulo Gallegos, Edif Costa PB Local 5, Chacao

Telefono: 0212-3416255 Email: Sellini@gmail.com

Pedido N°: 55			
Cliente: Rivera Company Vendedor: Vanessa Cuzzi			
Codigo	Producto	Cantidad	Precio
2301	Wix	9	124.00
6123	Nke	5	25.00
8945	Fleetguard	2	721.00

Figura 4.44: Reporte PDF del Pedido

Fuente: Elaboración Propia

✓ CÓDIGO FUENTE DEL PDF

```
<?php
```

```
if(isset($_POST['imprimir'])){
```

```
    $idPedido=$_POST['idpedido'];
```

```
    $cliente=$_POST['cliente'];
```

```
    $vendedor=$_POST['vendedor'];
```

```
@include '../fpdf/PDF.php';
```

```
@include '../conexion.php';
```

```
$pdf=new PDF();
```

```

$obj=new MySQL();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont('Arial','',15);
$pdf->Cell(10);
$pdf->SetFillColor(0,5,6);
$pdf->Cell(174,1,"",0,1,'C',true);
$pdf->SetDrawColor(2,3,2);
$pdf->Ln();
$pdf->Ln();
$pdf->SetDrawColor(0,0,0);
$pdf->Ln();
$pdf->Ln();
$pdf->Ln();
$pdf->Cell(10,10);
$pdf->SetFont('Arial','B',12);
$pdf->Cell(174, 10, utf8_decode('Pedido N°: '.$idPedido) , 1,0,'C');
$pdf->Ln();
$pdf->Cell(10,10);
$pdf->Cell(174, 10, utf8_decode('Cliente: '.$cliente." ".$vendedor:
".$vendedor) , 1,0,");
$pdf->Ln();
$pdf->Cell(10,10);
$pdf->SetFont('Arial','B',10);
$pdf->Cell(50, 10, utf8_decode("Codigo") , 1,0);
$pdf->Cell(50, 10, utf8_decode("Producto") , 1,0);
$pdf->Cell(24, 10, utf8_decode("Cantidad") , 1,0);
$pdf->Cell(50, 10, utf8_decode("Precio") , 1,0);
$pdf->SetFont('Arial','',10);

```

```

        $lista=mysql_query("select * from detallesdepedidos where
idPedido='$idPedido");
        while($row=mysql_fetch_assoc($lista)){
            $sql2=mysql_query("select * from productos where
idproducto='".$row['idProducto']."'");
            $con=mysql_fetch_assoc($sql2);
            $pdf->Ln();
            $pdf->Cell(10,10);
            $pdf->Cell(50, 10, utf8_decode($row['idProducto']) , 1,0);
            $pdf->Cell(50, 10, utf8_decode($con['NombreProducto']) , 1,0);
            $pdf->Cell(24, 10, utf8_decode($row['Cantidad']) , 1,0);
            $pdf->Cell(50, 10, utf8_decode($row['PrecioUnidad']) , 1,0);
        }
        $pdf->Output("Pedido_Sellini_$idPedido.pdf","D");
    }
    if(isset($_POST['registrar'])){
?>

```

```
<html lang="es">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>...: Vendedor | Crear Pedido ::.</title>
```

```
<link rel='stylesheet' type='text/css' href='../css/style.css' />
```

```
<script language="javascript" type="text/javascript" charset="utf-8">
```

```
<!------- Función para el Frame ----->
```

```
function calcHeight()
```

```
{
```

```
//find the height of the internal page
```

var

```
the_height=document.getElementById('frame').contentWindow.document.body.scrollHeight;
```

```
//change the height of the iframe
```

```
document.getElementById('frame').height=the_height;
}
```

```
<!-- Función limpiar formulario y enfocar en el 1er campo de texto ---->
```

```
function limpiar()
{
document.buscando.reset();
document.buscando.cliente.focus();
}
```

```
<!-------Función validar submit ----->
```

```
function enviar (){
document.buscando.submit();
}
```

```
</script>
```

```
<!------- Script Botones de Menú ----->
```

```
<script language="javascript" type="text/javascript">
```

```
function MM_preloadImages() { //v3.0
var d=document;
```

```
if(d.images)
```

```
{
```

```
if(!d.MM_p)
```

```
d.MM_p=new Array();
```

```
var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
```

```
for(i=0; i<a.length; i++)
```

```
if (a[i].indexOf("#")!=0)
```

```
{ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
```

```

    }
    function MM_swapImgRestore() { //v3.0
var i,x,a=document.MM_sr;
for(i=0;a&& i<a.length&&(x=a[i])&&x.oSrc;i++)
    x.src=x.oSrc;
    }
    function MM_findObj(n, d) { //v4.01
var p,i,x; if(!d) d=document;
        if((p=n.indexOf("?"))>0&&parent.frames.length) {
d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
        if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++)
            x=d.forms[i][n];
for(i=0;!x&&d.layers&&i<d.layers.length;i++)
            x=MM_findObj(n,d.layers[i].document);
        if(!x && d.getElementById) x=d.getElementById(n); return x;
    }
    function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array;
for(i=0;i<(a.length-2);i+=3)
        if ((x=MM_findObj(a[i]))!=null)
            {document.MM_sr[j++]=x;
            if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
        }
</script>
</head>
<body>
    <div id="page-wrap">
    <header>

```

```

        <div class="banner">
            
        </div>
        <nav>
            <ul class="groupV">
                <li><a href="Vcrear_pedido.php">Pedidos</a></li>
                <li><a href="Vconsultas.php">Consultas</a></li>
                <li><a href="Vcerrar_sesion.php">Cerrar
                Sesi&oacute;n</a></li>
            </ul>
        </nav>
    </header>
    <div id="menu">
        <ul>
            <li>
                <a href="Vcrear_pedido.php" onMouseOut="MM_swapImgRestore()"
                onMouseOver="MM_swapImage('Crear','../boton/Crear-over.png',1)"></a>
            </li>
            <li>
                <a href="Veliminar_pedido.php" onMouseOut="MM_swapImgRestore()"
                onMouseOver="MM_swapImage('Eliminar','../boton/Eliminar-
                over.png',1)"></a>
            </li>
        </ul>
    </div>

```

```

<section id="main-content3">
<div id="guts" style="width:700px; height:500px;">
<div id="formPedido">
    <br> <br>
    <br> <br>
    <br> <br>
    <br> <br>
    <br> <br>
    <div align="center">
        <h2> El Pedido N° <?php echo $_POST['idpedido']; ?>
se ha registrado Exitosamente </h2>
        <a href="Vcrear_pedido.php">  </a>
    </div>
</div>
</div>
</div>
</section>
</div>
<footer>
    &copy;2010 Distribuidora Sellini Rif: J-02345678-0
</footer>
</body>
</html>
<?php
}
else {
?>
<script language="javascript" type="text/javascript" charset="utf-8">

```

```
alert ("Lo sentimos, debes iniciar sesión para ver este contenido");
```

```
window.location="../index.php";
```

```
</script>
```

```
<?php
```

```
}
```

```
?>
```

4.3.2 Flujo de Trabajo: Pruebas

El flujo de trabajo Pruebas, es una de las fases más importantes del ciclo de vida de desarrollo de software. Un producto de software que se desarrolla se debe entregar al cliente libre de defectos o de errores. La prueba es el proceso de ejecutar la aplicación con la intención de descubrir defectos en el programa.

En el ciclo de vida de desarrollo de la aplicación, la fase de prueba ocurre después de la fase de programación. En tal sentido se plantea la revisión del funcionamiento de los distintos módulos de la aplicación, sometiendo cada uno a una entrada, proceso y salida esperada, a fin de verificar la confiabilidad, seguridad e integridad del sistema. Las pruebas garantizan que el producto es de calidad.

4.3.2.1 Pruebas de Integración

Para la integración de los módulos del sistema, las pruebas se planifican utilizando la integración incremental (antítesis del enfoque del “big bang”). El programa se construye y prueba en pequeños incrementos, en los

cuales resulta fácil aislar y corregir errores. Se planea utilizar, como estrategia de integración, la integración incremental ascendente.

La prueba de integración ascendente, como su nombre lo indica, empieza la construcción y la prueba con módulos atómicos (es decir, componentes de los más bajos de la estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, siempre está disponible el procesamiento requerido para los componentes subordinados a un determinado nivel y se elimina la necesidad de resguardos.

4.3.2.2 Pruebas de Caja Negra

Las pruebas de caja negra se concentran en los requisitos funcionales del software; es decir, permiten derivar un conjunto de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. La prueba de caja negra es un enfoque complementario al enfoque de caja blanca, debido a que provee la posibilidad de descubrir una clase diferente de errores.

Las pruebas de caja negra tratan de encontrar errores derivados por funciones incorrectas o faltantes, errores de interfaz, errores en estructura de datos, errores de comportamiento, errores de inicialización y término.

Para realizar las pruebas de caja negra existe una técnica algebraica llamada "clases de equivalencia", que consiste en tratar a todas las posibles entradas y parámetros como un modelo algebraico y utilizar las clases de este modelo para probar un amplio rango de posibilidades.

A continuación se muestran las clases de equivalencias para las operaciones más importantes de **“Sellini”**.

En la **Tabla 4.10** se muestra la Clase de Equivalencia para la Operación Crear Cliente, cabe destacar que los campos de la Operación Crear Proveedor y son los mismos que posee la Operación Crear Cliente, por lo que no se describirán en esta prueba.

Tabla 4.10: Clase de Equivalencia para la Operación: Crear Cliente

NÚMERO	CAMPOS	CASO DE PRUEBA	SALIDA
1	RIF	Nulo	Ingresar Rif del Cliente
2	Nombre de la Empresa	Nulo	Ingresas nombre del cliente
3	Nombre del Contacto	Nulo	Ingresar nombre del contacto
4	Estado	Nulo	Ingresar el estado donde se ubica el cliente
5	Ciudad	Nulo	Ingresar la ciudad donde se ubica el cliente
6	Dirección	Nulo	Ingresar la dirección del cliente
7	Teléfono	Nulo	Ingresar el teléfono del cliente
8	Fax	Nulo	Ingresar el fax del cliente
9	Email	Nulo	Ingresar el Email del cliente

Fuente: Elaboración Propia

En la **Tabla 4.11** se muestra la Clase de Equivalencia para la Operación Modificar Cliente, cabe destacar que los campos de la Operación Modificar Proveedor y son los mismos que posee la Operación Modificar Cliente, por lo que no se describirán en esta prueba.

Tabla 4.11: Clase de Equivalencia para la Operación: Modificar Cliente

NÚMERO	CAMPOS	CASO DE PRUEBA	SALIDA
1	Nombre de la empresa	Seleccionar	Seleccionar el nombre de la empresa
2	RIF	Nulo	Ingresar Rif del Cliente
3	Nombre de la Empresa	Nulo	Ingresas nombre del cliente
4	Nombre del Contacto	Nulo	Ingresar nombre del contacto
5	Estado	Nulo	Ingresar el estado donde se ubica el cliente
6	Ciudad	Nulo	Ingresar la ciudad donde se ubica el cliente
7	Dirección	Nulo	Ingresar la dirección del cliente
8	Teléfono	Nulo	Ingresar el teléfono del cliente
9	Fax	Nulo	Ingresar el fax del cliente
10	Email	Nulo	Ingresar el Email del cliente

Fuente: Elaboración Propia

En la **Tabla 4.12** se muestra la Clase de Equivalencia para la Operación Crear Producto, cabe destacar que los campos de la Operación Modificar

Producto, son los mismos que posee la Operación Crear Cliente, por lo que no se describirán en esta prueba.

Tabla 4.12: Clase de Equivalencia para la Operación: Crear Producto

NÚMERO	CAMPOS	CASO DE PRUEBA	SALIDA
1	Código	Nulo	Ingresar Rif del Cliente
2	Proveedor	Seleccionar	Selecciona el proveedor
3	Nombre del Producto	Nulo	Ingresar nombre del contacto
4	Categoría del Filtro	Nulo	Ingresar el estado donde se ubica el cliente
5	Tipo de Filtro	Seleccionar	Seleccione el tipo de filtro
6	Marca del Filtro	Seleccionar	Seleccione la marca del filtro
7	Cantidad por Unidad	Nulo	Indique la cantidad que se está adquiriendo
8	Precio	Nulo	Indique el precio de venta del producto
9	Piso	Seleccionar	Seleccione el piso donde se ubicara el producto
10	Stock Máximo	Nulo	Ingresar la cantidad máxima de existencia que se puede tener del producto
11	Stock Mínimo	Nulo	Ingresar la cantidad mínima de existencia que se puede tener del producto

Fuente: Elaboración Propia

En la **Tabla 4.13** se muestra la Clase de Equivalencia para la Operación Crear Pedido.

Tabla 4.13 se muestra la Clase de Equivalencia para la Operación Crear Pedido

NÚMERO	CAMPOS	CASO DE PRUEBA	SALIDA
1	Nombre del Cliente	seleccionar	Seleccionar un cliente
2	Código	seleccionar	Seleccionar un código
3	Cantidad	Nulo	Ingresar la cantidad deseada

Fuente: Elaboración Propia

4.3.3 Conclusión de la Fase de Construcción

La implementación de los componentes del sistema, se llevó a cabo por medio de la construcción de un código que tuvo como base fundamental los diagramas logrados en los dos capítulos anteriores, lográndose así, la versión beta del sistema **“Sellini”**. Para obtener un producto con la calidad y confiabilidad necesaria, le fueron aplicadas distintas pruebas al sistema demostrando así que el producto es estable para ser usado.

4.4 Fase de Transición

El objetivo de esta fase es asegurar que los requisitos se han cumplido y que el software está disponible para los usuarios finales. Por eso, esta fase está dirigida por la retroalimentación de los usuarios a partir de la información que se deduzca de la versión beta del sistema en funcionamiento. Para lograrlo, se siguieron los siguientes pasos:

- ✓ Se corrigen los fallos que hubiera, para lo cual se realizan las pruebas.

- ✓ Se determinan los elementos que deban ajustarse como: problemas no detectados, refinamiento y mejora de algunas características con un desarrollo adicional.
- ✓ Se actualiza la documentación correspondiente.
- ✓ Se deben descubrir riesgos no detectados anteriormente.

Los ajustes que se hagan serán específicos y de corto alcance. Ajustes estructurales mayores debieron haberse resuelto anteriormente en el ciclo de vida y deberán documentarse para futuras ampliaciones.

Estando en marcha la versión beta del sistema, se reemplaza por el sistema definitivo que se despliega entre los usuarios.

La carga de trabajo de esta fase las soportan los programadores (que hacen los cambios necesarios) y el equipo de control de calidad (que prueba los cambios).

Al final de la fase, se decide si se han cumplido los objetivos previstos y se puede determinar si es necesario empezar otro ciclo de desarrollo. Por otra parte, se puede llevar registro de los avances, para aplicarlas en próximas actualizaciones del sistema.

4.4.1 Evaluación de la Fase de Transición

El software cumple con los requisitos obtenidos en las fases de inicio y elaboración, obteniéndose así el funcionamiento adecuado con respecto a lo requerido por los usuarios que realizan las operaciones en el sistema.

4.4.2 Flujo de Trabajo Pruebas

Este flujo de trabajo tiene la finalidad de comprobar el correcto funcionamiento de cada uno de los módulos que se codificaron e integraron para obtener una aplicación robusta y confiable.

Se aplicaron pruebas a las operaciones más relevantes para determinar si la aplicación está trabajando correctamente y cumpliendo a cabalidad los requisitos capturados en las fases previas.

4.4.2.1 Aplicación de las Pruebas

Se hicieron un conjunto de pruebas con el personal seleccionado de la Empresa Distribuidora Sellini con la versión instalada en los servidores de desarrollo, estas pruebas se llevaron a cabo tomando como base los mismos casos de pruebas realizados en la fase de construcción.

4.4.3 Conclusión de la Fase de Transición

En la Fase de Transición no fue necesario realizar modificaciones o correcciones importantes sobre los componentes del software, demostrando así, la robustez y estabilidad de la línea base de la arquitectura, alcanzadas en fases anteriores.

Finalmente, se demostró que el sistema ha cubierto las expectativas y los requerimientos de la Empresa Distribuidora Sellini, lo que quiere decir, que se han alcanzado los objetivos fijados en la fase de inicio y que los usuarios principales están satisfechos.

CONCLUSIÓN

Una vez cumplidas en su totalidad con todas las fases del Proceso Unificado de Desarrollo de Software, utilizado para el desarrollo del sistema de información, denominado **“Sellini”**, dirigido a la Empresa Distribuidora Sellini, con la finalidad de automatizar los distintos procesos relacionados con la compra y venta de mercancía, entre los estos procesos se encuentran, la gestión de usuarios, productos, clientes, proveedores, registro de pedidos y realización de consultas.

Como punto de partida, fue necesaria la comprensión del estado actual del sistema de distribución de información de la Empresa, lo que permitió determinar, todos los procesos necesarios y sus actividades relacionadas, para la obtención de los requerimientos del nuevo sistema y de los distintos módulos que éste posee.

Cuando la información que se maneja en un sistema requiere un alto grado de confidencialidad y a su vez, tiene un tráfico constante de usuarios, es necesario implementar medidas de seguridad que limiten a los usuarios a tener acceso sólo a la información que les corresponda. El sistema **“Sellini”**, aplica estas medidas de seguridad por medio de la diferenciación de tipos de usuarios, dándoles distintos niveles de acceso a la información.

Es importante señalar que para la realización de un software robusto, flexible y escalable, es necesaria la utilización de una buena metodología y por esto se utilizó el Proceso Unificado de Desarrollo de Software (RUP), y

su herramienta UML, debido a que ofrece un modo estándar de visualizar, especificar, construir, documentar y comunicar los módulos de un sistema.

Para guardar todos los registros, cambios de estado y rutas de información, "**Sellini**" cuenta con una Base de Datos Relacional, que garantiza su almacenamiento en el tiempo que sea necesario.

Las interfaces se diseñaron con gran énfasis para que posean diversas utilidades de valor, como: fácil acceso, terminologías comunes que cualquier usuario puede conocer e identificar de manera inequívoca, haciéndolas intuitivas y de aprendizaje rápido.

El uso del lenguaje de programación PHP y del manejador de bases de datos MySQL, como piezas clave en la construcción del sistema, facilitó el desarrollo del sistema de forma precisa y eficaz, debido a que estas herramientas poseen requerimientos de funcionamiento mínimo y permiten crear aplicaciones complejas con una curva de aprendizaje corta.

Por medio de la aplicación de pruebas como las de caja negra, se pudo corroborar la efectividad del sistema, obteniéndose un funcionamiento eficiente de cada uno de sus módulos.

El diseño modular que presenta el sistema "**Sellini**", permite futuras y diversas ampliaciones en su funcionalidad.

RECOMENDACIONES

1. Es recomendable que el servidor a utilizar posea una cantidad considerable de espacio en disco (mínimo 100GB), para el hospedaje del sistema y almacenamiento de los archivos que se manejan, aunque estos no son de gran tamaño, su permanencia será prolongada, acumulándose en grandes cantidades.
2. Para garantizar la disponibilidad de la información, se sugiere que los archivos y registros del sistema permanezcan en el disco por un tiempo mínimo de un (1) año y un tiempo máximo de tres (3) años.
3. Desarrollar futuras actualizaciones al sistema, que permitan la automatización de otros procesos realizados por la empresa, se recomienda incorporar un sistema de facturación.
4. La implementación de este sistema sería de gran beneficio para la Empresa Distribuidora Sellini, en virtud de que este permite la automatización de distintos procesos relacionados con la compra y venta de mercancía, entre los cuales se encuentran, la gestión de usuarios, productos, clientes, proveedores, registro de pedidos y realización de consultas.

BIBLIOGRAFÍA

Alvares, R. y Márquez, Y. (2004). Diseño de un Plan Estratégico parte una Empresa de Venta y Servicios de Equipos de Computación en la Zona de Lechería. Trabajo de Grado no publicado, Universidad de Oriente, Barcelona.

Arias, F. (2006). El proyecto de investigación (5ta ed.). Caracas: Episteme.

Borjas, A. (2007). Diseño de un sistema de información para el registro de entrada y salida de vehículos de transporte del instituto de deportes del estado Anzoátegui. Trabajo de Grado no publicado, Universidad de Oriente, Barcelona.

Cohen, D. y Asín, E. (2000). Sistemas de Información para los Negocios (3ra ed.). Ciudad de México: Mc Graw Hill.

Elmasri, R y Navathe, S. (2000). Sistemas De Bases De Datos. Conceptos Fundamentales (2da ed.). Ciudad de México: Addison Wesley Longman,

Fowler, M. y Scott, K. (1997). UML Gota a Gota. Ciudad de México: Pearson Addison Wesley.

Joyanes, L. (1996), Programación Orientada a Objetos. Madrid: Mc Graw Hill.

- Larman, C. (1999). UML Y Patrones, Introducción Al Análisis Y Diseño Orientado A Objetos. Ciudad de México: Prentice Hall.
- Martín, J. y Odell, J. (1994). Análisis Y Diseño Orientado A Objeto. Madrid: Prentice Hall.
- Noriega, M., (1993), Diccionario Enciclopédico Universal (10ma ed.). Madrid: Cultural.
- Olivier, N. y El Nemer, R. (2007). Desarrollo de un software para la automatización del flujo de información asociado a los clientes y proveedores de un local comercial. Trabajo de Grado no publicado, Universidad de Oriente, Barcelona.
- Ortiz, R. (2009). Diseño de un sistema de información que permita la evaluación, cuantificación e identificación de las fallas de las bombas ubicadas en la superintendencia de movimientos de crudos de la refinería Puerto la Cruz. Trabajo de Grado no publicado, Universidad de Oriente, Barcelona.
- Pino, C. (2009). Diseño de un sistema de información para el control, evaluación y estimación de las horas hombre invertidas en el proceso de desarrollo de software. Trabajo de Grado no publicado, Universidad de Oriente, Barcelona.
- Post, G. (2006). Sistemas de Administración de Bases de Datos (3ra ed.). Ciudad de México: Mc Graw Hill.
- Pressman, R. (2002). Ingeniería del Software (5ta ed.). Madrid: Mc Graw Hill.

Rumbaugh, J., Jacobson, I. y Booch, G. (2000). El Proceso Unificado Del Desarrollo De Software. Madrid: Pearson Educación.

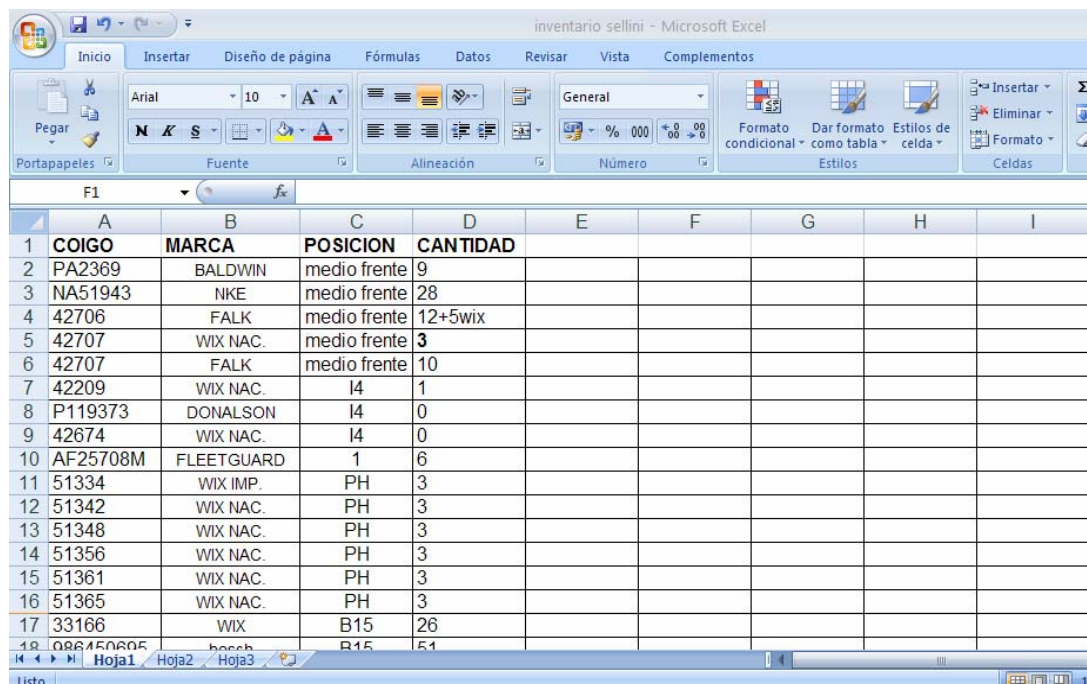
Schach, S (2005). Análisis y Diseño Orientado a Objetos con UML y el Proceso Unificado. Madrid: Mc Graw Hill.

Senn, James A. (1992). Análisis y Diseño de *Sistemas de Información* (2da ed.). Ciudad de México: McGrawHill.

Weitzenfeld, A., (2004). Ingeniería de Software Orienta a Objetos con UML, Java e Internet (3ra ed.). Ciudad de México: Thompson International.

ANEXOS

- ❖ **Información sobre el sistema de manejo de datos que posee actualmente la Empresa Distribuidora Sellini.**
- ✓ Hoja de Excel donde se almacena la información de los productos. Se muestra en la **Figura 4.45**.



	A	B	C	D	E	F	G	H	I
1	COIGO	MARCA	POSICION	CANTIDAD					
2	PA2369	BALDWIN	medio frente	9					
3	NA51943	NKE	medio frente	28					
4	42706	FALK	medio frente	12+5wix					
5	42707	WIX NAC.	medio frente	3					
6	42707	FALK	medio frente	10					
7	42209	WIX NAC.	I4	1					
8	P119373	DONALSON	I4	0					
9	42674	WIX NAC.	I4	0					
10	AF25708M	FLEETGUARD	1	6					
11	51334	WIX IMP.	PH	3					
12	51342	WIX NAC.	PH	3					
13	51348	WIX NAC.	PH	3					
14	51356	WIX NAC.	PH	3					
15	51361	WIX NAC.	PH	3					
16	51365	WIX NAC.	PH	3					
17	33166	WIX	B15	26					
18	086450605	hscch	B15	51					

Figura 4.45: Hoja de Excel manejada por la Empresa para almacenar la información de los productos

Fuente: Distribuidora Sellini

- ✓ Ficha personalizada donde se almacenan la información de los clientes y proveedores **Figura 4.46** (primera cara).

Cliente No. <u>147</u>		Tipo de Cliente A1 <input type="checkbox"/> A2 <input type="checkbox"/> A3 <input type="checkbox"/>		Referencias Comerciales						
Razón Social <u>Inversiones Tech Services 2010, e.A.</u>		Propietario		Referencias Bancarias						
Arrendatario		Jefe de Compras		Referencias Personales						
Dirección <u>Av. Diego Cisneros, Edf. Epicentro, piso 1, P1c2 Los Ruides. Caracas.</u>		Rif. <u>J-312.12.059-3</u>		Teléfono: <u>416-02-30.</u>						
Tipo de Negocio		Capital aproximado Bs.		Crédito otorgado Bs.		Condiciones de Pago: Crédito <input type="checkbox"/> Contado <input type="checkbox"/> 30 <input type="checkbox"/> 60 <input type="checkbox"/> 90 <input type="checkbox"/>				
FECHA	Orden de Compra	No. Factura	Descripción Mercancías Despachadas	Precio Mayor	Documento	Precio Especial	C U E N T A C O R R I E N T E			
							OBSERVACIONES	DEBE	HABER	SAL
13-5-10	490	11708	80-909151001	22,55		+121-	2.020,48.			
25-5-10	491	11718	30-P551670	134,10						
			20-GF481	26,35						
			24-3267012620-71	136,65						
			3-22544738=C271340	545,35						
			24-1774126600-7	158,75		+127-	14.877,69			
11-06-10	529	11840	38-5620-41010	26,30						
			57-909151001	22,55						
			33-1774126600-7	136,35						
			32-GF481	26,35						
			19-AC1519C	63,15						
			10-1541065DCD	23,35						
			10-137807300	37,90		+127-	11.452,95.			
11-06-10	530	11871	8-FP828633/47001	160,65						
			6-51640=PF47	19,40						
			12-PE47	19,40						
			12-51334	22,60						
			8-GF618	21,80						
			9-11A51943	61,35						
			10-P527482	286,50						
			5-42126	141,40						
			15-502154	141,20						
			38-33268	48,20						
			6-46440	93,85						
			38-675022.0540-71	225,50		+127-	20.136,31			
			33-3267012620-71	136,65						
11-06-10	531	11872	34-PF89	24,80						→ Sigue

Figura 4.46 Ficha personalizada donde se almacenan la información de los clientes y proveedores (primera cara)

Fuente: Distribuidora Sellini

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	Desarrollo de un sistema de información para la gestión de una empresa dedicada a la comercialización de repuestos para vehículos automotores en la ciudad de Caracas.
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Cuzzi D., Vanessa S.	CVLAC: 17.411.186 E MAIL: cuzzi_13@hotmail.com
Palomino W., Mitchel K.	CVLAC: 82.245.765 E MAIL: shadowfinal@gmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALABRAS O FRASES CLAVES:

Filtros, PHP, MySQL, JavaScript, UML, RUP, Sistema de Información, Ingeniería de Software, Desarrollo de Software

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
Ingeniería y ciencias aplicada	Ingeniería de Sistemas

RESUMEN (ABSTRACT):

Esta investigación tiene como objetivo desarrollar un sistema de información denominado **“Sellini”**, que permita a la Empresa Distribuidora Sellini actualizar el registro de información de las actividades que realiza diariamente para lograr mayor efectividad y eficiencia y hacerlo en menos tiempo. La metodología empleada en el proyecto está basada en el Proceso Unificado de Desarrollo de Software (RUP), las herramientas usadas para el desarrollo del sistema fueron el Lenguaje de Programación Orientado a Objetos PHP y el Sistema Manejador de Base de Datos MySQL. El sistema **“Sellini”** fue realizado en cuatro fases: Inicio, Elaboración, Construcción y Transición. Se determinaron las necesidades de la empresa para la automatización de los procesos de: la elaboración de pedidos, registro de productos, registro de proveedores, registro de clientes y consultas sobre la información de los mismos. La investigación se fundamentó en las teorías de: Elmasri, R y Navathe, S. sobre sistemas de Base de Datos, Larman, C. sobre el Modelo de Lenguaje Unificada (UML), Pressman, R. sobre la Ingeniería de Software.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Carrasquero, Manuel	ROL	CA	AS X	TU	JU
	CVLAC:	V-7.374.987			
	E_MAIL	manuelscm@hotmail.com			
	E_MAIL				
Rodríguez, Rhonald	ROL	CA	AS	TU	JU X
	CVLAC:	V-14.077.185			
	E_MAIL	rhoen2003@hotmail.com			
	E_MAIL				
Mujica, Victor	ROL	CA	AS	TU	JU X
	CVLAC:	V-14.054.907			
	E_MAIL	victormujica@yahoo.com			
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	07	26
AÑO	MES	DÍA

LENGUAJE. SPA**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

ARCHIVO (S): TESIS

NOMBRE DE ARCHIVO	TIPO MIME
TESIS: Comercialización de repuestos.doc	Aplication/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F
 G H I J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t
 u v w x y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

_____ Ingeniero Sistemas _____

NIVEL ASOCIADO CON EL TRABAJO:

_____ Pre-Grado _____

ÁREA DE ESTUDIO:

_____ Computación y Sistema _____

INSTITUCIÓN:

_____ Universidad de Oriente – Núcleo de Anzoátegui _____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

De acuerdo con el artículo 41 del reglamento de trabajo de grado:

“Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario, para su autorización”.

Vanessa S, Cuzzi D

..... **AUTOR**

Mitchel K, Palomino W

AUTOR

Manuel Carrasquero

ASESOR

Rhonald Rodríguez

JURADO

Mujica Victor

JURADO

POR LA SUBCOMISIÓN DE TESIS