



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**“SIPAL: SISTEMA INTEGRAL PARA LA
ADMINISTRACION DE UNA LIBRERÍA.”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

BRENDA SALAS SANTIAGO

GLORIA ANGELICA NOGUEZ BARAJAS

ROMAN ARANGO DIAZ

DIRECTOR DE TESIS: ING. LUCILA PATRICIA ARELLANO MENDOZA



MÉXICO, D.F.

MAYO 2007



Universidad Nacional
Autónoma de México

Biblioteca Central

Dirección General de Bibliotecas de la UNAM



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Agradezco a Dios por darme la sabiduría.

A mis padres que siempre estuvieron conmigo incondicionalmente, y que, sin ellos no hubiera llegado hasta aquí.

A mis hermanas por todo lo que hemos compartido.

A mi novio que siempre estuviste apoyándome.

A todos mis profesores por sus conocimientos.

A la Ing. Lucila por brindarnos su tiempo y consejos a lo largo de esta experiencia.

Gloria Angélica Noguez Barajas

Agradecimientos

A mis padres

Por su apoyo incondicional, cariño y comprensión en todo momento.

A mis hermanas

Por brindarme su compañía, su apoyo y cariño. Se que cuento con ellas siempre.

A mis maestros

Por su disposición y ayuda brindadas.

Brenda Salas Santiago

Agradecimientos

A mis Padres

Por su cariño y comprensión. Por brindarme su apoyo en todos los momentos buenos y malos a lo largo de mi vida , y por ser mi ejemplo de fortaleza y persistencia para alcanzar este objetivo tan importante para mi.

A mis hermanos

Gracias por estar conmigo en los momentos que los he necesitado.

A mis Amigos

Por todos los momentos especiales que hemos compartido a lo largo de mi vida.

A mis maestros

Por haberme brindado todos sus conocimientos de una manera incondicional.

A la facultad de Ingeniería

Por permitirme transitar esta etapa tan importante de mi vida dentro de sus aulas.

Román Arango Díaz

SIPAL

**Sistema Integral Para la Administración
de una Librería**

CONTENIDO

| | |
|--|-----|
| 1. Marco Teórico | 1 |
| 1.1 Ingeniería de Software | 4 |
| 1.1.1 Metodologías | 5 |
| 1.2 Sistemas web y cliente | 16 |
| 1.3 Bases de datos | 24 |
| 2. Planteamiento del problema y propuesta de solución | 29 |
| 2.1 Lenguajes de programación | 34 |
| 2.2 Manejadores de bases de datos | 40 |
| 2.3 Web server | 46 |
| 3. Análisis de requisitos (Información relacionada con la librería) | 51 |
| 3.1 Estructura actual de la librería | 52 |
| 3.2 Requerimientos del negocio | 53 |
| 3.3 Pantallas de interfaz de Usuario-Sistema (Vistas) | 60 |
| 4. Diseño | 64 |
| 4.1 Especificación de la arquitectura | 65 |
| 4.2 Diagramas de casos de uso | 69 |
| 4.3 Narrativas de casos de uso | 74 |
| 5. Implementación (codificación) y pruebas | 105 |
| 5.1 Matriz de pruebas | 111 |
| 6. Verificación, Liberación y Control | 114 |
| 6.1 Guía de implementación | 115 |
| 6.2 Instructivo de administración | 125 |
| Conclusiones generales | 135 |
| Glosario de términos | 136 |
| Bibliografía | 139 |

CAPÍTULO 1

MARCO TEÓRICO

Aunque el término Software es utilizado diariamente, es importante definirlo con claridad; realizando una minuciosa búsqueda encontramos la siguiente definición: “El software son las instrucciones electrónicas que van a indicar a la computadora qué es lo que tiene que hacer. También se puede decir que son los programas usados para dirigir las funciones de un sistema de computación o un hardware.”

Una vez que el término ha sido definido se procede a describir la situación actual del desarrollo de software. Haciendo una búsqueda se encontrará que muchos autores han hablado de las oportunidades que han surgido en la industria de desarrollo de software, al respecto, Booch (1997) comenta “A nivel mundial existe una demanda insaciable de software, a primera vista es una buena noticia. Se pensaría que son los tiempos adecuados para que los desarrolladores de software tengan éxito, debido a que existirán un sinnúmero de oportunidades, pero son malas noticias, debido a que ningún número de programadores será suficiente para satisfacer esa demanda”.

Además de esta demanda insaciable, la tasa de cambio en la tecnología de desarrollo de software es casi instantánea. Dieciocho meses en el calendario es un siglo en años de software. Una sola distracción y se pierde el próximo gran cambio. Sólo unos años atrás, C++ era lo novedoso, en estos días es la tecnología .NET, la programación visual y la programación basada en componentes han cambiado sus reglas. Es difícil predecir lo que vendrá después.

Además, la ingeniería de software es dominada por las actividades intelectuales que son enfocadas a resolver problemas de inmensa complejidad con numerosa incertidumbre en perspectivas competitivas.

Debido a lo anterior, ha surgido la necesidad de buscar nuevas metodologías y tecnologías que traten de controlar el crecimiento de la complejidad del software, y que ayuden a que los proyectos se terminen en el tiempo y costo asignado inicialmente, lo que llevará a incrementar el porcentaje de proyectos exitosos de software.

Según Zermeño (2003) en México existen oportunidades de desarrollar una industria de desarrollo de software debido a:

- Ubicación privilegiada
- Costos competitivos
- Cultura empresarial similar a la de estados unidos

Que generará beneficios como:

- Atraer inversionistas de TI
- Fuerte efecto multiplicador, lo que mejora la economía de la región
- Puente tecnológico entre Estados Unidos y Latinoamérica
- Fomenta la desconcentración de población
- Puede llegar a cambiar el perfil exportador
- Promueve el empleo masivo
- Evita fuga de cerebros
- Permitirá ser jugadores importantes en el mundo de las tecnologías de TI
- Permite pasar al lado correcto de la vertiente digital.

Sin embargo, actualmente en México existe poca oferta de desarrollo de software, el nivel de calidad o certificación en las empresas dedicadas al desarrollo es muy bajo o casi inexistente, y la industria se encuentra muy dispersa, sólo una empresa en México tiene más de 1,000 empleados, muy pocas más de 100 y muchas debajo de 50.

Para que México pueda exportar software de calidad y escapar de la brecha respecto al desarrollo de software, se tiene que promover el desarrollo de compañías de excelencia que usen procesos reconocidos, que puedan crecer para lograr las economías de escala, que desarrollen el mejor personal, y además que generen experiencia para la industria.

Es así como México también cuenta con la oportunidad de “encontrar mejores metodologías de desarrollo de software que ayuden a producir un producto de calidad a bajo costo, que

permita competir en un mercado global y lo impulse como un país exportador de conocimiento”. Zermeño (2003)

Por ultimo Booch (1997) concreta “Es importante pasar de alguien que sólo escribe líneas de código a un ingeniero que hace mucho mas al preguntarse que estoy haciendo con y por el proyecto con el que estoy trabajando.” Para lograr lo anterior es necesario que se aplique una ingeniería de software capaz de asegurar un producto de calidad.

1.1 Ingeniería de Software

Al igual que otros tipos de ingeniería, la ingeniería de software no solamente consiste en producir productos, más bien significa producir los productos de la mejor relación costo-beneficio posible. Asignando una cantidad de recursos ilimitados, la mayoría de los problemas de software podrían ser resueltos, pero el reto de la ingeniería de software es producirlo con alta calidad y una cantidad finita de recursos en el tiempo estimado.

Sommerville (1989) también menciona cuatro atributos que cualquier producto de software bien diseñado debe tener:

- **Fácil Mantenimiento:** Un software con una larga vida útil, esta sujeto a un cambio regular, por lo que es importante que el software sea escrito y documentado de modo que los cambios puedan ser hechos sin costos innecesarios.
- **Confiable:** Un nivel apropiado de confiabilidad es esencial, si el software es de uso general.
- **Eficiente:** Esto no necesariamente significa que el desempeño debe ajustarse al hardware donde el software se ejecuta. Ciertamente, tratar de maximizar la eficiencia puede hacer al software mucho más difícil de cambiar. Sin embargo, que sea eficiente significa que el software no debe desperdiciar el uso de recursos del sistema como la memoria y el procesador.

- **Fácil de Usar:** La mayoría de los usuarios no utilizan el potencial completo del software, porque la interfaz en la cual se ofrece, es difícil de utilizar. En el diseño de la interfaz se debe de tomar en cuenta las capacidades y los antecedentes de los usuarios a los que va dirigido el software, es decir se debe estar hecho a la medida de los usuarios.

En este aspecto, la ingeniería de software a través de las metodologías de desarrollo de software provee una guía que sirve como base para el desarrollo de software de manera exitosa.

1.1.1 Metodologías

La función primaria de una metodología de proceso de desarrollo de software es el determinar el orden de las etapas envueltas en el desarrollo, evolucionar y establecer el criterio de transición para procesar de una etapa a la siguiente.

El Modelo de Cascada

El modelo de cascada es atribuido a Walter Royce, (Van 2000) el cual es descrito por Pressman a continuación:

La Figura 1.1.1-1 ilustra el paradigma del ciclo de vida clásico para la ingeniería del software. Algunas veces llamado 'modelo en cascada', el paradigma del ciclo de vida exige un enfoque sistemático y secuencial del desarrollo de software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

Conceptualizado a partir del ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

- **Ingeniería y análisis del sistema.** Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos del software. Este planteamiento del sistema es esencial cuando el software debe

interrelacionarse con otros elementos, tales como hardware, personas y bases de datos. La ingeniería y el análisis del sistema abarcan los requisitos globales a nivel del sistema con una pequeña cantidad de análisis y diseño a nivel superior.

- **Análisis de los requisitos del software.** El proceso de recopilación de requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software (analista) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.
- **Diseño.** El diseño del software es realmente un proceso múltiple que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.
- **Codificación.** El diseño debe traducirse en forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- **Prueba.** Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, se realizan pruebas que aseguren que la entrada definida produce los resultados que realmente se quieren.
- **Mantenimiento.** El software, indudablemente sufrirá cambios después de que se entregue al cliente una posible excepción es el software empotrado. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo por ejemplo, un cambio solicitado debido a que se tiene

un nuevo sistema operativo o dispositivo periférico, o debido a que el cliente requiera ampliaciones funcionales o de rendimiento. El mantenimiento del software aplica cada uno de los pasos procedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

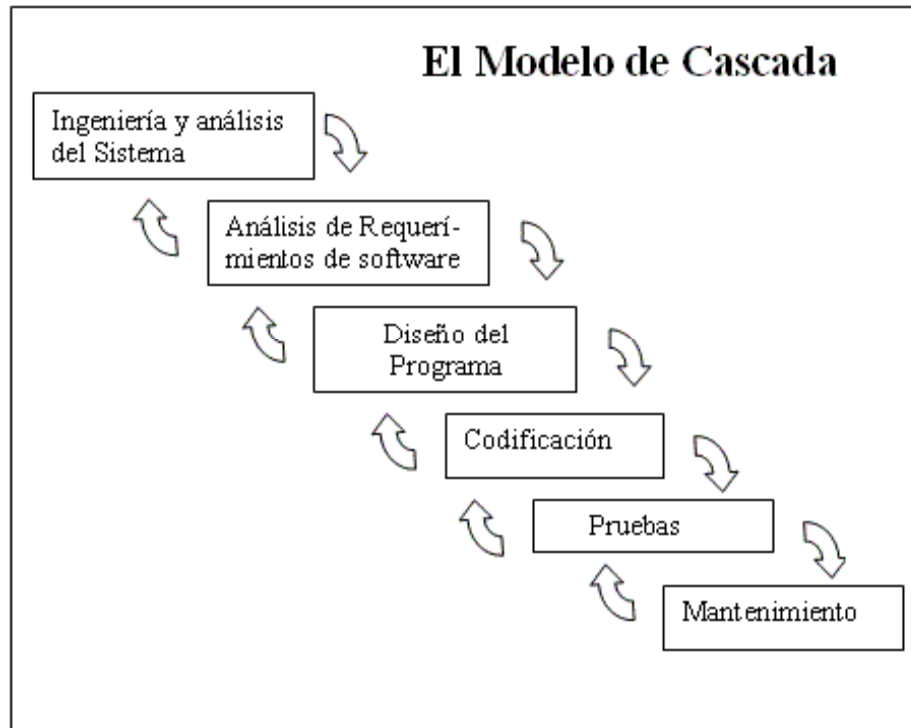


Figura 1.1.1-1 Esquema de Fases del Modelo de Cascada

El Modelo Incremental

En el modelo incremental el software es construido, no escrito, esto es, el software va siendo construido paso a paso, de la misma manera en que se construye un edificio.

Mientras un producto de software esta en el proceso de ser desarrollado, cada paso adiciona algo sobre lo que se ha hecho anteriormente. Día a día los módulos del sistema son añadidos, el desarrollo del producto se procesa incrementalmente hasta la conclusión del producto.

Por supuesto no es muy cierto que el progreso se hará diariamente, así como un constructor que tiene que tirar una pared mal posicionada o remplazar una ventana que un pintor ha quebrado, algunas veces es necesario re-especificar, re-diseñar, re-codificar o en el peor de los casos, eliminar lo que ya se había completado y empezar de nuevo.

El hecho de que este software es construido incrementalmente, ha llevado a los desarrolladores del modelo que explotan ese aspecto del desarrollo de software, a llamarlo modelo incremental. El producto es diseñado, implementado, integrado y probado como una serie de construcciones incrementales, donde la construcción consiste en piezas de código de varios módulos interactuando para proveer de una capacidad funcional específica.

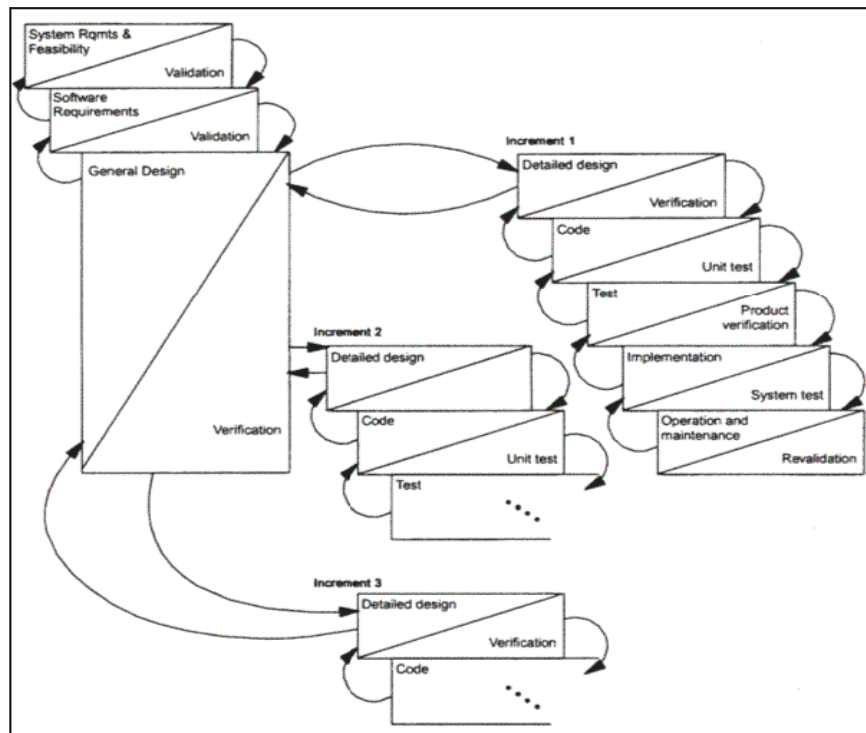


Figura 1.1.1-2 - Fases del Modelo Incremental.

Modelo de Espiral

Por lo general un elemento de riesgo está envuelto en el desarrollo de software, por ejemplo, personal clave puede renunciar antes que el producto haya sido adecuadamente documentado. También que el proveedor de hardware en el cual el producto es críticamente dependiente puede irse a la bancarrota; o la compañía puede investigar y desarrollar un sistema de administración de base de datos, pero antes que el producto pueda ser vendido, un paquete funcionalmente equivalente y a precio más bajo es anunciado por la competencia.

El modelo de espiral completo se muestra en la figura 1.1.1-3. La dimensión radial representa el costo acumulativo al día. La dimensión angular representa el progreso a través de la espiral. Cada ciclo de la espiral representa una fase, una fase empieza (en la parte izquierda superior del cuadrante) definiendo los objetivos de esa fase, alternativas para alcanzar esos objetivos, y obligaciones impuestos a esas alternativas. Este proceso resulta en una estrategia para alcanzar esos objetivos, esta estrategia es analizada desde el punto de vista de riesgo. Muchos intentos son hechos para resolver cada riesgo potencial, en algunos casos construyendo un prototipo. Si el riesgo no puede ser resuelto, el proyecto deberá terminar inmediatamente, bajo algunas circunstancias, una decisión podría ser continuar el proyecto pero a una escala menor, si todos los riesgos son resueltos exitosamente, el próximo paso en el desarrollo debe ser iniciado (cuadrante superior derecho). Este cuadrante del modelo de espiral es similar al modelo de cascada. Finalmente, los resultados de esta fase son evaluados y se planea la siguiente fase.

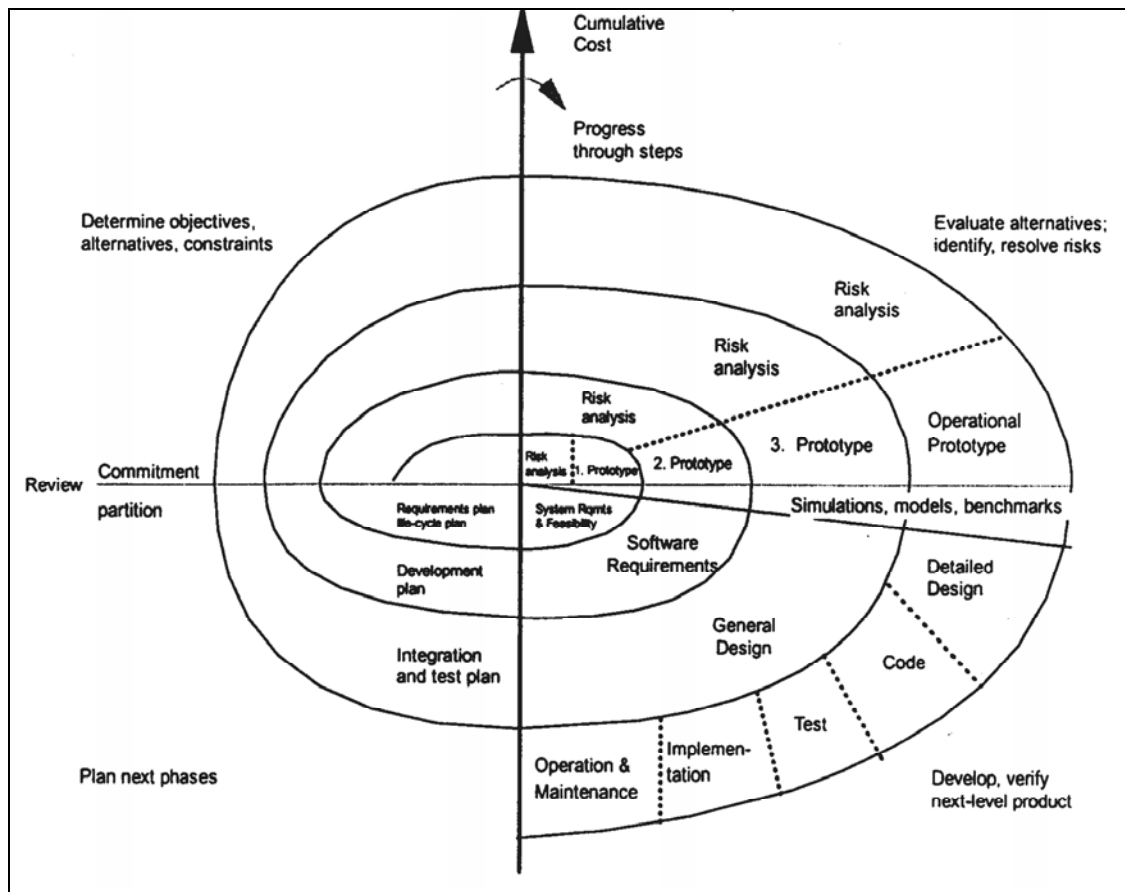


Figura 1.1.1-3 - Esquema de las fases del Modelo Incremental.

Pero, al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede ser difícil convencer a grandes clientes (particularmente en situaciones bajo control) de que el enfoque evolutivo es controlable. Si no se descubre un riesgo importante, indudablemente surgirán problemas. Por último, el modelo en sí mismo es relativamente nuevo y no se ha usado tanto como el ciclo de vida o la creación de prototipos. Pasarán unos cuantos años antes de que se pueda determinar con absoluta certeza la eficacia de este importante nuevo paradigma.

Proceso Unificado (RUP)

La metodología del Proceso unificado de Rational (RUP) es un compendio de ideas y experiencias de los líderes de la industria, socios y literalmente cientos de proyectos reales cuidadosamente sintetizados dentro de un conjunto de mejores prácticas, flujos de trabajo y objetos para un desarrollo de software iterativo. Cuando son usadas en combinación las mejores practicas promovidas por la metodología del RUP, las cuales incluyen desarrollo iterativo, administración de requerimientos, uso de arquitecturas de componentes, modelado visual, administración de cambios y continua verificación de calidad, golpea a la raíz de los problemas de desarrollo de software, ayudando a evitar caídas comunes con el apalancamiento de nuevas tecnologías y herramientas. Usando esta metodología probada y compartiendo un proceso compresivo, el equipo será capaz de comunicarse más efectivamente y trabajar más eficientemente.

El Proceso Unificado guía a los equipos de proyecto en la administración del desarrollo iterativo de un modo controlado, mientras se balancean los requerimientos del negocio, el tiempo del mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y el establecimiento de una guía arquitectónica para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones.

El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a proyectos que varían en tamaño y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología de objetos, en el desarrollo de software de misión crítica en una variedad de industrias. Uno de los componentes clave es el UML.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

- Interacción continua con el usuario desde un inicio.
- Mitigación de riesgos antes de que ocurran.
- Liberaciones frecuentes de ejecutables.
- Aseguramiento de la calidad.
- Participación del equipo en todas las decisiones del proyecto.
- Anticiparse al cambio de requerimientos.

Siendo las características primordiales del Proceso Unificado las siguientes:

- Proceso Iterativo e incremental.
- Proceso Centrado en la arquitectura.
- Proceso Guiado por casos de uso.
- Confrontación de riesgos.

La estructura del RUP se muestra en dos dimensiones:

El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida de procesos como su desarrollo.

El eje vertical representa disciplina, la cuál agrupa actividades lógicamente por naturaleza.

La primera dimensión representa el aspecto dinámico del proceso como es promulgado, y es expresado en términos de fases, iteraciones y pequeñas metas (Milestones). La segunda dimensión representa el aspecto estático del proceso: Cómo está descrita en términos de componentes de proceso, disciplina, actividades, flujos de trabajo, artefactos, y documentos.

El ciclo de vida del software esta compuesto de ciclos. Cada ciclo trabaja en una nueva generación de un producto. El RUP divide un ciclo de desarrolló en 4 fases consecutivas como se muestra en la figura 1.1.1-4:

- La fase Inicial.
- La fase de Elaboración.
- La fase de Construcción.
- La fase de Transición.

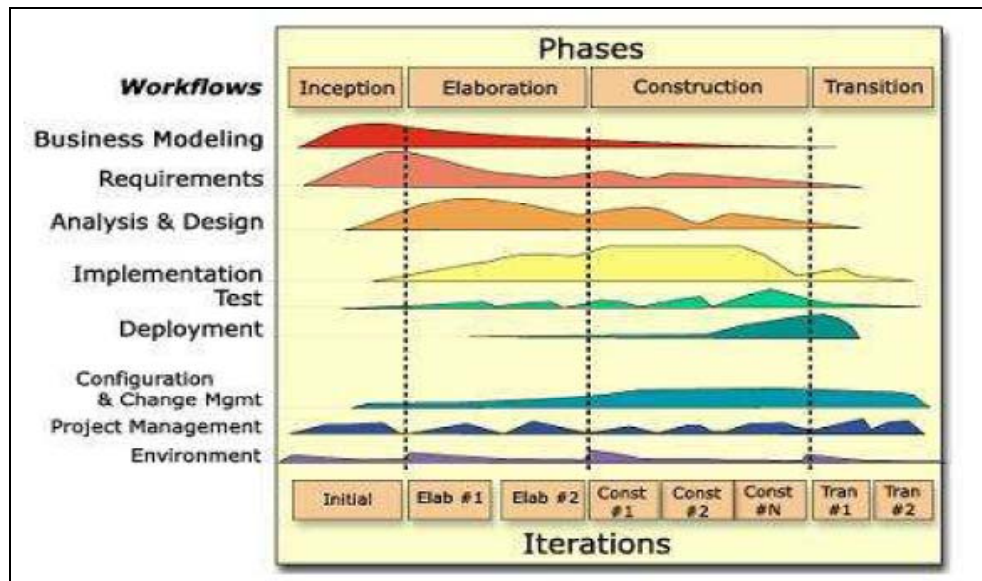


Figura 1.1.1-4 - Esquema de las fases del Proceso Unificado.

Cada fase es concluida con una meta bien definida (milestone) – la cual es un punto en el tiempo y se centra en las decisiones críticas que se deben tomar y qué metas clave deben ser alcanzadas.

La Fase de Inicio.- Durante la fase de inicio se establece un caso de negocios del sistema y se delimita el alcance del proyecto. Para completar esto, se debe de identificar todas las entidades externas con las cuales el sistema interactuara (actores) y se debe definir la naturaleza de estas interacciones a un alto nivel. El caso de negocios incluye un criterio de éxito, valoración de riesgos, estimado de recursos necesarios y un plan de la fase mostrando las fechas principales de las metas.

El resultado de esta fase inicial es:

Una visión de documento: Una visión general de los requerimientos centrales del proyecto, las características claves y las restricciones principales.

- Un modelo de uso inicial.
- Un glosario inicial del proyecto.
- Un caso inicial de negocios.
- Una valoración inicial de riesgos.
- Un plan de proyecto, mostrando fases e iteraciones.
- Un modelo de negocios.
- Uno o varios prototipos.

Al término de la fase inicial se encuentra la primera meta del proyecto. La meta de objetivo de ciclo de vida Los criterios de evaluación de la fase inicial son:

- Los accionistas deben estar de acuerdo con el alcance de la definición de las estimaciones de costo y calendario.
- El entendimiento de los requerimientos como evidencia de fidelidad de los usos principales.
- Credibilidad de los estimados de costo y calendario, prioridades, riesgos y proceso de desarrollo.
- Gastos actuales contra gastos planeados.

La Fase de Elaboración.- El propósito de la fase de elaboración es analizar el dominio del problema, establecer una sólida arquitectura, desarrollar el plan del proyecto y eliminar los elementos de mayor riesgo del proyecto. Las decisiones sobre arquitectura deben ser hechas con un entendimiento de todo el sistema, sus alcances, su función principal y los requerimientos no funcionales, así como los requerimientos de desempeño.

Es fácil entender que la fase de elaboración es la fase mas crítica de las 4 fases. Al término de esta fase la “ingeniería dura” es considerada completa y el proyecto experimenta su más importante día en los cálculos; la decisión sobre continuar o no con las fases de construcción o transición. Mientras el proceso debe siempre acomodarse a los cambios, las actividades de la fase de elaboración se aseguran que la arquitectura, los requerimientos y el plan sean lo suficientemente estables, y el riesgo sea lo suficientemente mitigado, así que se puede

predecir el costo y el calendario para completar el desarrollo. Conceptualmente, este nivel de fidelidad corresponde al nivel necesario para que una organización contemple costo de la fase.

En la fase de elaboración un prototipo ejecutable de la arquitectura es construido en una o más iteraciones, dependiendo del alcance, el tamaño, el riesgo y la novedad del proyecto.

Este esfuerzo por lo menos deberá manejar los usos críticos identificados en la fase de inicio, la cual típicamente expone la mayoría de los riesgos del proyecto. Mientras un prototipo evolucionado de un componente de producción de calidad es la meta, este no lo excluye del desarrollo de uno o más prototipos exploratorios, a través de prototipos que mitiguen los riesgos específicos como los de diseño, requerimientos intercambiados, estudios de factibilidad de componentes, o la demostración a los inversionistas, clientes y usuarios finales.

Los resultados de la fase de elaboración son:

- Un modelo de usuario (80% por lo menos)
- La captura de requerimientos suplementarios, los no funcionales y cualquier requerimiento que sea no asociado con un caso específico de uso.
- Una descripción de la arquitectura del software.
- Un prototipo de la arquitectura ejecutable.
- Una lista de riesgos y un caso de negocios revisado.
- Un plan de desarrollo para el proyecto en general, incluyendo mostrando las iteraciones y la evaluación y los criterios de evaluación para cada iteración
- Un manual de usuario preliminar

Meta: Ciclo de vida de la arquitectura

Al final de la fase de elaboración se encuentra la segunda meta importante. La meta de arquitectura del ciclo de vida. En este punto se examinan a detalle los objetivos y alcance del sistema, la selección de la arquitectura y la resolución de los riesgos mayores.

La Fase de Construcción.- Durante la fase de construcción, todos los componentes restantes y las características de las aplicaciones son desarrolladas e integradas al producto y todas las características son ampliamente probadas. La fase de construcción es en un sentido un proceso de manufactura, donde el énfasis esta colocado en la administración de recursos y el control de las operaciones para optimizar costos, calendarios y calidad. En este sentido, la administración mental experimenta una transición del desarrollo de la propiedad intelectual durante la fase de inicio y elaboración, al desarrollo de un producto destacable durante las fases de construcción y transición.

El resultado de la fase de construcción es un producto listo para ponerse en las manos del usuario final, y consisten en:

- El producto del software integrado en la adecuada plataforma.
- Los manuales de usuario.
- Una descripción de la revisión actual.

Meta: Capacidad operacional inicial.

Al termino de la fase de construcción se encuentra la tercera mayor meta del proyecto (la meta de capacidad operacional inicial) en este punto, se decidirá si el software, el sitio y los usuarios están listos para ser operacionales sin exponer al proyecto a riesgos mayores.

Esta liberación es frecuentemente llamada versión “Beta”.

La Fase de Transición.- El propósito de la fase de transición es la transición del producto de software a la comunidad de usuarios. Una vez que el producto ha sido dado al usuario final, usualmente se requiere del desarrollo de nuevos entregables, corrigiendo algunos problemas o el terminar las características que fueron pospuestas.

La fase de transición es introducida cuando el producto es lo suficientemente maduro como para ser entregado al usuario final. Esto típicamente requiere que algunos subconjuntos del sistema hayan sido completados a un nivel aceptable de calidad y que la documentación del

usuario esté disponible, así la transición del usuario proveerá de resultados positivos para todas las partes. Esto incluye:

Una prueba beta que valide que el nuevo sistema cumple con las expectativas del usuario.

Una operación paralela junto al sistema que se esta remplazando.

Conversión de las bases de datos operacionales.

Entrenamiento de los usuarios y administradores.

Entrega del producto a los equipos de mercadotecnia, distribución y ventas.

Un esfuerzo considerable es hecho para desarrollar documentación orientada al usuario, entrenamiento de usuarios y reaccionar a la retroalimentación de usuario. En este punto del ciclo de vida del producto, la retroalimentación del usuario debe ser tomada en cuenta para afinar el producto, configurarlo e instalarlo.

Los objetivos primarios de la fase de transición incluyen:

- Alcanzar el soporte de usuario.
- Lograr la concurrencia de los accionistas que el desarrollo esta completo y es consistente con el criterio de evaluación de la visión.
- Lograr que los lineamientos de producto final tanto en efectividad de tiempo y costo, como práctica.

Al final de la fase de transición se encuentra la cuarta meta en importancia del proyecto.

La meta de liberación de producto. En este punto, se decide si los objetivos fueron cumplidos o si se debe de empezar otro ciclo de desarrollo, en algunos casos, estas metas pueden coincidir con el fin de la fase de inicio del siguiente ciclo.

1.2 Sistemas Web y Cliente

Sistemas WEB

Entre las principales necesidades que enfrentan las organizaciones debido a su constante transformación se encuentran las siguientes:

- La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.
- Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones Web, que por su idiosincrasia, cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios Web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, un interés creciente por el desarrollo de propuestas metodológicas que ofrezcan un marco de referencia adecuado cuando se desarrolla un sistema de información Web.

Términos como aplicación Web y sitio Web tienen distintos significados para distintos autores. Varios autores han definido taxonomías para tratar de clasificar los distintos tipos de aplicaciones Web. Por ejemplo, Ginige y Murugesan definen las siguientes categorías de aplicaciones Web: informativas (noticias en línea, servicios de noticias, catálogos, manuales), interactivas (formularios de registro, servicios en línea), transaccionales (compras electrónicas, bancos en línea), sistemas de workflow (planificación en línea, administración de inventarios, monitoreo), ambientes de trabajo colaborativo (sistemas distribuidos de autoría, herramientas colaborativas), comunidades en línea (grupos de Chat, sistemas de recomendación), portales Web (centros comerciales electrónicos, intermediarios en línea).

Por su parte, Pressman define las siguientes categorías de aplicaciones Web: informativas (se proporciona un contenido sólo de lectura con navegación y enlaces simples), de descarga (el usuario descarga información desde el servidor), personalizable (el usuario personaliza el contenido según sus propias necesidades), interactivas (comunicación entre comunidades de usuarios), con entradas de usuario (el ingreso de información por parte del usuario a través de formularios es el mecanismo primario), orientadas a transacciones (solicitudes del usuario al servidor Web), orientadas a servicios (proporcionan servicios a los usuarios), portales (el usuario navega en la aplicación y esta lo lleva a servicios fuera del

dominio del portal), con acceso a bases de datos (el usuario extrae información de una base de datos), almacén de datos (el usuario consulta información en una colección grande de datos).

Jim Conallen trata de establecer algunas diferencias entre los sistemas Web desde el punto de vista de la modificación de la lógica del negocio. Según Conallen, una aplicación *Web* es un sistema Web (servidor Web, red, protocolo HTTP, y browser) en el cual el usuario, a través de navegación y entrada de datos afecta el estado del negocio. De este modo, un sitio Web es un sistema Web donde la navegación del usuario no modifica la lógica de negocio, o un sistema Web donde no hay lógica del negocio. Sin embargo, esta clasificación es un poco confusa. Una buena parte de los sistemas Web existentes extraen parte de la información que presentan a los usuarios desde bases de datos, y ocasionalmente modifican esta información, dependiendo de las acciones del usuario. De las cuatro operaciones básicas para el manejo de datos: crear (insert), recuperar (select), modificar (update) y borrar (delete), es común que los sistemas Web utilicen la segunda para desplegar información. Sin embargo, son las otras tres las que modifican el estado de una base de datos (el estado de la lógica del negocio), y es la presencia de estas operaciones, según la definición de Conallen, lo que define la funcionalidad de un sistema Web es: en qué proporción la interacción del usuario con el sistema permite modificar el estado de los datos del sistema. Por otra parte, la clasificación dada por Conallen no contempla la complejidad de los sistemas Web, la cual es de vital importancia en la Ingeniería Web para la construcción de las aplicaciones y para su futuro mantenimiento.

Por ejemplo, según la clasificación de Conallen, algunos motores de búsqueda serían catalogados como “sitios Web”, ya que la navegación e interacción de los usuarios con el sistema no afecta el estado de la lógica del negocio, pues solamente son ejecutadas instrucciones de tipo *select* para obtener la información presentada. Y claramente un sitio Web de este tipo es mucho más complejo que un portal donde se muestre, por ejemplo, la cartelera cinematográfica de la semana. No obstante, estos dos sitios Web tienen algo en común, y es que la información que despliegan a sus usuarios cambia periódicamente, es decir, el contenido del sitio cambia.

El usuario interactúa con las aplicaciones Web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente. Esta distribución se conoce como el modelo o arquitectura de tres capas.

En la mayoría de los casos, el navegador suele ser un mero presentador de información (modelo de cliente delgado), y no lleva a cabo ningún procesamiento relacionado con la lógica de negocio. No obstante, con la utilización de applets, código javascript y DHTML la mayoría de los sistemas se sitúan en un punto intermedio entre un modelo de cliente delgado y un modelo de cliente grueso (donde el cliente realiza el procesamiento de la información y el servidor sólo es responsable de la administración de datos). No obstante el procesamiento realizado en el cliente suele estar relacionado con aspectos de la interfaz (como ocultar o mostrar secciones de la página en función de determinados eventos) y nunca con la lógica de negocio.

En todos los sistemas de este tipo y ortogonalmente a cada una de las capas de despliegue comentadas, se puede dividir la aplicación en tres áreas o niveles:

- Nivel de presentación: es el encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.
- Nivel de negocio: contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.
- Nivel de administración de datos: encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio.

Los dos primeros y una parte del tercero (el código encargado de las actualizaciones y consultas), suelen estar en el servidor mientras que la parte restante del tercer nivel se sitúa

en la base de datos (notar que, debido al uso de procedimientos almacenados en la base de datos, una parte del segundo nivel también puede encontrarse en la misma).

Además, es muy importante diferenciar los aspectos de estos tres componentes, pues el perfil de la persona encargada de implementar cada uno de éstos, es distinto. De acuerdo con estos tres nuevos criterios, se define una nueva clasificación para sistemas Web. Los sistemas Web se clasificarían dependiendo de la cantidad y tipo de código (HTML, scripts y otros lenguajes de programación) que posean, según la cantidad de información que contengan, y según la calidad y complejidad del diseño.

Los sistemas que no contienen código ejecutable, es decir, los sistemas que no poseen funcionalidad son llamados portales o sitios Web. Por el contrario, los sistemas que contienen mucha funcionalidad, independientemente del contenido y del diseño son llamados aplicaciones Web. Pero, ¿qué pasa con un sistema Web que no cae en ninguna de las dos regiones señaladas?, ¿sería considerado un sitio o una aplicación Web? En realidad lo más importante es estimar el tiempo y esfuerzo en el desarrollo del sistema Web. El límite que hace la diferencia entre sitio y aplicación, dependerá de las métricas de estimación de cada empresa.

En resumen una aplicación Web es aquella que requiere suficiente esfuerzo en cuanto a su funcionalidad, como para que se requiera un ingeniero de software.

Sistemas Cliente - Servidor

El modelo Cliente - Servidor reúne las características necesarias para proveer una infraestructura de procesamiento de información que permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización, independientemente del tamaño y complejidad de las operaciones de ésta.

El modelo cliente – servidor se basa en la utilización de terminales (cliente), conectados a una o varias computadoras que proveen recursos (servidor). De esta manera los clientes son los elementos que necesitan servicios y el servidor es la entidad que posee los recursos proporcionando una forma eficiente de compartir y manejar los recursos disponibles entre equipos computacionales en un ambiente seguro y confiable a través de la red de datos. En

la definición del modelo cliente – servidor se manejan tres elementos importantes: presentación, procesamiento y datos.

Un sistema Cliente/Servidor suele ejecutarse en al menos dos sistemas distintos (uno hace de cliente y el otro de servidor). No obstante, es posible que tanto el cliente como el servidor se encuentren en un único sistema. Generalmente un servidor proporciona servicios a varios sistemas clientes, aunque puede haber un único sistema. La función del cliente suele llevarla a cabo un servidor de archivos, excepto cuando se necesita el máximo rendimiento y debe utilizarse un servidor especializado. A menudo el cliente es un sistema de escritorio conectado en red. Siempre que el usuario necesite recuperar o almacenar información, la parte cliente de la aplicación ejecuta una solicitud, que se envía (generalmente por una red) al servidor. El servidor ejecuta entonces la solicitud y devuelve la información al cliente.

A continuación se presenta un resumen de las principales funciones de los elementos involucrados en el modelo cliente – servidor.

Servidor

- Administración de la red, de los mensajes, del control y entradas al sistema.
- Administración, auditoria, recuperación y contabilidad de los recursos compartidos.
- Maneja el control de accesos concurrentes a base de datos compartidas.
- Activación de procesos encargados de atender las peticiones y enviar respuestas al cliente.
- Presentación a todos sus clientes una interfaz única y bien definida.
- Establecimiento de enlaces de comunicaciones con otras redes de área local o extensa.
- Administración, auditoria de bloqueos internos y recuperación de fallas.

Cliente

- Captura y validación de datos de entrada y salida.
- Maneja la interfaz del usuario.
- Genera consultas e informes sobre la información que reside en los servidores.
- Genera peticiones de recursos o aplicaciones en específico.

Capas en un sistema de Cliente – Servidor

En una arquitectura monolítica no hay distribución; los tres niveles tienen lugar en el mismo equipo. Ejemplo de este tipo de arquitectura tenemos los mainframes que concentran la funcionalidad de almacenamiento y lógica en un mismo equipo.

En un esquema Cliente/Servidor clásico Figura 1.2-1 existen dos capas, el cliente y el servidor: éste está ubicado normalmente en otra máquina, y suele ser un gestor de base de datos, de servidor de archivos o servidor de transacciones, a los que accedemos directamente desde nuestra aplicación cliente. El gestor de bases de datos es el que proporciona la conectividad, así como capacidades tan fundamentales como el soporte de transacciones.

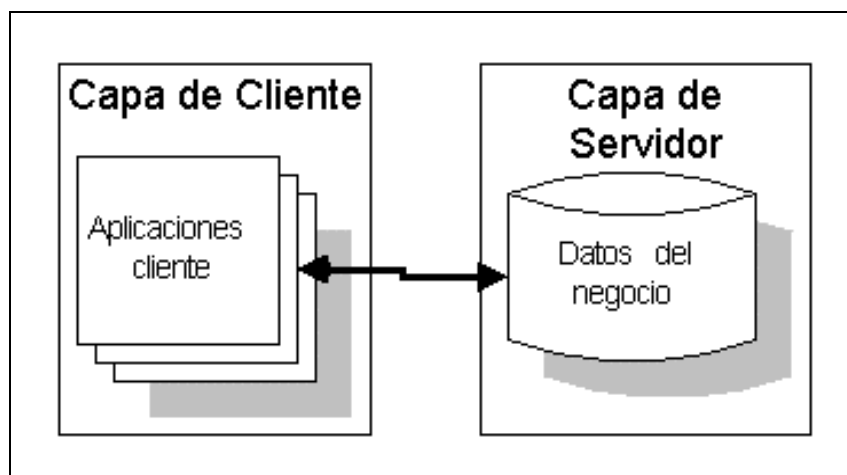


Figura 1.2-1 Esquema de arquitectura Cliente/Servidor clásica.

Para establecer la comunicación entre el cliente y el servidor se requiere de una infraestructura capaz de proporcionar mecanismos básicos de direccionamiento y transporte; de modo que todos los clientes pidan o envíen información a la misma, no al gestor de base de datos en el servidor. La capa intermedia o *middle-tier* (Figura 1.2-2), hace que nuestro sistema sea un sistema de tres capas (*three-tiered*).

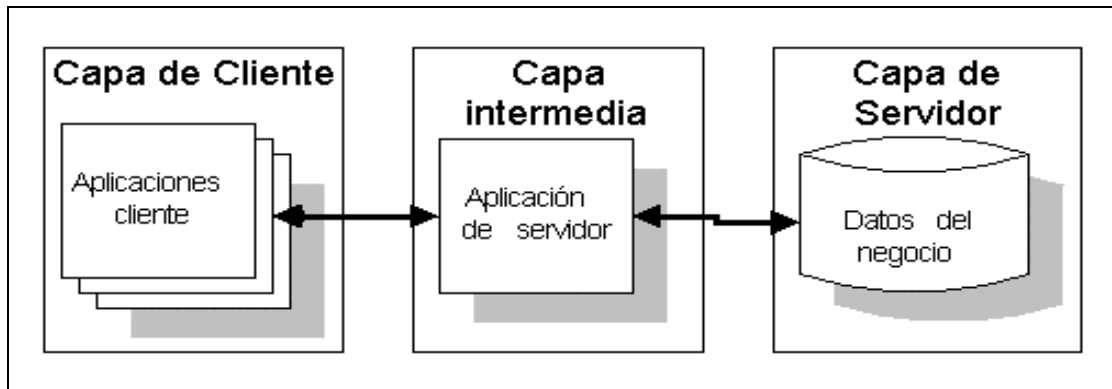


Figura 1.2-2 Arquitectura Cliente/Servidor en tres capas (three-tier)

Ventajas y desventajas del modelo Cliente – Servidor

Existen diferentes ventajas de manejar un modelo cliente – servidor, a continuación se pueden enunciar:

- División de transacciones. Las transacciones son divididas en procesos independientes capaces de cooperar entre sí para poder facilitar el intercambio de información, servicios o recursos, de tal manera que permiten que las aplicaciones sean distribuidas de una forma transparente y con la capacidad de manejar múltiples procesos de diferentes tipos.
- Aplicaciones independientes de la plataforma. El diseño de las aplicaciones debe ser autónomo de la plataforma donde residen para manejar una migración cuando sea requerida sin tener que realizar modificaciones a los programas que invocan cada aplicación.
- Administración Centralizada. Cada servidor se encarga de gestionar procesos o recursos que deben ser compartidos por uno o varios usuarios, de tal manera que se facilite la asignación apropiada de recursos disponibles y manteniendo:
 - La información del cliente disponible para más de una aplicación en un mismo tiempo.
 - La información se debe relacionar coherentemente y no almacenarse de forma redundante.

- Modularidad. Permite que con una simple adición de otro equipo se pueda multiplicar la capacidad de cómputo sin afectar a los demás procesos, favoreciendo la escalabilidad de las soluciones. También es más sencillo implantar una aplicación nueva por medio de construir una extensión con un servidor y estaciones de trabajo nuevos.
- Acceso a la información. Los clientes tienen la capacidad de comunicarse con procesos auxiliares encargados de establecer la conexión con el servidor, enviar peticiones de los usuarios, recibir y presentar respuestas para los usuarios, manejar fallas y realizar actividades de sincronización y seguridad habilitando a los clientes para realizar sus propios procesamientos, como por ejemplo el desplegar e interpretar la información.

Así como existen ventajas de los sistemas cliente – servidor encontramos desventajas algunas de las cuales son:

- El principal problema es con la red de comunicación, ya que en sistemas robustos se presentan problemas de saturación en el tráfico, pérdida de mensajes, etc.
- Al compartir datos a través de la red, podría surgir el problema de la seguridad de los mismos.
- Incompatibilidad. Encontramos en la práctica que las incompatibilidades mayores o menores entre computadoras, sistemas operativos, lenguajes, protocolos, interfaces y programas de aplicación son elevadas.

1.3 BASES DE DATOS.

Administración de datos

Sin los datos y la capacidad para procesarlos, una organización no podría completar con éxito la mayor parte de sus actividades de negocios. No estaría en posibilidad de pagar a los empleados, enviar facturas, solicitar nuevas existencias o producir información que ayude a los gerentes a tomar decisiones. Los datos consisten de hechos sin procesar, por ejemplo la cantidad de empleados y cifras de ventas. Para que los datos se transformen en información útil, primero se deben organizar en forma significativa.

Modelado de datos.

Entre las consideraciones fundamentales al organizar los datos en una base de datos se incluyen la determinación de los datos que se deben recopilar en la base de datos, de quien tendrá acceso a ella y como podrían desear utilizarlos. Con base en estas determinaciones se puede crear entonces una base de datos. Su creación requiere de dos tipos de diseño diferentes: un diseño lógico y uno físico.

El modelo lógico de una base de datos muestra un modelo abstracto de cómo se debe estructurar y ordenar los datos para cumplir con las necesidades de información de la organización. El diseño lógico de una base de datos incluye la identificación de las relaciones entre las diferentes sesiones de datos y su agrupamiento en una forma ordenada. Puesto que las bases de datos proporcionan al mismo tiempo entradas y salidas para el sistema de información de toda la empresa, los usuarios de todas las áreas funcionales deben de ayudar en la creación del diseño lógico para asegurar que se identifiquen y resuelvan sus necesidades.

El diseño físico de la base de datos se inicia a partir del diseño lógico de la base de datos y lo perfecciona para consideraciones de desempeño y costos (por ejemplo, menor tiempo de respuesta, mejor espacio de almacenamiento, menores costos de operación). La persona encargada de perfeccionar el diseño físico debe de tener un conocimiento profundo de los DBMS que se emplearán para poner en práctica la base de datos.

Una de las herramientas que usan los diseñadores de las bases de datos para mostrar la relación lógica entre los datos, la constituye el modelo de datos. Un modelo de datos es un mapa o diagrama de entidades y sus relaciones.

Por lo general, el modelado de datos incluye la comprensión de un problema de negocios específico y el análisis de los datos y la información necesaria para producir una solución. Cuando se realiza a nivel de toda la organización se denomina modelado de datos de la empresa. El modelado de datos de la empresa es un método que se inicia con la investigación de los datos generales y las necesidades de información, a nivel estratégico, de la organización; después lleva a cabo un análisis de las necesidades de datos e información

más específico para las diversas áreas y departamentos funcionales dentro de la organización. Existen varios modelos que se desarrollan para ayudar a los diseñadores de bases de datos a analizar las necesidades de datos e información. Un ejemplo de dicho modelo de datos lo constituye el diagrama entidad –relación.

Los diagramas de entidad-relación (ER) utilizan símbolos gráficos básicos para mostrar la organización de los datos y las relaciones entre ellos. En estos diagramas, por lo general, se usan cuadros para señalar objetos o entidades de datos, y un símbolo en forma de rombo para mostrar la relación entre los objetos y entidades de datos.

Los diagramas de ER pueden mostrar diferentes tipos de relaciones, por ejemplo, la relación uno a uno, uno a varios, muchos a muchos. Los diagramas ER ayudan a asegurar que las relaciones entre las entidades de datos de una base de datos estén estructuradas en forma lógica para que se puedan desarrollar programas de aplicaciones que satisfagan mejor las necesidades del usuario. Además, los diagramas ER pueden usarse como documentos de referencia una vez que la base de datos se puso en marcha. Si fuera necesario hacer cambios en la base de datos, los diagramas ER pueden ayudar a diseñarlos.

Sistemas de Administración de Bases de Datos (DBMS).

La creación y puesta en práctica del correcto sistema de base de datos asegura que éste dará soporte a las actividades y objetivos de la empresa. Un sistema de administración de base de datos es un grupo de programas que se usan como una interfaz entre una base de datos y programas de aplicaciones, o entre una base de datos y el usuario. Con independencia del modelo que respalden, los DBMS comparten algunas funciones comunes como proporcionar una vista de datos para el usuario, almacenar y recuperar físicamente los datos en una base de datos, permitir la modificación de la base de datos, manipular datos y elaborar informes.

Puesto que el DBMS tiene la responsabilidad del acceso a una base de datos, uno de los primeros pasos para instalar y usar una base de datos incluye informarle a aquél la estructura lógica y física de los datos y de las relaciones entre éstos en la base de datos. A esta descripción se le conoce como esquema (como un diagrama esquemático). El esquema

puede ser parte de la base de datos o un archivo independiente. El DBMS actúa también como una interfaz del usuario al proporcionar una vista de la base de datos. Una vista del usuario es la parte de la base de datos a la que puede tener acceso el usuario. La creación de diferentes vistas del usuario requiere del desarrollo de subesquemas. Un subesquema es un archivo que contiene la descripción de un subgrupo de la base de datos y que identifica que usuarios pueden realizar modificaciones a los conjuntos de datos en ese subgrupo. Mientras que un esquema es una descripción de la totalidad de la base de datos. Un subesquema sólo muestra algunos registros y sus relaciones en la base de datos.

Normalización de una Base de Datos

Cuando se esta estructurando una base de datos, poner las columnas correctas en las tablas adecuadas puede ser una tarea algo fastidiosa. Cuando finalmente se ha completado esta tarea, puede encontrar algunos problemas de lógica en su base de datos, especialmente si no se tiene experiencia con las bases de datos relacionales. Existe un conjunto de reglas para ayudar a los diseñadores de base de datos a desarrollar una verdadera base de datos relacional sin problemas de lógica. A la aplicación de estas reglas en la estructura de su base de datos se le conoce como normalizar o normalización.

Qué es la normalización

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede.

Grados de normalización

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos

conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En la tabla 1.3-1 se describe brevemente en que consiste cada una de las reglas.

| Regla | Descripción |
|-----------------------------------|--|
| Primera Forma Normal (1FN) | Incluye la eliminación de todos los grupos repetidos. |
| Segunda Forma Normal (2FN) | Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK). |
| Tercera Forma Normal (3FN) | Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. |
| Cuarta Forma Normal (4FN) | La cuarta forma normal tiene por objetivo eliminar las dependencias multivaluadas. Una relación está en 4NF si y sólo si, en cada dependencia multivaluada $X \twoheadrightarrow Y$ no trivial, X es clave candidata. Una dependencia multivaluada $A \twoheadrightarrow B$ es trivial cuando B es parte de A. Esto sucede cuando A es un conjunto de atributos, y B es un subconjunto de A. |

Tabla 1.3-1 Descripción de las Formas Normales

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Planteamiento del Problema

La necesidad de desarrollar un sistema que permita llevar un control administrativo de la empresa surgió a raíz de una inquietud del dueño por incrementar sus ventas y de tener un mejor control de su empresa. Esta inquietud dio como resultado la creación de una aplicación que permitiera a los clientes consultar el producto a través de la red, así como la administración de la misma, incrementado así su poder adquisitivo.

Actualmente la librería no tiene sistema informático alguno el cual maneje el 100% de su administración, es decir, hoy en día maneja parte de su información con paquetería comercial, la cual ya no es suficiente para cubrir las necesidades actuales que demanda este negocio. Además no se pueden obtener reportes estadísticos, los cuales muestren información detallada de sus ventas, compras a proveedores, libros en existencia, catálogo de proveedores, precio por productos.

Otro punto que la empresa necesita cubrir, es la falta de seguridad en la confidencialidad de su información, debido a que dicha información se encuentra almacenada en documentos comunes a todo el personal, tales como papel o archivos electrónicos (Word, Excel).

Debido a que la librería no cuenta con un repositorio central de su información, la actualización de la información en tiempo real es un requisito a cubrir importante en este desarrollo. Es decir, hay veces en que el personal cuenta con información errónea, ya que a pesar de tener la información necesaria, ésta ya podría haber sido modificada por otra persona, por lo que es muy demandante la comunicación entre el personal de la librería.

Además, considerando las perspectivas de crecimiento del negocio es necesaria la utilización de una plataforma que permita compartir la información en todas las sucursales que la empresa tenga.

Descripción del problema

Se requiere diseñar un sistema que permita al usuario registrar las ventas que se generan diariamente en la librería, además de que el administrador de la librería pueda consultar la información de esas ventas. También se requiere que el sistema permita llevar un control de las cantidades de libros que se tiene en almacenamiento para un determinado título. Se podrá consultar la información referente a un libro como autor, título, editorial, resumen y precio; esta información será mostrada en pantalla al usuario, para que el usuario pueda realizar la compra del artículo. Una vez que el usuario hizo su compra, se requiere que se lleve un control de los pedidos. Por el momento no se requiere que la compra se pueda pagar con tarjeta de crédito, sólo se manejarán pagos en efectivo. El administrador de la librería podrá controlar a los empleados asignándoles permisos de usuario para poder operar en las áreas que le correspondan: ventas, pedidos, etc. Como un crecimiento posterior del sistema se pretende que más adelante, el sistema sea capaz de manejar la información referente a proveedores y compras.

En resumen se necesita diseñar un sistema que le permita al negocio resolver las siguientes tareas:

- Llevar el control de las ventas
- Llevar el control del almacén.
- Llevar el control del personal.

| | |
|------------------------------------|---|
| El problema de | <ul style="list-style-type: none"> ✓ Controlar el almacenamiento existente en bodega, de forma que se puedan servir los pedidos que se reciben. ✓ Gestionar las órdenes de compra realizadas por los clientes. ✓ Gestionar la facturación de la empresa. |
| Afecta a | <ul style="list-style-type: none"> ✓ Gerente ✓ Administrador ✓ Encargado ✓ Secretaria ✓ Cliente ✓ Cajero/vendedor |
| El impacto asociado es | Almacenar toda la información referente a las ventas y pedidos recibidos, y que ésta información esté al instante accesible y actualizada en lugares físicamente muy distantes, éste es un proceso prácticamente imposible de realizar en el caso de que no esté informatizado. |
| Una solución adecuada sería | Informatizar el proceso, usando una red local con una base de datos accesible desde los distintos nodos de la red y generar interfaces amigables y sencillas con las que se pueda acceder a dicha base de datos. |

Descripción del Producto

| | |
|-------------------------------|---|
| Para | Gerente Administrador Encargado Secretaria Cliente Cajero/vendedor |
| Quiénes | Controlan los pedidos, las ventas, las órdenes de pedido y la facturación. |
| El nombre del producto | Es una herramienta software. (SIPAL) |
| Que | Almacena la información necesaria para administrar una librería, con posibilidades de agrandarse. |
| No como | El sistema actual. |
| Nuestro producto | Permitirá controlar las distintas actividades de la librería mediante una interfaz gráfica sencilla y amigable. Además proporciona un acceso rápido y actualizado de información desde cualquier punto que tenga acceso a la base de datos. |

Para proponer una solución que realmente satisfaga las necesidades de la librería, necesitamos analizar las diversas opciones técnicas actuales. La decisión principal en este momento se enfoca en el lenguaje de programación que emplearemos para implementar la solución, así como en el servidor web y el manejador de base de datos a utilizar.

Enseguida evaluaremos algunas herramientas que podremos utilizar para dar una solución al problema definido anteriormente.

2.1 Lenguajes de programación

Estos son algunos lenguajes de programación que se pueden utilizar en cuanto a programación Web:

ASP

Es la versión previa a la tecnología ASP.NET, y por tanto, también la solución inicial de Microsoft a la programación Web. Sus siglas se corresponden con Active Server Pages (Páginas Activas de Servidor) en su versión 3, y funcionaban bajo las versiones del servidor Web de Microsoft, IIS (Internet Information Server, Servidor de Información de Internet) en sus versiones 3, 4 y 5. Este sistema se basaba en la ejecución de una serie de lenguajes de script (principalmente, VBScript y Jscript) embebidos en páginas HTML. La extensión de estas páginas es .asp (son las comúnmente llamadas páginas ASP, y que aún hoy siguen utilizándose en la programación de multitud de sitios Web).

Su proceso de desarrollo es relativamente rápido, está integrado con el modelo COM (Modelo de Objetos Componentes) de Windows y se encuentra muy extendido (ha sido uno de los más utilizados en los últimos tiempos). Sin embargo, su modelo de desarrollo es más bien funcional, y no totalmente orientado a objetos.

JAVA (JSP, SERVLETS)

Es la solución de SUN Microsystems para el desarrollo de aplicaciones Web. Consta de un conjunto de librerías y API (Interfaz de Programación de Aplicaciones) de desarrollo, que incluye un conjunto de objetos de servidor o servlets, y un sistema de programación de páginas con un modelo similar al de ASP (introduciendo código java embebido en páginas HTML), páginas JSP (Java Server Pages, Páginas de Servidor Java).

Este modelo de servlets y páginas JSP, al estar directamente basado en Java, presenta todas las ventajas de este lenguaje, como la orientación a objetos, disponibilidad de todas las librerías adicionales de Java (generación de imágenes, comunicaciones, seguridad, encriptación, etc). Podríamos decir que esta solución es más potente y completa para desarrollar aplicaciones Web que la solución de páginas ASP.

PHP

PHP corresponde a las iniciales de Personal Home Page (Procesador de Hipertexto). Este lenguaje de programación tiene una sintaxis similar a los lenguajes C y Perl, se interpreta por un servidor Web Apache bajo sistemas Unix/Linux (también han salido al mercado versiones para sistemas Windows, aunque no siempre podremos utilizar todas sus características bajo este sistema operativo).

Presenta una técnica similar a las páginas ASP de Microsoft, es decir, las páginas HTML incluyen el código PHP embebido en ellas, y al ejecutarlas, se genera código HTML dinámicamente.

Por otro lado, es preciso mencionar que soporta clases, pero no está orientado a objetos, aunque dispone de un gran número de librerías de funciones para realizar operaciones avanzadas como acceso a bases de datos, correo electrónico, transferencia de ficheros, comunicaciones, etc.

Las páginas PHP también se han extendido mucho en los últimos años (son las páginas Web que vemos con extensión .php o .phtml), aunque parece ser que se tiende a utilizar las soluciones de Java y ASP.NET en las nuevas aplicaciones.

DHTML

Sus siglas significan Dinamic HTML (HTML dinámico). Se forma de una conjunción de HTML, hojas de estilo en cascada (CSS) y lenguajes de script como JavaScript o VBScript. Estos lenguajes de script se ejecutan en la máquina cliente (en el navegador del usuario que solicita una página). Normalmente, un desarrollador Web, utilizará alguna de las tecnologías anteriores, basadas en servidor, que implementa la lógica principal y fundamental de la aplicación, en conjunción con el HTML dinámico, ya que gracias a éste, podemos conseguir efectos muy vistosos en nuestras páginas, como formateo más completo al HTML, alternancia de imágenes, capas (contenido de la página cuya posición y tamaño puede cambiar), con lo que obtenemos desplazamiento de imágenes, banners, cambios de

visibilidad en los elementos de la página (partes de la página que aparecen o desaparecen en función de los eventos o acciones del usuario), validaciones de entradas del usuario antes de enviar los datos al servidor, etc.

ASP .NET

A partir de la versión 5 de su servidor Web, IIS, Microsoft da soporte a su nueva solución de programación Web, ASP.NET, (aunque existe compatibilidad con el código ASP). ASP.NET mejora en los lenguajes y herramientas de desarrollo, asume el modelo de programación orientada a objetos, mejora el rendimiento general del sistema, etc.

Como podemos observar, no existe una solución única para resolver todos los problemas. Cada lenguaje web ofrece diferentes características al usuario y debemos analizarlas, y definir cuál de las soluciones se adecua mejor a nuestro problema, ver la siguiente Tabla 2.2.1-1.

| <i>Lenguaje web</i> | <i>Compilado o interpretado</i> | <i>Orientado a objetos</i> | <i>Lenguaje de programación</i> | <i>Multiplataforma</i> |
|----------------------------|--|-----------------------------------|--|-------------------------------|
| ASP | Interpretado | No | VBScript, Jscript | No |
| JSP | Compilado | Si | Java | Si |
| PHP | Interpretado | Parcialmente | Php, sintaxis similar a la del lenguaje c. | No |
| DHTML | Interpretado | No | Javascript, css | No |
| ASP .NET | Compilado | Si | C#, VB .Net | Si |

Tabla 2.2.1-1 Comparación de los distintos lenguajes web.

Basándonos en la tabla 2.2.1-1 comparativa de los distintos lenguajes web. Se eligió la plataforma de desarrollo .NET por las siguientes características:

- ASP. NET soporta varios lenguajes de programación, los cuales, se pueden usar para construir sitios web(C#, VB.Net)
- Las páginas ASP son compiladas no interpretadas. En lugar de leer e interpretar el código cada vez que una página dinámica es solicitada, ASP.NET compila las páginas dinámicas en eficientes archivos binarios, los cuales el servidor puede ejecutar mas rápidamente.
- ASP. NET tiene acceso completo a la funcionalidad del .NET Framework. Soporte para XML, Servicios Web, interacción con base de datos, e-mail, expresiones regulares, y muchas otras tecnologías están incorporadas correctamente dentro de .NET, lo cual ahorra tiempo.
- ASP. NET permite separar el código del lado del servidor de la capa de HTML. Cuando se esta trabajando con un equipo compuesto de programadores y diseñadores, esta separación es de gran ayuda, ya que permite a los programadores modificar el código del lado del servidor sin afectar el HTML creado por los diseñadores y viceversa.
- Debido a que soporta lenguajes de programación como C#, se puede hacer uso de las ventajas de la programación orientada a objetos.
- A diferencia de jsp, ASP. NET soporta la interoperabilidad de lenguajes.
- C# incluye delegados, propiedades, indexadores y eventos como elementos del lenguaje, racionaliza la creación de componentes y ofrece una cantidad significativa de verificación de errores en tiempo de ejecución, seguridad y gestión de la ejecución.

Plataforma .NET

Es una tecnología en la cual los Servicios Web son un medio que permitirán a distintas tecnologías inter-operar entre sí, así como conectar diversos sistemas operativos, dispositivos, información y usuarios dando a los desarrolladores las herramientas y

tecnologías necesarias para desarrollar soluciones de negocios de manera rápida sin importar que involucren diversos medios y tecnologías. En la plataforma .NET el software es mas concretamente de servicio y se aboca a construir, instalar, consumir, integrar o agregar de manera federada servicios que pueden ser accedidos mediante Internet, la idea central es que un usuario de Internet con un explorador pueda acceder a contenidos, no sólo en forma de texto, imágenes o sonido, sino también pueda hacer uso de Servicios Web, los cuales se utilizan como bloques de construcción en este nuevo modelo de computación distribuida en Internet. El punto central es que los usuarios deben poder acceder a la funcionalidad que los servicios Web proveen desde cualquier dispositivo, sistema operativo y ubicación geográfica.

Componentes de la plataforma .NET

Los componentes principales de la plataforma .NET son:

- **Clientes Inteligentes:** Son dispositivos muy variados. Lo que los hace inteligentes es su capacidad para hacer uso de servicios Web. Pueden ser computadoras de bolsillo, personales, portátiles, handhelds (de mano), teléfonos inteligentes e incluso consolas de juegos como XBOX.
- **Servidores:** Proveen de la infraestructura para implementar el modelo de computación distribuida en Internet. Son sistemas operativos y de aplicación.
- **Servicios Web basados en XML (eXtensible Markup Language) (Lenguaje ampliable de composición):** Son los bloques de construcción de la tercera generación de Internet. Permiten a las aplicaciones compartir datos y están basados en XML.
- **Herramientas de desarrollo:** Visual Studio .NET y el .NET Framework. Ambos permiten al desarrollador hacer servicios Web basados en XML además de otro tipo de aplicaciones. El .NET Framework viene incorporado directamente en la nueva línea de sistemas operativos Windows .NET. Para los dispositivos móviles se llama .NET Compact Framework.

.NET ofrece una potente plataforma para el desarrollo de aplicaciones Internet que pueden ejecutarse sobre cualquier dispositivo .NET. El núcleo de esta plataforma es el CLR (Common Language Runtime (Entorno de Ejecución multilenguaje)). El CLR proporciona una forma independiente del dispositivo y del lenguaje para expresar los datos y el comportamiento de las aplicaciones.

Lenguaje C#

C# es un nuevo lenguaje de propósito general y estandarizado con el organismo ECMA, que aprovecha al máximo las características de la plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella.

La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que su intención es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic.

A diferencia de Java, en C# todo es un objeto, así ganamos en consistencia y además los tipos básicos (int, float,..) están optimizados para que su uso no repercuta en la eficiencia del programa.

También dispone de un Garbage collector (recolector de basura) que facilita y optimiza la gestión de la memoria, así es posible evitar el consumo en exceso de memoria por errores de programación y facilita la vida al desarrollador que ya no es necesario que controle esos aspectos (en C++ encontramos estos problemas).

Como en la mayoría de lenguajes orientados a objetos dispone de mecanismos de encapsulamiento, herencia (simple), implementación de interfaces (simples o múltiples) y polimorfismo. Además se ofrece la posibilidad de utilizar delegados, que no son más que

apuntadores a funciones, y eventos muy útiles para realizar notificaciones entre clases o componentes.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo.

2.2 Manejadores de Bases de Datos

El mercado de manejadores de bases de datos es bastante grande y ofrece demasiadas alternativas a la hora de elegir un software en que confiar, en el momento de tomar una decisión con respecto a por cual herramienta inclinarnos, cuál es la optima, cuál me ofrece mayores garantías en mi desarrollo específico, qué detalles de implementación debemos tener en cuenta para elegir nuestro sistema de gestión de bases de datos, se convierte en una gran preocupación y responsabilidad el conocer las características, ventajas y desventajas, no desconocemos que las herramientas constituyen un aspecto fundamental a la hora de desarrollar un proyecto o una implementación, las características de los proyectos, de las compañías o las necesidades hacen prioritario que estas herramientas se ajusten a esos requerimientos específicos, por estas razones se debe profundizar e investigar las diferentes alternativas que se tienen al alcance, evitando inconvenientes posteriores como son, pérdida de tiempo, pérdida de dinero o aún más grave comprometer la credibilidad profesional al avalar un concepto técnico sin el suficiente soporte y conocimiento.

Es por tal motivo, conocer muy de fondo las diferentes plataformas o manejadores de bases de datos para poder optar por la más adecuada para ser implantada, en la librería, como lo son MySQL, SQL Server, ORACLE e INFORMIX, los cuales ofrecen diferentes formas de licenciamiento tanto libre como comercial.

MySQL

Es la base de datos relacional de código libre más usada en el mundo. Su arquitectura, la reutilización del código, produce unas características funcionales que permite una administración de BD incondicionalmente inigualable, dado por su velocidad, conectividad, y facilidad de despliegue. Unas de las características de MySQL son:

- Costo. Mysql es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico. Es software de fuente abierta. Una de las ventajas que ofrece hoy en día las nuevas tecnologías de desarrollo es poder desarrollar aplicaciones cuyo objetivo es acceder a dicha información desde cualquier parte del mundo utilizando una red como Internet.
- Sistema de administración relacional de bases de datos. Esto permite velocidad y flexibilidad.
- Facilidad de Uso. Existen diferentes maneras de extraer la información almacenada en una base de datos, por ejemplo, un ingeniero lo puede realizar por medio de instrucciones SQL, los usuarios lo pueden realizar por medio de aplicaciones desarrolladas dentro de una organización o pueden extraer la información con herramientas que existen en el mercado, todo esto con dos valores agregados, Velocidad y precisión.
- Velocidad en la utilización de joins y procesos de optimización.
- Procesos MultiHilo. Capacidad de trabajar servidores con varios procesadores.
- Trabaja bajo diferentes plataformas: AIX 4x 5x, Amiga, BSDI, Digital Unix 4x, FreeBSD 2x 3x 4x, HP-UX 10.20 11x, Linux 2x, Mac OS, NetBSD, Novell NetWare 6.0, OpenBSD 2.5, OS/2, SCO OpenServer, SCO UnixWare 7.1.x, SGI Irix 6.x, Solaris 2.5, SunOS 4.x, Tru64 Unix y Windows 9x, Me, NT, 2000, XP, 2003.
- Manejo de la memoria a través de manejo del buffer y caché.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

INFORMIX

Informix ha sido un líder en tecnología de BD relacionales. Arquitectura OLP, lenguajes de cuarta generación, entre otros.

Algunas características de Informix son:

- Costo. Informix es software propietario por lo cual tienes que pagar una licencia para su uso.

- Capacidad de optimizar el rendimiento de una forma inteligente gracias a la ejecución en paralelo de las principales operaciones de la base de datos (particularmente, consulta o compactación de índices, y también salvaguarda y recuperación de datos).
- Ejecución de múltiples hilos y procesadores virtuales que trabajan en paralelo, atienden a usuarios concurrentes, esto consigue que todo el rendimiento se aproveche de acuerdo a su prioridad.
- El objetivo es conseguir escalabilidad total y lineal.
- Tiene la capacidad de relación de datos en múltiples lugares físicos (dentro de la citada arquitectura DSA), suponiendo un eficaz sistema de distribución y compartir datos dentro de un mismo sistema organizativo.
- Incluye de forma integrada el servidor Netscape FaxTrap, para soporte y conectividad de aplicaciones Web e Intranet. Ofrece servicios escalables Web/Inet de alto rendimiento, entorno de desarrollo basado en Java/JavaScript seguridad basada en SSL con autenticación.
- Web Integration Option, facilita la generación de Aplicaciones orientas a Internet. Se trata de un Runtime que enlaza al servidor *Web* y al servidor de base de datos de forma que las aplicaciones puedan incorporar de manera sencilla los datos suministrados dinámicamente por Informix. También es posible incorporar una consulta SQL dentro de una página Web.
- En cuanto a la seguridad e integridad de los datos, incluye duplicación, espejo de disco, volcado de seguridad y restauración de alta velocidad.
- Informix utiliza los mismos usuarios de Linux (los que creas con adduser), simplificando la administración.

ORACLE

Es manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas. Algunas características son:

- Costo. Oracle es software propietario por lo cual tienes que pagar una licencia para su uso.
- Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.
- Corre automáticamente en más de 80 arquitectura de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código.
- Se puede almacenar, en forma independiente, funciones y procedimientos sin tener que escribirlos repetidamente para cada forma, y pudiendo compilarlos independientemente de las formas que lo usen. Pero, además, las funciones y procedimientos se pueden agrupar en un paquete para compartir definiciones, variables globales, constantes, cursores y excepciones, así como garantizar y revocar los permisos a nivel de paquete.
- Oracle posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Estos porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.
- Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.
- Oracle provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso.
- En Oracle, se definen los usuarios dentro la base de datos (gestión interna de usuarios).

SQL Server

Es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

SQL Server disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo.

Las otras características tales como bloqueo a nivel de línea dinámica, el paralelismo intra-query, query distribuido, y mejoras para las bases de datos muy grandes (VLDB) hacen que

el SQL Server sea la elección ideal para sistemas que procesan las transacciones en tiempo real (OLTP) de alta tecnología.

Características de SQL Server:

- Costo. Oracle es software propietario por lo cual tienes que pagar una licencia para su uso.
- Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- Multithread real para mejor rendimiento transaccional y escalabilidad.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows. Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.
- Soporta grandes bases de datos.
- Procedimientos almacenados para generar páginas HTML o actualizar datos en plantillas Web.
- Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Orientada al desarrollo: Visual Basic, Visual C, C#, Visual J, Visual Interdev y muchas otras herramientas son compatibles con Microsoft SQL Server.
- Mínimo coste de Propiedad: La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor coste.
- Disponibilidad, fiabilidad y tolerancia a fallos
- Llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Configuración de replicación gráfica y características de gestión.
- Compatibilidad con Microsoft Internet Information Server y otros servidores Web populares.
- Seguridad. Password y encriptación de datos en red para mejorar la seguridad.

En la tabla siguiente 2.2-1 podemos observar algunas características de los diferentes manejadores de bases datos.

| Nombre del sgbd | Facilidad de uso | Escalabilidad | Replicación | Seguridad | Plataforma |
|------------------------|--|----------------------|--|---|--|
| Informix | Amigable | Dinámica | Lo permite, limitadamente, ya que es, dependiente de la PC | De acuerdo con C2 y OLTP | Windows, unix, entre otros. |
| Mysql | Excelente | Dinámica | Independiente del hardware de la PC. | Excelente, ya que resguarda cada bd y por la accesibilidad en el código fuente se puede ajustar a las necesidades del DBA, ya que está escrito en C/C++ | Windows, unix, linux, novell, solaris, entre otros. |
| Oracle | Bastante complejo para aquellos que no estén bastante familiarizados con las bases de datos, su alto rendimiento es directamente proporcional a su nivel de complejidad, esto es lógico, ya que toda aquella aplicación sobre la que se desea tener un control total suele estar | Multithreaded | Permitido, es independiente de la PC. | Mas flexible en lo niveles C2 y B2 | Windows, unix, entre sus otras variedades al igual que los anteriores no le hace ningún feo como SQL Server. |

| Nombre del sgbd | Facilidad de uso | Escalabilidad | Replicación | Seguridad | Plataforma |
|------------------------|---|--|--|---|-------------------|
| | bastante poco adaptada al usuario tradicional | | | | |
| Sqlserver | Fácil de manejar pero los expertos recomiendan usarlas sólo en caso de que la bd sea local. | Se adapta a la “empresa”, permitiendo la actualización de aplicación y concentración de datos. | Respalda toda el contenido de la base de datos en un fichero *.sql, y solo lo puede abrir SQL Server | Maneja un nombre de usuario y un password | Windows |

Tabla 2.2-1 Lista de manejadores de Bases de Datos

Para el desarrollo de nuestra aplicación seleccionamos SQL Server como manejador de la base de datos ya que el costo de licencias para su uso es el más bajo entre los productos propietarios, y una ventaja es que cuenta con soporte técnico garantizado al cual acudir si los manuales y ayuda en línea no son suficientes en comparación con el software libre. Otra razón es que su administración es fácil, no se requiere de gente especializada para su uso, ya que cuenta con una interfaz gráfica amigable; es la base de datos más confiable. Cubre la necesidad de almacenamiento de una gran cantidad de datos y seguridad, que facilite la automatización de sucursales. También otra ventaja es la sencillez de la instalación en comparación con los demás manejadores de BD. SQL Server tiene mejor rendimiento sobre Internet. Una característica importante por la cual se seleccionó fue que se integra en la nueva plataforma .NET.

2.3 Web Server

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso. En este artículo nos referiremos siempre a la aplicación.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear `www.unam.mx` en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio web clásico podemos disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse mas lenguajes mediante el uso de plugins.
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

En la siguiente tabla 2.3-1 se enlistan las características de los principales tipos de servidores del mercado actual:

| Nombre Web Server | Plataformas en las que se utiliza | Características Principales | Desventajas |
|--------------------------|--|---|--|
| Apache | Unix, Windows | Mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. | Falta de una interfaz gráfica que ayude en su configuración. |
| IIS | Windows | Servicios que ofrece: FTP, SMTP, NNTP y HTTP/HTTPS. Convierte a un computador en un servidor de internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web). | Windows XP Profesional incluye una versión limitada de IIS. |

| Nombre Web Server | Plataformas en las que se utiliza | Características Principales | Desventajas |
|--------------------------|--|--|---|
| LIGHTTPD | GNU/Linux y otros sistemas operativos tipo UNIX. | Diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores. Por todo lo que ofrece, lighttpd es apropiado para cualquier servidor que tenga problemas de carga. | No se adapta a plataformas soportadas por ambientes Windows |
| THHTTPD | | Simple, pequeño, portátil, rápido, y seguro, ya que utiliza los requerimientos mínimos de un servidor http. | Solo tiene la posibilidad de utilizar servidores estándar (APACHE). |

Tabla 2.3-1 Lista de Webservers

Por supuesto que existe otra gran cantidad de servidores web además de estos que se mencionaron, sin embargo, éstos son los principales servidores que podemos encontrar en el mercado.

Para nuestro desarrollo se usará el Servidor Web IIS debido a que se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl, además de que como se mencionaba en las características de la tabla las computadoras que tienen este servicio instalado pueden publicar páginas web tanto local como remotamente; también los servicios de Microsoft Internet Information Server (IIS) proporcionan capacidades de servidor Web integrado,

confiable, escalable, seguro y administrable en una intranet, una extranet o en Internet lo que nos proporciona los siguientes tres aspectos importantes que necesitamos en nuestro desarrollo:

- Confiabilidad y escalabilidad
- Seguridad y capacidad de administración
- Mejor desarrollo y compatibilidad internacional

CAPÍTULO 3

ANÁLISIS DE REQUISITOS (INFORMACIÓN RELACIONADA CON LA LIBRERÍA)

La librería requiere de una estructura con la cual dé un servicio adecuado al público que a ella asiste, así como tener un sistema que le permita internamente registrar sus operaciones con la finalidad de obtener un reflejo de la situación de la empresa para tomar decisiones que conduzcan a un mejor funcionamiento de la misma.

Las actividades que se realizan en la librería son:

- Ventas
- Pedidos
- Consultas
- Reportes

3.1 Estructura actual de la librería.

Para analizar los requisitos debemos evaluar la estructura actual de la librería, de esta manera se podrá dar una solución o propuesta para que el funcionamiento del sistema sea óptimo, para posteriormente pasar al diseño e implementación del mismo.

Como se mencionó en el capítulo anterior, actualmente la librería no tiene sistema informático alguno el cual maneje el 100% de su administración, es decir, hoy en día maneja parte de su información con paquetería comercial, la cual ya no es suficiente para cubrir las necesidades actuales que demanda este negocio.

En cuanto a la parte física, la librería esta estructurada de la siguiente manera:

- Una matriz
- Una sucursal
- Una bodega

El personal con el que cuenta es el siguiente:

- Un Gerente
- Un administrador
- Una secretaria
- Un encargado de almacén
- Dos cajeros
- Dos vendedores

Se cuenta con equipos Pentium IV con 512 en RAM y 160 GB de disco duro, Windows XP. Las computadoras tienen acceso a Internet mediante una conexión DSL.

3.2 Requerimientos del negocio.

Propósito

El propósito de este apartado es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de administración de una librería. Éste se centra en la funcionalidad requerida en el proyecto y los usuarios finales.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales, en el capítulo 4.

Descripción General

Desarrollar un sistema web que permita realizar las operaciones diarias de manera sencilla y clara, en la administración de la librería.

Objetivos Estratégicos de Negocio

Se pretende ir cambiando el sistema manual de captura actual por un sistema de captura digital y así reducir al mínimo los tiempos de captura y búsqueda en la librería, así como, de agilizar el proceso de ventas y entrega de mercancía.

Alcance del Proyecto

El sistema permitirá a los encargados de la librería controlar todo lo relativo a la administración y gestión de los productos.

La lista a continuación describe el alcance del proyecto en términos de requerimientos de negocio de alto nivel.

1. Administración de catálogos
 - 1.1. De Usuarios
 - 1.2. De Clientes
 - 1.3. De Libros
 - 1.4. De Proveedores
 - 1.5. De Categorías de libros
2. Consultas y reportes
 - 2.1. De Libros
 - 2.1.1. Clasificación por categoría
 - 2.1.2. Asignación de precios
 - 2.2. De ventas
 - 2.2.1. Por ventas de libros (decrementar existencias)
 - 2.2.2. Libros vendidos (histórico)
 - 2.3. De Proveedores
 - 2.4. De Clientes

El sistema se realizará en 3 capas.

- Front END: Donde estará comprendida la interfaz grafica, mediante la cual el usuario interactuará con el sistema.
- Negocio: Donde estará comprendido los módulos del sistema, los cuales definen el funcionamiento del negocio.
- Datos: Donde estará comprendido los módulos del sistema, los cuales interactúan con el motor de base de datos.

Descripción de Usuarios y tareas principales.

A continuación se describen las principales responsabilidades de cada uno de los actores en la librería, de acuerdo con los roles que se desempeñan en RUP.

| Usuario | Descripción |
|------------------------|--|
| Gerente | Se encarga de Administrar el Catálogo de Usuarios. |
| Administrador | Se encarga del manejo del sistema a nivel de root o superusuario. Puede controlar la administración de los siguientes catálogos: Usuario, Proveedores, Categorías, Libros; además de consultar la clasificación de libros por categoría. |
| Encargado | Controla la administración de catálogos de libros y de categorías. Asigna precios a los libros, y Clasifica libros por categoría. |
| Secretaria | Administra catálogo de proveedores. |
| Cliente | Se refiere a la persona que podrá realizar consultas de libros, |
| Cajero/vendedor | Se encarga de llevar a cabo el proceso de ventas de libros. También realiza el cierre de caja al término de su turno y al inicio de su turno abre la caja. |

Tabla 3.2.-1 Usuarios y tareas

Diagrama de contexto.

Define el ámbito del sistema (qué queda fuera y qué queda dentro del sistema). Dicho diagrama describe al sistema en su conjunto, es decir, proporciona una descripción general de la actividad de la institución. Ver figura 3.2-1.

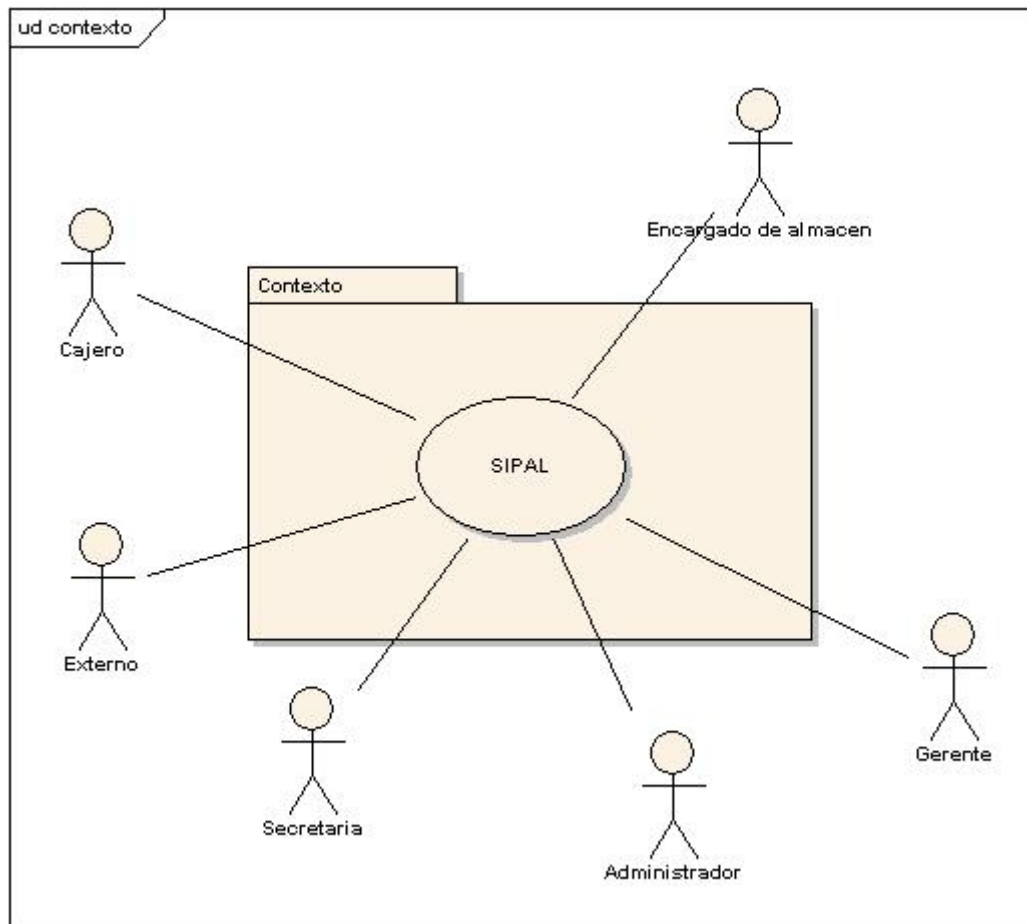


Figura 3.2.-1 Diagrama de contexto

A continuación la tabla 3.2-2 muestra las tareas principales de cada uno de los actores.

| Actor | Tareas principales |
|----------------------------------|----------------------------------|
| Cajero | Procesar venta |
| | Abrir caja |
| | Cerrar caja |
| Administrador del sistema | Gestionar las tablas del sistema |
| | Añadir usuarios |
| | Eliminar usuarios |
| | Gestionar seguridad |
| | Consultar ventas |
| | Consultar pedidos |

| Actor | Tareas principales |
|-----------------------------|---|
| Encargado de Almacén | Dar de baja material. |
| | Control de entradas y salidas de mercancía. |
| Cliente | Obtener información de libros en pantalla. |
| | Realizar pedido |

Tabla 3.2.-2 Actores y tareas

Requerimientos Funcionales

Son las cosas que el sistema puede hacer, su funcionalidad.

- El usuario del sistema podrá actualizar diariamente los movimientos realizados a las actividades antes mencionadas.
- Los usuarios podrán realizar una búsqueda de información de forma rápida y sencilla.
- El sistema brindará la capacidad de realizar modificaciones a la información previamente capturada.
- El sistema asignará niveles de seguridad para que la información sólo sea modificada por el usuario competente para esa actividad.
- El usuario tendrá la posibilidad de crear reportes.
- El sistema notificará el fin de vigencias de los pedidos por medio de un reporte semanal.
- Se deberá tener especial cuidado en que el sistema de captura sea lo más ágil posible

Requerimientos No Funcionales

Representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de seguridad, eficiencia, portabilidad, reutilización, entorno, desarrollo, disponibilidad, usabilidad, etcétera.

Factibilidad del Sistema

El sistema debe analizarse desde tres perspectivas:

- Factibilidad técnica. Los recursos disponibles como son el software, hardware y el personal capacitado son suficientes para el desarrollo del sistema, el proyecto puede realizarse.
- Factibilidad económica. Los beneficios del sistema en la mejora de la librería hace factible el proyecto.
- Factibilidad operativa. Los futuros usuarios no ofrecerán resistencia a llevar el control de la librería por medio de este sistema.

Requisitos de Facilidad de uso y Aprendizaje

- Facilidad de Uso. No se va a descuidar proporcionar un sencillo funcionamiento y manejo del sistema. Además el sistema ofrecerá ayuda al usuario cuando esté trabajando.
- Facilidad de Aprendizaje. El Sistema debe ser fácil de aprender para todo tipo de usuario sin que haya recibido un entrenamiento previo.

Requisitos de Funcionamiento

- Requisitos de Velocidad. El sistema responderá en tiempo real a las peticiones del usuario. La velocidad no es un factor primordial aunque el sistema no debe ser demasiado lento.
- Requisitos de Seguridad Crítica. Los datos personales que muestra el sistema están restringidos al administrador de sucursal y ningún dato de otro grupo puede ser visto a menos que se posea el rol de administrador de sucursal.
- Requisitos de Precisión. La clave de identificación contendrá como mínimo 6 caracteres.

- Requisitos de fiabilidad y disponibilidad. La disponibilidad es una responsabilidad del administrador del sistema global que debe asegurar que los componentes estén funcionando correctamente. El servidor es sobre el que recae mayor responsabilidad ya que sin él, el sistema no estará disponible. En principio, supondremos que el servidor estará en funcionamiento todos los días del año ininterrumpidamente.
- Requisitos de Capacidad. El sistema deberá asegurar capacidad de almacenamiento para todos los documentos de los distintos módulos que es capaz de asistir. No obstante, la información irá centralizada en el servidor, por lo que la gestión de la memoria no será un problema.
- Requisitos de Escalabilidad. El funcionamiento del sistema será el mismo aunque exista un número indeterminado de archivos. Así mismo, podrán almacenarse archivos de años anteriores.

Requisitos Operacionales

- Entorno Físico. Hay características y facilidades ofrecidas por el servidor como son espacio en memoria, velocidad de procesamiento, acceso a los datos a través de Internet, etcétera. Además debe ser utilizado usando cualquier dispositivo con acceso a Internet.
- Entorno Tecnológico. Servidor Windows proporciona como gestor de base de datos SQL Server, como servidor de páginas web IIS y un generador de contenido web ASP.NET.
- Aplicaciones Relacionadas. El diseño de este sistema garantiza la posibilidad de interaccionar con cualquier navegador HTML.
- Soporte. Con el sistema se entregará un manual detallado de uso.

Requisitos de Mantenimiento y Portabilidad

- ¿Cuál es la dificultad de mantenimiento de este sistema? Al estar basado en la obtención de datos de un servidor, el mantenimiento se resume en la correcta gestión del servidor.
- Requisitos de Portabilidad. Se espera que el sistema pueda ejecutar sin problema bajo plataformas Windows. Se podrá ejecutar en servidores web o cualquier estación que tenga acceso a Internet.

Requisitos de Seguridad

- ¿El sistema es fiable? Todos los usuarios al conectarse se deberán identificar y el sistema pedirá una contraseña. Una vez conectados, en función de qué persona lo haga, tendrá unos accesos u otros. Por ejemplo, El administrador tendrá acceso a todo mientras que el cajero no.
- Requisitos de Integridad de Ficheros. Ninguna persona ajena podrá modificar los datos de nuestro sistema. Garantizamos su total integridad.

Requisitos de Documentación

- Manual de Usuario. Contendrá información de cómo emplear el producto y un apéndice técnico orientado al administrador.
- Ayuda. Se proporcionará ayuda en el manual de usuario..
- Guías de Instalación y Configuración. Evitaremos que el usuario final necesite realizar instalación alguna, para ello instalaremos el producto en el servidor.
- Empaquetado y Etiquetado. Presentación del producto en un CD.

3.3 Pantallas de Interfaz de Usuario-Sistema (Vistas).

En esta sección se enlistan las pantallas que serán desarrolladas para la creación del sistema descrito anteriormente.

3.3.1 Pantalla de inicio de sesión

Para poder ingresar a la aplicación el usuario deberá firmarse en una pantalla la cual le solicitará dos datos: usuario y contraseña. Lo anterior se ejemplifica con la pantalla de la figura 3.3.1-1



Figura 3.3.1 -1 Pantalla de inicio de sesión

3.3.2 Pantalla de catálogo de usuarios de la librería

El usuario podrá tener acceso a consultar la información de proveedores, usuarios de la librería así como información del personal que labora dentro de ésta, siempre y cuando tenga permisos para realizar dicha consulta.

El prototipo de la pantalla se observa en la figura 3.3.2 -1.

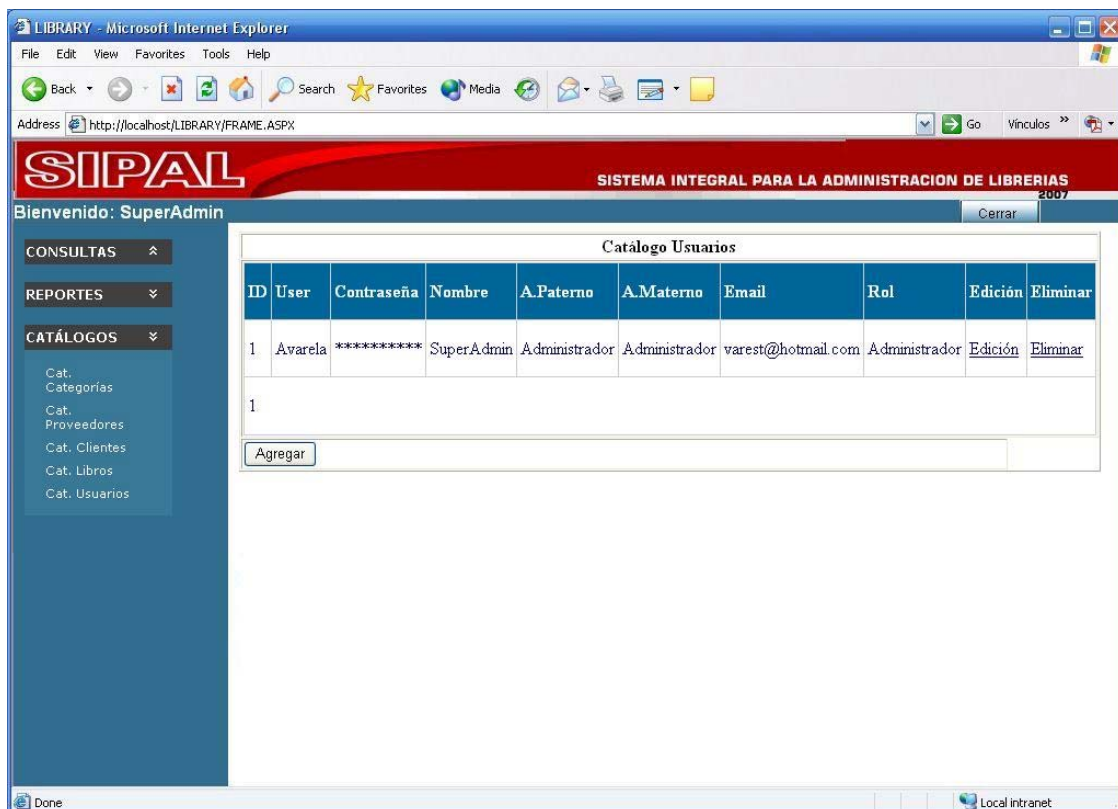


Figura 3.3.2 -1 Catálogo de usuarios de la librería.

3.3.3 Pantalla de consulta de información de libros

Se dispondrá de una pantalla que muestre el catálogo de libros con los que cuenta la librería. En la siguiente figura 3.3.3 -1 se muestra la imagen de la consulta de los libros.

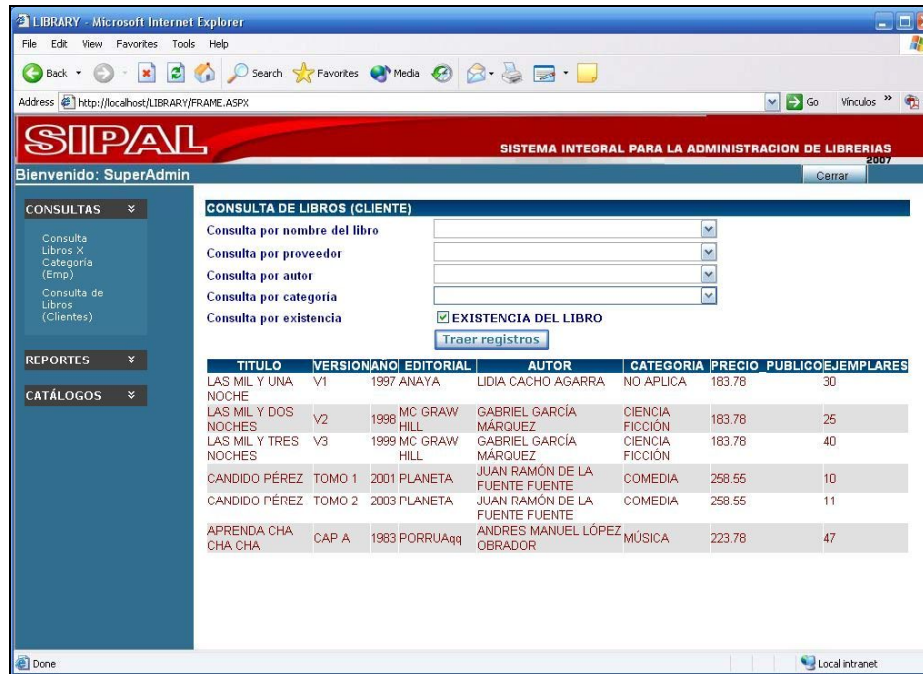


Figura 3.3.3 -1 Catálogo de libros

CAPÍTULO 4

DISEÑO

4.1 Especificación de la arquitectura.

Modelo-Vista-Controlador (Model-View-Controller MVC)

El sistema se divide en tres partes: procesamiento, entradas y salidas.

- Modelo – encapsula los datos y la funcionalidad de la aplicación.
- Vista – despliega la información contenida en el modelo (pueden existir varias vistas).
- Controlador – está asociado a cada vista, recibe entradas que traduce en invocaciones de métodos del Modelo o de Vista. El usuario interactúa con el sistema solamente vía controladores.

Para poder usar este modelo en un entorno Web, se tienen que hacer algunas modificaciones. La razón fundamental es que en un servidor http no se pueden notificar los cambios en el modelo a la vista para que ésta se actualice, ya que es el navegador quien debe hacer una nueva petición al servidor (e.g. petición de recarga de la página), para comprobar si ha cambiado el estado del modelo. También hay que tener en cuenta que en general la vista usa distintas tecnologías (HTML/JavaScript) que el modelo-controlador (en nuestro caso ASP.NET).

Esto lleva a la definición de un modelo MVC específico para la red conocido como modelo MVC 2, que tiene la estructura descrita en la figura 4.1-1. En este esquema cuando el usuario realiza una acción sobre el interfaz en la parte del cliente (navegador Web) que origina una nueva petición al servidor, ésta es procesada por el controlador. El controlador, en función de la petición que le llega, puede realizar alguna modificación sobre el modelo de datos, y redireccionar a la página adecuada; desde esta página se puede realizar alguna consulta sobre el estado del modelo de datos para completar la respuesta, y el resultado se envía al cliente; de esta forma se cierra el ciclo de operación.

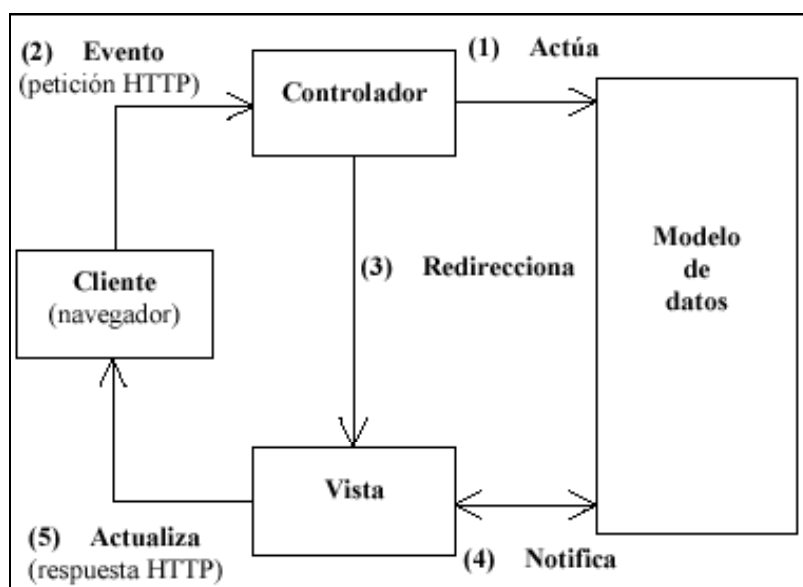


Figura 4.1-1 MVC 2 Resumen del Documento

Propósito

El propósito de este documento es la de brindar información sobre la arquitectura empleada en el Diseño y Codificación de la aplicación de generación del sistema SIPAL (SISTEMA INTEGRAL PARA LA ADMINISTRACIÓN DE UNA LIBRERÍA).

La audiencia para este documento es la tabla 4.1-1:

| <i>Audiencia</i> | <i>Propósito</i> |
|--|--|
| Integrantes y empleados de la librería donde será montado el sistema. | Conocer la arquitectura de la aplicación SIPAL |

Tabla 4.1-1 Tabla de audiencia

Tabla de Revisión

La siguiente tabla 4.1-2 lista las revisiones hechas a este documento. Se usa para describir los cambios y adiciones cada vez que este documento es actualizado. La descripción deberá incluir tanto detalle como sea posible, así como los revisores que solicitaron los cambios.

| <i>Fecha</i> | <i>Autor</i> | <i>Descripción de Cambios</i> |
|--------------------------|---------------------|--------------------------------------|
| Noviembre de 2006 | Usuario de Cambios | Creación |

Tabla 4.1-2 Revisiones

Descripción

La aplicación del sistema SIPAL consiste en un sistema que administre una librería, principalmente en cuestión del inventario de libros con que cuenta.

Así también la administración de qué proveedores son los que surten la librería, qué clientes son los que han comprado en la librería, además de reportes por búsquedas selectivas donde podrán filtrarse reportes como:

- Autor del libro
- Editorial
- Categoría del libro

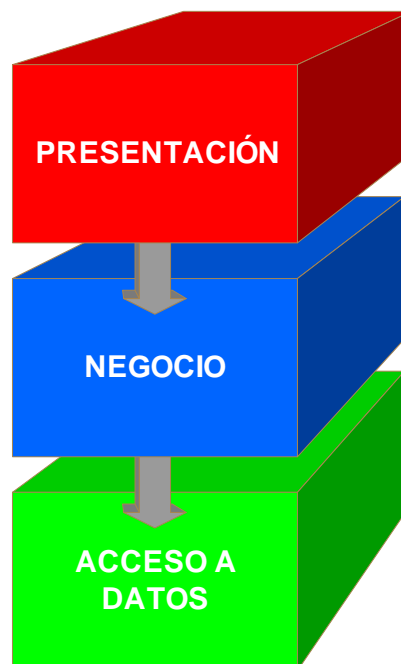


Figura 4.1-2 Arquitectura de 3 capas

Descripción de la Arquitectura del Sistema

La aplicación de Administración y recepción de mensajes utiliza una arquitectura de 3 capas (Presentación, Negocio y Acceso a datos), como se muestra en la figura 4.1-2. Algunas de las capas se concentran en un proyecto aparte, integrándose a través de referencias.

Fases del Proyecto

| Versión | Descripción |
|----------------|--|
| 1.0.0 | Versión base del Sistema Integral Para la Administración de una Librería |

Diseño de la Arquitectura del Sistema

Modelo de la Aplicación

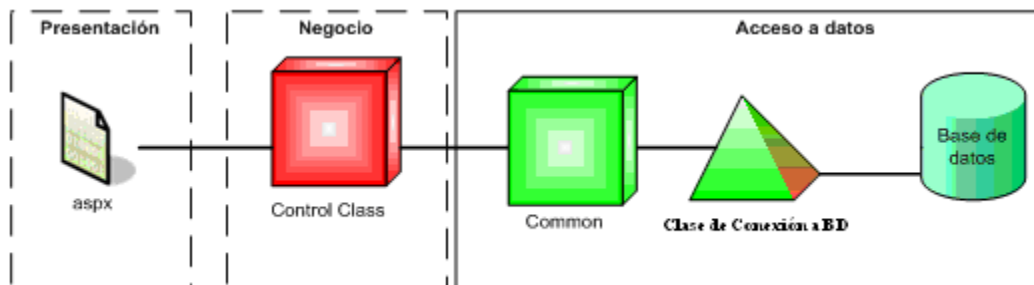


Figura 4.1-3 Modelo de la Aplicación

Capa de Presentación

Esta capa contempla las páginas HTML, aspx, imágenes, controles web, etc. Son las páginas que observa el usuario y su principal funcionalidad es la de solicitar o mostrar información; también en esta capa se realizan ciertas validaciones de la captura de información.

La capa de presentación existe en el proyecto Web Sistema Integral Para la Administración de una Librería. Ver figura 4.1-3

Capa de Negocio

Esta capa contiene un conjunto de clases de *Control* que contienen las reglas de negocio que contempla la aplicación. Aquí se realizan validaciones más avanzadas, se aplican reglas sobre ciertas funciones del negocio, transacciones, manejo de errores, etc. Ver figura 4.1-3

Capa de Acceso a datos

Esta capa contiene una serie de clases generales por entidad que contienen las principales funciones de acceso a datos del sistema.

Algunas de las funciones de datos más importantes, están concentradas en *stored procedures* que se encuentran en la base de datos de la aplicación, con lo que se facilita la creación y modificación de ciertos módulos como son las consultas de Proveedores y clientes. Ver figura 4.1-3

4.2 Diagramas de casos de uso.

Los casos de uso especifican un comportamiento deseado, no imponen como se llevará a cabo ese comportamiento. Describen un conjunto de secuencias, donde cada secuencia representa la interacción de los elementos externos al sistema (sus actores) con el propio sistema (y sus abstracciones clave)

Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. Son importantes para visualizar, especificar y documentar el comportamiento de un elemento; facilitan que los sistemas, subsistemas o clases sean abordables y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado.

A continuación presentamos los diagramas de casos de uso.

La figura 4.2-1 Representa el Diagrama de casos de uso general. Este diagrama conforma a todos los actores actuando juntos, posteriormente se mostrarán individualmente.

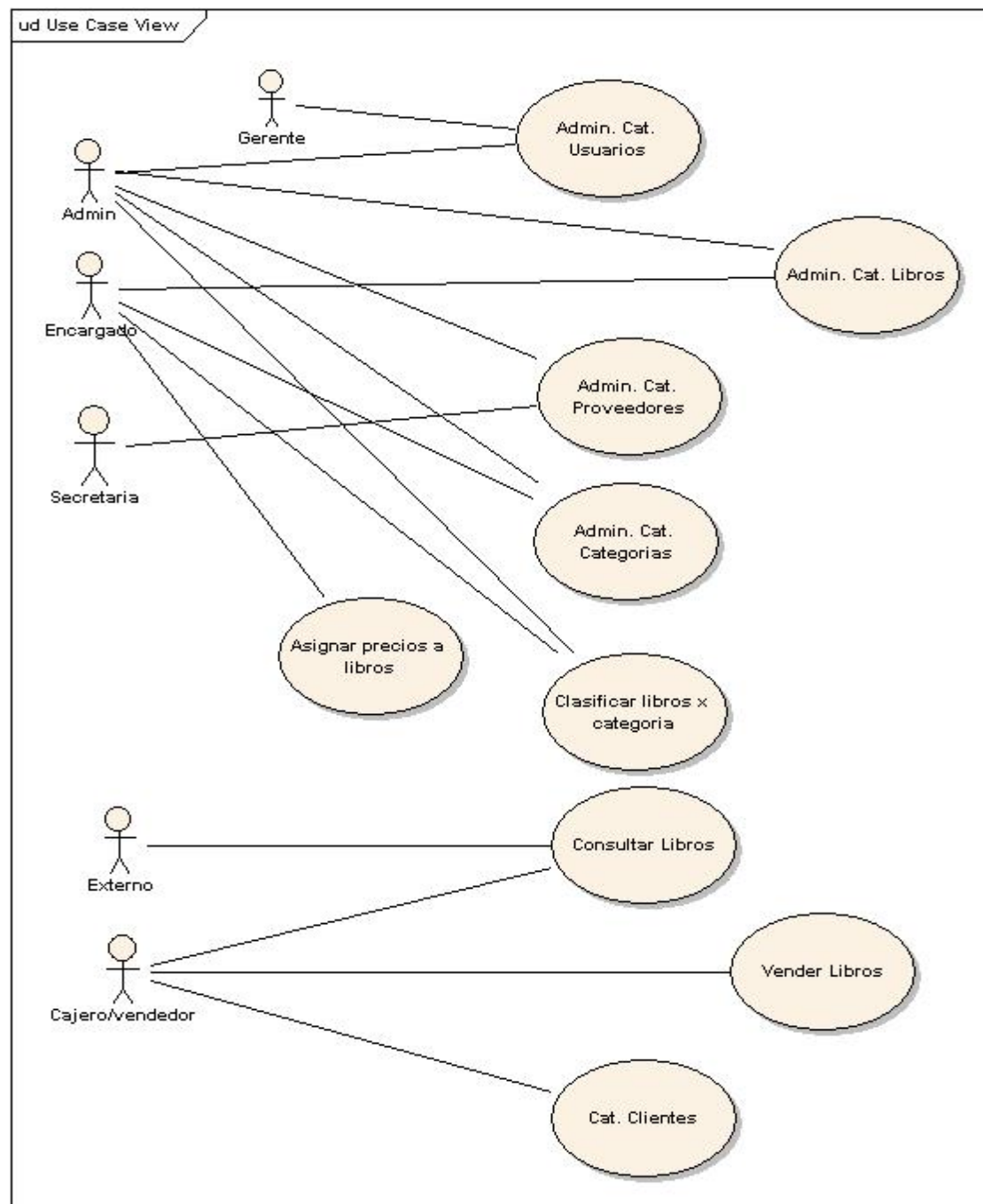


Figura 4.2.1 Diagrama de casos de uso general

La figura 4.2-2 Representa el Diagrama de casos de uso Gerente

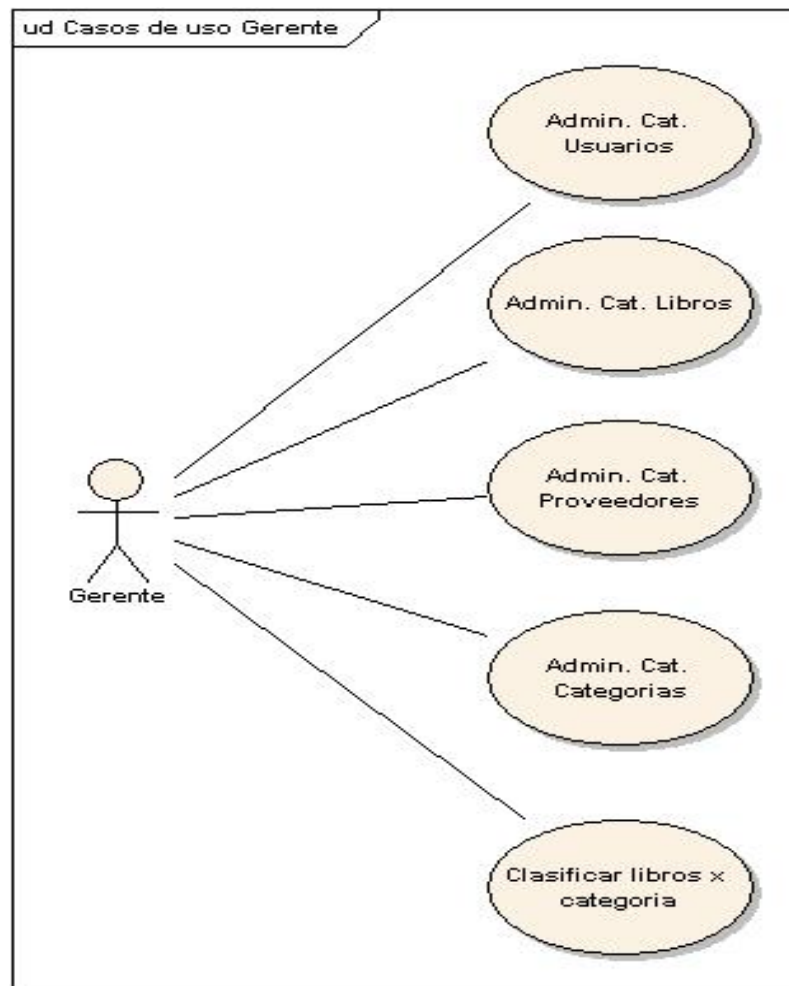


Figura 4.2-2 Diagrama de casos de uso Gerente

La figura 4.2-3 Representa el Diagrama de casos de uso Administrador

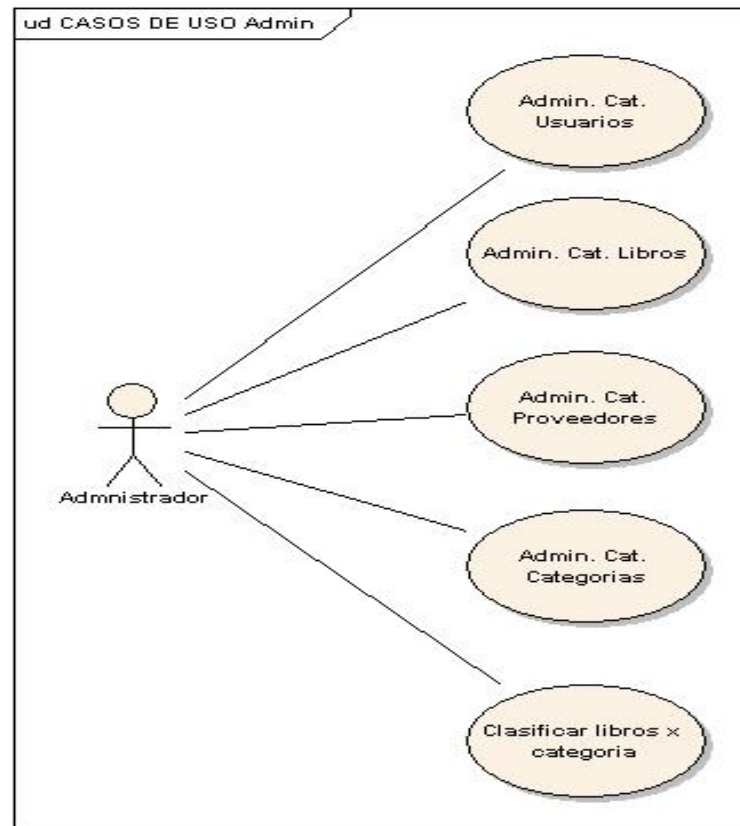


Figura 4.2-3 Diagrama de casos de uso Administrador

La figura 4.2-4 Representa el Diagrama de casos de uso Encargado

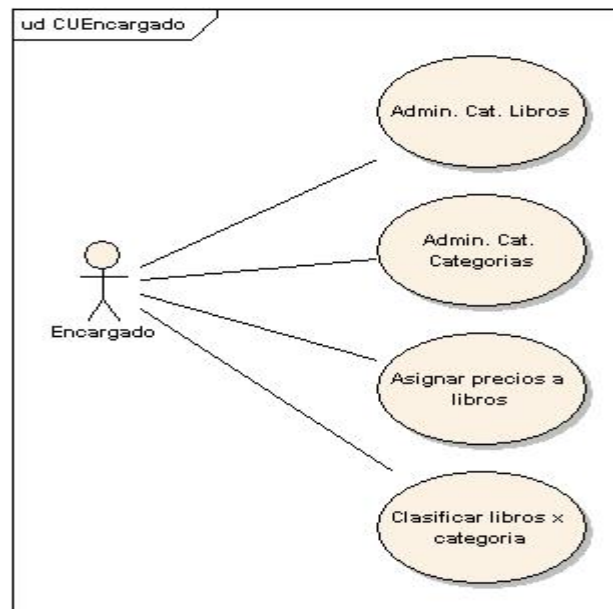


Figura 4.2-4 Diagrama de casos de uso Encargado

La figura 4.2-5 Representa el Diagrama de casos de uso Secretaria

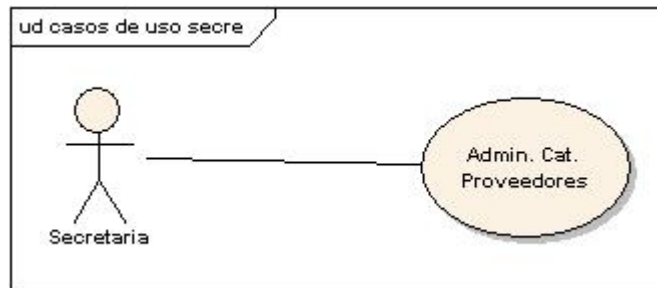


Figura 4.2-5 Diagrama de casos de uso Secretaria

La figura 4.2-6 Representa el Diagrama de casos de uso Externo

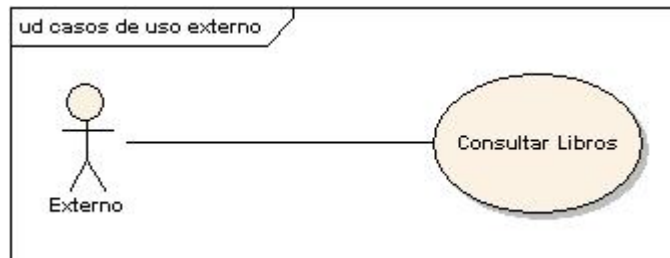


Figura 4.2-6 Diagrama de casos de uso Externo

La figura 4.2-7 Representa el Diagrama de casos de uso Cajero/vendedor

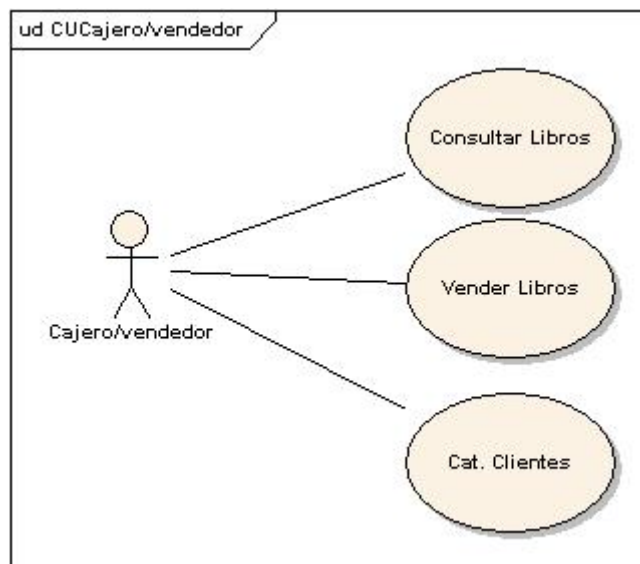


Figura 4.2-7 Diagrama de casos de uso Cajero/vendedor

4.3 Narrativas de casos de uso.

| | | | |
|---|---------------------------------|--------------|------------|
| Nombre Caso Uso | SP-AUSU-01 Administrar Usuarios | | |
| Creación | Román Arango Díaz | Fecha | 16/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| Realizar la administración de los usuarios que interactúan con el sistema. La administración de los usuarios incluye el agregar, listar, borrar, y modificar algunos atributos de los usuarios. | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none"> Administrador del Sistema, Gerente | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none"> Usuario autenticado en el sistema Usuario con perfil de Administrador del Sistema | | | |

Escenarios

Escenario primario:

| Paso | Acción |
|------|--|
| 1. | El caso de uso comienza cuando el usuario selecciona en el Menú de "Catálogos" la opción de " Cat. Usuarios" |
| 2. | <p>El sistema muestra una lista de los "Usuarios" que existen en el sistema con los siguientes datos:</p> <ul style="list-style-type: none"> Identificador de Usuario Clave de Usuario Contraseña Nombre Apellido Paterno Apellido Materno Mail |

| Paso | Acción |
|------|--|
| | <ul style="list-style-type: none"> • Rol de Usuario • Una opción para modificar el usuario • Una opción para eliminar el usuario <p>Se muestra una opción para agregar usuarios</p> |
| 3. | <p>El sistema solicita se realice alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Si el usuario solicita agregar un usuario, se ejecuta el escenario alternativo EA1 <Agregar Usuario> • Si el usuario solicita modificar usuario, se ejecuta el escenario alternativo EA2 <Modificar Usuario> • Si el usuario solicita eliminar usuario, se ejecuta el escenario alternativo EA3 <Eliminar Usuario> |
| 4. | Termina caso de uso |

Escenario Alternativo EA1 <Agregar Usuario>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita agregar un usuario |
| 2. | <p>El sistema solicita el ingreso de;</p> <ul style="list-style-type: none"> • Clave de usuario • Contraseña • Nombre • Apellido Paterno • Apellido Materno • Mail • Rol <p>El sistema solicita la selección de;</p> <ul style="list-style-type: none"> • Los tipos de roles; <ul style="list-style-type: none"> ✓ Administrador ✓ Consultas |
| 3. | El usuario ingresa; |

| Paso | Acción |
|------|---|
| | <ul style="list-style-type: none"> • Clave de usuario • Contraseña • Nombre • Apellido Paterno • Apellido Materno • Mail • Rol |
| 4. | <p>El usuario selecciona los tipos de rol, que desee;</p> <ul style="list-style-type: none"> ✓ Administrador ✓ Consultas |
| 5. | <p>El sistema muestra 2 opciones;</p> <ul style="list-style-type: none"> • Una para guardar los datos ingresados. • Una para cancelar y no guardar nada. |
| 6. | El usuario solicita guardar los datos para agregar el nuevo usuario |
| 7. | VA1 <Valida usuario> |
| 8. | El sistema agrega el nuevo usuario |
| 9. | Se continúa en el paso donde fue invocado el escenario alterno |

Escenario Alterno EA2 <Modificar Usuario>:

| Paso | Acción |
|------|--|
| 1. | El usuario solicita modificar los datos del usuario |
| 2. | <p>El sistema permite modificar los datos de;</p> <ul style="list-style-type: none"> • Contraseña • Nombre • Apellido Paterno • Apellido Materno • Mail • Los tipos de rol; <ul style="list-style-type: none"> ✓ Administrador |

| Paso | Acción |
|------|--|
| | ✓ Consultas |
| 3. | El usuario realiza las modificaciones pertinentes |
| 4. | El usuario solicita guardar las modificaciones |
| 5. | VA1 <Valida usuario> |
| 6. | El sistema guarda las modificaciones |
| 7. | Se continúa en el paso donde fue invocado el escenario alterno |

Escenario Alterno EA3 <Eliminar Usuario>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita eliminar un usuario |
| 2. | El sistema despliega un mensaje preguntado al usuario si, está seguro de eliminar el usuario, con 2 posibles opciones; <ul style="list-style-type: none"> • Si • No |
| 3. | El usuario selecciona la opción; <ul style="list-style-type: none"> • Si, continúa escenario alterno • No, Se continúa en el paso donde fue invocado el escenario alterno |
| 4. | Se elimina del sistema el usuario. |
| 5. | Se continúa en el paso donde fue invocado el escenario alterno |

Validación VA1 <Valida usuario>

| Paso | Acción |
|------|---|
| 1. | Se valida el usuario <ul style="list-style-type: none"> ▪ Si la Clave de usuario es repetida, enviar mensaje GE002 y regresar al paso donde fue invocada la validación ▪ Si falta ingresar una dirección de correo electrónico al usuario, enviar mensaje GE003 y regresar al paso donde fue invocada la validación |
| 2. | Se regresa al paso del escenario donde fue invocada la validación |

| |
|---------------------------|
| Reglas de negocio: |
| Ninguna |

| Mensajes de Error | |
|--------------------------|---|
| Id | Descripción del Error |
| GE0002 | No se pueden ingresar registros Duplicados. |
| GE0003 | Debe asignar un mail al usuario. |

| |
|---|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Postcondiciones |
| <ul style="list-style-type: none"> Las modificaciones del usuario son guardadas en el sistema, o bien la agregación del nuevo usuario o bien la eliminación del usuario. |
| Notas para implementación |
| Ninguna |

| | | | |
|-----------------------------|--|--------------|------------|
| Nombre Caso Uso | SP- CATL- 02 Administrar Libros | | |
| Creación | Gloria Angélica Noguez Barajas | Fecha | 14/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | Realizar la administración del catálogo de libros incluye el actualizar, agregar, listar, borrar, y modificar algunas características de los libros. | | |
| Actores involucrados | <ul style="list-style-type: none"> Administrador del Sistema Encargado | | |

Precondiciones

- Usuario autenticado en el sistema

Escenarios**Escenario primario:**

| Paso | Acción |
|------|---|
| 1 | El caso de uso comienza cuando el Administrador del Sistema o el encargado selecciona en el Menú de "Catálogos" la opción de "Cat. Libros" |
| 2 | <p>El sistema muestra una lista de los libros que existen en el sistema con los siguientes datos:</p> <ul style="list-style-type: none"> • Clave o Identificador de Libro • Proveedor • Autor • Categoría • Título • Año de Edición • Versión • Precio de Libro • Una opción para modificar el libro • Una opción para eliminar el libro <p>Se muestra una opción para agregar libros</p> |
| 3 | <p>El sistema solicita realizar alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Si el usuario solicita agregar un libro, se ejecuta el escenario alternativo EA1 <Agregar Libro> • Si el usuario solicita modificar libro, se ejecuta el escenario alternativo EA2 <Modificar Libro> • Si el usuario solicita eliminar libro, se ejecuta el escenario alternativo EA3 <Eliminar Libro> |
| 4 | Termina caso de uso |

Escenario Alternativo EA1 <Agregar Libro>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita agregar un libro |
| 2. | El sistema solicita el ingreso de: <ul style="list-style-type: none"> • Clave o Identificador • Proveedor • Autor • Categoría • Título • Año de Edición • Versión • Precio venta por unidad |
| 3. | El usuario ingresa: <ul style="list-style-type: none"> • Clave o Identificador • Proveedor • Autor • Categoría • Título • Año de Edición • Versión • Precio venta por unidad |
| 4. | El sistema muestra 2 opciones: <ul style="list-style-type: none"> • Una para Guardar los datos ingresados. • Una para cancelar y no guardar nada. |
| 5. | El usuario solicita guardar los datos para agregar el nuevo libro |
| 6. | VA1 <Valida libro> |
| 7. | El sistema agrega el nuevo libro |
| 8. | Se continúa en el paso donde fue invocado el escenario alternativo |

Escenario Alternativo EA2 <Modificar libro>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita modificar los datos del libro |
| 2. | El sistema permite modificar los siguientes datos: <ul style="list-style-type: none"> • Clave o Identificador • Proveedor • Autor • Categoría • Título • Año de Edición • Versión • Precio venta por unidad |
| 3. | El usuario realiza las modificaciones pertinentes |
| 4. | El usuario solicita guardar las modificaciones |
| 5. | El sistema guarda las modificaciones |
| 6. | Se continúa en el paso donde fue invocado el escenario alternativo |

Escenario Alternativo EA3 <Eliminar Libro>:

| Paso | Acción |
|------|--|
| 1. | El usuario solicita eliminar un libro |
| 2. | El sistema despliega un mensaje preguntado al usuario si, está seguro de eliminar el libro, con 2 posibles opciones; <ul style="list-style-type: none"> • Si • No |
| 3. | El usuario selecciona la opción; <ul style="list-style-type: none"> • Si, Se continúa escenario alternativo • No, Se continúa en el paso donde fue invocado el escenario alternativo |
| 4. | Se elimina del sistema el libro. |
| 5. | Se continúa en el paso donde fue invocado el escenario alternativo |

Validación VA1 <Valida libro>

| Paso | Acción |
|------|---|
| 1. | Se valida el usuario <ul style="list-style-type: none"> Si la clave de libro es repetida, enviar mensaje GE001 y regresa al paso donde fue invocada la validación |
| 2. | Se regresa al paso del escenario donde fue invocada la validación |

Reglas de negocio:

Ninguna

Mensajes de Error

| Id | Descripción del Error |
|---------------|---|
| GE0001 | No se pueden ingresar registros Duplicados. |

Riesgos

Ninguno

Requerimientos no funcionales

Ninguno

Postcondiciones

- Las modificaciones del catálogo son guardadas en el sistema, o bien la agregación del nuevo libro o bien la eliminación del libro.

Notas para implementación

Ninguna

| | | | |
|----------------------------|--------------------------------------|-------|------------|
| Nombre Caso Uso | SP- PROV -03-Administrar Proveedores | | |
| Creación | Brenda Salas Santiago | Fecha | 15/11/2006 |
| Ultima modificación | | Fecha | |

| |
|---|
| Objetivo |
| Realizar la administración de los proveedores que interactúan con el sistema integral de administración de una librería. La administración de los proveedores incluye el agregar, listar, borrar, y modificar algunos atributos de los proveedores. |
| Actores involucrados |
| <ul style="list-style-type: none"> • Administrador del Sistema • Secretaria |
| Precondiciones |
| <ul style="list-style-type: none"> • Usuario autenticado en el sistema • Usuario con perfil de Administrador del Sistema |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|--|
| 1. | El caso de uso comienza cuando el usuario selecciona en el Menú de "Catálogos" la opción de "Cat. Proveedores" |
| 2. | <p>El sistema muestra una lista de los "Proveedores" que existen en el sistema con los siguientes datos:</p> <ul style="list-style-type: none"> • Clave o Identificador de Proveedor • Nombre • Una opción para modificar el proveedor • Una opción para eliminar el proveedor <p>Se muestra una opción para agregar proveedores</p> |
| 3. | <p>El sistema solicita se realice alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> • Si el usuario solicita agregar un proveedor, se ejecuta el escenario alternativo EA1 <Agregar Proveedor> • Si el usuario solicita modificar proveedor, se ejecuta el escenario alternativo EA2 <Modificar Proveedor> • Si el usuario solicita eliminar proveedor, se ejecuta el escenario alternativo EA3 <Eliminar Proveedor> |

| Paso | Acción |
|------|---------------------|
| 4. | Termina caso de uso |

Escenario Alternativo EA1 <Agregar Proveedor>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita agregar un proveedor |
| 2. | El sistema solicita el ingreso de; <ul style="list-style-type: none"> • Clave o Identificador • Nombre |
| 3. | El usuario ingresa: <ul style="list-style-type: none"> • Clave o Identificador • Nombre |
| 4. | El sistema muestra 2 opciones; <ul style="list-style-type: none"> • Una para Guardar los datos ingresados. • Una para Cancelar y no guardar nada. |
| 5. | El usuario solicita guardar los datos para agregar el nuevo proveedor |
| 6. | VA1 <Valida proveedor> |
| 7. | El sistema asigna un identificador único para el proveedor |
| 8. | El sistema agrega el nuevo proveedor |
| 9. | Se continúa en el paso donde fue invocado el escenario alternativo |

Escenario Alternativo EA2 <Modificar Proveedor>:

| Paso | Acción |
|------|--|
| 1. | El usuario solicita modificar los datos del proveedor |
| 2. | El sistema permite modificar los datos de; <ul style="list-style-type: none"> • Clave o Identificador • Nombre |
| 3. | El usuario realiza las modificaciones pertinentes |
| 4. | El usuario solicita guardar las modificaciones |

| Paso | Acción |
|------|--|
| 5. | VA1 <Valida proveedor> |
| 6. | El sistema guarda las modificaciones |
| 7. | Se continúa en el paso donde fue invocado el escenario alterno |

Escenario Alterno EA3 <Eliminar Usuario>:

| Paso | Acción |
|------|---|
| 1. | El usuario solicita eliminar un proveedor |
| 2. | El sistema despliega un mensaje preguntado al usuario si, está seguro de eliminar el proveedor, con 2 posibles opciones; <ul style="list-style-type: none"> • Si • No |
| 3. | El usuario selecciona la opción; <ul style="list-style-type: none"> • Si, Continúa escenario alterno • No, Se continúa en el paso donde fue invocado el escenario alterno |
| 4. | Se elimina del sistema el proveedor |
| 5. | Se continúa en el paso donde fue invocado el escenario alterno |

Validación VA1 <Valida proveedor>

| Paso | Acción |
|------|--|
| 1. | Se valida el proveedor <ul style="list-style-type: none"> ▪ Si falta ingresar nombre, enviar mensaje GE002 y regresar al paso donde fue invocada la validación ▪ Si falta ingresar un nombre de contacto, dirección de correo electrónico, teléfono de contacto al proveedor, enviar mensaje GE003 y regresar al paso donde fue invocada la validación |
| 2. | Se regresa al paso del escenario donde fue invocada la validación |

| |
|---------------------------|
| Reglas de negocio: |
| Ninguna |

| Mensajes de Error | |
|--------------------------|---|
| Id | Descripción del Error |
| GE0002 | Debe asignar nombre al proveedor |
| GE0003 | Debe asignar campo obligatorio del proveedor. |

| |
|---|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Postcondiciones |
| <ul style="list-style-type: none"> Las modificaciones del proveedor son guardadas en el sistema, o bien la agregación del nuevo proveedor o bien la eliminación del proveedor. |
| Notas para implementación |
| Ninguna |

| | | | |
|-----------------------------|---|--------------|------------|
| Nombre Caso Uso | SP-CAT-04 Administrar Categorías | | |
| Creación | Sonia Estrada Mejía | Fecha | 16/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | <ul style="list-style-type: none"> Mostrar catálogo de categorías con los parámetros ingresados previamente. | | |
| Actores involucrados | <ul style="list-style-type: none"> Usuario SIPAL | | |
| Precondiciones | <ul style="list-style-type: none"> Haber ingresado al sistema. | | |

Escenarios**Escenario primario:**

| Paso | Acción |
|------|--|
| 1. | <p>El sistema muestra las siguientes opciones para realizar la búsqueda:</p> <ul style="list-style-type: none"> • Clave o Identificador de categoría • Nombre de la categoría a buscar <p>Además se muestran 2 opciones adicionales.</p> <ul style="list-style-type: none"> • Una opción con la leyenda “Editar”, para agregar una categoría al sistema. • Una opción con la leyenda “Eliminar”, para eliminar categorías, siempre y cuando no existan libros dentro de ésta categoría |
| 2. | <p>El sistema muestra los resultados de la búsqueda, con los siguientes datos:</p> <ul style="list-style-type: none"> • Clave o Identificador de clasificación • Nombre de la categoría |
| 3. | Termina caso de uso. |

Reglas de negocio:

Ninguna

Mensajes de Error

| Id | Descripción del Error |
|----|-----------------------|
|----|-----------------------|

Riesgos

Ninguno

Requerimientos no funcionales

Ninguno

Postcondiciones

Ninguna

| | | | |
|--|-----------------------------------|--------------|----------|
| Notas para implementación | | | |
| Ninguna | | | |
| Nombre Caso Uso | SP-ACLI-05- Administrar Clientes. | | |
| Creación | Abraham Varela Benítez | Fecha | Nov/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| Administrar los clientes de la librería. <ul style="list-style-type: none"> • Agregar nuevos clientes. • Modificar datos del cliente | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none"> • Cajero/vendedor del sistema. | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none"> • Haber ingresado al sistema y tener perfil de Cajero/Vendedor del sistema. | | | |

Escenarios

Escenario primario:

| Paso | Acción |
|------|--|
| 1. | El caso de uso comienza cuando el usuario selecciona en el Menú de "Catálogos", la opción "Cat. de clientes" |
| 2. | <p>El sistema muestra los resultados de la búsqueda, con los siguientes datos:</p> <ul style="list-style-type: none"> • Identificador o clave de Cliente. • Nombre • Apellido paterno • Apellido materno • Dirección <p>Si no se encuentra ningún usuario en la búsqueda se mostrará el mensaje GE018</p> <p>Y una opción para editar cada cliente.</p> <p>Una opción para agregar un nuevo cliente.</p> |

| Paso | Acción |
|------|--|
| 3. | <p>El usuario podrá seleccionar la opción de, agregar nuevo cliente o editar cliente:</p> <ul style="list-style-type: none"> • Si el usuario selecciona la opción agregar nuevo cliente. Se ejecuta escenario alternativo EA1<Agregar nuevo cliente> • Si el usuario selecciona la opción editar cliente. Se ejecuta escenario alternativo EA2<Editar cliente> |
| 4. | <p>El usuario selecciona la opción de guardar.</p> <ul style="list-style-type: none"> • Si se esta agregando un nuevo Cliente, se valida la CveCliente R1 Crear Clave de cliente • . VA1 <Valida CveCliente Existente>, |
| 5. | Termina caso de uso. |

Escenario Alternativo EA1 <Agregar nuevo cliente>:

| Paso | Acción |
|------|--|
| 1. | <p>El sistema muestra los datos a ingresar;</p> <ul style="list-style-type: none"> • Identificador o clave • Nombre • Apellido paterno • Apellido materno • Dirección <p>El sistema muestra una opción para “cancelar”, y una opción para “Guardar datos”</p> |
| 2. | <p>El usuario ingresa los datos:</p> <ul style="list-style-type: none"> • Identificador o clave • Nombre • Apellido paterno • Apellido materno • Dirección |
| 3. | Regresa al paso 6 del flujo principal |

Escenario Alternativo EA2 <Modificar cliente>:

| Paso | Acción |
|------|--|
| 1. | <p>El sistema muestra los siguientes datos;</p> <ul style="list-style-type: none"> • Identificador o clave • Nombre • Apellido paterno • Apellido materno • Dirección <p>El sistema muestra una opción para “cancelar”, y una opción para “Guardar datos”</p> |
| 2. | El usuario modifica los datos que crea conveniente |
| 3. | Regresa al paso 6 del flujo principal |

Validación VA1 < Valida CveCliente Existente >:

| Paso | Acción |
|------|---|
| 1. | <p>Las siguientes condiciones;</p> <ul style="list-style-type: none"> • Que la clave cliente no exista en la base de datos. De no serlo así enviará el mensaje GE017 y regresará al paso 1 del escenario primario |

Reglas de negocio:**R1 Crear Clave de cliente**

Antes de validar la clave de cliente, es necesario crear una nueva, esta se formará como el RFC y se validará que no exista una igual.

| Mensajes de Error | |
|-------------------|--|
| Id | Descripción del Error |
| GE015 | La longitud de la contraseña debe ser de 8 caracteres. |
| GE017 | La clave de cliente ya existe en la base de datos, verifique si ya fue agregado dicho cliente. |
| GE018 | No se encontraron clientes que cumplan el criterio de búsqueda. |

| |
|---|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Poscondiciones |
| Habrán sido agregados nuevos clientes o bien habrán sido modificados. |
| Notas para implementación |
| Ninguna |

| | | | |
|-----------------------------|--|-------|------------|
| Nombre Caso Uso | SP-CLC-06 Consulta Libros por Categoría | | |
| Creación | Román Arango Díaz | Fecha | 16/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | <ul style="list-style-type: none"> Consultar la categoría de cada libro | | |
| Actores involucrados | <ul style="list-style-type: none"> Administrador, Encargado de almacén | | |
| Precondiciones | <ul style="list-style-type: none"> Haber ingresado al sistema. | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|---|
| 1. | <p>El sistema muestra la siguiente opción para realizar la búsqueda:</p> <ul style="list-style-type: none"> Nombre de Libro Proveedor Autor Categoría |

| Paso | Acción |
|------|--|
| | <ul style="list-style-type: none"> Existencia <p>Además se muestra:</p> <ul style="list-style-type: none"> Una opción con la leyenda “IR”, para realizar la búsqueda con los parámetros ingresados. Dicha opción realizara una búsqueda en la BD para mostrar los registros encontrados. |
| 2. | <p>El usuario ingresará los criterios de búsqueda ^{VA1} que desee y podrá seleccionar la opción de, “Ir”:</p> <ul style="list-style-type: none"> Si el usuario selecciona la opción “Ir”. Continúa Caso de uso. |
| 3. | <p>El sistema muestra los resultados de la búsqueda, con los siguientes datos:</p> <ul style="list-style-type: none"> Título del libro Versión Año Editorial Autor Categoría Precio de libro |
| 4. | <p>El sistema solicita se realice alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> Si el usuario solicita consultar, se ejecuta el escenario alternativo EA1 <consultar clasificación> |
| 5. | Termina caso de uso |

Escenario Alternativo EA1 <modificar clasificación>:

| Paso | Acción |
|------|--|
| 1. | El usuario solicita consultar un libro clasificado |
| 2. | <p>El sistema solicita la selección de:</p> <ul style="list-style-type: none"> Nombre de Libro Proveedor Autor Categoría |

| Paso | Acción |
|------|--|
| | <ul style="list-style-type: none"> Existencia |
| 3. | El usuario selecciona el filtro, que desee. |
| 4. | Se continúa en el paso donde fue invocado el escenario alternativo |

| |
|---------------------------|
| Reglas de negocio: |
| Ninguna |

| |
|---|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Postcondiciones |
| Al dar clic en la opción Ir , el valor de la variable clasificado cambia a trae |
| Notas para implementación |
| Ninguna |

| | | | |
|-----------------------------|--|-------|------------|
| Nombre Caso Uso | SP - VTAL – 07 – Venta de Libros | | |
| Creación | Brenda Salas Santiago | Fecha | 15/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | <ul style="list-style-type: none"> Realizar la venta de libros en caja. | | |
| Actores involucrados | <ul style="list-style-type: none"> Cajero / Vendedor. | | |
| Precondiciones | <ul style="list-style-type: none"> Haber ingresado al sistema. | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|---|
| 1. | El caso de uso comienza cuando el usuario selecciona en el Menú de "Reportes" la opción de "Reporte de Libros Vendidos" |
| 2. | <p>El sistema muestra las siguientes opciones para realizar la venta:</p> <ul style="list-style-type: none"> • Código de Identificación del libro (Código de barras). • Número de libros. • Clave de Cliente <p>Además se muestran 3 opciones adicionales.</p> <ul style="list-style-type: none"> • Una opción con la leyenda "Limpiar datos", para borrar los datos ingresados. • Una opción con la leyenda "Ir", para consultar el libro en una lista de venta. Dicha opción consultara los libros que se quieran vender. • Una opción con la leyenda "Vender", para realizar la venta con los parámetros ingresados. Dicha opción realizará una baja de existencia de los libros seleccionados en la BD. |
| 3. | <p>El usuario ingresará los criterios de búsqueda ^{VA1} que desee y podrá seleccionar la opción de, "Agregar", "Limpiar datos" o "Vender":</p> <ul style="list-style-type: none"> • Si el usuario selecciona la opción "Agregar". Continúa Caso de uso. • Si el usuario selecciona la opción "Limpiar datos". Se borran los datos ingresados en los parámetros de búsqueda; <ul style="list-style-type: none"> ➤ Código de Identificación del libro (Código de barras) ➤ Número de libros ➤ Si el usuario selecciona la opción "Vender". Continúa caso de Uso. |
| 4. | <p>El sistema muestra los resultados de la compra, con los siguientes datos del libro:</p> <ul style="list-style-type: none"> • Nombre • Clasificación • Categoría • Precio • Nombre de proveedor |

| Paso | Acción |
|------|---|
| | En el que el campo Nombre de Proveedor deberá ser un link, hacia el visor del PROV. |
| 5. | El sistema realizará una baja de existencia de cada uno de los libros de la venta. |
| 6. | Termina caso de uso. |

Validación VA1 < Valida Datos Ingresados >:

| Paso | Acción |
|------|--|
| 1. | <p>Las siguientes condiciones;</p> <ul style="list-style-type: none"> Que los datos ingresados no contengan caracteres inválidos (“,#, /,*.- .[,}), es decir solamente permitirá caracteres numéricos, GE000 y regresar al paso 1 del escenario primario |

Reglas de negocio:

Ninguna

Mensajes de Error

| Id | Descripción del Error |
|--------------|--|
| GE000 | Favor de ingresar únicamente caracteres numéricos. |

Riesgos

Ninguno

Requerimientos no funcionales

Ninguno

Postcondiciones

Al dar clic en el link del Nombre de Proveedor, tendrá que aparecer la pantalla de consulta de proveedores.

| |
|----------------------------------|
| Notas para implementación |
| Ninguna |

| | | | |
|---|-------------------------------|-------|------------|
| Nombre Caso Uso | SP-CON- 09 Consulta de Libros | | |
| Creación | Sonia Estrada Mejía | Fecha | 16/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| <ul style="list-style-type: none"> Mostrar catálogo de libros con los parámetros ingresados previamente. | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none"> Usuario SIPAL | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none"> Haber ingresado al sistema. | | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|---|
| 1. | <p>El sistema muestra las siguientes opciones para realizar la búsqueda:</p> <ul style="list-style-type: none"> Nombre del libro que desea consultar, o bien , Autor Proveedor Categoría Existencia Si el usuario no inserta ningún valor el sistema mostrará todos los libros que estén almacenados en el sistema. Dicha opción realizará una búsqueda en la BD para mostrar los libros encontrados. <p>Además se muestran 2 opciones adicionales.</p> <ul style="list-style-type: none"> Una opción con la leyenda "Insertar Libro", para agregar un libro nuevo |

| Paso | Acción |
|------|---|
| | <p>al sistema.</p> <ul style="list-style-type: none"> • Una opción con la leyenda “Eliminar Libro”, para eliminar libros, siempre y cuando no se encuentren solicitados por clientes. |
| 2. | <p>El usuario ingresará los criterios de búsqueda que desee y podrá seleccionar la opción de, “Buscar” o “Limpiar datos” :</p> <ul style="list-style-type: none"> • Si el usuario selecciona la opción “Buscar”. Continúa Caso de uso. • Si el usuario selecciona la opción “Limpiar datos”. Se borran los datos ingresados en los parámetros de búsqueda; <ul style="list-style-type: none"> ➤ Identificador o clave de libro ➤ Nombre del libro ➤ Autor ➤ Proveedor ➤ Categoría ➤ Existencia |
| 3. | <p>El sistema muestra los resultados de la búsqueda, con los siguientes datos:</p> <ul style="list-style-type: none"> • Clave de libro • Nombre del libro • Versión • Año • Editorial • Autor • Categoría • Precio |
| 4. | Termina caso de uso. |

Reglas de negocio:

Ninguna

| Mensajes de Error | |
|-------------------|-----------------------|
| Id | Descripción del Error |

| |
|--------------------------------------|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Postcondiciones |
| |
| Notas para implementación |
| Ninguna |

| | | | |
|--|--|-------|----------|
| Nombre Caso Uso | SP-CRPR-10- Reporte de Estadísticas de Venta | | |
| Creación | Abraham Varela Benítez. | Fecha | Nov/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| Visualizar el reporte estadístico de los proveedores | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none">Consulta | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none">Usuario autenticado en el sistemaUsuario con permisos de visualizar reporte de proveedoresExistencia de la encuesta a consultar sus datos estadísticos | | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|---|
| 1. | El caso de uso comienza cuando el usuario selecciona en el Menú de "Reportes" la opción de "Estadísticas de Venta" |
| 2. | El sistema muestra unas opciones para realizar una Búsqueda y delimitar el número de proveedores mostrados, así como también una opción de "Búsqueda". |
| 3. | El usuario ingresa parte o alguno de los datos solicitados, o también puede solamente seleccionar la opción de "Búsqueda" |
| 4. | El sistema muestra el reporte de los libros de la LIBRERÍA conteniendo los siguientes datos: <ul style="list-style-type: none"> • Total vendido • Clave • Título • Versión • Año |
| 5. | Termina caso de uso |

Reglas de negocio:

Ninguna

Mensajes de Error**Id****Descripción del Error****Riesgos**

Ninguno

Requerimientos no funcionales

Ninguno

Postcondiciones

- La visualización del reporte de los proveedores.

| |
|----------------------------------|
| Notas para implementación |
| Ninguna |

| | | | |
|---|---|--------------|------------|
| Nombre Caso Uso | SP-CRCLI-11 Consulta de Libros por Clientes | | |
| Creación | Sonia Estrada Mejía | Fecha | 16/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| <ul style="list-style-type: none"> Mostrar una consulta de los libros vendidos por clientes. | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none"> Usuario SIPAL | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none"> Haber ingresado al sistema. | | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|-------------|---|
| 1. | <p>El sistema muestra las siguientes opciones para realizar la búsqueda:</p> <ul style="list-style-type: none"> Nombre de Libro Proveedor Autor Categoría Existencia Si el usuario no inserta ningún valor el sistema mostrará todos los clientes que se encuentren registrados en el sistema. Dicha opción realizara una búsqueda en la BD para mostrar los clientes encontrados, siempre y cuando no tenga una compra asignada. |
| 2. | <p>El usuario ingresará los criterios de búsqueda que desee y podrá seleccionar la opción de, “Buscar” o “Limpiar datos”:</p> <ul style="list-style-type: none"> Si el usuario selecciona la opción “Buscar”. Continúa Caso de uso. |

| Paso | Acción |
|------|---|
| | <ul style="list-style-type: none"> • Si el usuario selecciona la opción “Limpiar datos”. Se borran los datos ingresados en los parámetros de búsqueda; • Nombre de Libro • Proveedor • Autor • Categoría • Existencia |
| 3. | <p>El sistema muestra los resultados de la búsqueda, con los siguientes datos:</p> <ul style="list-style-type: none"> • Título • Versión • Año • Editorial • Autor • Categoría • Cliente |
| 4. | Termina caso de uso. |

Reglas de negocio:

Ninguna

Mensajes de Error**Id****Descripción del Error****Riesgos**

Ninguno

Requerimientos no funcionales

Ninguno

Postcondiciones

| |
|----------------------------------|
| |
| Notas para implementación |
| Ninguna |

| | | | |
|---|---|-------|------------|
| Nombre Caso Uso | SP – REPVTA - 12 – REPORTE DE LIBROS VENDIDOS | | |
| Creación | Brenda Salas Santiago | Fecha | 15/11/2006 |
| Ultima modificación | | Fecha | |
| Objetivo | | | |
| Reporte de Libros Vendidos | | | |
| Actores involucrados | | | |
| <ul style="list-style-type: none"> Gerente | | | |
| Precondiciones | | | |
| <ul style="list-style-type: none"> Usuario autenticado en el sistema Usuario con perfil de Gerente o Administrador de Sistema | | | |

| |
|-------------------|
| Escenarios |
|-------------------|

Escenario primario:

| Paso | Acción |
|------|--|
| 1. | <p>El caso de uso comienza cuando el usuario selecciona en el Menú de "Reportes" la opción de "Libros Vendidos"</p> <p>El sistema muestra las siguientes opciones para realizar el repote:</p> <ul style="list-style-type: none"> Nombre de Libro Proveedor Autor <p>Además se muestran 2 opciones adicionales.</p> <ul style="list-style-type: none"> Una opción con la leyenda "Limpiar datos", para borrar los datos ingresados. Una opción con la leyenda "Generar", para presentar el reporte. |

| Paso | Acción |
|------|---|
| 2. | El sistema valida si el rango de fechas es correcto. ^{VA1} . El sistema valida si el proveedor es correcto. ^{VA2} . |
| 3. | El sistema despliega varias secciones con información de las ventas de libros <ol style="list-style-type: none"> La primer sección corresponde a datos generales del periodo de venta: <ul style="list-style-type: none"> Nombre del reporte Rango de Fechas La segunda sección corresponde a un listado de todos los libros vendidos. Dicho reporte se agrupa en las columnas. <ul style="list-style-type: none"> Fecha (Fecha de venta) Clasificación/Categoría Nombre Precio de venta Identificador de Proveedor Nombre de Proveedor |
| 4. | Termina Caso de Uso. |

Validación VA1 <Verificación de Fechas correctas>

| Paso | Acción |
|------|---|
| 1. | El sistema verifica; <ul style="list-style-type: none"> Si la fecha inicial es menor a la fecha final del rango GE001. Termina Caso de uso. |

Validación VA2 <Verificación de existencia de proveedor>

| Paso | Acción |
|------|---|
| 1. | El sistema verifica si existe en la BD el proveedor; <ul style="list-style-type: none"> Si no existen, enviar el mensaje GE002. |
| 2. | Prosigue caso de uso |

| |
|---------------------------|
| Reglas de negocio: |
| Ninguna |

| Mensajes de Error | |
|-------------------|---------------------------|
| Id | Descripción del Error |
| GE001 | Rango de fechas inválido. |
| GE002 | No existe el proveedor. |

| |
|--------------------------------------|
| Riesgos |
| Ninguno |
| Requerimientos no funcionales |
| Ninguno |
| Postcondiciones |
| Ninguna |
| Notas para implementación |
| Ninguna |

CAPÍTULO 5

IMPLEMENTACIÓN Y PRUEBAS

Para el desarrollo del sistema se utilizó el ambiente de desarrollo .NET. A continuación se describe la estructura general del sistema y se proporciona una descripción de las pantallas que lo conforman.

La estructura general del sistema se muestra en la figura 5.0-1 siguiente:

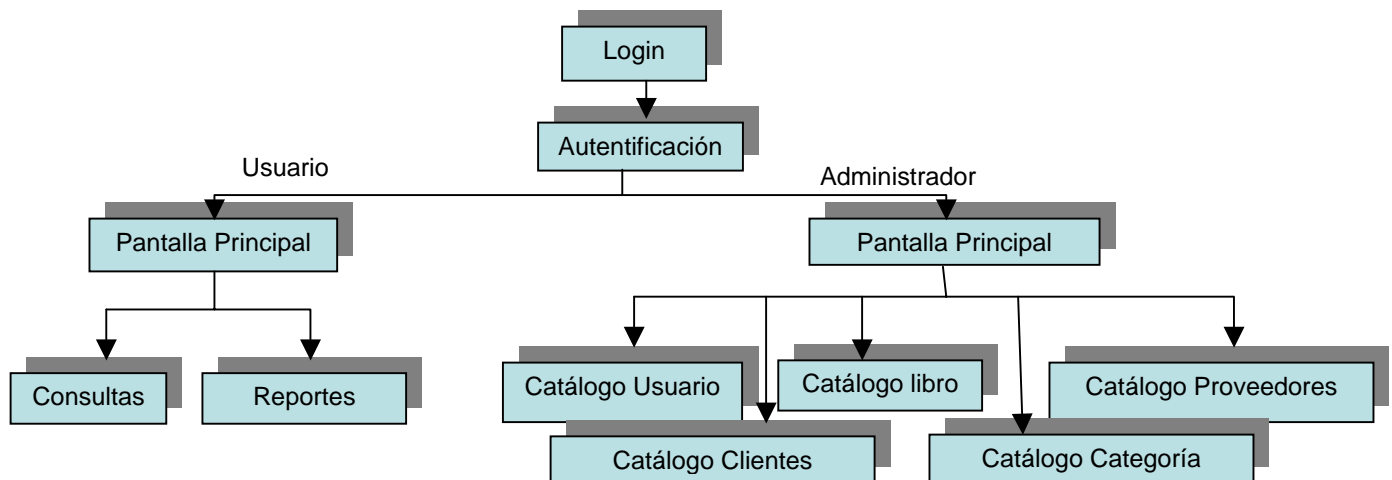


Figura 5.0-1 Diagrama del menú y pantallas del sistema

Para la construcción del sistema, primero se crea el proyecto en entorno de desarrollo, una vez que se muestra la pantalla principal del IDE seleccionamos del menú Archivo->Nuevo->Proyecto ..., se mostrará un cuadro de diálogo para escoger el lenguaje a usar y el tipo de aplicación que se desea construir, se selecciona Proyectos de C# -> Aplicación Web ASP.Net a continuación de le asigna un nombre al proyecto (SIPAL) y la ruta donde se guardará, se muestra a continuación.

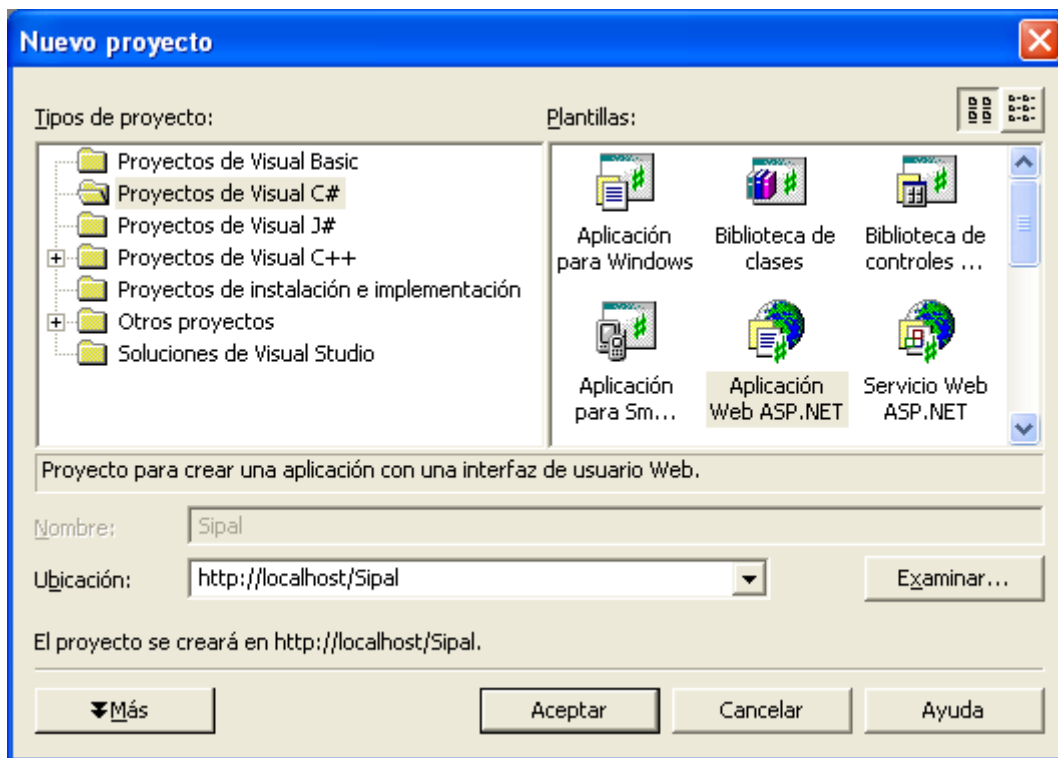


Figura 5.0-2 Imagen que muestra la creación de un proyecto WEB sobre .NET

Lo siguiente es establecer la conexión con la base de datos, esto se hace mediante código utilizando el archivo Web.config. Se indica el servidor y el origen de datos a continuación se muestra en la figura 5.0-3:

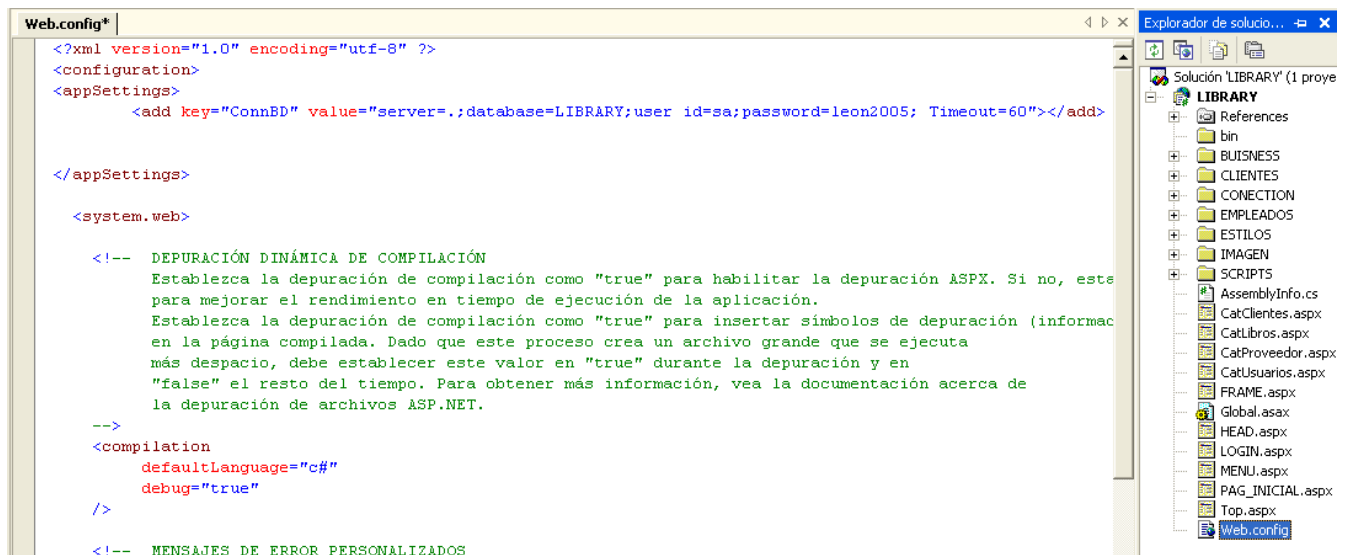


Figura 5.0-3 Muestra el archivo Web.config

Codificación de la pantalla Catálogo de Libros

Para crear la pantalla, se selecciona del menú la opción de Archivo -> Agregar nuevo elemento -> Web Form.

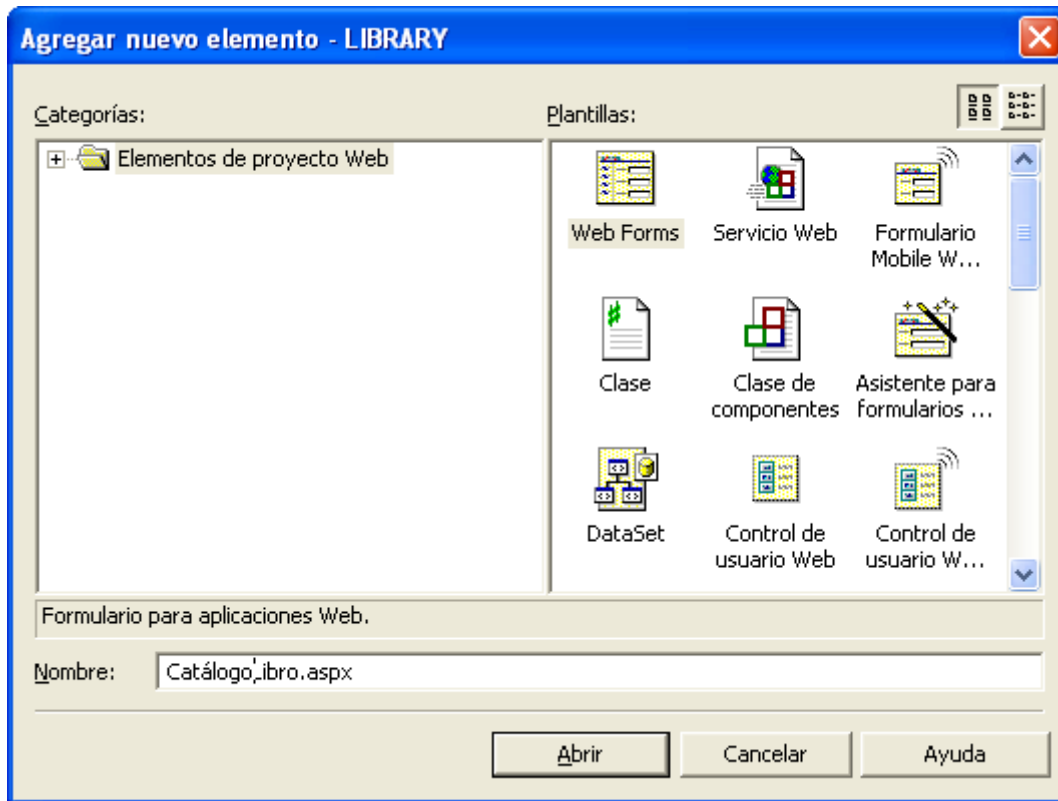


Figura 5.0-4 Muestra las opciones para agregar una clase al proyecto

Para mostrar los datos utilizamos un control Datagrid, dentro de sus propiedades, existe la opción para crear las columnas que necesitamos. Dentro podemos indicarle al control si son columnas de datos o en todo caso si son links para editar, eliminar y crear registros.

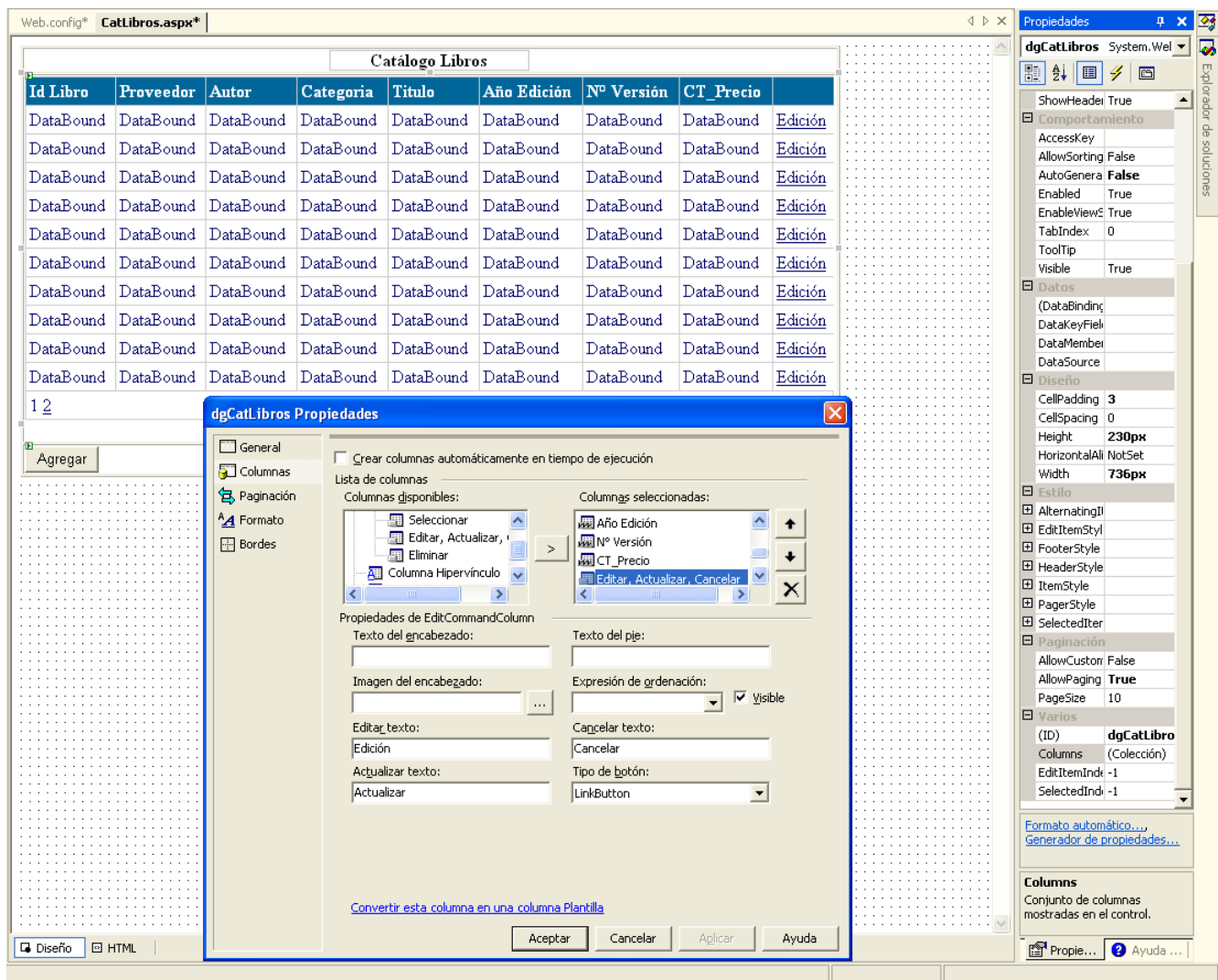


Figura 5.0-5 Muestra la creación de columnas y configuración de las mismas

Una vez finalizado el diseño visual de la forma, se escribe el código necesario para darle la funcionalidad a la pantalla, En la Figura 5.0-6 se muestra el código asociado al botón Agregar de esta pantalla.

```
private void btnAgregar_Click(object sender, System.EventArgs e)
{
    Session["AgregarNuevoLibro"] = true;
    DataView dvLibros = (DataView)Session["dvLibros"];
    dvLibros.AddNew();
    int intCPIndex = dvLibros.Count % dgCatLibros.PageSize;
    dgCatLibros.CurrentPageIndex = intCPIndex == 0 ?
        (dvLibros.Count/dgCatLibros.PageSize) - 1 : dvLibros.Count/dgCatLibros.PageSize;
    int intCont = (dvLibros.Count - (dgCatLibros.CurrentPageIndex * dgCatLibros.PageSize) - 1);
    dgCatLibros.EditItemIndex = intCont;
    dgCatLibros.DataSource = dvLibros;
    dgCatLibros.DataBind();
}
```

Figura 5.0 -6 Muestra código de Agregar

Para poder Editar un registro de la tabla de libro lo que debemos codificar es lo siguiente, en la figura 5.0-7 se muestra el código asociado al botón de Editar.

```
private void dgCatLibros_UpdateCommand(object source,
    System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataView dvLibros = (DataView)Session["dvLibros"];

    Libros clsLibros=new Libros();

    clsLibros.CV_PROVEEDOR = ((DropDownList)e.Item.Cells[1].Controls[1]).SelectedValue;
    clsLibros.ID_AUTOR = ((DropDownList)e.Item.Cells[2].Controls[1]).SelectedValue;
    clsLibros.ID_CATEGORIA = ((DropDownList)e.Item.Cells[3].Controls[1]).SelectedValue;
    clsLibros.TX_LIBRO = ((TextBox)e.Item.Cells[4].Controls[1]).Text;
    clsLibros.ANIO_EDICION = ((TextBox)e.Item.Cells[5].Controls[1]).Text;
    clsLibros.NU_VERSION = ((TextBox)e.Item.Cells[6].Controls[1]).Text;
    clsLibros.CT_PRECIO = ((TextBox)e.Item.Cells[7].Controls[1]).Text;

    if ((bool)Session["AgregarNuevoLibro"])
    {
        clsLibros.ID_LIBRO = ((TextBox)e.Item.Cells[0].Controls[1]).Text;
        dvLibros= new DataView(clsLibros.InsertCliente());
    }
    else
    {
        clsLibros.ID_LIBRO=dvLibros.Table.Rows[e.Item.ItemIndex][0].ToString();
        dvLibros= new DataView(clsLibros.UpdateLibros());
    }

    dgCatLibros.EditItemIndex = -1;
    if (dvLibros.Count <= dgCatLibros.PageSize)
        dgCatLibros.CurrentPageIndex = 0;
    dvLibros.Table.AcceptChanges();
    dgCatLibros.DataSource = dvLibros;
    dgCatLibros.DataBind();
    Session["dvLibros"] = dvLibros;
    Session["AgregarNuevoLibro"] = false;
}
```

Figura 5.0-7 Muestra código de Modificar

En el caso de que se quiera eliminar un registro o cancelar la operación se ejecuta el código que se ve en la figura 5.0-8

```

private void dgCatLibros_CancelCommand(object source,
    System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataView dvLibros = (DataView)Session["dvLibros"];
    if ((bool)Session["AgregarNuevoLibro"])
        dvLibros.Delete(dvLibros.Count - 1);
    dvLibros.Table.RejectChanges();
    dgCatLibros.DataSource = dvLibros;
    dgCatLibros.EditItemIndex = -1;
    dgCatLibros.DataBind();
    Session["AgregarNuevoLibro"] = false;
}

private void dgCatLibros_DeleteCommand(object source,
    System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataView dvLibros = (DataView)Session["dvLibros"];
    int itemIndex = e.Item.ItemIndex + (dgCatLibros.PageSize * dgCatLibros.CurrentPageIndex);
    DataRowView row = dvLibros[itemIndex];
    string id_libro = ((Label)e.Item.Cells[0].Controls[1]).Text;

    if ((bool)Session["AgregarNuevoLibro"])
        dvLibros.Delete(itemIndex);
    else
    {
        Libros clsLibros = new Libros();
        clsLibros.ID_LIBRO = row.Row["ID_Libro"].ToString();
        dvLibros = new DataView(clsLibros.DeleteRegLibros());
    }

    if (dvLibros.Count <= dgCatLibros.PageSize)
        dgCatLibros.CurrentPageIndex = 0;

    Session["dvLibros"] = dvLibros;
    dgCatLibros.DataSource = dvLibros;
    dgCatLibros.DataBind();
}

```

Figura 5.0-8 Muestra código de cancelar y eliminar.

La creación de las otras pantallas es similar al descrito anteriormente.

5.1 Matriz de Pruebas

Las pruebas del sistema es una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan, se registran y se realiza una evaluación al respecto. La tabla 5.1.1 muestra las pruebas realizadas a nuestro sistema.

Check List de Prueba Unitaria Reporte

| No | Descripción | Si | No | Na |
|-----------------------------------|---|----|----|----|
| No Funcional (Estructural) | | | | |
| Estática | | | | |
| 1 | Verificar que se cuenta con acceso a los repositorios de información que requiere el componente en el ambiente de pruebas unitarias definido por Arquitectura | x | | |
| 2 | Verificar que se cuenta con el perfil (autoridades) para la ejecución del componente | x | | |
| 3 | Verificar que los datos requeridos, por diseño para la prueba, están cargados en los repositorios de información (Archivos y Bases de Datos) | x | | |
| 4 | Verificar que el componente considere las reglas de codificación de desempeño | x | | |
| | | | | |
| Dinámica | | | | |
| 1 | Verificar que el componente aplique seguridad transaccional (que solo se muestre la información permitida para el que lo ejecuta) | x | | |
| | | | | |
| Funcional | | | | |
| Estática | | | | |
| 1 | Verificar que el componente está codificado completamente y su compilación libre de errores | x | | |
| 2 | Verificar estándares de codificación | x | | |
| 3 | Verificar la estructura interna del componente (Modularidad, Encapsulamiento) | x | | |
| 4 | Verificar la utilización de los componentes de infraestructura (subsistemas de soporte) | x | | |
| | | | | |

| No | Descripción | Si | No | Na |
|-----------------------------------|---|----|----|----|
| No Funcional (Estructural) | | | | |
| Estática | | | | |
| 1 | Verificar que se cuenta con acceso a los repositorios de información origen y destino que requiere la Interfaz (Work Flow del ETL, PAE o SQR del ERP) en el ambiente de pruebas unitarias definido por Arquitectura | x | | |
| 2 | Verificar que la conexión a el repositorio origen / destino utilice una conexión nativa | x | | |
| 3 | Verificar que se cuenta con el perfil (autoridades) para la ejecución de la Interfaz | x | | |
| 4 | Verificar que los datos requeridos, por diseño para la prueba, están cargados en los repositorios de información origen / destino | x | | |
| 5 | Verificar que el componente considere las reglas de configuración para desempeño | x | | |
| 6 | Verificar que la Interfaz cumple con las especificaciones de manejo de volumen requerido | x | | |
| | | | | |

| | | | |
|------------------|--|---|--|
| Dinámica | | | |
| 1 | Verificar precisión de cálculo decimal en todo el componente | x | |
| | | | |
| Funcional | | | |
| Estática | | | |
| 1 | Verificar que la Interfaz está configurada completamente | x | |
| 2 | Verificar aplicación de reglas de codificación y configuración estándares | x | |
| 3 | Verificar la estructura interna de la Interfaz (Modularidad, Encapsulamiento) | x | |
| 4 | Verificar la utilización de los componentes de infraestructura (subsistemas de soporte) | x | |
| 5 | Verificar que la validación y transformación de datos cumpla con los lineamientos establecidos | x | |
| 6 | Verificar que la Interfaz considera el reproceso de información | x | |
| 7 | Verificar que la Interfaz considera el proceso de extracción parcial de Información | x | |
| | | | |
| Dinámica | | | |
| 1 | Verificar que la Interfaz procesa extracción parcial de información | x | |
| 2 | Verificar que la Interfaz reprocesa información | x | |
| 3 | Verificar que la transformación de la información cumpla con lo requerido por el diseño | x | |
| 4 | Verificar que todos los registros que fueron seleccionados y procesados se carguen | x | |
| 5 | Verificar que todos los datos cargados sean correctos (tipo, tamaño, formato, valor) | x | |
| 6 | Verificar que el formato del reporte de cifras cumple con todas las especificaciones (Layout) | x | |
| 7 | Verificar que el formato de las cifras reportadas sea el correcto (tipo, tamaño, mascara y posición) | x | |
| 8 | Verificar que las cifras reportadas se presenten en el orden correcto | x | |
| 9 | Verificar que se reporten todos las cifras que cumplen las condiciones de la Interfaz | x | |
| 10 | Verificar la congruencia en cálculos (datos calculados, sumarios, subtotales y totales) | x | |
| 11 | Verificar Encabezado, Pie de Página y conteo de página, en cada hoja del reporte de cifras | x | |
| 12 | Verificar las cifras con casos frontera (vacío y campos a su máximo valor) | x | |
| | | | |

Tabla 5.1-1 Matriz de Pruebas.

CAPÍTULO 6

VERIFICACIÓN, LIBERACIÓN Y PRUEBAS

6.1 Guía de Implementación

Resumen del Documento

Propósito

El objetivo de este documento es proporcionar una guía para dejar una aplicación en producción, entregando cuales son los requisitos mínimos indispensables para la correcta instalación.

La audiencia de este documento se muestra en la tabla 6.1-1:

| Audiencia | Propósito |
|------------|--|
| PROFESORES | Conocer los requerimientos mínimos para la puesta en producción del SIPAL, así como los pasos para su configuración. |

Tabla 6.1-1 Audiencia

Tabla de revisiones

La siguiente tabla lista las revisiones hechas a este documento. Usarla sólo para describir los cambios y adiciones cada vez que el documento sea re-publicado. La descripción debe incluir el mayor detalle posible, así como los revisores que requirieron los cambios.

| Fecha | Autor | Descripción de cambios |
|--------------|---------------|------------------------|
| Febrero 2007 | Administrador | Creación |

Contenido

Requerimientos

- Requerimientos de Hardware
- Requerimientos de Software

Descripción de versión de software

- Inventario de materiales
- Inventario de contenidos de software
- Consideraciones de conectividad
- Requerimientos de falla manual o automática

Plan de Implementación

- Equipo de implementación
- Plan de trabajo de la implementación
- Entrenamiento interno

Instrucciones de instalación

- Primera instalación
- Instrucciones de configuración
- Configuración para Administración de requerimientos
- Instrucción de instalación de aplicaciones
- Errores comunes

Aceptación**Requerimientos**

Listar los requerimientos generales para completar una satisfactoria implementación a producción:

- El servidor que hospede la aplicación deberá tener acceso a la Intranet.
- Las personas o sistemas que accedan al SIPAL deberán tener acceso a la Intranet.

Requerimientos de Hardware.

En la tabla 6.1.2 se muestra la descripción y configuración de los requerimientos del hardware.

| Hardware | Descripción | Configuración |
|----------------|--|---|
| Servidores WEB | Procesador: 800 MHz Pentium o superior Memoria: 256 MB Espacio en disco duro: 500 MB | Configuración estándar para el servidor WEB |

| Hardware | Descripción | Configuración |
|------------------------------|---|--|
| | Video: 800 x 600 High Color – 16 bits Red: Ethernet | |
| Servidores de aplicaciones | N/A | |
| Servidores de bases de datos | Procesador: 800 MHz Pentium o superior Memoria: 256 MB Espacio en disco duro: 500 MB Video: VGA Red: Ethernet | Configuración estándar para las bases de datos. 1. Staging_EC 2. Vista |
| Servidores Proxy | N/A | |
| Servidores de respaldo | N/A | |

Tabla 6.1-2 Requerimientos de Hardware

Requerimientos de Software

En la tabla 6.1.3 se muestra la categoría y nombre de los requerimientos del software

| Categoría de software | Nombre del software | Versión | Proveedor |
|---------------------------|------------------------|----------------------------|-----------|
| Sistema operativo | Windows | 2000 Server 2003 Server | Microsoft |
| Servidor WEB | IIS | 5.0 ó superior | Microsoft |
| Framework | .NET Framework | 1.1 | Microsoft |
| Herramienta | Internet Explorer | 6 ó superior | Microsoft |
| Herramienta | Data Access Components | 2.6 ó superior | Microsoft |
| Servidor de base de datos | SQL Server 2000 | Enterprise | Microsoft |

Tabla 6.1-3 Requerimientos de Software

Descripción de versión de software**Inventario de materiales**

Ninguno

Inventario de contenidos de software

En la tabla 6.1.4 se muestra la descripción del inventario de contenidos de software

| Descripción | Contenidos |
|-------------------------------|--|
| Setup.Exe | Ejecutable para instalación de la aplicación |
| Setup.Ini | Configuración para instalación de la aplicación |
| Setup_AdmonRequerimientos.msi | Instalador de la aplicación (es ejecutado por Setup.Exe), está es la aplicación central del sistema de administración de requerimientos. |

Tabla 6.1-4 Inventario de contenidos de software

Consideraciones de conectividad

- La aplicación WEB requiere establecer una conexión directa y constante con el servidor que hospeda las bases de datos de la aplicación
- El servidor WEB debe poder ser visto en la intranet.
- Los usuarios que deseen ingresar al SIPAL, deberán tener acceso a la intranet del Instituto.

Requerimientos de falla manual o automática

En la tabla 6.1-5 se muestra el Hardware y sus requerimientos, para alguna falla que pudiera presentarse.

| Hardware | Requerimientos | Notas |
|--------------|--|-------|
| Servidor Web | Tener un servidor temporal alternativo con las mismas especificaciones descritas anteriormente | |
| Servidor de | N/A | |

| Hardware | Requerimientos | Notas |
|---------------------------|--|-------|
| aplicaciones | | |
| Servidor de Base de datos | Contar con un servidor de respaldo para la base de datos del sistema | |
| Servidor proxy | N/A | |
| Servidor de respaldo | N/A | |

Tabla 6.1-5 Requerimientos de falla manual o automática.

Plan de implementación

Equipo de implementación

| Tipo de miembro | Min # Miembros | Nombre de miembros | Responsabilidades |
|----------------------------------|-------------------|-----------------------|---|
| Líder de proyecto | 1 | Román Abraham | Supervisión de proceso Proporcionar dirección técnica. Suplir de recursos adecuados Reportar a gerencia |
| Administrador del sistema | 1 | Brenda | Asegurar que el ambiente de producción y valoración estén disponibles Administrar el sistema de producción Instalar y administrar el acceso a sistema de producción |
| Representante de desarrollo | 1 | Sonia | Implementar pruebas unitarias Crear clases y paquetes de prueba implementadas en el modelo de pruebas |
| Representante de infraestructura | 1 | Gloria | Ejecuta pruebas Lleva la bitácora de resultados Recuperación de errores |
| Representante de | 1 | Román | Genera plan de pruebas |

| Tipo de miembro | Min # Miembros | Nombre de miembros | Responsabilidades |
|-----------------------------------|-------------------|-----------------------|--|
| Aseguramiento de calidad | | | Genera modelo de pruebas Identifica, prioriza e implementa casos de prueba Evalúa efectividad del esfuerzo de prueba |
| Representante de operaciones | 1 | Abraham | Identifica y define las clases de prueba Identifica y define los paquetes de prueba. |
| Administrador de base de datos | 1 | Brenda | Administra los datos de prueba |

Tabla 6.1-6 Equipo de implementación

Plan de trabajo de la implementación

La tabla 6.1-7 detalla el esfuerzo requerido para cada tarea en el plan de trabajo de la implementación, así como las fechas de su realización.

| Tarea | Esfuerzo requerido | Fecha de inicio | Fecha de fin | Recursos |
|---|-----------------------|--------------------|--------------|----------|
| Plan de implementación | N/A | 24/01/07 | 24/01/07 | |
| Diseño de estrategia de implementación | N/A | 27/01/07 | 27/01/07 | |
| Implementación de estrategia de implantación | N/A | 30/01/07 | 30/01/07 | |
| Ejecuta implementación | 3 hrs | 03/02/07 | 03/02/07 | 1 |
| Evalúa implementación | 1 hr | 05/02/07 | 05/02/07 | 1 |

Tabla 6.1-7 Plan de trabajo de la implementación.

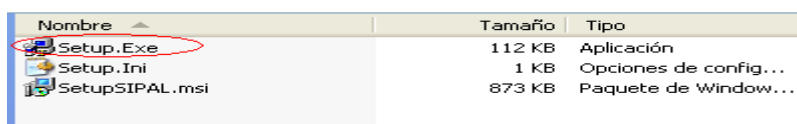
Entrenamiento interno

- No aplica

Instrucciones de instalación

Primera instalación

1. Realizar la verificación de que el(los) equipo(s) que hospedará la aplicación y la base de datos cumplen los requerimientos mínimos de hardware descritos anteriormente
2. Verificar que el servidor WEB cuenta con el siguiente software de acuerdo a los requerimientos de software previamente descritos:
 - a. Sistema operativo
 - b. Servidor WEB
 - c. Internet Explorer
 - d. Data Access Components
3. En caso de que el servidor WEB no cuente con el .NET Framework 1.1, se podrá obtener de la siguiente liga .NET Framework 1.1. Se recomienda que la versión que se instale sea en idioma inglés, sin importar el idioma del sistema operativo. Se realiza la instalación del .NET Framework en caso de ser requerida.
4. Teniendo todos los requisitos mínimos completos se prosigue con la instalación del sistema de Administración de Requerimientos. Para esto, se debe ejecutar el instalador de la aplicación dando doble click sobre SetupSIPAL.msi. Como se muestra en la figura 6.1-1



| Nombre | Tamaño | Tipo |
|----------------|--------|-----------------------|
| Setup.Exe | 112 KB | Aplicación |
| Setup.Ini | 1 KB | Opciones de config... |
| SetupSIPAL.msi | 873 KB | Paquete de Window... |

Figura 6.1-1 Ejecución del instalador de la aplicación

5. Se mostrará la pantalla de instalación del sistema. Como se muestra en la figura 6.1-2

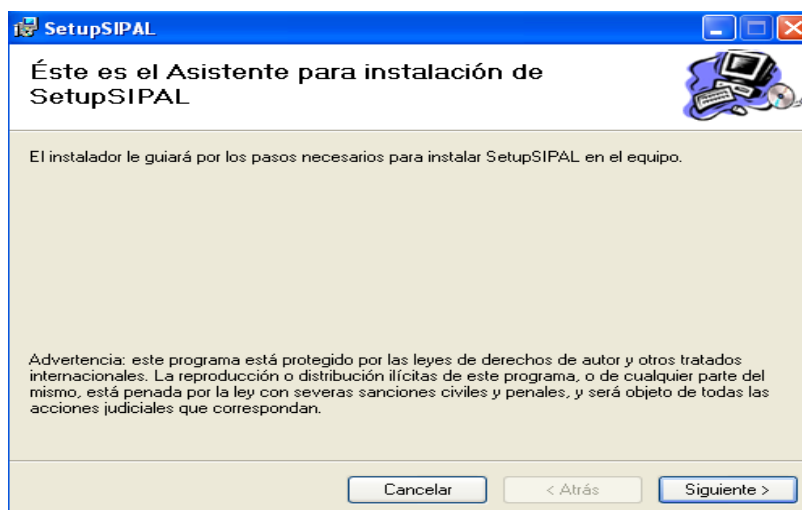


Figura 6.1-2 Pantalla de Instalación

Se presiona el botón de "Siguiente >" para continuar a la próxima pantalla, como se muestra en la figura 6.1-3:

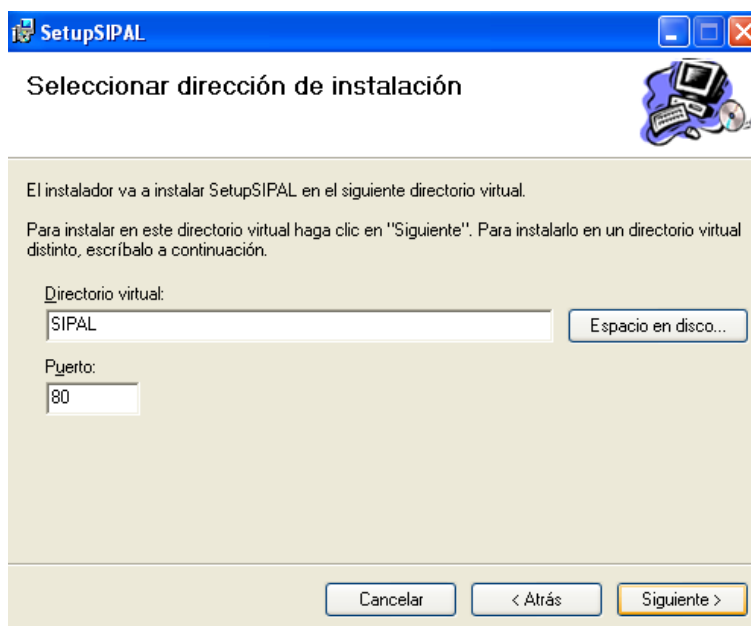


Figura 6.1-3 Seleccionar dirección de instalación

En esta pantalla es posible configurar el Directorio virtual de la aplicación (es el nombre del directorio WEB con el que se hará referencia para acceder a la aplicación) y el Puerto (es el puerto que utiliza el servidor WEB; generalmente se utiliza el 80). Para detalles del manejo de esta información, consulte a su Administrador Web. Recuerde que por default el nombre del directorio virtual es "SetupSIPAL".

Se presiona el botón "Siguiente >" para mostrar la próxima pantalla, que es la figura 6.1-4:

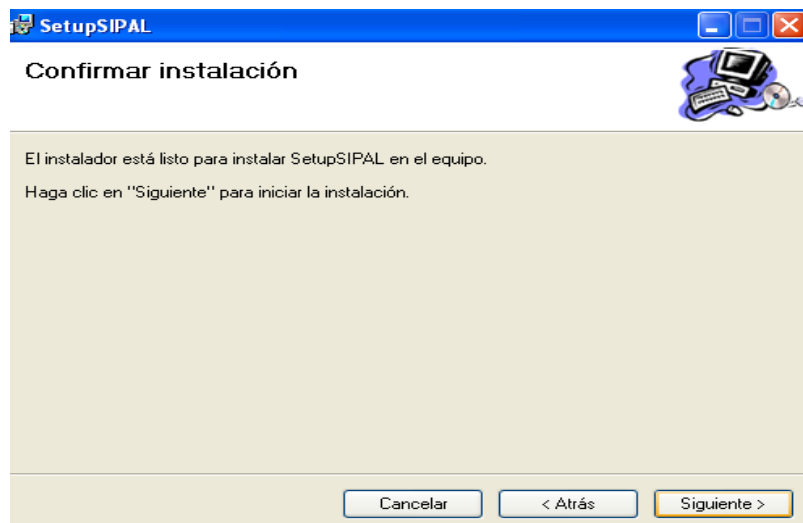


Figura 6.1-4 Confirmar instalación

En esta pantalla se solicita la confirmación para la instalación de la aplicación, una vez estando seguro de realizar la instalación se presiona el botón "Siguiente >":

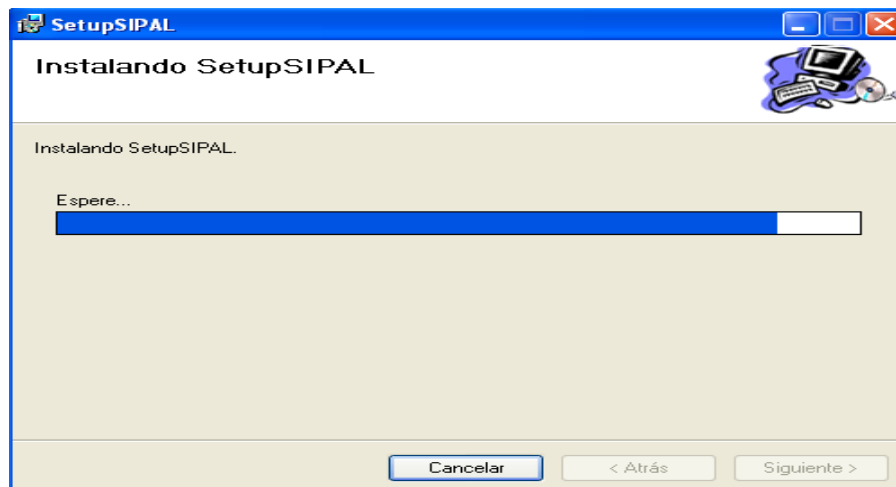


Figura 6.1-5 Instalando SetupSIPAL

Se muestra la pantalla de progreso de la instalación (ver figura 6.1-5) y una vez culminada, se mostrará la pantalla de finalización de la instalación (ver figura 6.1-6):

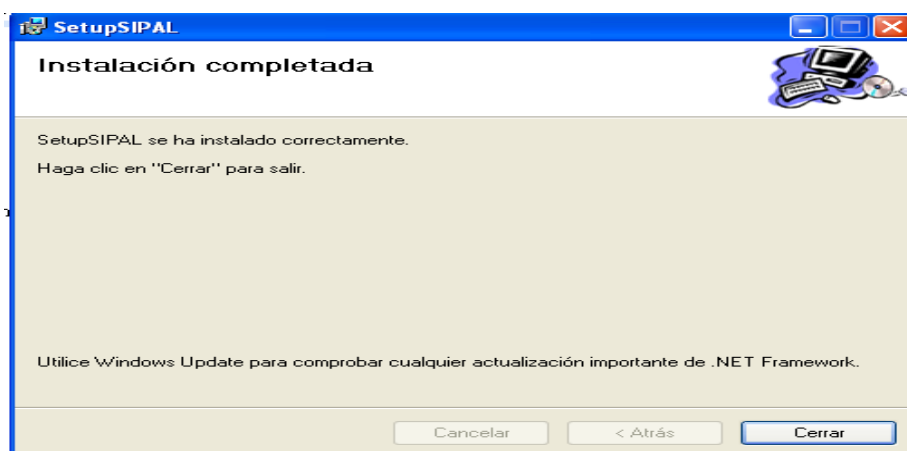


Figura 6.1-6 Instalación Completa

Al mostrar esta pantalla quiere decir que la instalación se ha realizado exitosamente. Se presiona el botón "Cerrar" para culminar el proceso de instalación.

Instrucciones de configuración

| Nombre de software | Instrucciones de configuración |
|--------------------|-----------------------------------|
| SIPAL | Descritas en la siguiente sección |

Configuración para Administración de requerimientos.

1. En caso de que se cambie el usuario o password de la BD, se tendría que modificar el archivo web.config, en la sección de appSettings

Instrucción de instalación de aplicaciones

1. Para la instalación de aplicaciones necesarias (ejemplo: SQL Server, .NET Framework, etc.) se recomienda consultar al administrador de los servidores para que sea él quien lleve a cabo dichas instalaciones.

Errores comunes

- Que los datos en el archivo de configuración de la base de datos (web.config) no sean los correctos. Verificar con el administrador de la base de datos la información proporcionada para establecer la comunicación.

6.2 Instructivo de Administración

Propósito

El objetivo de este documento es proporcionar una guía práctica para entender el correcto funcionamiento del sistema SIPAL, y poder administrarlo correctamente, explicando el como se opera.

La audiencia de este documento se muestra en la tabla 6.2-1:

| Audiencia | Propósito |
|------------|--|
| PROFESORES | Conocer como es la administración del sistema SIPAL. |

Tabla 6.2-1 Audiencia

Tabla de revisiones

La siguiente tabla lista las revisiones hechas a este documento. Usarla sólo para describir los cambios y adiciones cada vez que el documento sea re-publicado. La descripción debe incluir el mayor detalle posible, tabla 6.2-2

| Fecha | Autor | Descripción de cambios |
|--------------|---|------------------------|
| Febrero 2007 | Román Arango Díaz Gloria Angélica Noguez Barajas Brenda Salas Santiago Sonia Estrada Mejía Abraham Varela Benítez | Creación |

Tabla 6.2-2 Revisiones

Ingresando al sitio

El sitio se ubica en la dirección <http://localhost/sipal/login.aspx>

Entiéndase por localhost, el servidor raíz donde radica la aplicación, también puede ser una dirección IP.

Al ingresar aparecerá la figura 6.2-1.



Figura 6.2-1 Pantalla Inicial

Ingresa su Usuario y Contraseña. Este usuario deberá tener el rol de Administrador, cualquier usuario que tenga asignado este rol, podrá tener permisos necesarios para:
Usuario= "Dato" y contraseña= "Clave".

Por seguridad este usuario será permanente y no será editable.

Cabe recalcar que la contraseña es sensitiva, lo cual significa que hay que respetar las mayúsculas y minúsculas en la misma.

Una vez ingresado su usuario y contraseña válidos, dé clic en Entrar, como se muestra en la figura 6.2-2.



Figura 6.2-2 Pantalla para Entrar

A continuación se mostrará la figura 6.2-3 Pantalla Principal, donde podrá elegir 2 opciones a realizar.

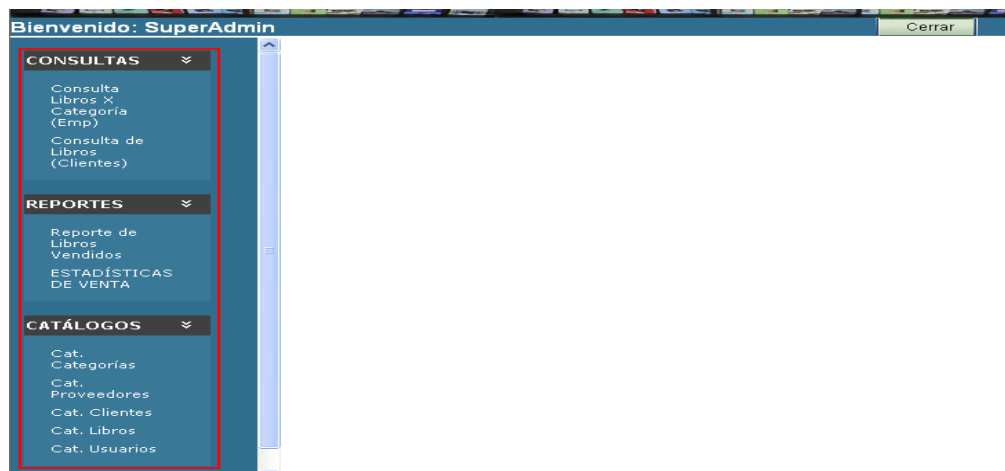


Figura 6.2-3 Pantalla Principal

- Saliendo de la aplicación

Al presionar el botón de cerrar, se le hará la pregunta si está seguro de salir del sistema (como se muestra en la figura 6.2-4). Elija la opción que desee.

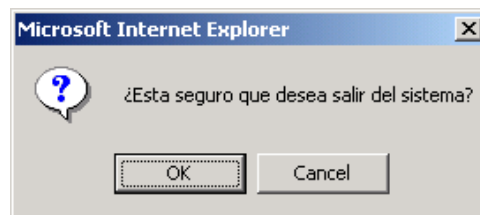


Figura 6.2-4 Saliendo de la Aplicación

- Menú. Figura 6.2-5



Figura 6.2-5 Menú

El usuario “Administrador” podrá tener permisos sobre todos los módulos del sistema, que son los correspondientes al menú, como se observa en la figura 6.2-6.

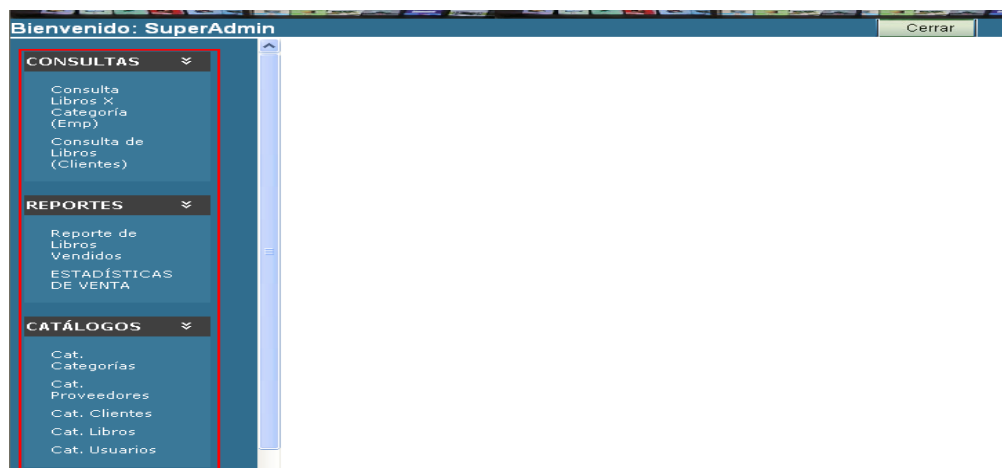


Figura 6.2-6 Permisos sobre módulos

- **Consultas**
 - **Consulta de Libros por categoría**
 - En esta pantalla (figura 6.2-7.) podrá elegir de los combos anidados los criterios de búsqueda, también podrá elegir si los libros buscados están en existencia o no.
 - Una vez elegidos los criterios de búsqueda dé clic al botón de ir para realizar la búsqueda.

CONSULTA DE LIBROS (EMPLEADO)

Consulta por nombre del libro

Consulta por proveedor

Consulta por autor

Consulta por categoría

Consulta por existencia ☒ EXISTENCIA DEL LIBRO

| CLAVE | TÍTULO | VERSION | AÑO | EDITORIAL | AUTOR | CATEGORIA | PRECIO_PUBLICO |
|-------------|-----------------------|---------|------|--------------|--------------------------------|-----------------|----------------|
| MGH11400001 | LAS MIL Y UNA NOCHE | V1 | 1997 | ANAYA | LIDIA CACHO AGARRA | NO APLICA | 183.78 |
| MGH11400002 | LAS MIL Y DOS NOCHES | V2 | 1998 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCIÓN | 183.78 |
| MGH11400003 | LAS MIL Y TRES NOCHES | V3 | 1999 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCIÓN | 183.78 |
| PLT53200004 | CANDIDO PÉREZ | TOMO 1 | 2001 | PLANETA | JUAN RAMÓN DE LA FUENTE FUENTE | COMEDIA | 258.55 |
| PLT53200005 | CANDIDO PÉREZ | TOMO 2 | 2003 | PLANETA | JUAN RAMÓN DE LA FUENTE FUENTE | COMEDIA | 258.55 |
| POR35500006 | APRENDA CHA CHA CHA | CAP A | 1983 | PORRUAAQ | ANDRES MANUEL LÓPEZ OBRADOR | MÚSICA | 223.78 |

Figura 6.2-7 Consulta de libros por Categoría

○ Consulta de Libros para los clientes

- En esta pantalla (figura 6.2-8) podrá elegir de los combos anidados los criterios de búsqueda, también podrá elegir si los libros buscados están en existencia o no.
- Una vez elegidos los criterios de búsqueda dé clic al botón de ir para realizar la búsqueda.
- Esta pantalla es casi idéntica a la anterior con la peculiaridad de que está hecha para poder ser mostrada a los clientes ya que se omiten algunos datos importantes como la clave de los libros.

CONSULTA DE LIBROS (CLIENTE)

Consulta por nombre del libro

Consulta por proveedor

Consulta por autor

Consulta por categoría

Consulta por existencia ☒ EXISTENCIA DEL LIBRO

| TÍTULO | VERSION | AÑO | EDITORIAL | AUTOR | CATEGORIA | PRECIO_PUBLICO | EJEMPLARES |
|-----------------------|---------|------|--------------|--------------------------------|-----------------|----------------|------------|
| LAS MIL Y UNA NOCHE | V1 | 1997 | ANAYA | LIDIA CACHO AGARRA | NO APLICA | 183.78 | 30 |
| LAS MIL Y DOS NOCHES | V2 | 1998 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCION | 183.78 | 25 |
| LAS MIL Y TRES NOCHES | V3 | 1999 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCION | 183.78 | 40 |
| CANDIDO PÉREZ TOMO 1 | | 2001 | PLANETA | JUAN RAMÓN DE LA FUENTE FUENTE | COMEDIA | 258.55 | 10 |
| CANDIDO PÉREZ TOMO 2 | | 2003 | PLANETA | JUAN RAMÓN DE LA FUENTE FUENTE | COMEDIA | 258.55 | 11 |
| APRENDA CHA CHA CHA | CAP A | 1983 | PORRUAGG | ANDRES MANUEL LÓPEZ OBRADOR | MÚSICA | 223.78 | 47 |

Figura 6.2-8 Consulta de libros por Clientes

○ Reportes

○ Reporte de libros vendidos, figura 6.2-9

- En esta pantalla podrá elegir de los combos anidados los criterios de búsqueda, también podrá elegir si los libros buscados están en existencia o no.
- Una vez elegidos los criterios de búsqueda y dados los resultados, podrá realizar una venta.
- Para realizar dicha venta el cliente deberá de tener una clave de cliente, dicha clave tendrá que ser dada de alta en el sistema en el catálogo de usuarios, una vez ingresada la clave de libro a vender, así como la cantidad de los libros a vender de clic al botón de ir para realizar la búsqueda.

VENTA DE LIBROS

Consulta

Consulta por nombre del libro

Consulta por proveedor

Consulta por autor

Consulta por categoría

Consulta por existencia ☒ EXISTENCIA DEL LIBRO

| CLAVE | TITULO | VERSION | AÑO | EDITORIAL | AUTOR | CATEGORIA | PRECIO_PUBLICO |
|-------------|---------------------|---------|------|-----------|--------------------|-----------|----------------|
| MGH11400001 | LAS MIL Y UNA NOCHE | V1 | 1997 | ANAYA | LIDIA CACHO AGARRA | NO APLICA | 183.78 |

VENTA

Clave del Libro

Cantidad de Libros

Clave del Cliente

Figura 6.2-9 Venta de libros

- **Estadísticas de venta**
- Ésta pantalla (figura 6.2-10) tiene el fin de dar el reporte de los libros vendidos.
- Podrá elegir entre el total de los libros vendidos o bien por titulo del libro.

ESTADÍSTICAS DE VENTA

Venta total de libros 18

Venta de libros por parámetros

| TOTAL_VENDIDO | CLAVE | TITULO | VERSION | AÑO |
|---------------|-------------|-----------------------|---------|------|
| 2 | MGH11400001 | LAS MIL Y UNA NOCHE | V1 | 1997 |
| 1 | MGH11400002 | LAS MIL Y DOS NOCHES | V2 | 1998 |
| 7 | MGH11400003 | LAS MIL Y TRES NOCHES | V3 | 1999 |
| 2 | PLT53200005 | CANDIDO PÉREZ | TOMO 2 | 2003 |
| 5 | POR35500006 | APRENDA CHA CHA CHA | CAP A | 1983 |
| 1 | POR35500007 | APRENDA CHA CHA CHA | CAP B | 1980 |

Figura 6.2-10 Estadísticas de Venta.

- **Catálogos**

- **Catálogo de Categorías**

- En este catálogo podrá;
 - Ingresar nuevas categorías
 - Modificar categorías ya ingresadas.
 - Eliminar categorías.

Como se muestra en la figura 6.2-11.

ADMINISTRACIÓN DEL CATÁLOGO DE CATEGORÍAS

| CLAVE_CATEGORIA | DESCRIPCION_CATEGORIA | EDITAR | ELIMINAR |
|-----------------|-----------------------|--|--|
| 0 | NO APLICA |  |  |
| 1 | CIENCIA FICCION |  |  |
| 2 | TERROR |  |  |
| 3 | COMEDIA |  |  |
| 4 | TRAGEDIA |  |  |
| 5 | MÚSICA |  |  |
| 6 | PROGRAMACION |  |  |

Si desea agregar una categoría extra, sólo llene los campos de la parte inferior sin dejar espacios y oprima el botón ACEPTAR

Clave de la Categoría Escriba un número que no esté en el catálogo
 Descripción de la Categoría Escriba la descripción de la categoría. De preferencia verifique que no exista en la Base de Datos

Figura 6.2-11 Catálogo de Categorías.

- **Catálogo de Proveedores**

- En este catálogo podrá;
 - Ingresar nuevos proveedores
 - Modificar proveedores ya ingresados.

Como se muestra en la figura 6.2-12.

| CATALOGO DE PROVEEDORES | | |
|--|--------------|---|
| CLAVE | NOMBRE | |
| ALF | ALFA | Edición |
| ANA | ANAYA | Edición |
| MGH | MC GRAW HILL | Edición |
| New | NuevaEdicion | Edición |
| PLT | PLANETA | Edición |
| POR | PORRUAgg | Actualizar Cancelar |
| < > | | |
| <input type="button" value="Agregar"/> | | |

Figura 6.2-12 Catálogo de Proveedores

○ Catálogo de Clientes

- En este catálogo podrá:
 - Ingresar nuevos clientes
 - Modificar clientes ya ingresados.

Como se muestra en la figura 6.2-13.

| CATALOGO DE CLIENTES | | | | | |
|--|------------------|-----------|-----------|-----------|---|
| CLAVE | NOMBRE | A.PATerno | A.MATerno | DIRECCIÓN | |
| ***** | ABRAHAM | VARELA | BENÍTEZ | | Edición |
| ***** | DANIEL | MUNÍZ | MARTÍNEZ | | Edición |
| ***** | GABRIELA EDUARDA | RIVERA | GARIBALDI | | Edición |
| ***** | NOÉ MIGUEL | VÉLEZ | GUTIÉRREZ | | Edición |
| ***** | Adriana | Varela | Basuto | México DF | Edición |
| XIOMABCDEFGHIJKLMN | XIMENA MIROSLAVA | OSORNO | MARTÍNEZ | | Actualizar Cancelar |
| < > | | | | | |
| <input type="button" value="Agregar"/> | | | | | |

Figura 6.2-13 Catálogo de Clientes

○ Catálogo de Libros

- En este catálogo podrá:
 - Ingresar nuevos libros
 - Modificar libros ya ingresados.

Como se muestra en la figura 6.2-14.

| CATALOGO DE LIBROS | | | | | | | | |
|--------------------|--------------|--------------------------------|-----------------|---------------------------------------|-------------|------------|-----------|-------------------------|
| ID LIBRO | PROVEEDOR | AUTOR | CATEGORIA | TITULO | AÑO EDICIÓN | N° VERSIÓN | CT_PRECIO | |
| ARA360002563 | ALFA | JUAN RAMÓN DE LA FUENTE FUENTE | PROGRAMACION | ANALISIS Y DISEÑO ORIENTADO A OBJETOS | 2006 | 1 | 300 | Edición |
| MGH11400001 | ANAYA | LIDIA CACHO AGARRA | NO APLICA | LAS MIL Y UNA NOCHE | 1997 | V1 | 183.78 | Edición |
| MGH11400002 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCIÓN | LAS MIL Y DOS NOCHES | 1998 | V2 | 183.78 | Edición |
| MGH11400003 | MC GRAW HILL | GABRIEL GARCÍA MÁRQUEZ | CIENCIA FICCIÓN | LAS MIL Y TRES NOCHES | 1999 | V3 | 183.78 | Edición |
| MGH455113365 | MC GRAW HILL | ANDRES MANUEL LÓPEZ OBRADOR | TERROR | Presidencia2006 | 2006 | 1 | 250 | Edición |

Figura 6.2-14 Catálogo de Libros

○ Catálogo de Usuarios

- En este catálogo podrá:
 - Ingresar nuevos usuarios
 - Modificar usuarios ya ingresados.

Como se muestra en la figura 6.2-15.

| CATALOGO DE USUARIOS | | | | | | | | | |
|--|---------|------------|------------|---------------|---------------|--------------------|---------------|-------------------------|--------------------------|
| ID | USER | CONTRASEÑA | NOMBRE | A.PATERNO | A.MATERNO | EMAIL | ROL | EDICIÓN | ELIMINAR |
| 1 | Avarela | ***** | SuperAdmin | Administrador | Administrador | varest@hotmail.com | Administrador | Edición | Eliminar |
| 9 | Sonia | ***** | admin | | | | Consultas | Edición | Eliminar |
| < > | | | | | | | | | |
| <input type="button" value="Agregar"/> | | | | | | | | | |

Figura 6.2-15 Catálogo de Usuarios

CONCLUSIONES GENERALES

Posteriormente a la implementación del sistema, se requiere evaluarlo para determinar si se alcanzaron los objetivos deseados al principio.

Después de algunos meses de trabajo, el proyecto finalizó exitosamente, se realizó la instalación del mismo, llevando a cabo las pruebas necesarias, se puso en marcha el sistema, se comprobó que cumple con los requerimientos señalados por el cliente, tanto a corto, como a mediano plazo.

Al dar por terminado el proyecto, podemos concluir que se cumplieron los siguientes objetivos:

- Se creó un sistema capaz de manejar y controlar el flujo diario de información que se genera en la micro librería.
- Se utilizaron tecnologías web recientes.

Además, tomando en cuenta las necesidades y el crecimiento a futuro de la empresa, se preparó el sistema para soportar la integración de nuevos módulos, como la venta de artículos vía Internet.

Al trabajar en este proyecto, nos dimos cuenta de las necesidades reales de las empresas para actualizarse en el ambiente tecnológico. Aunque nos tomó tiempo de investigación y preparación, finalmente reforzamos nuestros conocimientos adquiridos en la carrera y tuvimos la oportunidad de aplicarlos, dándonos cuenta que aunque no fue fácil llevar a cabo la teoría a la práctica, nos hubiera resultado imposible diseñar el sistema sin fundamentos teóricos.

Por lo mencionado anteriormente, se puede concluir que los resultados son satisfactorios, ya que se alcanzaron las metas propuestas para este trabajo de tesis.

GLOSARIO DE TÉRMINOS

Active Server Pages (ASP). Tecnología del lado del servidor de Microsoft que permite combinar elementos HTML y componentes reutilizables de ActiveX con documentos scripts para crear páginas dinámicas en la Web.

ActiveX. Controles propios del entorno Windows, utilizados en programación. Denominados antiguamente OCX, son módulos separados del programa principal, que se distribuyen con él mismo. Tienen la ventaja de ser portables entre unos y otros lenguajes y su contenido puede ser cualquiera, no tienen una utilidad específica para un desarrollo concreto.

Application Programming Interface (API). Colección de funciones que un entorno concreto (por ejemplo, un Sistema Operativo) pone a disposición del usuario para poder interactuar con él.

Base de Datos. Conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

Cliente/Servidor. Arquitectura a dos capas, es decir, una capa servidor, u computadora que contendrá los datos y los programas gestores asociados, y capas cliente, y computadoras que se dirigirán al anterior para obtener información.

DataBase Management System (DBMS). Es un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

Diagrama. Son una herramienta para el modelado de datos de un sistema de información. Estos diagramas expresan entidades relevantes para el sistema, sus interrelaciones y propiedades.

Framework. Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio

Hardware. Equipo utilizado para el funcionamiento de una computadora. El hardware se refiere a los componentes materiales de un sistema informático. La función de estos componentes suele dividirse en tres categorías principales: entrada, salida y almacenamiento. Los componentes de esas categorías están conectados a través de un conjunto de cables o circuitos llamado bus con la unidad central de proceso (CPU) del ordenador, el microprocesador que controla la computadora y le proporciona capacidad de cálculo.

HTML. HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

HTTP. El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).

Integrated Development Environment (IDE). Programa compuesto por un conjunto de herramientas para el desarrollo de aplicaciones, puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios. Sus principales son un editor de texto, un compilador, un intérprete y un depurador.

Programación Orientada a Objetos (POO). Es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son todas miembro de una jerarquía de clases unidas mediante relaciones de herencia.

Query. Consulta a una base de datos generalmente utilizando sentencias SQL.

Servidor. Computadora o sistema que comparte sus recursos con otras máquinas, denominadas clientes, que se los solicitan.

Software. Conjunto de instrucciones que una computadora emplea para manipular datos: por ejemplo, un procesador de textos o un videojuego. Estos programas suelen almacenarse y transferirse a la CPU a través del hardware de la computadora. El software también rige la forma en que se utiliza el hardware, como por ejemplo la forma de recuperar información de un dispositivo de almacenamiento.

UML. Se le denomina así al Lenguaje unificado de Modelado, basados en los primeros métodos de programación Orientada a Objetos (POO) y está pensado para realizar análisis completos para desarrollo de aplicaciones de dimensiones amplias.

WEB. World Wide Web, la Web o WWW, es un sistema de navegador web para extraer elementos de información llamados "documentos" o "páginas web". Puede referirse a "una web" como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a "la Web", que es la enorme e interconectada red disponible prácticamente en todos los sitios de Internet.

XML. Acrónimo de eXtensible Markup Language (lenguaje ampliable de composición). Permite a los diseñadores y desarrolladores web crear etiquetas personalizadas que ofrecen una mayor flexibilidad a la hora de organizar y presentar la información de la que resulta posible con el sistema HTML de codificación de documentos, que es más antiguo. XML está definido como lenguaje estándar por el W3C y goza de un amplio soporte en la industria.

BIBLIOGRAFÍA

LIBROS:

- AMITI. Desarrollo de software una oportunidad para México Asociación Mexicana de la Industria de Tecnologías de Información, A.C. [En línea] Disponible en: <http://www.amiti.org.mx/biblioteca/EI%20desarrollo%20de%20software%20Una%20oportunidad%20para%20M%C3%A9xico.PDF>
- Booch, Grady (1997) "Quality Software and the Unified Modeling Language" paper Rational Software, 1997
- Castro Tapia, Iván Sebastián (2000) Tesis "Guía Híbrida para la Ingeniería de Software" 2000.
- Date, C. J., Introducción a los Sistemas de Bases de Datos, Volumen I, quinta edición, Editorial Addison Wesley Longman, México 1998, ISBN: 968-444-220-3.
- Laplante Phillip (1999) "Key to Successful Software Development: Selected Readings", IEEE Networking Group 1999.
- Michael Abbey, Michael j. Corey, Ian Abramson. Oracle 8i Guía de aprendizaje. Editorial Oracle Press Osborne. España 2001. ISBN: 0-07-212204-8.
- Paper New Dimensions of Project Management by Rational (www.rational.com)
- Piattini Mario, Adoración de Miguel, Marcos Esperanza. DISEÑO DE BASES DE DATOS RELACIONALES. Ed. Alfaomega.
- Pressman, Roger S. (1998), "Ingeniería de Software, un enfoque practico", Mc Graw Hill. España 1998

- Pressman, Roger S. Ingeniería de Software. Quinta Edición. McGraw-Hill Interamericana. 2002.
- Ralph M. Stair, George W. Reynolds. Principios de sistemas de información. Editorial: Thomson, Cuarta Edición.
- Raya González, Laura. Sistemas operativos en entornos monousuarios y multiusuarios. Alfaomega Grupo Editor
- Royce, Walter (1998) "Software Project Management, a Unified Framework", Addison-Wesley, 1998.
- Silberschatz, Abraham, Korth, Henry F.. y Sudarshan, S., Fundamentos de Bases de Datos, tercera edición, Editorial McGraw Hill, España 1998, ISBN: 0-07-044756-X.
- Sommerville Ian (1989). "Software Engineering", Ed. Addison-Wesley, 1989.
- Ullman, Jeffrey y Widom, Jennifer, Introducción a los Sistemas de Bases de Datos. Editorial Prentice Hall, México 1999, ISBN: 970-17-0256-5.
- Van Vliet Hans (2000). "Software Engineering: Principles and practice", Second Edition, by John Wiley 2000
- Zermeño González Ricardo (2003), "Mercado de Desarrollo de Software", Presentación Select, 2003

SITIOS DE INTERNET:

- http://fmc.axar.net.es/redes/tema_04.htm (Sistemas Operativos).
- <http://www.netwareinformatica.com.ar/ClienteServidor.Asp> (Arquitectura C/S)
- <http://www.netwareinformatica.com.ar/ClienteServidor.Asp>
- http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseno_de_sistemas_web_moder-nos.pdf#search=%22%22sistemas%20web%22%22
- <http://www.informandote.com/jornadasIngWEB/articulos/jiw01.pdf>
- <http://www.dcc.uchile.cl/~luguerre/papers/IS-2003.pdf>
- <http://www.mysql-hispano.org/page.php?id=16&pag=5>
- http://en.wikipedia.org/wiki/Web_server
- <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vsintro7/html/vxconF-older-BasedReference-BasedProjects.asp>
- <http://zarza.fis.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/Mono.pdf>
- <http://di002.edv.uniovi.es/~dani/publications/omnivm.PDF>
- <http://www.mysql.org/>
- <http://office.microsoft.com/es-mx/FX010857913082.aspx>
- www.monografias.com
- www.abcdatos.com
- www.espanol.yahoo.com
- www.ibm.com
- www.zonaoracle.com/manuales-tutoriales-oracle/?id=185
- www.minedu.gob.pe/transparencia/2006/PDFs/008_E_SW_OINFE_S10.pdf -
- www.ilustrados.com/documentos/sghbd.pdf
- www.foroswebgratis.com/tema-manejadores_de_bases_de_datos-20225-125232.htm

ÍNDICE DE FIGURAS

| Figura | Descripción | Página |
|---------|--|--------|
| 1.1.1-1 | Esquema de Fases del Modelo de Cascada..... | 7 |
| 1.1.1-2 | Fases del Modelo Incremental..... | 8 |
| 1.1.1-3 | Esquema de las fases del Modelo Incremental..... | 9 |
| 1.1.1-4 | Esquema de las fases del Proceso Unificado..... | 12 |
| 1.2-1 | Esquema de arquitectura Cliente/Servidor clásica..... | 22 |
| 1.2-2 | Arquitectura Cliente/Servidor en tres capas (three-tier)... | 23 |
| 3.2.-1 | Diagrama de contexto..... | 56 |
| 3.3.1-1 | Pantalla de inicio de sesión..... | 61 |
| 3.3.2-1 | Catálogo de usuarios..... | 62 |
| 3.3.3-1 | Catálogo de libros..... | 63 |
| 4.1-1 | MVC 2 | 66 |
| 4.1-2 | Arquitectura de 3 capas..... | 67 |
| 4.1-3 | Modelo de la Aplicación..... | 68 |
| 4.2-1 | Diagrama de casos de uso general..... | 70 |
| 4.2-2 | Diagrama de casos de uso Gerente..... | 71 |
| 4.2-3 | Diagrama de casos de uso Administrador..... | 72 |
| 4.2-4 | Diagrama de casos de uso Encargado..... | 72 |
| 4.2-5 | Diagrama de casos de uso Secretaria..... | 73 |
| 4.2-6 | Diagrama de casos de uso Externo..... | 73 |
| 4.2-7 | Diagrama de casos de uso Cajero/vendedor..... | 73 |
| 5.0-1 | Diagrama del menú y pantallas del sistema..... | 106 |
| 5.0-2 | Imagen que muestra la creación de un proyecto WEB sobre .NET | 107 |
| 5.0-3 | Muestra el archivo Web.config..... | 107 |
| 5.0-4 | Muestra las opciones para agregar una clase al proyecto | 108 |
| 5.0-5 | Muestra la creación de columnas y configuración de las mismas | 109 |
| 5.0-6 | Muestra código de Agregar..... | 109 |

| | | |
|--------|--|-----|
| 5.0-7 | Muestra código de Modificar..... | 110 |
| 5.0-8 | Muestra código de cancelar y eliminar. | 111 |
| 6.1-1 | Ejecución del instalador de la aplicación..... | 121 |
| 6.1-2 | Pantalla de Instalación..... | 122 |
| 6.1-3 | Seleccionar dirección de instalación..... | 122 |
| 6.1-4 | Confirmar instalación..... | 123 |
| 6.1-5 | Instalando SetupSIPAL..... | 123 |
| 6.1-6 | Instalación Completa..... | 124 |
| 6.2-1 | Pantalla Inicial..... | 126 |
| 6.2-2 | Pantalla para Entrar..... | 127 |
| 6.2-3 | Pantalla Principal..... | 128 |
| 6.2-4 | Saliendo de la Aplicación..... | 128 |
| 6.2-5 | Menú..... | 128 |
| 6.2-6 | Permisos sobre módulos..... | 129 |
| 6.2-7 | Consulta de libros por Categoría..... | 129 |
| 6.2-8 | Consulta de libros por Clientes..... | 130 |
| 6.2-9 | Reporte de Libros Vendidos..... | 131 |
| 6.2-10 | Estadísticas de Venta..... | 131 |
| 6.2-11 | Catálogo de Categorías..... | 132 |
| 6.2-12 | Catálogo de Proveedores..... | 133 |
| 6.2-13 | Catálogo de Clientes..... | 133 |
| 6.2-14 | Catálogo de Libros..... | 134 |
| 6.2-15 | Catálogo de Usuarios..... | 134 |

ÍNDICE DE TABLAS

| Tabla | Descripción | Página |
|---------|--|--------|
| 1.3-1 | Descripción de las Formas Normales..... | 28 |
| 2.2.1-1 | Comparación de los distintos lenguajes web.... | 36 |
| 2.2-1 | Lista de manejadores de Bases de Datos..... | 46 |
| 2.3-1 | Lista de Web servers..... | 49 |
| 3.2-1 | Usuarios y tareas..... | 55 |
| 3.2-2 | Actores y tareas..... | 57 |
| 4.1-1 | Tabla de audiencia..... | 66 |
| 4.1-2 | Revisiones..... | 66 |
| 5.1-1 | Matriz de Pruebas..... | 113 |
| 6.1-1 | Audiencia | 115 |
| 6.1-2 | Requerimientos de Hardware | 117 |
| 6.1-3 | Requerimientos de Software | 117 |
| 6.1-4 | Inventario de contenidos de software | 118 |
| 6.1-5 | Requerimientos de falla manual o automática.. | 119 |
| 6.1-6 | Equipo de implementación..... | 120 |
| 6.1-7 | Plan de trabajo de la implementación..... | 120 |
| 6.2-1 | Audiencia..... | 125 |
| 6.2-2 | Revisiones..... | 125 |