



Universidad Nacional Autónoma De México



Facultad de Ingeniería

**“SISTEMA DE SUPERVISIÓN DE CLIENTES CREDITICIOS
PARA UNA INSTITUCIÓN FINANCIERA”**

T E S I S P R O F E S I O N A L

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

Presentan:

Galicia González Marco Antonio

García Velasco Juan Gabriel

Juárez Briones Adolfo

Martínez Hernández Omar

Muñoz Carrillo Héctor Adrián.

Director de tesis:

M.I. Juan Carlos Roa Beiza

Mayo 2011



Universidad Nacional
Autónoma de México



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:

El agradecer es una palabra que se queda pequeña en estas instancias cuando uno voltea atrás y visualiza a todas aquellas personas que de una u otra forma influyeron y tomaron parte en este meta cumplida, para todos ellos gracias y que los tengo siempre presentes.

Tema aparte es la familia ya que ellos son la base sobre las cuales cimentaremos nuestras propias familias, por ello pilares como la madre, el padre y los hermanos son difíciles de describir pero sin duda cada uno de ellos ha dejado algo en mí ser y también han sido parte del motor que mueve mi vida. De cada uno he aprendido algo y espero seguir haciéndolo, de verdad gracias a todos por que ustedes son el inicio de lo que escribiremos cada uno de nosotros, mamá te amo.

Nunca he creído en la suerte pero si en el trabajo diario y por ello he reescrito las líneas de mis manos para cambiar lo que puedo que es lo que tengo a mi alcance, pero sin dudar un solo momento sin ti no hubiera podido hacerlo, tu dedicación y perseverancia son tus principales virtudes si cabe la palabra en esta descripción y ese ejemplo acompañado de tu amor y presencia son un motivo más para seguir adelante y cambiar lo que está a mi alcance y por qué no visualizar como posible lo que pareciera que no lo es.

Hoy agradezco además el tener conmigo una chispa adecuada que es un buen motivo para ser mejor y esforzarme más para brindarle no solo un futuro mejor si no una mejor persona hoy de la cual se sienta orgullosa mañana, las amo a cada una con todo mi ser.

If you've ever been reversed,
know very well that your feet smell

Adolfo Juárez Briones



Este trabajo se lo dedico a todas las personas involucradas en ella a mis padres por educarme y darme todo su amor a mi papá Sebastián Martínez Cástulo por darme el ejemplo de siempre trabajar para ser alguien de bien a mi mamá Julia Hernández Ruiz que me da fortaleza y siempre me apoya en todos los momentos a mis hermanas Jazmín y Vanessa Martínez Hernández que siempre están en todas mis aventuras no importa de que se traten a mis amigos que siempre estarán ahí para los momentos de la vida, a mis compañeros de tesis que me tuvieron paciencia para la realización de este trabajo, a mi niño Dieguito a ese pequeño pedacito de vida que vino a darme alegría, fuerza y motivos para querer seguir adelante y pensar en un mejor futuro para todos.

Ñayii satniñu
ñukuun itu i ma
chi koo ja kaa ja
ko'o i.

*La persona que se dedica a
Trabajar bien en su terreno,
Siempre tendrá alimento.*

Omar Martínez Hernández



A mi esposa Aydee:

Por compartir su vida conmigo y porque gracias al amor y apoyo que me ha brindado en estos años hemos logrado realizar muchas metas.

A mis Hijos Leo, Emi y Héctor.

Por ser el motor de mi vida y porque ellos son la principal inspiración para poder cerrar este ciclo en mi vida.

A mis padres Raquel y Roberto:

Porque gracias a su esfuerzo siempre me dieron lo necesario para poder tener una educación profesional, a ellos les pertenece gran parte de este logro.

A mis hermanos, Beto, Karina y Rodri.

Porque siempre han sido mis mejores amigos y siempre están junto a mí cuando los necesito.

A las Familias **Reyes Cantero, Rivera Espinoza y Pérez Aguirre** por ocupar un lugar tan especial en mi vida.

Héctor Adrian Muñoz Carrillo



Dedico este logro, como muestra de mi sincero agradecimiento por todo su apoyo, a:

A mi papá y mamá.

A mi hermano y hermana.

A mis amigos.

A mi novia.

Seres queridos que han llenado mi vida y me han impulsado a alcanzar mis metas.

A todos ellos y a la Facultad de Ingeniería de la UNAM, gracias totales.

García Velasco Juan Gabriel



ÍNDICE TEMÁTICO

Capítulo 1	8
“ENTORNO DEL PROBLEMA”	8
1.1 INTRODUCCIÓN. BOSQUEJO A NIVEL GENERAL.....	9
1.2 LOGÍSTICA DE OPERACIÓN ACTUAL.	12
1.3 REGLAS DE NEGOCIO QUE SE DEBEN CUMPLIR PARA MANTENER UNA CARTERA CREDITICIA SANA.	19
1.4 IMPORTANCIA DE LA SUPERVISIÓN DEL CUMPLIMIENTO DE LOS CLIENTES CREDITICIOS.	23
1.5 TOMA DE DECISIONES CON BASE EN LA INFORMACIÓN HISTÓRICA DE CADA CLIENTE CREDITICIO.	27
Capítulo 2	35
“MARCO TEÓRICO”	35
2.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES	36
2.2 METODOLOGÍAS PARA DESARROLLO DE SISTEMAS ORIENTADOS A OBJETOS.	43
2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE ORACLE	56
2.5 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE WEBLOGIC 10.....	71
2.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA ARQUITECTURA CLIENTE-SERVIDOR.....	78
Capítulo 3.....	97
“ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA”	97
3.1 ANÁLISIS DEL PROCESO ACTUAL DE SUPERVISIÓN DE CLIENTES CREDITICIOS.....	98
3.2 RECOPIACIÓN Y ANÁLISIS DE LA INFORMACIÓN.....	103
3.3 LEVANTAMIENTO DE LOS REQUERIMIENTOS GENERALES Y PARTICULARES QUE DEBE CUBRIR LA SOLUCIÓN TECNOLÓGICA A CONSTRUIR.....	113
3.4 IDENTIFICACIÓN DE POSIBLES BLOQUES	117
3.5 JUSTIFICACIÓN DE LA METODOLOGÍA Y DEL SOFTWARE A UTILIZAR	123

Capítulo 4.....	131
“DISEÑO Y CONSTRUCCION DE LA APLICACIÓN”.....	131
4.1 DIAGRAMACIÓN.....	132
4.1.1 DIAGRAMA DE CASOS DE USO.....	132
4.1.2 DIAGRAMA DE MODELOS DE DATOS.....	135
4.1.3 DIAGRAMA ENTIDAD-RELACIÓN.....	136
4.1.4 DIAGRAMA DE CLASES.....	140
4.2 Diccionario de datos.....	144
4.3 DISEÑO Y CONSTRUCCIÓN DEL BACK END.....	150
4.4 DISEÑO Y CONSTRUCCIÓN DEL SITIO WEB.	170
4.5 DISEÑO Y CONSTRUCCIÓN DEL FRONT END.....	184
4.6 INTEGRACIÓN, PRUEBAS Y MANTENIMIENTO.....	193
4.7 GENERADOR DE REPORTES.	200
CONCLUSIONES.....	204
BIBLIOGRAFÍA.	206

Capítulo 1

"ENTORNO DEL PROBLEMA"



1.1 INTRODUCCIÓN. BOSQUEJO A NIVEL GENERAL

Desde la existencia del dinero como moneda de cambio en la vida del hombre siempre ha sido necesario realizar cálculos que nos permitan establecer una relación entre lo que se tiene y lo que se debe. Dicha necesidad ha desembocado en complejos sistemas que van desde el simple préstamo prendario hasta complejas ecuaciones para determinar intereses moratorios, tasas de rendimiento porcentuales, etc.

Durante varios años las instituciones bancarias han implementado diferentes esquemas que les han permitido solventar sus procesos y que con el tiempo fueron adecuados para resolver los problemas que planteaban las reglas del negocio. Las condiciones dinámicas del mercado obligaron a actualizar los sistemas para poder enfrentar los diferentes tipos de operaciones que se fueron manifestando debido a la creciente demanda de créditos.

De esta forma los sistemas que en un inicio cumplieron con su propósito para el cual fueron diseñados, hoy día y ante las necesidades cambiantes del mercado, exigen no sólo mejorarlos si no que también obligan a la creación de nuevos esquemas basados en las tecnologías de punta. La existencia de factores internos y externos que no se pudieron tomar en cuenta en los sistemas que se operaban hace 10 años ha desembocado en la necesidad de llevar a cabo sistemas más flexibles y dinámicos.

De esta forma han surgido sistemas y/o programas que nos permitan tener al día la información para con ello apoyarnos en la toma de decisiones. De ahí la importancia de contar con un sistemas que nos permita el adecuado seguimiento de dichos créditos otorgados y que en función de los parámetros establecidos nos permita ayudar en la toma de decisiones para rechazar, avalar o ampliar según sea el caso una solicitud de crédito, esto es, avalar la solicitud realizada, darle seguimiento a lo largo de la vida del crédito y aportar elementos informativos que nos permita tener una mejor visión sobre la viabilidad de retorno del crédito otorgado.

Con base en el estudio realizado por el BANMEX es necesario llevar a cabo la construcción de un nuevo sistema de créditos, el cual contemplará los diferentes problemas, como son:

- Otorgamiento
- Seguimiento
- Cumplimiento

El cual contemplará los problemas antes mencionados y permitirá una pronta respuesta tomando en cuenta las necesidades del mercado en busca de ofrecer productos y servicios de alta calidad y así mantener el nivel competitivo de la institución.

En esta tesis se presenta el proceso de análisis y desarrollo, así como el uso de diversas herramientas que se utilizará para la creación del módulo de supervisión de clientes crediticios, el cual le permitirá al banco mantener al día el proceso de otorgamiento del crédito consiguiendo subsanar las deficiencias actuales como:

- La cartera vencida
- La pérdida en la captación de recursos (clientes)
- La falta del control del proceso
- La duplicación de información
- La falta de reportes en línea.

Y que nos permite generar la integración institucional de la correcta información para con ello apoyar de una manera adecuada en la toma de decisiones.

Nuestra tesis está dividida en cuatro capítulos; En este primer capítulo presentamos una serie de conceptos básicos sobre los cuales se sustenta el resto del trabajo, como son las reglas del negocio que se deben cumplir para mantener una cartera crediticia sana, la importancia de la supervisión en el cumplimiento de los clientes crediticios y la toma de decisiones en función de la información histórica de cada cliente crediticio. En el capítulo dos se presenta un marco teórico sobre la evolución de los sistemas computacionales y su situación actual, las metodologías

disponibles para el desarrollo de software, la teoría sobre la administración de proyectos tecnológicos y las arquitecturas existentes para la implementación del sistema.

En el capítulo tres se realiza el análisis y planteamiento del problema de acuerdo a las características u áreas de oportunidad detectadas en el proceso actual de la supervisión de clientes, así como la optimización de dicho proceso utilizando el levantamiento de los requerimientos funcionales que debe cubrir la solución al construirla para de esta forma poder elaborar los casos de uso respectivos y definir la arquitectura de la solución dotándolo de un prototipo de la misma.

Para finalmente concluir con el diseño y construcción de la aplicación justificando la metodología utilizada en el desarrollo, la determinación de las tecnologías a utilizar en la implementación de la solución, el porqué del diseño de dicha arquitectura, así como el diseño del modelado de los datos para proseguir con un correcto modelado de los componentes del negocio que se vea reflejado en una interfaz limpia, sobria, pero sobre todo intuitiva y de fácil navegación y que nos permita la construcción de la funcionalidad analizada y diseñada para cada caso.

Con todo esto diremos que el objetivo de este trabajo es describir el proceso a seguir para el desarrollo de un sistema de supervisión de clientes crediticios para una institución financiera, el cual deberá de apoyar y dotar de mayores elementos que les permitan desempeñar su trabajo con mayor eficiencia y rapidez tomando en cuenta los elementos proporcionados para una correcta toma de decisiones.

Por otro lado debemos de tomar en cuenta los componentes involucrados en la construcción del sistema como lo es la base de datos, el lenguaje de programación, el tipo de acceso que nos dará las bases correctas para el desarrollo de este trabajo de tesis.

1.2. LOGÍSTICA DE OPERACIÓN ACTUAL.

Ciclo del crédito.

Etapas del ciclo de crédito

- **Presentación de solicitud y carpetas crediticias.**

El ejecutivo de créditos es el responsable de la veracidad de la información que se anexa en el expediente del solicitante, el cual incluye: la solicitud de crédito, la información económica financiera del negocio o empresa, los documentos de identidad tanto del deudor como del codeudor.

- **Evaluación del crédito por el oficial del crédito.**

La evaluación se hace a través del análisis cuantitativo y cualitativo del crédito. El ejecutivo de crédito prepara y anexa su informe de evaluación, tomando como base la situación económica financiera del cliente, así como el análisis de la moral de pago y sus garantías.

- **Elaboración de informe y recomendaciones.**

Luego de evaluar la documentación, el ejecutivo prepara el informe final debidamente ordenado para ser presentado al comité de créditos.

- **Presentación de solicitud al comité de crédito.**

El ejecutivo presenta al comité de créditos el expediente debidamente documentado y expone al comité evaluador los principales elementos de la solicitud del cliente y su respectiva recomendación.

- **Aprobación del crédito por la institución y notificación al cliente.**

Luego de la resolución del comité, se notifica al cliente sobre la aprobación o denegación del crédito, para proceder a la formalización o archivo del expediente.

- **Elaboración del plan de pago.**

En el plan de pago del cliente se establecen las fechas de pago y de vencimiento.

- **Elaboración de un archivo o carpeta de crédito.**

Se abre un expediente al solicitante con su nombre completo y con los anexos del mismo.

- **Previsión de fondos.**

Es la reserva que establece la institución para la colocación efectiva de los créditos, con el propósito de contar con la disponibilidad de efectivo inmediata al momento del desembolso.

- **Elaboración del contrato de crédito.**

La elaboración del contrato se hace con base a las condiciones negociadas originalmente.

- **Firma del contrato.**

El contrato es firmado por el solicitante, fiadores y representantes de la institución que otorga el crédito.

- **Desembolso.**

Acto de entrega del efectivo a los solicitantes.

Análisis del crédito.

El análisis del crédito implica:

- Conocer la capacidad de pago de los prestatarios, mediante técnicas contables y financieras aplicadas a este tipo de usuarios de crédito.
- Analizar al máximo la moral de pago del prestatario
- Que mediante un mayor y mejor conocimiento del prestatario, el análisis de la garantía sea un elemento complementario al otorgamiento del crédito y no la condición que lo determine.

Primer paso: Visita al prestatario.

Una vez iniciado el proceso de crédito, se concreta la solicitud y se inicia la verificación de los datos proporcionados por el cliente. En esta etapa, se puede identificar si se tiene capacidad de pago o no.

En este contexto, la visita al prestatario constituye la primera fase del análisis del crédito y tiene como objetivo general conseguir las informaciones para el análisis y evaluación de la voluntad y capacidad de pago del solicitante y al mismo tiempo, buscando minimizar el riesgo crediticio.

Finalidades de la visita al prestatario:

- Estrechar los vínculos con el cliente y, a veces, para conocerlo, si es que no se ha tenido la oportunidad de verlo en la institución financiera. Este caso podría ocurrir cuando al ejecutivo de crédito se le asignan los clientes según zonas geográficas o cuando el oficial asume transitoriamente la cartera de clientes de los oficiales que se encuentran de vacaciones.
- Apreciar la importancia verdadera del negocio en la unidad económica del solicitante y sus interrelaciones financieras, la amplitud del local, la cantidad y calidad de los inventarios, de los vehículos, de la maquinaria, de las instalaciones, el orden, la limpieza, que revelan el cuidado que se pone en el manejo del negocio.
- Para el caso de créditos sucesivos, a través de la visita del oficial podría controlar las cifras de los estados financieros del prestatario. Si aún no se contara con informaciones actualizadas, la visita es una magnífica oportunidad para obtener las cifras de los activos tangibles.
- La visita es una buena oportunidad para averiguar los problemas que afectan a la empresa o a la familia: las dificultades que tiene para las ventas, la fabricación, modos de distribución, vacíos de la unidad económica donde se descubren los problemas potenciales de repago. También es una oportunidad para conocer las proyecciones del negocio y detectar las probables dificultades para su crecimiento en el futuro.

Segundo paso: Recopilación de datos.

Lo que se busca es indagar a través de simples aproximaciones sobre su probable capacidad de pago y algunos signos que indiquen la estabilidad de su negocio o empresa. Las informaciones más relevantes en esta primera entrevista son las siguientes:

- Verificar si cumple los requisitos
- Verificar la actividad económica
- La ubicación de la empresa y el domicilio
- El tiempo que lleva funcionando la empresa
- El tamaño aproximado del negocio
- La estrategia de ventas del producto o del servicio
- El monto solicitado
- Destino del crédito
- Cuota deseada de reembolso del crédito
- Experiencia crediticia con otras instituciones financieras
- Nivel de competencia en relación a otros similares

Tercer paso: Análisis de la moral de pago.

En términos crediticios, la moral de pago es la capacidad y voluntad de un prestatario de querer pagar su deuda; ello significa el conocimiento por parte del empresario dueño de una empresa de las ventajas que adquiere al cumplir con todos los compromisos crediticios de acuerdo a lo pactado en los contratos respectivos.

Metodología para determinar la moral de pago:

- Cruce de información.

Toda la información recopilada en la visita a la empresa y el hogar del empresario o empresaria, debe cruzarse con la de otras fuentes o incluso, la misma que haya

ofrecido el prestatario o la prestataria en las diferentes consultas que se le hayan realizado.

La poca consistencia de la información recolectada implica poca seriedad del solicitante de crédito respecto a su voluntad para que conozcamos su negocio, asumiéndose por lo tanto que se esconde información del negocio que pueda perjudicar el otorgamiento del mismo. Esta puede ser: malas referencias crediticias, ventas muy bajas, negocios ilícitos, entre otros.

- Buró de crédito

El buró de crédito es una institución privada que proporciona servicios de recopilación, manejo y entrega o envío de información relativa al historial crediticio de personas físicas y morales.

- Indagación personal

Algo insustituible en un ejecutivo de créditos es su actitud hacia la indagación personal: el ejecutivo de créditos no debe quedarse solamente con la información que le contaron la primera vez o dar por consumados algunos eventos. Es necesario que investigue y compruebe los hechos y conozca las causas de los mismos.

Cuarto paso: Determinación de la capacidad de pago.

El objetivo de esta fase del análisis es minimizar el riesgo crediticio y conocer concretamente si la empresa a financiar arroja un margen de beneficio que le permita autofinanciar su endeudamiento. Otro objetivo es explorar la posibilidad de que el crédito a otorgarse no sea pagado según las condiciones pactadas.

En el análisis de la capacidad de pago, también nos interesa conocer cuál sería el efecto del financiamiento en la empresa del cliente desde el punto de vista económico y financiero.

Alcance y utilidad de los estados financieros:

Las instituciones financieras, a través del análisis de los estados financieros, tratan de detectar los riesgos y estudiar la situación de los negocios de los clientes. Normalmente enfocan los aspectos: patrimonial, financiero, económico y comercial. Por otro lado, los estados financieros se han convertido en exigencia con la finalidad de que las instituciones financieras estudien sistemáticamente los riesgos de sus prestatarios. Estas exigencias se refieren también a ciertas normas para la evaluación, como por ejemplo, indicadores de liquidez, solvencia, rentabilidad y endeudamiento.

Quinto paso: concepto de análisis de garantías.

Características del análisis de garantía:

No cualquier bien puede servir de garantía de un préstamo. En términos generales una garantía debería reunir seis atributos básicos:

- Ser apropiable
- Ser vendible o tener la posibilidad de efectivizarse en dinero líquido para cubrir la operación de crédito
- Tener sentido o valor de pérdida para el prestatario
- Ser durable o sostenible en el tiempo de duración del contrato
- Tener costo de transacción accesible a los prestatarios, al monto y a las condiciones del préstamo
- Que ejerza presión psicológica y moral para el repago de sus obligaciones

Es importante mencionar que en todo crédito es necesario solicitar garantías, pero la presentación de una garantía no implica la concesión de un crédito.

Tipos de garantías:

- Garantía fiduciaria

En este tipo de garantía es donde uno o más fiadores se obligan frente al prestamista a cumplir determinada obligación, en garantía de una obligación ajena

si ésta no es cumplida por el prestatario deudor principal. Es decir, se trata de una obligación accesoria que viene a adherirse a una obligación principal, en virtud de la cual una o más personas responden por una obligación ajena; comprometiéndose para con el prestamista o acreedor a cumplirla en todo o en parte si el deudor principal no cumple.

- Garantía prendaria

La prenda constituye un derecho real que adquiere el prestamista sobre un bien mueble del prestatario. La prenda puede ser con desplazamiento o sin desplazamiento.

El uso de una prenda como garantía requiere de un contrato que explicita claramente el bien que se otorgará como prenda, su localización y datos del prestatario. El valor asignado a la misma es dado tomando en cuenta su valor de mercado, para lo cual puede hacer uso de herramientas como: anuncios clasificados, ventas de muebles usados y la experiencia del analista de crédito.

- Garantía hipotecaria

Se entiende por hipoteca el derecho real que adquiere el prestamista sobre un bien inmueble que el prestatario o deudor otorga como garantía.

Este derecho permanece en manos del prestamista hasta que la deuda sea cancelada, independientemente de quien sea el propietario en dicho momento. El prestatario mantiene el derecho de usufructo del bien hipotecado. Para que un bien sea hipotecado, es imprescindible tener adecuadamente definidos sus derechos de propiedad.

- Concepto de garantía mixta

Consiste en presentar como garantía de un crédito una combinación de los diferentes tipos de garantías antes mencionadas.

1.3 REGLAS DE NEGOCIO QUE SE DEBEN CUMPLIR PARA MANTENER UNA CARTERA CREDITICIA SANA.

Para el desarrollo de una correcta administración de la función de cobro en una empresa, es necesario contar con una estrategia clara y bien definida que nos permita hacer frente a las posibles causas de un retraso en pago, pero también a tomar acciones concretas cuando se presentan a fin de que éstas no se repitan.

La mayoría de las empresas tienen como objetivo principal las ventas y las políticas comerciales están dirigidas a aumentar constantemente la cifra de negocio de las compañías. Por desgracia este objetivo se convierte en el único norte de muchas compañías que destinan grandes esfuerzos a conseguir vender cada vez más pero se descuidan aspectos tan importantes como la rentabilidad de las ventas y los riesgos que suponen los créditos

El análisis de crédito

La estructura de la política de cartera consiste en hacer un análisis estricto y a fondo del cliente, los antecedentes crediticios son en parte la fuente de información que determinará en porcentaje, la venta a crédito

La responsabilidad y control de la cartera de clientes es para ambas áreas, la de ventas y cobranza ya que están ligadas.

Las medidas preventivas tienen su piedra angular en saber seleccionar sabiamente la cartera de clientes. Antes de otorgar un crédito a un nuevo cliente, el responsable de créditos debe filtrar previamente a todos los solicitantes de crédito y descartar los morosos en potencia.

Por consiguiente las empresas proveedoras no pueden permitirse el lujo de confiar en todos los clientes que desean comprar a crédito, ya que la confianza de que se cobrarán las ventas hechas a un nuevo comprador no puede ser otorgada de manera automática. Las empresas antes de realizar una venta a crédito a un nuevo cliente deben comprobar siete puntos básicos:

- 1) Solvencia
- 2) Liquidez
- 3) Resultados
- 4) Nivel de endeudamiento
- 5) Arraigo
- 6) Buenos hábitos de pago
- 7) Historial de los referenciados

- **Solvencia y liquidez**

Tener o contar con medios suficientes para satisfacer las propias deudas y cumplir con los compromisos adquiridos. Prestigio comercial, intelectual o moral.

Se entiende por solvencia a la capacidad financiera (capacidad de pago) de la persona para cumplir sus obligaciones de vencimiento a corto plazo y los recursos con que cuenta para hacer frente a tales obligaciones, o sea una relación entre lo que una persona tiene y lo que debe.

Para que una persona cuente con solvencia, debe estar capacitada para liquidar las deudas contraídas, al vencimiento de los mismos y demostrar también mediante el estudio correspondiente que podrá seguir una trayectoria normal que le permita conservar dicha situación en el futuro.

Se debe diferenciar el concepto de solvencia con el de liquidez, debido a que liquidez es tener el efectivo necesario en el momento oportuno que nos permita hacer el pago de los compromisos anteriormente contraídos.

Solvencia es contar con los bienes y recursos suficientes para respaldar los adeudos que se tengan contraídos, aun cuando estos bienes sean diferentes al efectivo.

Es decir, liquidez es cumplir con los compromisos y solvencia es tener con que pagar esos compromisos, para tener liquidez se necesita tener solvencia previamente.

- **Resultados**

Los resultados de una persona son fundamentales a la hora de conocer su viabilidad y continuidad en el futuro. Para saber si una persona es sólida y tiene posibilidades de futuro es conveniente comprobar su cuenta de pérdidas y ganancias, averiguando sus beneficios y su rentabilidad.

Otro dato de gran interés es el flujo de dinero que ha generado la persona, lo que podemos obtener sumando sus ingresos menos gastos y servirá para conocer su capacidad para generar fondos a través de su actividad ordinaria.

- **Endeudamiento**

El nivel de endeudamiento es un dato muy importante para evaluar la salud financiera de toda persona. Las deudas de una persona no deben por regla general superar el 60 % de sus ingresos, puesto que las personas con mayor índice de endeudamiento están gastando más de lo que ganan.

- **Arraigo**

El arraigo de una persona es el grado de solidez y antigüedad en el medio crediticio, es un dato que sirve como garantía de continuidad en el futuro. Es un elemento cuya cuantificación no siempre es fácil de realizar, no obstante el año de adquisición de su primer crédito es el dato más objetivo que se puede utilizar.

- **Hábitos de pago**

Los hábitos de pago de las personas son un elemento muy importante para decidir la venta a crédito, puesto que una persona puede ser muy solvente y tener una liquidez muy sana, pero retrasarse al máximo en los pagos.

La mejor manera de conocer los hábitos de pago de un posible cliente, es solicitando informes a otras instituciones, ya que son los únicos que podrán informar de manera fiable sobre sus costumbres de pago y su reputación comercial.

- **Historial de los referenciados**

Es conveniente conocer las personas que está referenciando nuestro posible cliente, averiguar sus antecedentes empresariales y su honorabilidad. Así como el currículum crediticio del referenciado son unas buenas pistas para conocer cuál será el comportamiento de esta persona.

En la actualidad Las instituciones Bancarias pueden acceder fácilmente a las informaciones mencionadas en los puntos anteriores gracias a las numerosas empresas de información comercial que existen. Estas empresas de información disponen de bancos de datos, gracias a los cuales pueden proporcionar toda la información existente sobre un posible cliente en muy pocos minutos y a un coste muy reducido. Cuando el riesgo es elevado es recomendable solicitar un informe comercial completo en el que exista investigación añadida sobre la situación actual del posible cliente.

Una vez verificados estos puntos, se puede tomar la decisión final sobre si es conveniente o no conceder un crédito a un nuevo cliente. Gracias a esta simple investigación para cada nuevo cliente, la institución se ahorrará mucho dinero en impagados y evitará vender a morosos o defraudadores.

En caso de existir dudas razonables sobre la solvencia o buen comportamiento de pagos de un posible cliente, es mejor venderle al contado, o en todo caso exigirle un aval bancario que cubra un posible incumplimiento.

Una vez que se ha aceptado a un nuevo cliente como merecedor de un crédito, el segundo paso que ha de dar la empresa suministradora es el de determinar su límite de crédito.

Existen varias maneras de fijar el límite de crédito, y es conveniente que cada empresa desarrolle el más adecuado según las características de su sector y mercado

En realidad las políticas y su estructura va de acuerdo con las normas establecidas de cada empresa lo que se escribe aquí es solo una forma general de tomar una buena decisión para definir a un nuevo cliente.

1.4 IMPORTANCIA DE LA SUPERVISIÓN DEL CUMPLIMIENTO DE LOS CLIENTES CREDITICIOS.

Elementos del control poscrédito

Análisis en los compromisos adquiridos entre la institución y el cliente

Éste es el primer paso que debe dar todo ejecutivo de crédito para conocer de la evolución financiera de la empresa y sobre la base de esta información, detectar la necesidad de realizar un estudio más efectivo acerca de la posibilidad de pago de la empresa en cuestión.

Informes periódicos

A la luz de los cuales se compara la evolución real con la esperada por el cliente. Este tipo de informe no requiere la realización de visitas programadas.

Visitas al cliente

Las visitas al cliente se realizan en forma periódica. Se considera este tipo de control para realizar informes de gestión con un grado mayor de estudio. La definición de las fechas de las visitas se canaliza en la etapa de negociación.

Verificación física de las garantías

La verificación de garantías debe estar acordada en la etapa de negociación, pero, además la institución financiera se debe reservar el derecho de efectuar verificaciones no anunciadas.

Las visitas periódicas del ejecutivo a la empresa deudora o cliente permiten que la institución financiera obtenga una apreciación real de la situación del negocio y por lo tanto, la probabilidad de pago de créditos. Si el ejecutivo mantiene visitas periódicas al cliente, se logra que se identifique la posibilidad de nuevos negocios entre el cliente y la institución.

Ante situaciones que indiquen que la empresa no podrá hacer frente a créditos en los términos establecidos, el comité de créditos, a sugerencia del ejecutivo a cargo de la colocación, puede verse obligado a tomar medidas especiales de control tales como:

- Aumentar garantías, con el fin de mejorar la posible recuperación de créditos, a través de su tercera fuente de pago
- Disminuir la exposición o cuantía de montos otorgados al cliente, por medio de no renovar los créditos de corto plazo
- Exigir la entrega de informes periódicos de la gestión de la empresa y nombrar a un auditor o interventor
- Realizar la ejecución de garantías, como última de las medidas que puede adoptar la institución financiera

Las actividades que pueden llegar a implementar el comité de créditos no tienen por qué seguir algún orden establecido; ellas serán dependientes del nivel del problema en que se encuentre y, en algunas oportunidades, se podrá llevar adelante más de una de las medidas señaladas.

Cabe recordar que una vez que se entregan los fondos al cliente, la acción de control no termina, sino todo lo contrario. Sin embargo, la intensidad de su ejecución dependerá única y exclusivamente del comportamiento que presente el cliente, en conjunto con la apreciación del ejecutivo de créditos respecto a la recuperación de los montos invertidos durante el período establecido para la operación.

Uno de los objetivos del proceso en discusión es contar con los mecanismos tendientes a que los ejecutivos puedan prever situaciones de no pago. Sin

embargo, para que este propósito se pueda cumplir, el control y seguimiento de créditos deben estar basados en una información amplia y verdadera. Lo anterior se basa en que no tiene sentido detectar oportunamente una situación irregular si no se aplican en forma rápida las medidas correctoras que permitan salvaguardar los intereses de la institución financiera.

Un correcto seguimiento y control de los riesgos debe estar basado en hechos concretos, fundados en fuentes de información que permitan al ejecutivo defender su posición frente a sus superiores. Por ello, una de las primeras medidas que se deben tomar ante la sospecha de una situación irregular de un crédito es corroborar la información a la mayor brevedad posible y contar con elementos tanto cuantitativos como cualitativos, que ratifiquen la situación particular de la colocación.

Una vez confirmada una situación irregular es necesario realizar ciertas acciones, tales como:

- Analizar la situación económica de la empresa, al entregar una conclusión respecto a si los problemas detectados son temporales o son consecuencia de alguna situación extraña arrastrada en el tiempo y que se mantendrá.

En caso de ser un problema circunstancial, el ejecutivo dispondrá de elementos cualitativos y cuantitativos que le permitan prever si esta situación se mantendrá, y prestar el apoyo que requiera la empresa, en conjunto con las medidas tendientes a fortalecer la posición de la institución financiera.

- De ser una situación irregular de carácter permanente, se deberá cuantificar y establecer la calidad de las garantías vigentes a favor de la institución, además de estudiar de nuevo la situación y calidad de los documentos ejecutivos y de cobertura que fueron elaborados en la etapa de negociación

Una adecuada vigilancia permite mantener una cartera de clientes sana y selecta, e influye de forma decisiva en una recuperación de las colocaciones como también en los plazos y condiciones inicialmente establecidos.

La unidad operativa es la encargada de los créditos que presentan algún grado de pérdida para la institución; su tarea y finalidad es el inicio de cobro judicial, la renegociación de deuda a corto o largo plazo, los avenimientos judiciales y la mejora de garantías.

Cabe señalar que si, a juicio del ejecutivo de créditos, existe algún peligro para la buena conclusión de una operación crediticia, el comité de crédito decidirá pasar el control de esta colocación a ejecutivos especializados en el manejo de estas situaciones, y tratará de evitar mayores deterioros en la posición de la institución.

Finalidad del seguimiento del crédito

¿Cuál es la finalidad del seguimiento del crédito?

La finalidad del seguimiento es mantener un bajo nivel de morosidad que no afecte la rentabilidad de la cartera ni los niveles de liquidez de la institución.

También debe tener métodos o formas de seguimiento que no tienen que ser uniformes y mucho menos de igual intensidad. Deben ser adaptados a las diferentes circunstancias. Desde el punto de vista tradicional, el seguimiento es el requerimiento de información contable y financiera después de otorgado el crédito.

Cuando se trata de la micro y pequeña empresa, este seguimiento consiste en:

- Visitas sistemáticas al negocio y al domicilio del cliente, según crea conveniente el ejecutivo de crédito.
- Las visitas deberán complementarse con recordatorios escritos tales como cartas, mensajes cortos, entre otros, los que se archivarán en el expediente del prestatario.
- Establecer un sistema de sanciones a través de recargos y acciones judiciales como última fase de recuperación
- Uso eficiente del programa computarizado que diariamente emita un listado

de morosidad según cartera de cada ejecutivo de crédito

- Estar permanentemente actualizado sobre las necesidades de financiamiento, lo que permitirá al ejecutivo de crédito una ágil y oportuna presentación de las solicitudes de refinanciamiento
- El deber de los ejecutivos de crédito es salvaguardar los intereses de la institución financiera y de su propio prestigio. Por lo anterior debe continuar insistente en sus gestiones de cobro, aún cuando se haya iniciado un juicio, asistiendo en todo momento a las instancias correspondientes, con su especial conocimiento del cliente moroso.
- Para aquellos casos de cobranza judicial, deberá archivar en el expediente fotocopias del estado de la cobranza judicial, con la finalidad de sustentar la razonabilidad de las acciones de recuperación.

1.5 TOMA DE DECISIONES CON BASE EN LA INFORMACIÓN HISTÓRICA DE CADA CLIENTE CREDITICIO.

La necesidad del conocimiento de los principales indicadores económicos y financieros, así como su interpretación, son imprescindibles para el mercado competitivo, por lo que se hace necesario aplicar consecuentemente el análisis financiero como base esencial para el proceso de toma de decisiones financieras.

Actualmente los directivos deben contar con una base teórica de los principales métodos que se utilizan para lograr una mayor calidad de los estados financieros para optimizar la toma de decisiones.

El gerente financiero debe de tener en cuenta los siguientes factores:

- a.- La reputación crediticia del cliente
- b.- Referencia de crédito
- c.- Periodos de pago promedio

d.- Persona natural (promedio de ingresos)

e.- Persona jurídica (Estados Financieros)

En el proceso de evaluación de un crédito para una empresa se debe contemplar una evaluación profunda tanto de sus aspectos cuantitativos como cualitativos.

- Es necesario considerar el comportamiento pasado del cliente tanto como cliente de la misma institución como de las demás instituciones.
- La decisión crediticia se le debe tomar en base a antecedentes históricos o presentes.
- Es necesario considerar en los análisis de crédito diferentes consideraciones que se pueden dar con el fin de anticipar los problemas.
- Después de haber realizado un análisis concienzudo del crédito es necesario tomar una decisión por lo que se recomienda escoger 4 ó 5 variables de las tantas que se dieron para su elaboración

Los analistas de crédito usan con frecuencia las cinco C del crédito para centrar su análisis en los aspectos principales de la solvencia de crédito de un solicitante.

- **Reputación:** El registro del cumplimiento de las obligaciones pasadas del solicitante (financiero, contractual y moral). El historial de pagos anteriores, así como cualquier juicio legal resuelto o pendiente contra el solicitante, se utilizan para evaluar su reputación.
- **Capacidad:** La posibilidad del solicitante para Reembolsar el crédito requerido. El análisis de los Estados Financieros, destacando sobre todo la liquidez y las razones de endeudamiento, se lleva a cabo para evaluar la capacidad del solicitante.
- **Capital:** La solidez financiera del solicitante, que se refleja por su posición de propietario. A menudo, se realizan análisis de la deuda del solicitante, en relación con el capital contable y sus razones de rentabilidad, para evaluar su capital.

- **Garantía Colateral:** La cantidad de activos que el solicitante tiene disponible para asegurar el crédito. Cuanto mayor sea la cantidad de activos disponibles, mayor será la oportunidad de que una empresa recupere sus fondos si el solicitante no cumple con los pagos. Una revisión del balance general del solicitante, el cálculo del valor de sus activos y cualquier demanda legal presentada contra el solicitante ayudan a evaluar su garantía colateral.
- **Condiciones:** El ambiente de negocios y económico actual, así como cualquier circunstancia peculiar que afecte a alguna de las partes de la transacción de crédito. Por ejemplo si la empresa tiene un inventario excedente de los artículos que el solicitante desea comprar a crédito, estará dispuesta a vender en condiciones más favorables o a solicitantes menos solventes. El análisis de la situación económica y de negocios, así como el de las circunstancias especiales que pudieran afectar al solicitante o a la empresa, se lleva a cabo para evaluar las condiciones.

El analista de crédito centra su atención, sobre todo, en las dos primeras C (reputación y capacidad); porque representan los requisitos básicos para la concesión del crédito. Las tres últimas (capital, garantía colateral y condiciones), son importantes para elaborar el contrato de crédito y tomar la decisión de crédito final, que depende de la experiencia y el criterio del analista de crédito.

Los dos pasos básicos en el proceso de la investigación del crédito son obtener información de crédito y analizar la información para tomar la decisión del crédito.

- **Obtención de la Información de Crédito.**

Cuando un cliente que desee obtener crédito se acerca a una empresa, normalmente el departamento de crédito da comienzo a un proceso de evaluación del crédito pidiéndole al solicitante que llene diferentes formularios en los cuales se solicita información financiera y crediticia junto con referencias de crédito. Trabajando con base en la solicitud de crédito, la empresa obtiene entonces información adicional del crédito de otras fuentes.

Si la empresa ya le ha concedido anteriormente crédito al solicitante, ya tiene su información histórica acerca de los patrones de pago del solicitante. Las fuentes externas principales de información crediticia es la brindada por los estados financieros, por las oficinas de referencias comerciales, las bolsas de información crediticia, la verificación bancaria y la consulta de otros proveedores.

- **Estados Financieros**

Pidiéndole al solicitante que suministre sus estados financieros de los últimos años, la empresa puede analizar la estabilidad financiera de éste, su liquidez, rentabilidad y capacidad de endeudamiento. Aunque en un Balance o un Estado de resultados no aparece información con respecto a las normas de pago pasadas, el conocimiento de la situación financiera de la empresa puede indicar la naturaleza de la administración financiera total.

La buena voluntad por parte de la empresa solicitante para suministrar estos estados puede ser un indicador de su situación financiera. Los estados financieros auditados son una necesidad en el análisis de crédito de solicitantes que deseen hacer compras importantes a crédito o que deseen que se les abran líneas de crédito.

- **Oficinas de intercambio de referencias (CENTRALES DE RIESGO)**

Las empresas pueden obtener información de crédito por medio de los sistemas de intercambio de referencias, que es una red que cambia información crediticia con base en reciprocidad. Accediendo a suministrar información crediticia a esta oficina de créditos acerca de sus clientes actuales, una empresa adquiere el derecho de solicitar información a la oficina de créditos relacionada con clientes en perspectiva.

Los informes que se obtienen por medio de estas relaciones de intercambio de información crediticia más que analíticos son acerca de casos definidos. Comúnmente se cobran honorarios por cada solicitud.

- **Verificación bancaria**

Puede que sea posible que el banco de la empresa obtenga información crediticia del banco del solicitante. Sin embargo, el tipo de información que se obtiene probablemente es muy vago a menos que el solicitante ayude a la empresa, en su consecución. Normalmente se suministra un estimado del saldo en caja de la empresa.

- **Otros proveedores**

Este consiste en obtener información de otros proveedores que vendan al solicitante del crédito y preguntarles cuáles son las normas de pago y sus relaciones inter-empresariales.

- **Análisis de la Información de Crédito.**

Los estados financieros de un solicitante de crédito y el mayor de cuentas por pagar se pueden utilizar para calcular su plazo promedio de cuentas por pagar. Esta cifra se puede entonces comparar con las condiciones de crédito que la empresa ofrece actualmente. Un segundo paso puede ser el plazo de las cuentas por pagar del solicitante para obtener una idea mejor de sus normas de pago.

Para clientes que solicitan créditos grandes o líneas de crédito, debe hacerse un análisis de razones detallado acerca de la liquidez, rentabilidad y deuda de la empresa utilizando los estados financieros de ésta. Una comparación cíclica de razones similares en años diferentes debe indicar algunas tendencias del desarrollo. Una empresa puede establecer las razones o programas de evaluación de crédito a la medida de sus propios estándares de crédito. No hay procedimientos establecidos, pero la empresa debe encajar su análisis a sus necesidades. Esto da una sensación de confianza que se están tomando los tipos de riesgos de crédito deseados.

Una de las principales contribuciones en la decisión final del crédito es el juicio subjetivo del analista financiero acerca de los méritos que tenga una empresa para el crédito. Para determinar los méritos crediticios, el analista debe agregar sus conocimientos acerca de la índole de la administración del solicitante, las

referencias de otros proveedores y las normas históricas de pagos de la empresa a cualquier cifra cuantitativa que se haya establecido. Basado en su propia interpretación subjetiva de los estándares de crédito de la empresa, puede entonces tomar una decisión final acerca de si se debe conceder el crédito al solicitante y probablemente el monto de éste. Muy a menudo estas decisiones nos la toma una sola persona, sino un comité de revisión de créditos.

Para facilitar la toma de decisiones en una Institución Financiera es necesario contar con una Política de Crédito y Reglamento del Crédito, éstos son de gran importancia para llevar a cabo el proceso del otorgamiento de créditos; de esta manera, se evitan malos entendidos y problemas de insolvencia de los interesados.

Su importancia radica en que es el instrumento que define los lineamientos básicos en el otorgamiento o denegación de un crédito.

Política de Crédito

Las políticas de crédito representan la parte normativa y disciplinaria en una institución y tienen relación directa desde la identificación como sujeto de crédito hasta el otorgamiento del mismo. Por ello, es importante conocer qué es una política, su propósito y cuáles son las principales dentro de una organización.

Definición de política de crédito

Son los lineamientos que rigen y garantizan la eficiente gestión del crédito, los cuales emanan de los lineamientos de los directivos de cada entidad y que reflejan cómo se pretende llegar a los objetivos previstos.

Políticas

Para desarrollar todas las actividades crediticias, las instituciones deben definir su portafolio de políticas y normas de crédito con el propósito de:

- Definir la política que rige la normativa y los procedimientos de crédito.
- Definir las limitaciones del tratamiento a seguir en la atención de las solicitudes durante todo el ciclo crediticio.
- Definir la metodología para la evaluación de las solicitudes de créditos.
- Definir el procedimiento a seguir para la administración de los créditos y de todos los componentes para el control y seguimiento de la recuperación.
- Enmarcarse en los parámetros necesarios para la minimización del riesgo crediticio.

Diferentes políticas crediticias

- La política del manejo de riesgo.
- La política de recuperación.
- La política financiera.

Principios básicos de las políticas de crédito

Los principios básicos de las políticas de crédito son las siguientes:

- El tipo de cliente debe corresponder al mercado objetivo definido por la institución.
- El mercado objetivo de las instituciones financieras implica definir el perfil de los clientes con los que va a operar, así como evaluar el riesgo que está dispuesto a aceptar, la rentabilidad mínima con que se trabajará, el control y seguimiento que se tendrán.

Elementos principales de las políticas de crédito

Monto máximo de los créditos

Determinación de los montos a otorgar estipulando los montos mínimos y máximos de créditos.

Actividades a financiar

Actividades que consideren las instituciones de acuerdo a sus objetivos y principios institucionales; por ejemplo, comercio, industria y servicios.

Destinos

En cuanto a los destinos, éstos los determinará la utilización del dinero prestado.

Plazos

El plazo es el tiempo establecido para que se amortice el capital e intereses del préstamo otorgado. El plazo debe establecerse considerando las necesidades de la empresa y la capacidad de pago.

Forma de pago

Es la periodicidad que se establece para que sea amortizado el crédito. Esta forma de pago puede ser mensual, trimestral, semestral y hasta quincenal, dependiendo del destino del crédito y la capacidad de pago de la empresa.

Tasa de interés

Monto que deberá pagar el deudor por el dinero prestado.

Comisiones y accesorios

Costo en que incurrirá el deudor por el uso del dinero.

Garantías

Aval o respaldo por el dinero prestado.

Capítulo 2

“MARCO TEÓRICO”



2.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES

De forma sencilla podemos decir que una base de datos no es más que un conjunto de información relacionada que se encuentra agrupada o estructurada

El archivo por sí mismo, no constituye una base de datos, sino más bien la forma en que está organizada la información es la que da origen a la base de datos.

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto de datos.

Desde el punto de vista más formal, podríamos definir una base de datos como un conjunto de datos estructurados, fiables y homogéneos, organizados independientemente en una máquina, accesibles en tiempo real, compartibles por usuarios concurrentes que tienen necesidades de información diferente y no predecible en el tiempo. La idea general es que estamos tratando con una colección de datos que cumplen las siguientes propiedades:

- Están estructurados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene.
- Presentan la menor redundancia posible.
- Son compartidos por varios usuarios y/o aplicaciones.

En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una forma normal adecuada.

Características

- Tablas: unidad donde crearemos el conjunto de datos de nuestra base de datos. Estos datos estarán ordenados en columnas verticales. Más adelante se define el concepto de campo y sus características.
- Consultas: En este punto se definen las preguntas que formularemos a la base de datos con el fin de extraer y presentar la información resultante de diferentes formas (pantalla, impresora...).
- Formulario: elemento en forma de ficha que permite la gestión de los datos de una forma más cómoda y visiblemente más atractiva.
- Informe: permite preparar los registros de la base de datos de forma personalizada para imprimirlos.
- Macro: conjunto de instrucciones que se pueden almacenar para automatizar tareas repetitivas.

Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas. Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no están determinados.
- Para cada campo existe un conjunto de valores posible.

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, el ancho del campo, etc.

Los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, el apellido del paciente y la dirección de éste.

Un Campo clave es un campo que permite identificar y localizar un registro de manera ágil y organizada.

Generalmente los diferentes tipos de campos que se pueden almacenar son los siguientes:

- **Texto:** Para introducir cadenas de caracteres hasta un máximo de 255.
- **Memo:** Para introducir un texto extenso. Hasta 65.535 caracteres.
- **Numérico:** Para introducir números
- **Fecha/Hora:** Para introducir datos en formato fecha u hora
- **Moneda:** Para introducir datos en formato número y con el signo monetario
- **Auto numérico:** Con este tipo de campo se numera automáticamente el contenido.
- **Sí/No:** Campo lógico. Este tipo de campo es sólo si queremos un contenido del tipo Sí/No, Verdadero/Falso, etc.
- **Objeto OLE:** Para introducir una foto, gráfico, hoja de cálculo, sonido, etc.
- **Hipervínculo:** Podemos definir un enlace a una página Web
- **Asistente para búsquedas:** Crea un campo que permite elegir un valor de otra tabla o de una lista de valores mediante un cuadro de lista o un cuadro combinado.

Ventajas

- Simplicidad y generalidad.
- Facilidad de uso para el usuario final.
- Períodos cortos de aprendizaje.
- Las consultas de información se especifican de forma sencilla.
- Está su flexibilidad y soporte para el manejo de tipos de datos complejos.
- Puede ajustarse a usar siempre el espacio de los campos que son necesarios, eliminando espacio desperdiciado en registros con campos que nunca usan.
- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.
- Búsquedas más rápidas.
- Se crean formularios e informes que muestren sólo los datos que quiere ver.
- Puede crear cuestionarios para contestar preguntas que son difíciles o imposibles de ser contestadas en las bases de datos simples.

Desventajas

Al considerar la adopción de la tecnología relacional, la inmadurez del mercado constituye una posible fuente de problemas por lo que debe analizarse con detalle la presencia del proveedor para adoptar su producto en una línea de producción sustantiva, El segundo problema es la falta de estándares en la industria. La implantación de una nueva tecnología requiere que los usuarios iniciales acepten cierto riesgo. Aquellos que esperan resultados a corto plazo y con un costo reducido quedarán desilusionados. Sin embargo, para aquellos usuarios que planean a un futuro intermedio con una visión tecnológica avanzada, el uso de tecnología relacional, paulatinamente compensará todos los riesgos.

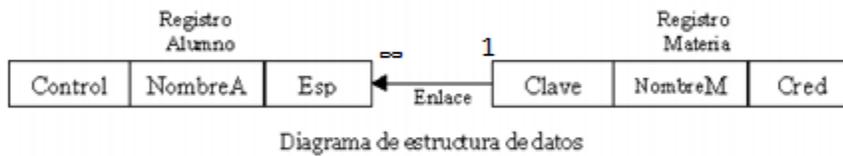
- Pueden ser de instalación compleja, usando muchas tablas.
- Es más difícil entender cómo se relaciona cada parte con la otra.

La cardinalidad es simplemente la forma en que se relacionan las Entidades, o expresa cuantas entidades se relacionan con otras entidades. Las estructuras de datos según la cardinalidad se representan en los siguientes casos como muestra la figura 2.1.

Caso 1. Cardinalidad Uno a Uno.



Caso 2. Cardinalidad Muchos a uno.



Caso 3. Cardinalidad Muchos a muchos.

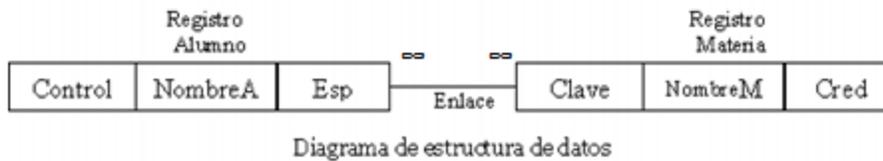


Figura 2.1 Diagrama de estructura de datos

Algebra y calculo relacional

El álgebra relacional y el cálculo relacional son formalismos diferentes que representan distintos estilos de expresión del manejo de datos en el ámbito del modelo relacional. El álgebra relacional proporciona una serie de operaciones que se pueden usar para decir al sistema cómo construir la relación deseada a partir de las relaciones de la base de datos. El cálculo relacional proporciona una notación para formular la definición de la relación deseada en términos de las relaciones de la base de datos.

Es un lenguaje teórico con operaciones que se aplican a una o más relaciones , con el fin de definir otras relaciones sin modificar las relaciones originales , existen operaciones fundamentales que son: selección, proyección, producto cartesiano, unión y diferencia de conjuntos, la operaciones de selección y proyección son operaciones unarias, ya que operan sobre una única relación, existen varias formas de operación de combinación: Combinación Theta, Equicombinación, Combinación Natural, Combinación Externa y Semicombinación.

Combinación Theta.-Define una relación que contenga tuplas del producto cartesiano

Equicombinación.- Produce una relación que contiene las tuplas del producto cartesiano.

Combinación Natural.- Es una equicombinación entre las dos relaciones sobre todo los atributos comunes es decir que tengan el mismo nombre.

Combinación Externa.- Es una combinación en la que también se incluyen en la relación resultante las tuplas de R que no tengan valores correspondientes en los atributos comunes S .La ventaja de una combinación externa es que se preserva la información es decir presera las tuplas que se habrían perdido si se utiliza otros tipos de combinación.

Semicombinación.- Utiliza una combinación de las dos relaciones y luego la proyecta sobre los atributos del primer operando la ventaja es que reduce el número de tuplas que es preciso manejar para formar la combinación.

Las Operaciones de División definen una relación sobre los atributos que está compuesta por el conjunto de tuplas, esta operación resulta útil para un tipo concreto de consulta que suele presentarse con bastante frecuencia en las aplicaciones de las BD.

Las operaciones de Agregación .- Emplea la lista de funciones de agregación definir una relación sobre la lista de agregación, las principales funciones son count, sum, avg, min, max y las Operaciones de Agrupación.- Son aquellas que agrupa las tuplas de la relación según los atributos de agrupación y luego aplica la lista de funciones de agregación.

Formas normales

La normalización es un proceso que pretende conseguir tablas con una estructura óptima y eficaz. El proceso de normalización está basado en lograr la independencia de los datos respecto a las aplicaciones que los usan.

Primera Forma Normal ó 1FN:

La Primera Forma Normal, ó 1FN, es la más elemental de todas. Una tabla está en 1FN si el valor que contiene un atributo de un registro, un campo, es único y elemental. En cada uno de los atributos sólo se puede incluir un dato, aunque sea compuesto, pero no se pueden incluir una lista de datos.

Segunda Forma Normal ó 2FN:

Se dice que un atributo o conjunto de atributos tiene dependencia funcional de otro u otros si a cada uno de los primeros le corresponde sólo uno de los segundos.

Una tabla está en Segunda Forma Normal ó 2FN cuando está en 1FN y todo atributo que no pertenece a la clave primaria tiene una dependencia funcional de la clave completa y no de parte de ella. Luego, si la clave principal está formada por un solo atributo y ya está en 1FN, ya estará en 2FN.

Tercera Forma Normal ó 3FN:

Se dice que hay dependencia funcional transitiva entre dos atributos cuando un atributo que no pertenece a la clave primaria permite conocer el valor de otro atributo.

Una tabla está en Tercera Forma Normal ó 3FN si está en 2FN y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forma parte de la clave primaria, es decir, no hay dependencias funcionales transitivas.

2.2 METODOLOGÍAS PARA DESARROLLO DE SISTEMAS ORIENTADOS A OBJETOS.

A partir del desarrollo del paradigma de orientación a objetos (O.O), son muchas las metodologías que comienzan a aparecer con el fin de facilitar la tarea del análisis y diseño de una aplicación que se va a desarrollar bajo el enfoque de la orientación a objetos. En este apartado explicaremos algunas de las metodologías y notaciones que han tenido un mayor éxito y que son más comúnmente utilizadas por los desarrolladores.

Una metodología está constituida por un proceso de modelado, una semántica, técnicas que lo ponen en práctica para un caso concreto y un lenguaje que la representa haciendo uso de una determinada notación. De esta forma a continuación expondremos un conjunto de metodologías orientadas a objeto que consideramos las más relevantes actualmente. El último subapartado lo hemos dedicado al estándar UML, no considerado como metodología, sino como lenguaje, al ser compendio de varias metodologías y perseguir como principal objetivo la estandarización de la notación a emplear a la hora de hacer ingeniería de software.

Los criterios que se consideran los más importantes para seleccionar una u otra metodología varían, por ejemplo una metodología debe proporcionar un alto grado de flexibilidad, rigor y creatividad. Cuatro criterios que una metodología debe cumplir para considerarse óptima son: versatilidad, funcionalidad, productividad y efectividad.

Toda metodología se apoya en un conjunto de diagramas que responden a una notación concreta. Las distintas metodologías O.O. que existen actualmente difieren entre sí en los aspectos que subraya cada una, como veremos oportunamente en una breve descripción de las técnicas y metodologías que hemos considerado más interesantes por su relevancia en sí y por haber contribuido de alguna forma al desarrollo del estándar UML.

Metodología de Coad-Yourdon

En este apartado se expone una breve descripción de cuatro técnicas orientadas a objetos pioneras en el desarrollo de metodologías. En primer lugar, veremos el "Análisis orientado a objetos" desarrollado por Coad-Yourdon y que se conoce como O.O.A. (Object Oriented Analysis). El objetivo fundamental de O.O.A. consiste en la extracción del modelo de análisis para descubrir la funcionalidad del sistema; este método tiene las siguientes fases de desarrollo:

- Fase 1. Consiste en la extracción de las clases y los objetos del sistema.
- Fase 2. Durante esta fase se identifican las relaciones de generalización/especialización del sistema y las relaciones de composición.
- Fase 3. Se trata de definir áreas, es decir, agrupar objetos según distintos criterios en una entidad superior denominada área.
- Fase 4. En ella se identifican los atributos, que definen cada objeto.
- Fase 5. Consiste en la identificación de los servicios que definen las operaciones de las clases.

Una vez llevadas a cabo estas fases, se dispone de un modelo de análisis listo para llevar a cabo la programación informática del sistema.

Metodología de Booch

La metodología denominada "Diseño orientado a objetos", también conocida por O.O.D. (Object-Oriented Design), desarrollada por Booch, presenta un gran parecido con la anterior, estando formada por una serie de fases, que se pueden resumir en las siguientes:

- Fase 1. Durante esta fase se identifican las clases y objetos a un nivel concreto de abstracción.
- Fase 2. Consiste en la identificación de las semánticas de dichas clases y objetos.
- Fase 3. En esta fase se identifican las relaciones entre clases y objetos.
- Fase 4. En ella se programan las clases y los objetos anteriormente extraídos.

Esta técnica ofrece un gran número de conceptos y diagramas para describir un

sistema, lo que la convierte en una de las metodologías más completas. Algunos de estos diagramas son por ejemplo el diagrama de clases, el de objetos, el de tiempos, el de transición o el de estados, entre otros. El que esta metodología precise tanta diagramación para llevarse a cabo, supone que resulta difícil de aplicar si no se dispone de una herramienta CASE que facilite la labor.

Metodología de Wirfs-Broek

La metodología "responsabilidad-colaboración", también conocida como R.D.D. (Responsability Driven Design) ha sido desarrollada por Rebeca Wirfs-Brock y emplea como técnica CRC (Class Responsibility Collaboration) o técnica de las tarjetas de clase.

Esta metodología consiste en la extracción de las clases del sistema a desarrollar y de la estructura jerárquica de las mismas, es decir, de sus superclases y de sus subclases. A partir de este punto se identifican las responsabilidades y colaboraciones de cada una. En la Figura 2.2.1 se muestra una tarjeta de clase, donde se irán plasmando todos los elementos extraídos, teniendo en cuenta que, entre las ventajas que presenta la elaboración de las tarjetas, se encuentran:

- Las tarjetas son independientes de máquina y del lenguaje.
- Las tarjetas son intuitivas y baratas.

Resumiendo las fases por las que se ha de pasar si empleamos esta metodología son las siguientes:

- Fase 1. En ella se identifican clases y responsabilidades.
- Fase 2. Esta fase asigna responsabilidades a cada clase.
- Fase 3. Durante esta fase se identifican colaboraciones entre las diferentes clases existentes.

La metodología CRC/RDD es insuficiente por sí misma, pero facilita la extracción de las entidades del modelo y sus interacciones, pudiendo servir de gran ayuda en tareas de formación y como complemento de otras metodologías.

Nombre de clase	Súper clase
Responsabilidades Descripción de métodos	
Colaboraciones Relación con otras clases	

Figura 2.2.1 Tarjeta de clase en CRC/RDD

Metodología de Jacobson

La metodología O.O.S.E. (Object-Oriented Software Engineering), desarrollada por Jacobson (1992) emplea como técnica los casos de uso, también llamados formas de utilización (use cases). Los casos de uso han sido posteriormente incorporados a otras metodologías debido a la gran utilidad que presentan.

La metodología O.O.S.E. es realmente la versión reducida de Objctory. Consiste a grandes rasgos en describir un sistema desde el punto de vista del usuario y consta de las siguientes fases:

- Fase 1. En ella se muestran gráficamente las formas de utilización.
- Fase 2. Tiene como objetivo describir las formas de utilización mediante una definición clara de las necesidades del sistema.
- Fase 3. Durante esta fase se lleva a cabo la elaboración de un diagrama de clases.
- Fase 4. Esta fase establece las interacciones existentes en cada forma de utilización.

La metodología OOSE/Objctory es uno de los enfoques más industriales del desarrollo orientado a objetos.

Metodología de Rumbaugh

La técnica de modelado de objetos OMT (Object Modelling Technique) es una metodología de orientación a objetos que cubre el análisis, diseño y programación informática de un sistema y que presenta la variedad más grande de conceptos de todas las metodologías hasta ahora descritas.

Consiste en cuatro fases:

- Fase de análisis.- Su objetivo es la modelización del mundo real, haciendo uso de los modelos que se detallarán más adelante.
- Fase de diseño del sistema.- Durante esta etapa se agrupan en subsistemas los modelos obtenidos en la fase de análisis.
- Fase de diseño de objetos. Consiste en definir cada objeto en detalle, lo que incluye definir su interfaz, algoritmos y operaciones. De nuevo los tres modelos se integran para diseñar los objetos.
- Fase de programación.- Su objetivo es traducir lo obtenido a un lenguaje de programación.

Según el autor, todo sistema tiene aspectos estáticos y dinámicos. El aspecto estático es fácilmente observable, mientras que el aspecto dinámico resulta más complejo de entender, ya que en él intervienen relaciones temporales. La evolución temporal de un sistema es lo que se denomina comportamiento. El desarrollo de cada una de las fases anteriormente expuestas, se da lugar a la formación de tres modelos que van completándose de manera iterativa. Estos tres modelos son el Modelo de Objetos, el Modelo Dinámico y el Modelo Funcional. A continuación se expone en qué consiste cada uno de los modelos anteriormente enumerados.

Modelo de Objetos

El modelo de objetos describe la estructura estática de los objetos de un sistema y sus relaciones. Su objetivo es la elaboración de los diagramas de objetos, consistentes en un grafo cuyos nodos son las clases y cuyos arcos son las relaciones entre las clases. Los elementos que componen el modelo de objetos se detallan en la tabla 2.2.2 Asimismo se matiza que, en la construcción del modelo de objetos, se suceden los siguientes pasos:

- Identificar objetos y clases: Los objetos incluyen entidades físicas abstractas. Todas las clases deben tener sentido en el dominio de la aplicación, y no todas las clases son explícitas del problema, algunas son implícitas. Se pueden cometer errores en este primer paso, considerando

clases a lo que realmente no lo son, por lo que habrá que eliminar clases redundantes, irrelevantes y difusas.

- Preparar un diccionario de datos: Un diccionario de datos comprende todas las entidades del modelo, junto con una pequeña descripción de las mismas.
- Identificar asociaciones (y agregaciones) entre objetos: Cualquier dependencia entre dos o más clases es una asociación. Una referencia de una clase a otra es una asociación. Se pueden cometer errores también en la identificación de las asociaciones, considerando como tales a las asociaciones irrelevantes, derivadas o las existentes entre clases eliminadas.
- Identificar atributos de los objetos: Los atributos son propiedades de los objetos individuales. Durante esta fase se procede a concretar los atributos de que constan las clases y objetos especificados en la primera fase del análisis.
- Organizar y simplificar las clases haciendo uso de la herencia: Se puede descubrir la herencia observando las clases desde arriba hacia abajo cuando tienen similares atributos, asociaciones u operaciones.
- Verificar que los accesos existen para las peticiones de datos: Siempre que se quiera obtener un dato, ha de existir la forma de conseguirlo.
- Iterar y refinar el modelo: Es necesario revisar varias veces el modelo, para asegurar su consistencia.
- Agrupar las clases en módulos: Un módulo es un conjunto de clases que comprende un subconjunto lógico de elementos del modelo. Este sería el momento de concretar los módulos de la aplicación si tuviera.

ELEMENTO	DESCRIPCIÓN
Clase	Grupo de objetos con propiedades y comportamiento idénticos, relaciones comunes con otros objetos y semántica común. Una clase puede ser abstracta si no tiene objetos directamente.
Atributo	Característica que define a los objetos pertenecientes a una clase determinada.
Operación	Función o transformación que se puede aplicar a los objetos de una clase.
Enlace	Conexión física o conceptual entre las instancias de los objetos.
Multiplicidad	Relación que especifica cuántos objetos de una clase se pueden relacionar con un objeto único de una de las clases con la que esté asociada.
Rol	Asociación específica entre dos clases.
Agregación	Asociación que representa una relación de composición.
Herencia múltiple	Herencia que detalla si un objeto hereda características de más de una clase.

Tabla 2.2.2 Elementos que componen el modelo de objetos

Modelo Dinámico

El modelo dinámico describe los aspectos de un sistema que cambia a lo largo del tiempo. Se emplea para especificar y programar los aspectos de control de un sistema. El objetivo del modelo dinámico es la elaboración de los diagramas de estado, que consisten en grafos cuyos nodos son estados y cuyos arcos son transiciones causadas por eventos externos que ocasionan los cambios de estado. El modelo dinámico muestra el comportamiento de los objetos a lo largo del tiempo y sus elementos se representan en la tabla 2.2.3

ELEMENTO	DESCRIPCIÓN
Evento	Transmisión de información en una dirección concreta, de un objeto a otro, en un instante de tiempo. Un evento no tiene duración.
Escenario	Secuencia de eventos que ocurre durante la ejecución particular de un sistema. El ámbito de un escenario puede variar, puede incluir a todos los eventos de un sistema, o a aquéllos que generan ciertos objetos del sistema.
Estado	Abstracción de los valores de los atributos y de Los enlaces de un objeto. El conjunto de valores se agrupa dentro de un estado de acuerdo a las propiedades que afectan al comportamiento del objeto.
Diagrama de estados	Diagrama que relaciona eventos y estados. Cuando se recibe un evento, el estado siguiente depende del estado actual, así como del evento. Un cambio de estado causado por un evento se le llama transición. Un diagrama de estado es un grafo cuyos nodos son estados y cuyos arcos son transiciones entre los eventos.
Condición	Función booleana válida en un intervalo de tiempo. Un estado se puede definir en términos de una condición.
Operación de Control	Operación que gestiona el comportamiento del sistema. Puede ser de dos tipos: actividades y acciones. Una actividad es una operación que para finalizarla lleva su tiempo. La actividad está asociada a un estado. Una acción es una operación instantánea que se asocia con un evento y representa una operación cuya duración es insignificante, comparada con la resolución del diagrama de estados.
Concurrencia	Ocurrencia instantánea de dos eventos sin que exista interacción entre los receptores de los mismos.

Tabla 2.2.3 Elementos que componen el modelo dinámico

La elaboración del modelo dinámico está determinada por las siguientes fases:

- Preparar los escenarios: Un escenario es una secuencia de eventos. Un evento ocurre siempre que existe un trasiego de información entre un objeto del sistema y un agente externo al sistema. Los valores intercambiados constituyen los parámetros del evento.
- Identificar los eventos entre objetos: Examinar los escenarios para identificar todos los eventos externos. Se pueden agrupar juntos los eventos que tienen el mismo efecto en el flujo de control, aunque sus parámetros difieran.
- Construir un diagrama de estados: Preparar un diagrama de estados para cada clase mostrando los eventos que el objeto envía y recibe; cada escenario, o secuencia de eventos, corresponde a un camino a través del diagrama de estados. Es importante destacar que no todas las clases tienen por qué disponer de un diagrama de estados.
- Validar eventos entre objetos: Chequear que los diagramas de estado, para cada clase, están completos y son consistentes.

Modelo Funcional

El modelo funcional describe las transformaciones en los valores de datos dentro del sistema; su objetivo es la creación de diagramas de flujo de datos, que son grafos cuyos nodos son procesos y cuyos arcos son datos. El modelo funcional muestra cómo son procesados los valores. En la tabla 2.2.4 se describen los elementos de que consta este modelo, en el que se siguen los siguientes pasos para la construcción del modelo funcional:

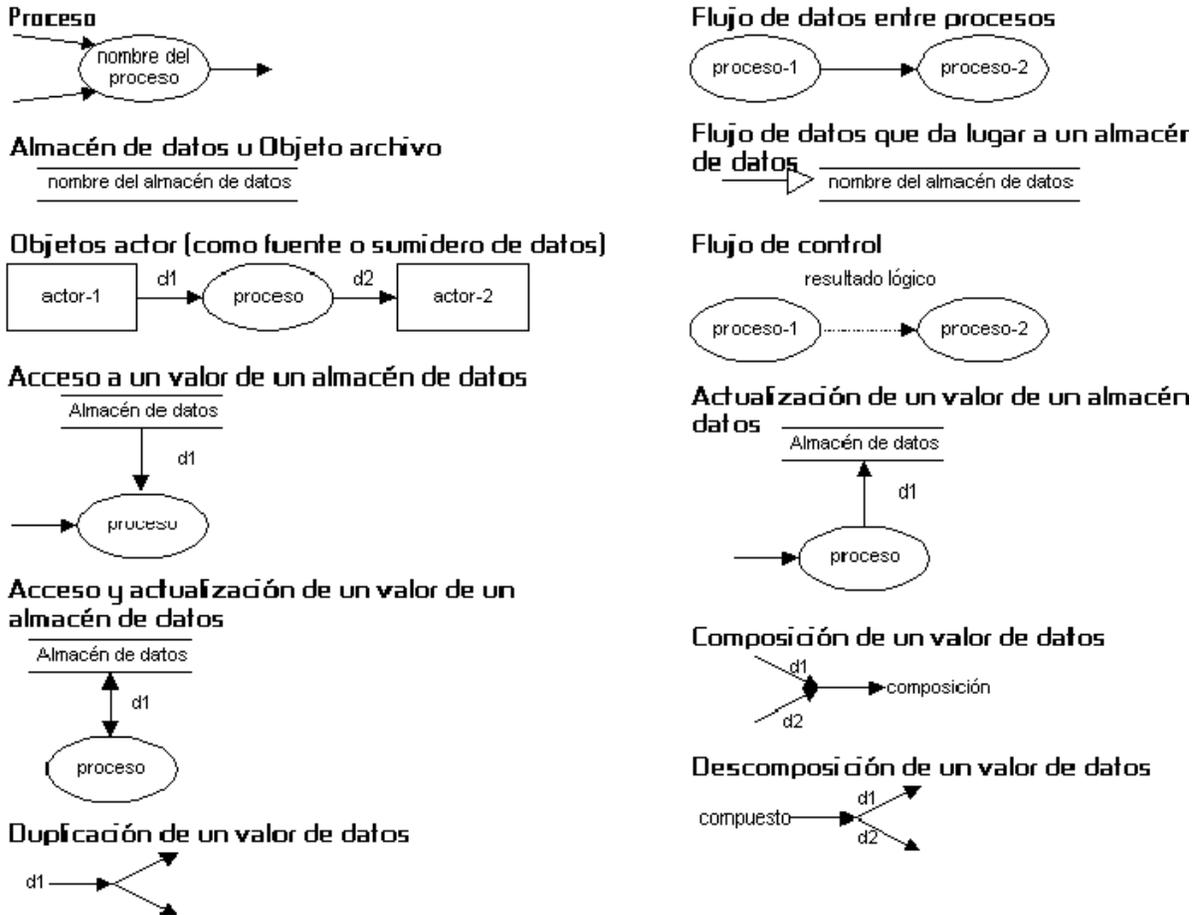


Tabla 2.2.4. Elementos del Modelo Funcional

- Identificar los valores de entrada/salida: Los valores de entrada/salida son parámetros de los eventos entre el sistema y el mundo exterior.
- Construir los diagramas de flujo de datos: Un diagrama de flujo de datos se construye de acuerdo a un conjunto de niveles. En él se detallan las entradas y salidas y los almacenes de datos de los que se recoge información o que se modifican.
- Describir las funciones: Una vez refinados los diagramas de flujo de datos, se escribe una descripción de cada una de las funciones, ya sea matemáticamente, en lenguaje natural o en pseudocódigo. Es necesario centrarse en lo que va a hacer la función.

- Identificar las restricciones: Las restricciones son dependencias funcionales entre objetos que no están relacionadas a una dependencia entrada/salida. Pueden actuar sobre dos objetos simultáneamente, entre instancias del mismo objeto o entre instancias de diferentes objetos.

Lenguaje unificado de modelado

El lenguaje de modelado unificado, UML, comienza su desarrollo en 1994, cuando Booch y Rumbaugh deciden desarrollar un estándar de modelado orientado a objeto, pero es en 1995 cuando aparece UML v.0.8. Posteriormente Jacobson se une aportando su **metodología** O.O.S.E. y creando en 1996 la versión 0.9 de U.M.L y en 1997 la versión 1.0. Las razones por las que se pensó en la posibilidad de elaborar un estándar para el análisis y diseño de un sistema fueron varias:

- Las tres metodologías que dieron lugar a UML (OOO; OMT; OOSE) estaban evolucionando por separado, creándose cada vez mayor distancia entre ellas, tanto en cuanto a semántica como en cuanto a notación.
- Se hacía necesario unificar la semántica y la notación en un estándar.
- Se iba a producir un proceso de retroalimentación entre los autores de las tres metodologías.

UML consiste en un metamodelo y en una notación. Decimos que es un metamodelo al facilitar todas las herramientas necesarias para describir formalmente los elementos de modelado y la sintaxis y la semántica de la notación que permite manipularlos. Decimos que es una notación porque proporcionan los elementos necesarios para representar conceptual y funcionalmente un problema. UML surge, por tanto, como elemento unificador de todas las metodologías orientadas a objetos desarrolladas hasta el momento; de esta forma, al aparecer como estándar entre varias metodologías, presenta una diagramación mixta de todas ellas. UML dispone de dos tipos de formas de ver los datos de un sistema: desde un punto de vista estático y desde un punto de vista dinámico.

El comportamiento particular que tienen los datos se engloban en vistas:

- Vistas estáticas. Responden a una enumeración de las entidades de que consta el sistema. Estas vistas tiene como diagramas: Diagramas de Casos

de Uso, Diagramas de Clases, Diagramas de Objetos, Diagramas de Despliegue y Diagramas de Componentes.

- Vistas dinámicas. Responden a una descripción del comportamiento dinámico de las entidades anteriormente expuestas. Los diagramas de que constan son: Diagramas de Secuencia, Diagramas de Colaboración, Diagramas de Estados y Diagramas de Actividad.

Las vistas estáticas y dinámicas en UML contienen un conjunto de diagramas. A continuación describimos cada uno de los diagramas de que disponen y en la tabla 2.2.5 se puede observar un esquema gráfico de cada uno de ellos.

- Diagramas de Casos de Uso.- Representan la funcionalidad final del sistema. Los diagramas de casos de uso sirven para capturar los requisitos de un sistema y generar los casos de prueba. Estos diagramas derivan de la metodología O.O.S.E.
- Diagramas de Clases. Representan las entidades del sistema y las colaboraciones entre ellas. Derivan de los diagramas de clases de OMT y de O.O.D.
- Diagramas de Objetos.- Representan los objetos del sistema.
- Diagramas de Despliegue.- Capturan la topología de un sistema hardware.
- Diagramas de Componentes. Representan la estructura física de la programación informática; su objetivo es organizar el código fuente, constituir un ejecutable y especificar una base de datos.
- Diagramas de Secuencia. Representan el comportamiento dinámico del sistema, teniendo en cuenta la dimensión temporal
- Diagramas de Colaboración. Representan el comportamiento dinámico del sistema teniendo en cuenta los objetos que intervienen y las interacciones entre ellos.
- Diagramas de Estados.- Representan el comportamiento dinámico del sistema teniendo en cuenta los eventos existentes.
- Diagramas de Actividad.- Se trata de un caso especial de los diagramas de estados. Un diagrama de actividad se corresponde con una clase, una operación o un caso de uso.

UML no es una notación propietaria, siendo accesible por todos de manera totalmente libre, es por lo tanto un estándar abierto. Por otra parte soporta el ciclo de vida de software completo y es suficientemente general como para ser aplicado a diferentes áreas, no sólo al ámbito informático.

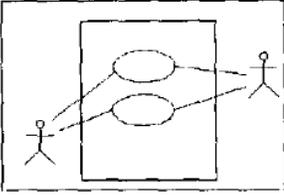
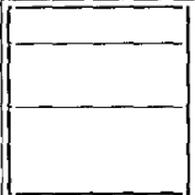
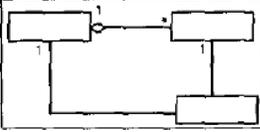
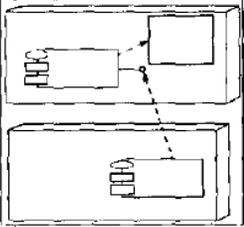
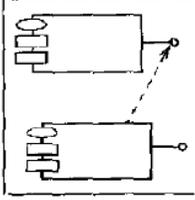
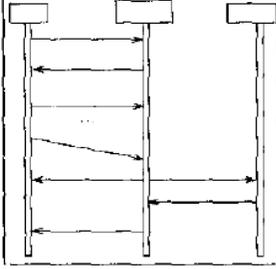
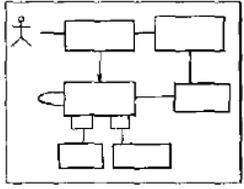
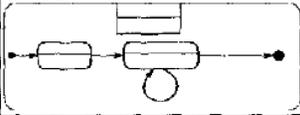
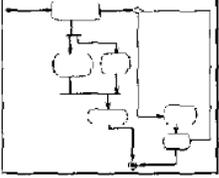
<p>Diagramas de Casos de Uso</p> 	<p>Diagramas de Clases</p> 	<p>Diagramas de Objetos</p> 
<p>Diagramas de Despliegue</p> 	<p>Diagramas de Componentes</p> 	<p>Diagramas de Secuencia</p> 
<p>Diagramas de Colaboración</p> 	<p>Diagramas de Estados</p> 	<p>Diagramas de Actividad</p> 

Tabla 2.2.5 Diagramas propios de UML

2.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE ORACLE

Introducción

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo, en inglés de Relational Data Base Management System), el cual 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, actualmente muchas empresas usan este manejador de base de datos por ser un sistema seguro, estable, escalable, confiable, optimo, multiplataforma, fácil de administrar con herramientas gráficas para la gestión de las bases de datos, monitoreo de datos, monitoreo de usuarios y monitoreo de CPU.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Características

A continuación se mencionan brevemente las características de Oracle y algunas mejoras de su última versión 11g:

Uso de Memoria

El uso de memoria en el RDBMS Oracle tiene como propósito:

- Almacenar los códigos de los programas para empezar a ejecutarse.
- Almacenar los datos necesarios durante la ejecución de un programa.

- Almacenar información sobre cómo es la transferencia entre procesos y periféricos.

Programas y Archivos que componen Oracle

Un RDBMS Oracle está compuesto por tres partes principales, que son:

- El Kernel de Oracle.
- Las instancias del Sistema de Base de Datos.
- Los Archivos relacionados al sistema de Base de Datos.

El Kernel de Oracle

El Kernel es el corazón del RDBMS Oracle, el cual maneja las siguientes tareas:

- Manejar el almacenamiento y definición de los datos.
- Suministra y limita el acceso a los datos y la concurrencia de los usuarios.
- Permite los backup y la recuperación de los datos.
- Interpretar el SQL y PL/SQL.

EL PGA (Programa Global Área)

El PGA contiene información acerca de las conexiones y los procesos que se realizan en Oracle, su tamaño es variable en longitud, pero no es dinámico. El PGA se activa al conectarse un usuario.

EL SGA (System Global Area)

Se podría definir como una serie de buffers en memoria residente, a través de la cual todas las transacciones y el almacenamiento de datos fluyen. El SGA es localizado en memoria al iniciarse una instancia y desaparece al bajarla. Su longitud está definida por los parámetros del archivo de iniciación INIT.ORA.

Está Compuesto por:

- Diccionario Cache
- Los Redo Log Buffers

- Los Database Buffer

Diccionario Cache

Es un conjunto de tablas que son usadas para proveer información asociada con la base de datos, donde se definen los nombres de usuarios, privilegios, objetos, retenciones y espacios que conforman un RDBMS Oracle.

Redo Log Buffers:

Es un espacio reservado en memoria que contiene los cambios realizados a los bloques de datos, permitiendo la reconstrucción de la base de datos en caso de ocurrir un accidente.

Database Buffers:

Es un espacio reservado en memoria para las operaciones normales de la base de datos, el cual depende del tamaño especificado en el archivo de inicialización (INIT.ORA).

Diferentes Herramientas de Oracle

SQLForms: es la herramienta de Oracle que permite, de un modo sencillo y eficiente, diseñar pantallas para el ingreso, modificaciones, bajas y consultas de registros. La herramienta fundamental de SQL es la sentencia SELECT, que permite seleccionar registros desde las tablas de la Base de Datos, devolviendo aquellos que cumplan las condiciones establecidas y pudiendo presentar el resultado en el orden deseado.

Índices

El índice es un instrumento que aumenta la velocidad de respuesta de la consulta, mejorando su rendimiento y optimizando su resultado. El manejo de los índices en ORACLE se realiza de forma inteligente, donde el programador sólo crea los índices sin tener que especificar, explícitamente, cuál es el índice que va a usar.

Manejo de Cursores

El conjunto de filas resultantes de una consulta con la sentencia SELECT, puede estar compuesto por ninguna, una o varias filas, dependiendo de la condición que define la consulta. Para poder procesar individualmente cada fila de la consulta debemos definir un cursor (que es un área de trabajo de memoria) que contiene los datos de las filas de la tabla consultada por la sentencia SELECT.

Los pasos para el manejo de cursores, son:

- Definir el cursor, especificando la lista de parámetros con sus correspondientes tipos de datos y estableciendo la consulta a realizar con la sentencia SELECT.
- Abrir el cursor para inicializarlo, siendo éste el momento en que se realiza la consulta.
- Leer una fila del cursor, pasando sus datos a las variables locales definidas a tal efecto.
- Repetir el proceso fila a fila hasta llegar a la última.
- Cerrar el cursor una vez que se terminó de procesar su última fila.

Disparadores (Triggers)

El módulo SQL*Forms tiene incorporado una colección de procedimientos y funciones llamados "empaquetados" que se pueden incluir en el código de procedimientos o disparadores definidos por el usuario.

El disparador es un bloque de código que se activa cuando se pulsa una determinada tecla u ocurre cierto evento, como puede ser:

- Mover el cursor hacia o desde un campo, registro, bloque o forma.
- Realizar una consulta.
- Validar un dato.
- Hacer una transacción al insertar, modificar o eliminar registros de la base de datos.

Oracle Real Application Clusters (RAC)

Es la principal tecnología para clustering de base de datos que permite a dos o más computadoras (también referidos como “nodos”) de un clúster acceder concurrentemente a una sola base de datos compartida. Esto crea efectivamente un sistema de base de datos único que abarca múltiples sistemas de hardware y aparece frente a la aplicación como una sola base de datos unificada. Esto trae consigo enormes beneficios de disponibilidad y escalabilidad para todas sus aplicaciones, como por ejemplo:

- Tolerancia a fallas dentro del clúster, en especial, a fallas de hardware.
- Flexibilidad y eficiencia de costos en la planificación de capacidad, de manera que un sistema pueda escalar a cualquier capacidad deseada a pedido y a medida que las necesidades del negocio cambian.

Tiempo de Recuperación de la Caída de la Base de Datos

Las fallas del sistema son el resultado de fallas de hardware, fallas de potencia y caídas del sistema operativo o de los servidores.

La tecnología Fast-Start Fault Recovery incluida en Oracle Database limita automáticamente el tiempo de recuperación de la caída de la base de dato. La base de datos ajustará el procesamiento del punto de control para proteger el objetivo de tiempo de recuperación deseado. Esto hace que el tiempo de recuperación sea rápido y previsible, y mejora la capacidad de cumplir con los objetivos del nivel de servicio. Oracle Fast-Start Fault Recovery puede reducir de varios minutos a menos de 10 segundos el tiempo de recuperación de una base de datos muy cargada.

Protección contra Fallas de Almacenamiento

ASM (Automatic Storage Management) se basa en el principio de distribuir y espejar todo. Las capacidades inteligentes de espejamiento permiten que los administradores definan espejos de 2 ó 3 direcciones para la protección esencial

de los datos de negocio críticos. Cuando se producen fallas en el disco, se evita el tiempo de baja del sistema al utilizar los datos disponibles en los discos espejados. Si el disco con fallas se elimina permanentemente de ASM, los datos subyacentes se distribuyen o se vuelven a estabilizar en los discos restantes para continuar brindando un alto desempeño.

Protección contra Fallas del Sitio

Eventos como desastres naturales y cortes de energía o comunicaciones son solo algunos ejemplos de escenarios que pueden causar efectos perjudiciales sobre el centro de datos.

La forma más básica de protección es el almacenamiento *off-site* (fuera del lugar) de los backups de base de datos. A pesar de ser integral para la estrategia HA, el proceso de restaurar backups cuando se produce un desastre que afecta todo el sitio puede llevar más tiempo que el que puede permitirse la empresa, y los backups pueden no contener las versiones más actuales de los datos. Una solución más expedita y completa es administrar una o más copias duplicadas de la base de datos de producción en centros de datos físicamente separados.

Data Guard

Data Guard ofrece la tecnología para implementar y administrar una o más copias standby de la base de datos de producción, ya sea en el centro de datos local o en un centro de datos remoto. Data Guard también funciona de manera transparente en los clúster de Grid ya que los servidores pueden agregarse dinámicamente en la base de datos standby en caso de que se requiera un failover. Data Guard respalda dos tipos de bases de datos standby– bases de datos standby físicas que utilizan la tecnología Redo Apply y las bases de datos standby lógicas que utilizan la tecnología SQL Apply.

En Oracle Database 11g se permite que la recuperación de medios continúe mientras la base de datos standby física se abre en modo solo lectura. Esta nueva capacidad, denominada standby física con Real Time Query, elimina las

desventajas anteriores respecto a abrir la base de datos standby para la actividad de solo lectura– ahora la base de datos standby física permanece en sincronía con la base de datos de producción incluso a medida que brinda servicios a las aplicaciones de solo lectura.

La nueva funcionalidad denominada "Snapshot Standby" permite que una base de datos standby física sea abierta temporalmente para las actividades de lectura-escritura para probar las actividades sin perder la protección ante desastres. Esto permite que los datos de producción permanezcan en un estado protegido en todo momento.

Finalmente, Oracle Database 11g puede aplicar cambios en la base de datos standby en paralelo, mejorando así drásticamente el desempeño.

Además de las mencionadas cuenta con un gran número de características para protección contra errores humanos (Tecnología Flashback), protección contra corrupción de datos (Oracle Hardware Assisted Resilient Data-HARD), reconfiguración online del sistema, parches y actualizaciones online y reorganización online de datos y esquemas.

Ventajas

- Oracle es el motor de base de datos relacional más usado a nivel mundial.
- Puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador.
- Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.

- Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
- ¿Qué hay de los objetos de Oracle? Este sistema ha comenzado a evolucionar en esta dirección, añadiendo tipos de clases, referencias, tablas anidadas, matrices y otras estructuras de datos complejas. Desafortunadamente, la implementación actual de las mismas no ofrece una ventaja clara en eficiencia, como sería de esperar, y sí provocan la incompatibilidad de los diseños que aprovechan las nuevas características con otras bases de datos.
- Oracle es la base de datos con más orientación hacia INTERNET
- Un aceptable soporte

Oracle Database 11g proporciona economías de escala en grids de bajo costo fáciles de administrar, lo que permite:

- Reducciones en los costos de inactividad con una arquitectura de máxima disponibilidad.
- Cambio más rápido de los sistemas de TI mediante Real Application Testing.
- Partición y compresión de los datos para ejecutar consultas más rápidas utilizando menos discos.
- Protección y auditoría seguras de datos, y total recuperación de la información.
- Eliminación de redundancias con Active Data Guard.

Desventajas

- Una de las versiones con más problemas fue Oracle 8. Y es que desde que se lanzó la original de la 8 se sucedieron varias versiones con correcciones, hasta alcanzar la estabilidad en la 8.0.3. El motivo de tantos fallos fue, al parecer, la remodelación del sistema de almacenamiento por causa de la introducción de extensiones orientadas a objetos.
- El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras.

- Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
- También es elevado el costo de la formación, el personal calificado para manejar todas las herramientas es escaso y prepararlos en Centros Oracle es muy costoso, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.
- El costo de mantenimiento es Alto, para lo estable que es.

2.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE JEE 1.6 (STRUTS, HIBERNATE, ECLIPSE, JASPER REPORTS, IREPORTS)

Java es un lenguaje de desarrollo de propósito general orientado a objetos, para realizar todo tipo de aplicaciones, fue introducido por Sun Microsystems en 1995. Uno de los principales elementos que hacen diferente a Java es que usa una máquina virtual para la ejecución de programas lo que permite que las aplicaciones escritas en java se puedan ejecutar en cualquier máquina, independientemente del sistema operativo y de la configuración de hardware. Java se distingue de otros lenguajes, en que es una plataforma completa de desarrollo, consta de un gran conjunto de componentes que se pueden reutilizar y mecanismos para extenderlos, facilitando la vida a los desarrolladores.

Java es una tecnología orientada al desarrollo de software con el cual podemos realizar toda una variedad de aplicaciones.

1. Aplicaciones empresariales.

- Aplicaciones Web usando tecnologías Java: Servlets, JSP.
- Servicios Web.
- Aplicaciones cliente/servidor.

- Aplicaciones distribuidas.

2. Aplicaciones de escritorio con AWT, Swing.

3. Aplicaciones móviles.

Java posee tres tipos de subestándares, cada una es un mundo de tecnología disponible para desarrollar familias de aplicaciones.

- JSE (Java Standard Edition). Plataforma base del lenguaje Java donde se puede hacer aplicaciones Stand Alone, tanto JEE y JME se basan en JSE.
- JEE (Java Enterprise Edition). Plataforma del lenguaje Java que provee una especificación de cómo debe construirse una aplicación empresarial. Esta especificación describe como los servidores de aplicación deben proporcionar seguridad, escalabilidad, portabilidad, consistencia, manejo transaccional robusto e independencia de la plataforma tanto de hardware como de sistemas operativos, y al mismo tiempo para que el desarrollador final pueda desarrollar aplicaciones empresariales con menor esfuerzo.
- JME (Java Micro Edition). Plataforma del lenguaje Java destinada al desarrollo de aplicaciones para dispositivos móviles. JME es una versión reducida del conjunto de clases de JSE aunque con algunas librerías extra pensando sobre todo en el desarrollo de aplicaciones para dispositivos móviles. Se trata de una edición especial debido a las limitaciones de recursos en diversas familias de dispositivos (celulares, PDAs, Pockets, PCs, televisiones, relojes, sistemas de ayuda para automóviles, tarjetas, robótica).

Ventajas de Java.

- Licencias: Java es Open Source (Software libre o Código abierto). La infraestructura para desarrollar en JAVA es gratuita.
- Multiplataforma: Es un lenguaje independiente de la plataforma.

- Documentación. JAVA ofrece mucha documentación de ayuda (herramientas, debuggers, editores), apoyado con un gran número de librerías que son soportadas por diversas empresas como IBM, Oracle, Jakarta.
- Java es un estándar abierto: Es abierto a todo el que quiera participar en la evolución de Java, con esa intención se creó el JCP (organismo que controla la evolución de Java).
- Reutilización. Los desarrollos de una aplicación permiten un alto grado de reutilización de componentes. Java implementa consigo un gran número de clases, incluidas gratuitamente en su entorno de desarrollo.
- Seguridad. Provee diversos mecanismos de seguridad tanto a nivel de código (un programa Java no se ve afectado por virus informáticos) como a nivel de autenticación y autorización de usuarios.
- Es Multihilo. El beneficio de ser multihilo consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.
- Extendible. Se puede extender para soportar otros servidores de aplicaciones y bases de datos.
- Se puede desarrollar con IDEs muy robustos. Algunos de ellos Open-Source como: NetBeans y Eclipse.
- Alta conectividad con Bases de Datos, ERP, Sistemas legados (AS-400, CICS) y otros sistemas.
- Es más solicitado en el mercado de software por su gran cantidad de frameworks (Spring, Struts, Hibernate, etc.) que facilitan el desarrollo de proyectos.
- Soporte. Esta soportado por empresas como Oracle, BEA, IBM, Sun.

Desventajas de Java.

- Curva de aprendizaje larga comparado con otros lenguajes.
- Java es una tecnología que posee una gran variedad de conceptos que pueden confundir (Servlets, JSP, Struts, Java Server Faces, Hibernate, EJB).
- Como existen muchas alternativas en la arquitectura de la aplicación es complicado evaluar cuál es la mejor combinación por lo que se requiere de un arquitecto que sepa elegir la arquitectura más adecuada para la problemática.
- La demanda de desarrolladores con un buen nivel de desarrollo en java es complicado en el caso del mercado Mexicano.
- El cambio de arquitectura puede afectar dramáticamente el desempeño.
- El desconocimiento de los frameworks puede llevar a desarrollar desde cero.
- Hay ambientes que no soportan la última versión de Java.

Struts: Es un framework que implementa el patrón de arquitectura MVC en Java.

Un framework es la extensión de un lenguaje mediante una o más jerarquías de clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas. El framework puede involucrar TagLibraries. El patrón de arquitectura MVC (Model-View-Controller) es un patrón que define la organización independiente del Model (Objetos de Negocio), la View (interfaz con el usuario u otro sistema) y el Controller (controlador del workflow de la aplicación). Vea figura 2.4.1

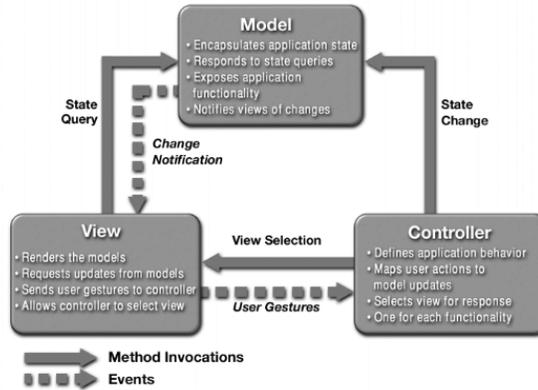


Figura 2.4.1. Arquitectura del MVC en Struts

Ventajas de Struts.

- Simplifica notablemente la implementación de una arquitectura según el patrón MVC.
- Separa la gestión del workflow de la aplicación, del modelo de objetos de negocio y de la generación de interfaz.

Desventajas.

- El controlador ya se encuentra implementado por Struts, lo que limita su utilización.
- Las acciones que se ejecutarán sobre el modelo de objetos de negocio se implementan basándose en clases predefinidas por el framework.

Hibernate: Es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO y convertirá los datos entre los tipos utilizados por Java y los definidos por SQL generando las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Ventajas de Hibernate.

- Ofrece un lenguaje de consulta de datos llamado **HQL** (*Hibernate Query Language*), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "*criteria*").
- Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
- Puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente **Hibernate Annotations** que implementa el estándar JPA, que es parte de esta plataforma.

Desventajas de Hibernate.

- Su configuración requiere de conocimientos avanzados.
- Requiere de un parámetro de configuración normalmente una clase que se encargará de comunicarse con la base de datos en el SQL que entienda la base de datos.
- Dependencia de otras herramientas para alcanzar su máxima utilidad.

Eclipse: Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ).

Ventajas de Eclipse:

- Dispone de un Editor de texto con resaltado de sintaxis.
- La compilación es en tiempo real.
- Requiere de conocimientos avanzados.
- Dependencia de Plugins.

Jasper Reports: Es una herramienta de código libre desarrollada en Java para generar reportes, puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML.

Ventajas de Jasper Reports:

- Está completamente escrita en Java.
- Se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web.
- Permite de forma sencilla la generación de contenido dinámico.

Desventajas de Jasper Reports:

- Se requiere tener instalado en el equipo el JDK 1.4 (SDK) o posterior. No basta con tener instalado el J2RE (Run Time Environment).

- Dependencia de librerías externas para generar los reportes.
- Es complejo en su utilización que se soluciona con la práctica.

iReports: Es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc.

Ventajas de iReports:

- Desarrollado completamente en Java.
- Permite que los usuarios corrijan visualmente informes.
- Está integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java.
- Es 100% compatible con Jasper Reports Server.

Desventajas de iReports:

- Requiere de conocimientos previos para su utilización.
- Requiere de software previo a su instalación.
- No es muy intuitivo en su funcionamiento.

2.5. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE WEBLOGIC 10.

La solución “WebLogic Server”.

Oracle WebLogic Server es una solución escalable, una plataforma Java lista para la empresa, un servidor de aplicaciones Java Enterprise Edition (Java EE). La infraestructura de WebLogic Server soporta el despliegue de muchos tipos de aplicaciones distribuidas y es una base ideal para crear aplicaciones basadas en Arquitecturas Orientadas a Servicios (Service Oriented Architecture SOA). SOA es una metodología de diseño orientado a maximizar la reutilización de los servicios de aplicación.

WebLogic Server es una implementación completa de la especificación Sun Microsystems Java EE 5.0, proporciona un conjunto estándar de APIs para crear aplicaciones Java distribuidas que pueden acceder a una amplia variedad de servicios, tales como bases de datos, servicios de mensajería y conexiones a sistemas empresariales externos. Los clientes usuarios finales acceden a estas aplicaciones usando exploradores web o clientes Java.

Además de la implementación Java EE, WebLogic Server permite a las empresas desplegar aplicaciones de misión crítica en un entorno robusto, seguro, altamente disponible y escalable. Estas características permiten a las empresas configurar grupos de servidores WebLogic (clusters) para distribuir la carga de trabajo y proporcionar capacidad adicional en caso de fallas de hardware u otro tipo. Las nuevas herramientas de diagnóstico permiten a los administradores del sistema monitorear y optimizar el rendimiento de las aplicaciones desplegadas y también el propio entorno de Weblogic. También puede configurar WebLogic Server para monitorear y optimizar el rendimiento de aplicaciones de forma automática sin intervención humana. Extensas funcionalidades de seguridad protegen el acceso a servicios, manteniendo seguros los datos de la empresa y previene ataques maliciosos.

Características de “WebLogic Server”.

WebLogic Server proporciona soporte completo para la especificación Java EE 5.0.

- **Aplicaciones web**

Proporcionan el mecanismo base de Java EE para el despliegue de páginas Web dinámicas basadas en los estándares Java EE Servlets y Java Server Pages (JSP). Las aplicaciones Web se utilizan también para servir contenido Web estático como páginas HTML y archivos de imágenes.

- **Servicios web**

Proporcionan un conjunto compartido de funciones que están disponibles para otros sistemas en una red y pueden ser utilizados como un componente de aplicaciones distribuidas basadas en web.

- **XML**

Incluye el intercambio de datos y un medio para almacenar contenido independiente de su presentación, y mucho más.

- **Java Messaging Service (JMS)**

Permite a las aplicaciones comunicarse entre sí mediante el intercambio de mensajes. Un mensaje es una solicitud, informe y/o evento que contiene información necesaria para coordinar la comunicación entre diferentes aplicaciones.

- **Java Database Connectivity (JDBC)**

Proporciona acceso combinado a recursos DBMS.

- **Adaptadores de recursos**

Proporciona conectividad a sistemas legados y otros sistemas externos de la empresa.

- **Enterprise JavaBeans (EJB)**

Ofrece objetos de Java para encapsular datos y lógica empresarial.

- **Remote Method Invocation (RMI)**

Es el estándar de Java para la computación de objetos distribuidos, permitiendo a las aplicaciones invocar métodos en objetos remotos a nivel local.

- **APIs de seguridad**

Permite integrar autenticación y autorización en las aplicaciones Java EE. También se pueden utilizar las APIs de seguridad para crear proveedores de seguridad propios.

- **WebLogic Tuxedo Connectivity (WTC)**

Proporciona interoperabilidad entre aplicaciones en WebLogic Server y servicios Tuxedo. WTC permite a los clientes de WebLogic Server invocar servicios Tuxedo y clientes Tuxedo invocar EJBs en respuesta a una solicitud de servicio.

Ventajas de “WebLogic Server”.

- **Alta Disponibilidad**

Las siguientes características de WebLogic Server y herramientas apoyan la implementación de aplicaciones altamente disponibles y escalables:

- **Clusters:** Proporciona escalabilidad y confiabilidad mediante la distribución de la carga de trabajo entre múltiples instancias de WebLogic Server. Las peticiones entrantes pueden ser dirigidas a una instancia de WebLogic Server en el clúster basado en el volumen de trabajo en proceso. En caso de fallas de hardware o de otro tipo, el estado de sesión está disponible para los otros nodos del clúster que puede reanudar el trabajo del nodo que ha fallado. Además, se puede implementar clusters para que los servicios pueden ser alojados en una sola máquina con opciones para migrar el servicio a otro nodo en caso de fallo.
- **Administradores de Trabajo:** Priorizan el trabajo basado en reglas definidas y mediante las estadísticas actuales del monitoreo del rendimiento en tiempo de ejecución. Esta información se utiliza para optimizar el rendimiento de las aplicaciones. Los administradores de trabajo se puede

aplicar a nivel mundial a un dominio de WebLogic Server o una aplicación específica o componente.

- Protección de sobrecarga: Proporciona la capacidad de detectar, evitar y recuperarse de condiciones de sobrecarga.
- Canales de redes: Facilita el uso eficaz de los recursos de red mediante la segregación del tráfico de la red en canales basados en el tipo de tráfico.
- Almacenamiento persistente: Es una solución de almacenamiento integrada y de alto rendimiento para subsistemas de WebLogic Server y los servicios que requieren persistencia.
- Servicios de almacenamiento y reenvío: Permite que WebLogic Server envíe mensajes de forma fiable entre aplicaciones que están distribuidas a través de servidores WebLogic. Si el destino del mensaje no está disponible en el momento en que se envían los mensajes, ya sea por problemas de red o por fallas en el sistema, los mensajes se guardan en una instancia local del servidor y se reenvían al destino remoto una vez que esté disponible.
- Herramientas de implementación listas para la empresa: Facilitan el despliegue y la migración de aplicaciones, desde la fase de desarrollo hasta un entorno de producción.
- Redistribución en producción: Permite a las empresas implementar una nueva versión de una aplicación sin interrumpir el trabajo en curso en la versión anterior.

- **Administración del sistema**

La administración del entorno del sistema de WebLogic Server incluye tareas tales como la creación de dominios, la implementación de aplicaciones, migración de dominios desde los entornos de desarrollo hacia entornos de producción, monitoreo y configuración del desempeño del dominio de WebLogic Server, y el diagnóstico y solución de problemas. WebLogic Server proporciona muchas herramientas para los administradores de sistemas y como ayuda en estas tareas,

incluyendo una consola de administración basada en el navegador, la herramienta de secuencias de comandos de WebLogic (WLST), un lenguaje de script para la automatización de tareas de administración del sistema WebLogic basado en Jython y SNMP, el asistente de configuración y las utilerías de línea de comando.

Debido a que el sistema de gestión de WebLogic Server se basa en Java EE y otros estándares, se integra bien con las herramientas utilizadas para administrar el software de la empresa y otros componentes de hardware. Además, WebLogic Server implementa la especificación Java EE Java Management Extension (JMX), la cual permite el acceso mediante programación al sistema de administración de WebLogic Server. Usando esta API se pueden crear utilerías propias de administración o automatizar tareas frecuentes usando clases Java.

- **Entorno de diagnóstico**

El entorno de diagnóstico de WebLogic es un servicio de monitoreo y diagnóstico que permite crear, recopilar, analizar, archivar y acceder a los datos de diagnóstico generados por un servidor en ejecución y sus aplicaciones implementadas. Estos datos proporcionan detalles sobre el rendimiento en tiempo de ejecución de las instancias de WebLogic Server y las aplicaciones implementadas, y permite aislar y diagnosticar fallas y cuellos de botella en el rendimiento.

- **Seguridad**

La arquitectura de seguridad de WebLogic Server proporciona una completa y flexible infraestructura de seguridad diseñada para hacer frente a los retos de seguridad de hacer las aplicaciones disponibles en la Web. La seguridad de WebLogic se puede utilizar de forma independiente para asegurar las aplicaciones WebLogic Server o a nivel de empresa, el sistema de gestión de seguridad representa la mejor solución en su clase.

- **Oracle JRockit JVM**

Oracle JRockit JVM es una máquina virtual de Java (JVM) de alto rendimiento optimizada para las arquitecturas Intel y desarrollada para garantizar la fiabilidad, escalabilidad, manejabilidad y flexibilidad para las aplicaciones Java. Oracle JRockit JVM optimiza el rendimiento de las aplicaciones Java en sistemas operativos Windows o Linux con arquitecturas de 32 bits o 64 bits. Oracle JRockit JVM se incluye con la instalación de WebLogic Server.

- **Diferentes opciones de clientes**

Además del soporte para clientes de aplicaciones basadas en navegador Web, WebLogic Server también soporta una gran variedad de tipos de clientes para crear ricas aplicaciones GUI o simples utilerías de línea de comando. Estos tipos de clientes son: RMI-IIOP, T3, clientes J2SE, clientes delgados Java EE, clientes CORBA/IDL y clientes C++ que se comunican con BEA Tuxedo.

- **Integración con otros sistemas**

WebLogic Server proporciona una variedad de herramientas para integrar sus aplicaciones con sistemas dispares. Estas herramientas incluyen Servicios Web, adaptadores de recursos, servicios de mensajes y RMI.

- **Integración con servidores web**

Los plugins proporcionados con la instalación de WebLogic Server le permiten funcionar con servidores Web de Apache, Microsoft y Sun Microsystems. Normalmente, estos servidores Web sirven contenido estático HTML mientras que las solicitudes para contenido Web dinámico, como JSP, se dirigen al entorno de WebLogic.

Desventajas de “WebLogic Server”.

La única desventaja que se puede considerar es que es un producto bajo licencia y cada instalación tiene un costo económico significativo comparado con servidores de libre distribución.

2.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA ARQUITECTURA CLIENTE-SERVIDOR.

La tecnología Cliente/Servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. Se trata pues, de la arquitectura más extendida en la realización de Sistemas Distribuidos.

Un sistema Cliente/Servidor es un Sistema de Información distribuido basado en las siguientes características:

- Servicio: unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- Recursos compartidos: Muchos clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- Protocolos asimétricos: Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.
- Transparencia de localización física de los servidores y clientes: El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- Independencia de la plataforma HW y SW que se emplee.
- Sistemas débilmente acoplados. Interacción basada en envío de mensajes.
- Encapsulamiento de servicios. Los detalles de la implementación de un servicio son transparentes al cliente.
- Escalabilidad horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).

- Integridad: Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

En el modelo usual Cliente/Servidor, un servidor, (daemon en la terminología sajona basada en sistemas UNIX/LINUX, traducido como "demonio") se activa y espera las solicitudes de los clientes. Habitualmente, programas cliente múltiples comparten los servicios de un programa servidor común. Tanto los programas cliente como los servidores son con frecuencia parte de un programa o aplicación mayores.

El Esquema de funcionamiento de un Sistema Cliente/Servidor sería:

- El cliente solicita una información al servidor.
- El servidor recibe la petición del cliente.
- El servidor procesa dicha solicitud.
- El servidor envía el resultado obtenido al cliente.
- El cliente recibe el resultado y lo procesa.

Componentes de Arquitectura Cliente/Servidor.

El modelo Cliente/Servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos.

De estas líneas se deducen los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas Cliente/Servidor: el proceso cliente que es quien inicia el diálogo, el proceso servidor que pasivamente espera a que lleguen peticiones de servicio y el middleware que corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

Para entender en forma más ordenada y clara los conceptos y elementos involucrados en esta tecnología se puede aplicar una descomposición o

arquitectura de niveles. Esta descomposición principalmente consiste en separar los elementos estructurales de esta tecnología en función de aspectos más funcionales de la misma:

- Nivel de Presentación: Agrupa a todos los elementos asociados al componente Cliente.
- Nivel de Aplicación: Agrupa a todos los elementos asociados al componente Servidor.
- Nivel de comunicación: Agrupa a todos los elementos que hacen posible la comunicación entre los componentes Cliente y servidor.
- Nivel de base de datos: Agrupa a todas las actividades asociadas al acceso de los datos.

Este modelo de descomposición en niveles, como se verá más adelante, permite introducir más claramente la discusión del desarrollo de aplicaciones en arquitecturas de hardware y software en planos.

Elementos Principales

Cliente

Un cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se le conoce con el término front-end.

Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

La funcionalidad del proceso cliente marca la operativa de la aplicación (flujo de información o lógica de negocio). De este modo el cliente se puede clasificar en:

- Cliente basado en aplicación de usuario. Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- Cliente basado en lógica de negocio. Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

Servidor

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Puede darse el caso que un servidor actúe a su vez como cliente de otro servidor. Existen numerosos tipos de servidores, cada uno de los cuales da lugar a un tipo de arquitectura Cliente/Servidor diferente.

El término "servidor" se suele utilizar también para designar el hardware, de gran potencia, capacidad y prestaciones, utilizado para albergar servicios que atienden a un gran número de usuarios concurrentes. Desde el punto de vista de la arquitectura cliente/servidor y del procesamiento cooperativo un servidor es un servicio software que atiende las peticiones de procesos software clientes.

MIDDLEWARE

El middleware es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red. En el caso que nos concierne, es el intermediario entre el cliente y el servidor y se ejecuta en ambas partes.

La utilización del middleware permite desarrollar aplicaciones en arquitectura Cliente/Servidor independizando los servidores y clientes, facilitando la interrelación entre ellos y evitando dependencias de tecnologías propietarias. El concepto de middleware no es un concepto nuevo. Los primeros “monitores de teleproceso” de los grandes sistemas basados en tecnología Cliente/Servidor ya se basaban en él, pero es con el nacimiento de la tecnología basada en sistemas abiertos cuando el concepto de middleware toma su máxima importancia. El middleware se estructura en tres niveles:

- Protocolo de transporte.
- Network Operating System (NOS).
- Protocolo específico del servicio.

Las principales características de un middleware son:

- Simplifica el proceso de desarrollo de aplicaciones al independizar los

entornos propietarios.

- Permite la interconectividad de los Sistemas de Información del Organismo.
- Proporciona mayor control del negocio al poder contar con información procedente de distintas plataformas sobre el mismo soporte.
- Facilita el desarrollo de sistemas complejos con diferentes tecnologías y arquitecturas.

Dentro de los inconvenientes más importantes destacan la mayor carga de máquina necesaria para que puedan funcionar.

Tipos de Arquitectura Cliente / Servidor.

Uno de los aspectos claves para entender la tecnología Cliente/Servidor, y por tanto contar con la capacidad de proponer y llevar a cabo soluciones de este tipo, es llegar a conocer la arquitectura de este modelo y los conceptos o ideas asociados al mismo. Más allá de entender los componentes cliente/middleware/servidor, es preciso analizar ciertas relaciones entre éstos, que pueden definir el tipo de solución que se ajusta de mejor forma a las estadísticas y restricciones acerca de los eventos y requerimientos de información que se obtuvieron en la etapa de análisis de un determinado proyecto. De hecho el analista deberá conocer estos eventos o restricciones del proyecto para que a partir de ahí, se puedan hacer las consideraciones y estimaciones de la futura configuración, teniendo en cuenta aspectos como por ejemplo, la oportunidad de la información, tiempo de respuesta, tamaños de registros, tamaño de bases de datos, estimaciones del tráfico de red, distribución geográfica tanto de los procesos como de los datos, etc.

En tal sentido se presenta, en primer lugar, un esquema de clasificación basado en los conceptos de Fat Client/Thin Client, Fat Server/Thin Server, es decir, basado en el tamaño de los componentes. En segundo lugar tenemos una clasificación según la naturaleza del servicio que nos ofrecen.

Por tamaño de componentes.

Este tipo de clasificación se basa en los grados de libertad que brinda el modelo Cliente/Servidor para balancear la carga de proceso entre los niveles de presentación, aplicación y base de datos. Dependiendo de qué segmento de las capas de software tenga que soportar la mayor o menor carga de procesamiento, se habla de Fat Client (Thin Server) o Fat server (Thin Client). Consideraciones de este tipo son importantes en el momento de decidir una plataforma de desarrollo, al mismo tiempo que pueden definir la viabilidad o no de las mismas para enfrentar un cierto número de restricciones impuestas por una problemática a resolver.

FAT CLIENT (THIN SERVER)

En este esquema de arquitectura el peso de la aplicación es ejecutada en el cliente, es decir, el nivel de presentación y el nivel de aplicación corren en un único proceso cliente, y el servidor es relegado a realizar las funciones que provee un administrador de base de datos.

En general este tipo de arquitectura tiene mejor aplicación en sistemas de apoyo de decisiones (DSS: Decision Support System) y sistemas de información ejecutiva (EIS: Executive Information System), y como se concluirá más adelante, tiene pocas posibilidades de aplicarse en sistemas de misión crítica.

FAT SERVER (THIN CLIENT)

Este es el caso opuesto al anterior, el proceso cliente es restringido a la presentación de la interfaz de usuario, mientras que el peso de la aplicación corre por el lado del servidor de aplicación.

En general este tipo de arquitectura presenta una flexibilidad mayor para desarrollar una gran variedad de aplicaciones, incluyendo los sistemas de misión crítica a través de servidores de transacciones.

Por naturaleza de servicio

Servidos de ficheros

Con un servidor de archivos, un cliente lo que hace es requerimientos de los mismos sobre una red. Esta es una forma muy primitiva de servicios de datos, la cual necesita intercambio de muchos mensajes sobre una red para hallar el dato requerido. Los servidores de archivos usan recursos compartidos sobre la red y son necesarios para crear repositorios de documentos, imágenes y archivos grandes sobre la red.

Servidores de bases de datos

La creación de aplicaciones Cliente/Servidor está asociada a la utilización de servidores de bases de datos relacionales SQL, y dependiendo de los requerimientos y restricciones se debe elegir entre una arquitectura dos o tres planos. Pero para una arquitectura centrada en un servidor de bases de datos, cualquiera de las modalidades dos planos, permite que un proceso cliente solicite datos y servicios directamente a un servidor de bases de datos. El servidor debe proveer un acceso compartido a los datos con los mecanismos de protección necesarios, así como proveer mecanismos para seleccionar resultados dentro de un conjunto de datos, posibilitando un ahorro en procesos de comunicación. El servidor debe también proveer mecanismos de concurrencia, seguridad y consistencia de datos, basado principalmente en el concepto de transacción en el que todo se realiza, y por lo tanto se hace permanente, o todo falla, anulándose la transacción en tal caso.

Los servidores de bases de datos actuales son una mezcla de SQL estándar más otras extensiones propias de cada proveedor. Por ejemplo casi todas las bases de datos están provistas con:

- Procedimientos almacenados (stored procedures): Una de las posibilidades de implementar de mejor forma un sistema Cliente/Servidor en dos planos

(two-tier), sin olvidar todas sus restricciones y limitaciones, es a través de procedimientos almacenados, que son funciones que agrupan un conjunto de instrucciones y lógica de procedimientos SQL, los cuales son compilados y almacenados en la misma base. El rol principal de los procedimientos almacenados es proveer la parte servidora de la lógica de una aplicación Cliente/Servidor, es decir vendría a reemplazar al servidor de aplicaciones en una arquitectura tres planos (three-tier).

- **Desencadenantes (triggers):** Son mecanismos que permiten realizar acciones automáticamente sobre los datos, las cuales están asociadas a algún evento definido. Normalmente son implementados a través de procedimientos almacenados. Los eventos a los cuales se hace referencia están asociados a las actualizaciones de tablas mediante sentencias delete, insert o update, y son llamados implícitamente al suceder cualquiera de estos eventos, a diferencia de los procedimientos almacenados que son llamados explícitamente por un proceso cliente.
- **Restricciones (constraints):** Al igual que los desencadenantes, son acciones que se realizan asociadas a algún evento determinado y están orientadas a llevar a cabo validaciones más simples de datos. Los tipos de eventos son los mismos que para los desencadenantes.

Servidores de transacciones

Estos tipos de sistemas se pueden implementar con cualquiera de las modalidades Cliente/Servidor en dos o tres planos, pero incorporan un elemento principal sobre el cual se elabora y basa toda la fortaleza de este modelo, el concepto de transacción.

Con un servidor de transacciones el proceso cliente llama a funciones, procedimientos o métodos que residen en el servidor, ya sea que se trate de un servidor de bases de datos o un servidor de aplicaciones. Lo importante es que el intercambio a través de la red se realiza mediante un único mensaje de solicitud/respuesta, es decir, independientemente de que se necesite ejecutar una

o más funciones, una o más instrucciones o sentencias SQL, éstas son agrupadas en una unidad lógica llamada transacción; evitando así el intercambio a través de la red de un mensaje solicitud/respuesta por cada sentencia SQL, el cual es el caso de los sistemas Cliente/Servidor dos planos, implementados a través de SQL remoto. Estas aplicaciones denominadas OLTP (On Line Transaction Processing) están orientadas a dar soporte a los procedimientos y reglas de los sistemas de misión crítica.

En la actualidad muchas aplicaciones tienen la necesidad de ser desarrolladas con la ayuda de transacciones, véase el ejemplo de los cajeros automáticos: imaginemos que queremos sacar dinero. Introducimos la cantidad adecuada y pulsamos el botón de enviar. El cajero manda una solicitud al banco para que descuente dicha cantidad de la cuenta del cliente, y recibe la respuesta. Si diera la casualidad que la respuesta del banco se perdiera en medio de la red, el cajero volvería a realizar la petición, por lo que el banco volvería a descontar dicha cantidad de la cuenta bancaria asociada. Este problema tan común se soluciona con la ayuda de las transacciones.

El modelo de comunicación más usado entre el cliente y el servidor a la hora de realizar transacciones planas distribuidas es el Two-Phase-Commit, el cual funciona de la siguiente manera:

La parte central de cualquier servidor de transacciones es el TP Manager. El núcleo del TP consta de diferentes partes:

- Gestor de transacciones (Transaction Manager):
 - Arranque de la transacción.
 - Registro de Gestores de Recursos que participan.
 - Gestión Commit/Rollback.
 - RPCs transaccionales: Permiten garantizar la integridad de la transacción cuando no se ejecuta en un solo ordenador.
 - RPCs con un identificador de transacción asociado, asignado por el TM.
 - Gestor de registro de modificaciones (Log Manager). Graba los cambios

realizados por los Gestores de Recursos. Permite reconstruir una versión consistente de todos los recursos.

- Gestor de bloqueos (Lock Manager). Proporciona mecanismos para regular el acceso concurrente a los recursos.

Servidores de objetos

Con un servidor de objetos, las aplicaciones Cliente/Servidor son escritas como un conjunto de objetos que se comunican. Los objetos cliente se comunican con los objetos servidores usando un Object Request Broker (ORB). El cliente invoca un método de un objeto remoto. El ORB localiza el método del objeto en el servidor, y lo ejecuta para devolver el resultado al objeto cliente. Los servidores de objetos deben soportar concurrencia. La parte central de la comunicación en los servidores de objetos es el ORB:

- Elemento central y principal de esta arquitectura.
- Bus de objetos. Permite la comunicación entre ellos.
- Middleware avanzado: Permite llamadas estáticas y dinámicas a objetos.
- Lenguaje de descripción de interfaces independiente del lenguaje de programación.

Servidores Web

La primera aplicación cliente servidor que cubre todo el planeta es el World Wide Web. Este nuevo modelo consiste en clientes simples que hablan con servidores Web. Un servidor Web devuelve documentos cuando el cliente pregunta por el nombre de los mismos. Los clientes y los servidores se comunican usando un protocolo basado en RPC, llamado HTTP. Este protocolo define un conjunto simple de comandos, los parámetros son pasados como cadenas y no provee tipos de datos. La Web y los objetos distribuidos están comenzando a crear un conjunto muy interactivo de computación Cliente/Servidor.

Modelos CLIENTE/SERVIDOR

Una de las clasificaciones mejor conocidas de las arquitecturas Cliente/Servidor se basa en la idea de planos (tier), la cual es una variación sobre la división o clasificación por tamaño de componentes. Esto se debe a que se trata de definir el modo en que las prestaciones funcionales de la aplicación serán asignadas, y en qué proporción, tanto al cliente como al servidor. Dichas prestaciones se deben agrupar entre los tres componentes clásicos para Cliente/Servidor: interfaz de usuario, lógica de negocios y los datos compartidos, cada uno de los cuales corresponde a un plano. Ni que decir tiene que la mala elección de uno u otro modelo puede llegar a tener consecuencias fatales.

Dentro de esta categoría tenemos las aplicaciones en dos planos (two-tier), tres planos (three-tier) y multi-planos (multi-tier). Dado que este término ha sido sobrecargado de significados por cuanto se lo utiliza indistintamente para referirse tanto a aspectos lógicos (Software) como físicos (Hardware), aquí se esquematizan ambas acepciones.

A nivel de software

Este enfoque o clasificación es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura Cliente/Servidor y en la madurez y popularidad de la computación distribuida. Por ejemplo, esto permite hablar de servidores de aplicación distribuidos a lo largo de una red, y no tiene mucho sentido identificar a un equipo de hardware como servidor, sino más bien entenderlo como una plataforma física sobre la cual pueden operar uno o más servidores de aplicaciones.

Modelo CLIENTE/SERVIDOR 2 Capas

Esta estructura se caracteriza por la conexión directa entre el proceso cliente y un administrador de bases de datos. Dependiendo de dónde se localice el grupo de tareas correspondientes a la lógica de negocios se pueden tener a su vez dos tipos distintos dentro de esta misma categoría:

Implementado con SQL remoto

En este esquema el cliente envía mensajes con solicitudes SQL al servidor de bases de datos y el resultado de cada instrucción SQL es devuelto por la red, no importando si son uno, diez, cien o mil registros. Es el mismo cliente quien debe procesar todos los registros que le fueron devueltos por el servidor de base de datos, según el requerimiento que él mismo hizo. Esto hace que este tipo de estructura se adecue a los requerimientos de aplicaciones orientadas a los sistemas de apoyo y gestión, pero resultan inadecuados para los sistemas críticos en que se requieran bajos tiempos de respuesta.

Ventajas:

- Presenta una estructura de desarrollo bastante simple ya que el programador maneja un único ambiente de desarrollo (es más simple respecto al Cliente/Servidor en tres planos, puesto que reduce una capa de programación, como se verá más adelante).

Desventajas:

- La gran cantidad de información que viaja al cliente congestiona demasiado el tráfico de red, lo que se traduce en bajo rendimiento.
- Por su bajo rendimiento esta estructura tiene un bajo espectro de aplicación, limitándose a la construcción de sistemas no críticos.

Implementado con procedimientos almacenados

En este esquema el cliente envía llamadas a funciones que residen en la base de datos, y es ésta quien resuelve y procesa la totalidad de las instrucciones SQL agrupadas en la mencionada función.

Ventajas:

- Presenta las mismas ventajas de una arquitectura dos planos con procedimientos almacenados, pero mejora considerablemente el rendimiento sobre ésta, dado que reduce el tráfico por la red al procesar los datos en la misma base de datos, haciendo viajar sólo el resultado final de un conjunto de instrucciones SQL.

Desventajas:

- Si bien la complejidad de desarrollo se ve disminuida, se pierde flexibilidad y escalabilidad en las soluciones implantadas. Obliga a basar el peso de la aplicación en SQL extendido, propios del proveedor de la base de datos que se elija. Debiera considerarse que sí bien los procedimientos almacenados (stored procedures), los desencadenantes (triggers) y las reglas (constraint) son útiles, en rigor son ajenos al estándar de SQL.

Modelo CLIENTE/SERVIDOR 3 CAPAS

Esta estructura se caracteriza por elaborar la aplicación en base a dos capas principales de software, más la capa correspondiente al servidor de base de datos. Al igual que en la arquitectura dos capas, y según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el nuevo proceso correspondiente al servidor de aplicación.

En este esquema el cliente envía mensajes directamente al servidor de aplicación el cual debe administrar y responder todas las solicitudes. Es el servidor, dependiendo del tipo de solicitud, quien accede y se conecta con la base de datos.

Ventajas:

- Reduce el tráfico de información en la red por lo que mejora el rendimiento de los sistemas (especialmente respecto a la estructura en dos planos).
- Brinda una mayor flexibilidad de desarrollo y de elección de plataformas sobre

la cual montar las aplicaciones. Provee escalabilidad horizontal y vertical.

- Se mantiene la independencia entre el código de la aplicación (reglas y conocimiento del negocio) y los datos, mejorando la portabilidad de las aplicaciones.
- Los lenguajes sobre los cuales se desarrollan las aplicaciones son estándares lo que hace más exportables las aplicaciones entre plataformas.
- Dado que mejora el rendimiento al optimizar el flujo de información entre componentes, permite construir sistemas críticos de alta fiabilidad.
- El mismo hecho de localizar las reglas del negocio en su propio ambiente, en vez de distribuirlos en la capa de interfaz de usuario, permite reducir el impacto de hacer mantenimiento, cambios urgentes de última hora o mejoras al sistema.
- Disminuye el número de usuarios (licencias) conectados a la base de datos.

Desventajas:

- Dependiendo de la elección de los lenguajes de desarrollo, puede presentar mayor complejidad en comparación con Cliente/Servidor dos planos.
- Existen pocos proveedores de herramientas integradas de desarrollo con relación al modelo Cliente/Servidor dos planos, y normalmente son de alto costo.

A nivel de hardware

Esta clasificación del modelo Cliente/Servidor se basa igualmente en la distribución de los procesos y elementos entre sus componentes, pero centrándose en la parte física del mismo, en el que la administración de la interfaz gráfica se asocia a los clientes PC y la seguridad e integridad de los datos quedan asociados a ambientes mainframe o por lo menos a servidores locales y/o centrales.

Modelo CLIENTE / SERVIDOR 2 CAPAS

Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual, dependiendo de la aplicación puede dar acceso a los datos administrados por él.

Modelo CLIENTE / SERVIDOR 3 CAPAS

Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación.

Modelos CLIENTE / SERVIDOR según el reparto de funciones entre cliente y servidor.

Separación de funciones.

Las distintas arquitecturas cliente/servidor presentan variaciones acerca de cómo son distribuidas las diferentes funciones de las aplicaciones de sistemas entre el cliente y el servidor, sobre la base de los conceptos de los tres componentes generales de cualquier SI:

- La lógica de acceso a datos. Funciones que gestionan todas las interacciones entre el SW y los almacenes de datos (archivos, bases de datos, etc.) incluyendo recuperación/consulta, actualización, seguridad y control de concurrencia.
- La lógica de presentación. Funciones que gestionan la interfaz entre los usuarios del sistema y el SW, incluyendo la visualización e impresión de formas y reportes, y la posibilidad de validar entradas del sistema.
- La lógica de negocio o lógica de la aplicación. Funciones que transforman entradas en salidas, incluyendo desde simples sumas hasta complejos modelos matemáticos, financieros, científicos, de ingeniería, etc.

Según cómo se distribuyan las funciones correspondientes a estas tres lógicas o funciones de un sistema entre cliente, middleware y servidor (los principales componentes de un sistema con arquitectura distribuida) nos podemos encontrar con los siguientes tipos de arquitectura cliente / servidor (conforme a la célebre clasificación hecha por el Gartner Group):

- Presentación Distribuida.
- Presentación remota.
- Acceso a datos remoto.
- Lógica o procesamiento distribuidos.
- Bases de datos distribuidas.

La importancia de esta clasificación radica en que permite jugar con el ancho de banda de la red y con la capacidad de proceso de los componentes hardware del sistema para repartir entre ellos la carga de proceso de las lógicas de la aplicación. Seguidamente entraremos en más detalle sobre estos tipos de arquitectura.

Presentación distribuida.

El cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor dedicados a esta tarea. El resto de funciones de la aplicación (negocio, acceso a datos) residen en el servidor.

Esta arquitectura se utiliza para construir emuladores de terminal, aplicaciones de control remoto, front ends gráficos de aplicaciones que residen en un host, etc. Algunos ejemplos de productos que siguen esta filosofía son VLC, Microsoft Terminal Server, Cytrix Metaframe, emulador de host para sistemas operativos modernos como Windows, etc.

La gran ventaja de esta arquitectura es que permite revitalizar sistemas antiguos. Así, las aplicaciones antiguas que funcionaban en entornos host pueden ser empleadas desde modernas estaciones de trabajo Windows o Microsoft (que solamente hacen la función de terminal del host incrustada en un entorno de trabajo de escritorio moderno).

El principal problema es que no se elimina la dependencia del host, no siendo posible la aplicación de los conceptos de downsizing o rightsizing.

Presentación remota.

Toda la lógica de negocio y acceso a datos se ejecuta en el servidor, que en esta ocasión no realiza ninguna función relacionada con la presentación. Todas las funciones de presentación son ejecutadas en el cliente. Un ejemplo de este tipo de aplicaciones son las aplicaciones web, las de los terminales de cajeros automáticos, etc.

La principal ventaja es que la interfaz de usuario se adapta bien a las capacidades del entorno cliente (en la presentación distribuida el servidor tenía que ejecutar funciones dentro de un entorno que podría no ser el más apropiado para el cliente). La principal desventaja es que toda la información necesaria para la presentación tiene que circular por la red desde el servidor al cliente.

Lógica o proceso distribuido.

La lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un proceso distribuido o cooperativo.

La principal ventaja de esta arquitectura es que cada uno de los nodos/servidores puede especializarse en un área determinada, de forma que cada proceso se ejecutará en el nodo más apropiado. Además, se pueden reutilizar los sistemas ya existentes (es una especie de antesala del concepto de SOA).

En contrapartida, este tipo de sistemas son más difíciles de diseñar, de mantener y de probar.

Acceso a datos remoto.

El cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

La principal ventaja de esta arquitectura radica en su sencillez de uso, y su proliferación al ser adaptada por lenguajes de cuarta generación (como Oracle Forms). La desventaja es la latencia de red introducida. Al descargarse toda la

lógica de proceso en los aplicativos clientes, estos necesitan descargar los datos necesarios (entradas al proceso) que circulan por la red.

Bases de datos distribuidas.

Este modelo es similar al de Acceso a Datos Remoto, pero además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como bases de datos distribuidas.

La principal ventaja de este modelo es que facilita el acceso a los datos desde entornos heterogéneos. Los componentes de acceso a datos ubicados en el cliente permiten independizar la base de datos del entorno en el que corren las aplicaciones cliente. Además, permite implementar la "transparencia de ubicación". Este sistema presenta dos importantes inconvenientes:

Las bases de datos distribuidas son más difíciles de implementar, y son dependientes del gestor de base de datos (siempre que no existan acuerdos y estándares)

La integridad de los datos puede verse comprometida.

Capítulo 3

"ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA"



3.1. ANÁLISIS DEL PROCESO ACTUAL DE SUPERVISIÓN DE CLIENTES CREDITICIOS.

En la figura 3.1.1 se presenta el diagrama del proceso actual.

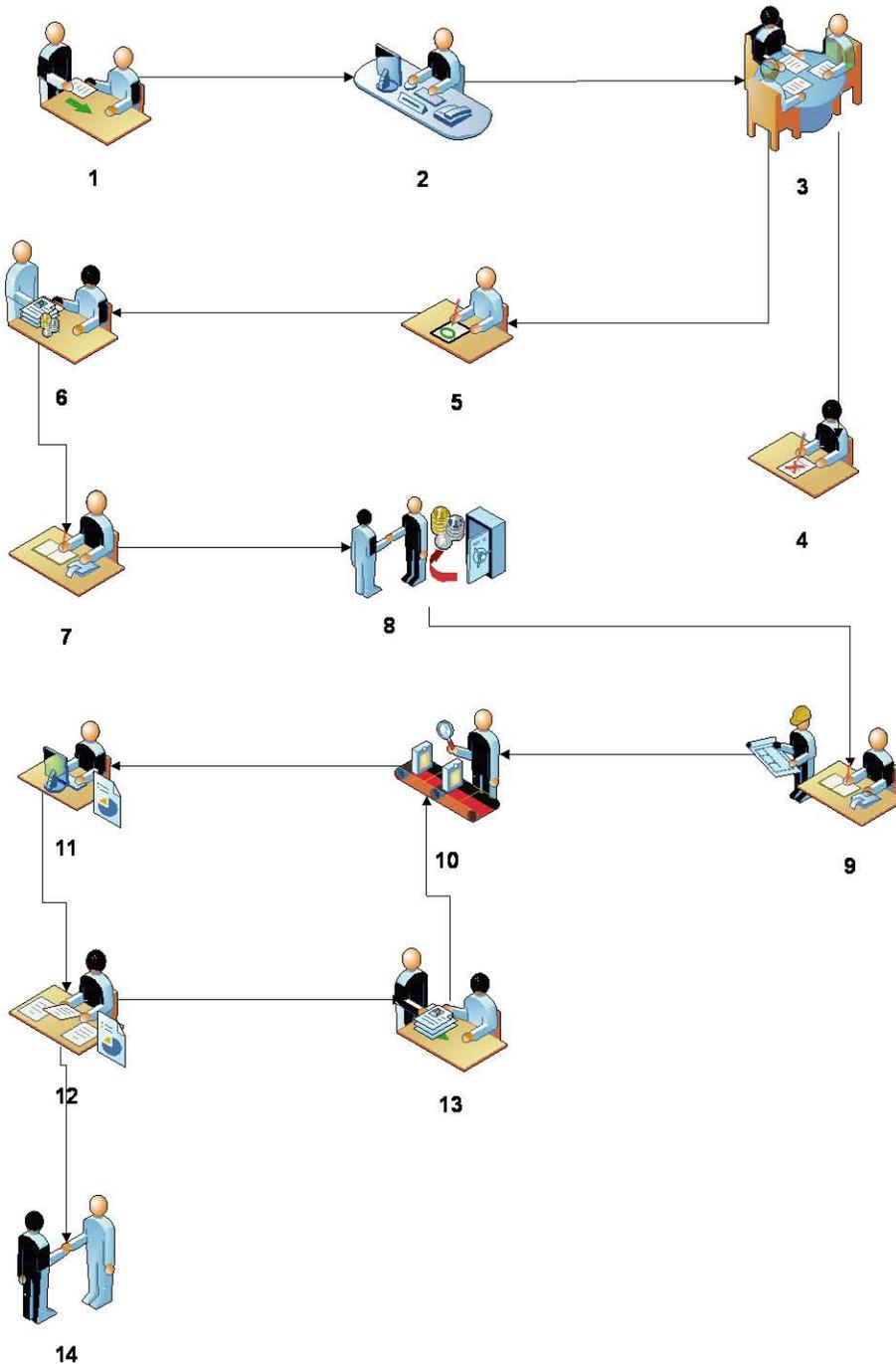


Figura 3.1.1 Diagrama del proceso actual.

Problemática actual.

A continuación se describe la problemática encontrada en las actividades de las diferentes etapas del proceso mostrado.

Recepción de la documentación inicial del cliente.

El registro de la documentación recibida se realiza en archivos de hojas de cálculo.

Recopilación de los datos del análisis crediticio del cliente.

La información recopilada se registra en archivos electrónicos.

La concentración de la información final toma bastante debido al número de archivos que se manejan.

Análisis y determinación de la solicitud de crédito.

No se identificaron problemas en esta actividad.

Rechazo de la solicitud de crédito.

La información relacionada con el rechazo se guarda en archivos electrónicos dispersos, lo que hace difícil consolidar la información.

Aprobación de la solicitud de crédito.

No se identificaron problemas en esta actividad.

Recepción de los requisitos contractuales y de las garantías del crédito.

No se identificaron problemas en esta actividad.

Registro de los requisitos contractuales y de las garantías del crédito.

El registro de la documentación recibida se realiza en archivos de hojas de cálculo.

El control de la información registrada se vuelve complejo debido al manejo de archivos individuales.

Elaboración y firma del contrato de crédito.

No se identificaron problemas en esta actividad.

Elaboración del plan de visitas de supervisión.

El plan de visitas se genera manualmente y de forma lenta.

La información de la programación de las visitas se guarda en archivos de hojas de cálculo lo que dificulta el intercambio de la información.

Realización de las visitas.

No se identificaron problemas en esta actividad.

Registro de la información de las visitas y creación del reporte de informe.

El registro de la información recopilada se hace en archivos de procesador de texto, se generan bastantes archivos y es difícil llevar un control sobre ellos.

Los reportes de los informes se generan manualmente con base en información almacenada en archivos electrónicos. Se invierte demasiado tiempo en consolidar la información para generar los informes.

Seguimiento de la vigencia de los requisitos contractuales y creación de reportes de cumplimiento.

El seguimiento se realiza a través de la información registrada en archivos de hojas de cálculo y es susceptible a errores.

Los reportes se crean manualmente con base en información guardada en varios archivos electrónicos. Se invierte demasiado tiempo en reunir la información requerida para generar los reportes.

Recepción de la actualización de los requisitos contractuales vencidos.

El control de las actualizaciones se hace de forma manual y se registra en archivos electrónicos difíciles de controlar.

Cierre del crédito y almacenamiento del historial crediticio.

El historial crediticio se guarda en documentos físicos y en archivos electrónicos dispersos difíciles de administrar.

Se dificulta el proceso de búsqueda y revisión de los historiales de cada cliente.

En resumen, el manejo de la información de los clientes durante todo el ciclo de vida del crédito se lleva a cabo de manera manual, usando archivos electrónicos independientes y dispersos; esto provoca que se invierta gran cantidad de trabajo horas/hombre en llevar un control efectivo de la información y éste es susceptible

a errores, la información no está disponible de forma inmediata y no es fácil de analizar.

Solución propuesta.

La solución propuesta consiste en un sistema único y centralizado para el registro, control y seguimiento interno de la información relacionada con el crédito; este sistema será accedido vía web para facilitar su disponibilidad a nivel empresarial. El sistema propuesto contribuirá a solucionar la problemática en las siguientes actividades:

Recepción de la documentación inicial del cliente.

La información de la documentación recibida se registrará en un formato único.

Los datos registrados estarán disponibles de forma inmediata, única y centralizada.

Recopilación de los datos del análisis crediticio del cliente.

La información del cliente se va a registrar y administrar en un formato único y centralizado.

Análisis y determinación de la solicitud de crédito.

No se identificaron cambios en esta actividad.

Rechazo de la solicitud de crédito.

La información relacionada con el rechazo quedará guardada automáticamente durante el proceso de registro de la información del cliente.

Aprobación de la solicitud de crédito.

No se identificaron cambios en esta actividad.

Recepción de los requisitos contractuales y de las garantías del crédito.

No se identificaron cambios en esta actividad.

Registro de los requisitos contractuales y de las garantías del crédito.

La información de los requisitos contractuales y de las garantías proporcionadas por el cliente, se van a capturar en un formato único y quedará almacenada de manera centralizada.

Elaboración y firma del contrato de crédito.

No se identificaron cambios en esta actividad.

Elaboración del plan de visitas de supervisión.

El plan de visitas se va a generar, registrar y administrar desde el sistema computacional, por lo cual la información estará centralizada y disponible fácilmente.

Realización de las visitas.

No se identificaron cambios en esta actividad.

Registro de la información de las visitas y creación del reporte de informe.

El seguimiento de las visitas programadas se hará a través de una interfaz única.

El registro de la información se realizará en un formato único.

El reporte del informe de la visita se creará automáticamente con base en la información almacenada en el sistema y estará disponible electrónicamente en cualquier momento que se requiera.

Seguimiento de la vigencia de los requisitos contractuales y creación de reportes de cumplimiento.

Los requisitos contractuales registrados tendrán un periodo de vigencia y una vez concluido se considerará en incumplimiento.

Los reportes de cumplimiento se generarán automáticamente con base en la información almacenada en el sistema y estará disponible en cualquier momento.

Recepción de la actualización de los requisitos contractuales vencidos.

La actualización de los requisitos contractuales se realizará en un formato único y centralizado.

Cierre del crédito y almacenamiento del historial crediticio.

El historial crediticio se guardará en el sistema centralizado por lo cual la información estará disponible de forma rápida y oportuna.

3.2 RECOPIACIÓN Y ANÁLISIS DE LA INFORMACIÓN

El crédito para las empresas es algo en que cada empresario tiene que pensar en algún momento. Puede ser al principio cuando una empresa está iniciando y necesita algo de capital para ponerla en marcha, o cuando la empresa quiere poner en marcha un nuevo proyecto y requiere de algunos fondos adicionales para impulsar el proceso. Mientras que las empresas se crean para ganar dinero, también cuestan dinero hacerlas funcionar y a veces, los dueños de negocios necesitan un poco de ayuda con los aspectos financieros. A menos que su empresa cuente con los recursos suficientes, siempre es probable que usted tenga que considerar, en algún momento, un crédito para su empresa

Luego de llenar el formulario de la solicitud, se pasa a la presentación de todos aquellos documentos que sirven como sustento a lo depositado en el formulario. Así entre los diversos papeles que se piden, se pueden encontrar:

Copia fotostática del Alta de Hacienda y Cédula de Identificación Fiscal. Ver figura 3.2.1

Copia fotostática del Acta Constitutiva incluyendo la hoja que contiene los datos de registro en el Registro Público de la Propiedad y del Comercio. Cuando dicha acta cuente con modificaciones y reformas, deberá anexarse una copia fotostática de cada una de ellas. Figura 3.2.2

Copia fotostática de las actas en la que conste la designación de apoderados. Debido a que se firmará un Contrato de Apertura de Crédito y el Pagaré correspondiente, comúnmente se solicita que los apoderados cuenten con poderes para actos de Administración, Dominio y Títulos de Crédito. Figura 3.2.3

Copia fotostática de una identificación oficial vigente del representante legal o del apoderado, del obligado solidario o aval (en su caso). Si el obligado o aval es una Persona Física casada por sociedad deberá presentarse la identificación del cónyuge. Las identificaciones de mayor aceptación son la credencial del Instituto Federal Electoral (IFE), el Pasaporte y el FM2 o FM3 en caso de ser extranjero. Figura 3.2.4

Copia fotostática de un comprobante de domicilio fiscal de la empresa y en su caso del obligado o aval, cuya antigüedad no sea mayor a 2 meses. Los comprobantes comúnmente aceptados son el recibo de agua, luz, predial y el teléfono. Figura 3.2.5

Información financiera

Copia fotostática de los estados financieros anuales de la empresa que incluyan Balance General y Estado de Resultados. Dependiendo del banco y tipo de crédito se podrá solicitar hasta los 3 últimos años y los parciales (con antigüedad no mayor a 90 días). Podrán solicitarse algunas relaciones analíticas, la relación de información específica de determinadas cuentas e información auditada. Figura 3.2.6

Copia fotostática de la última declaración anual de impuestos. Dependiendo del banco se podrá solicitar únicamente la del último ejercicio fiscal, definitiva y/o provisionales. Figura 3.2.7

Copia fotostática de la declaración patrimonial reciente del obligado solidario o aval (en su caso). Figura 3.2.8

Autorización del solicitante para solicitar una consulta al Buró de Crédito y en su caso del obligado solidario. Toda empresa o Persona Física con Actividad Empresarial tiene derecho a obtener su Reporte de Crédito Especial de manera gratuita cada 12 meses consultando la siguiente dirección (www.burodecredito.com.mx).

Por otra parte, en el análisis siempre influye el conocimiento del cliente, de tal modo que es bueno haber mantenido relaciones comerciales anteriores a la solicitud de la operación. Una buena trayectoria puede hacer que una operación, que objetivamente se denegaría a un cliente en particular, se le conceda. Una entidad, sin conocernos de nada, no vos va a financiar, pues no debemos olvidar que el banco lo componen personas.

Nunca debemos dar datos falsos a una entidad cuando le solicitemos una operación de activo. Si la realidad es mala, igual necesitamos avalistas, o nos debemos conformar con un préstamo menor; en cualquier caso, siempre podemos hablar y exponer nuestras necesidades y capacidad de compromiso. El intentar engañar al banco no suele dar buen resultado por varios motivos. Un analista ve cientos de operaciones al año, por lo que fácilmente va a detectar incongruencias; tienen datos de clientes de nuestro mismo estrato o sector, así como una serie de contactos que le van a poder informar sobre nosotros. Si la entidad en cuestión detecta que le intentamos engañar, no volverá a confiar en lo que decimos, de tal forma que lo más probable es que la operación planteada se deniegue y tengamos cerrada para siempre esa puerta.

Hasta ahora hemos planteado la operación como si de una operación aislada se tratara; sin embargo, cada vez son más las entidades que quieren clientes, no

operaciones. Nos forzarán a que nos vinculemos a ellos a través de otras operaciones que serán requisitos indispensables para la concesión de la principal. A esto lo denominan venta cruzada.

INSCRIPCIÓN EN EL R.F.C.

EL SERVICIO DE ADMINISTRACIÓN TRIBUTARIA, LE DA A CONOCER EL REGISTRO FEDERAL DE CONTRIBUYENTES, QUE LE HA SIDO ASIGNADO CON BASE EN LOS DATOS QUE PROPORCIONÓ, LOS CUALES HAN QUEDADO REGISTRADOS CONFORME A LO SIGUIENTE:

NOMBRE, DENOMINACIÓN O RAZÓN SOCIAL
PATRICIA TAZ BUENROSTRO

DOMICILIO
EUGENIA 15 DISTRITO FEDERAL BENITO JUAREZ NARVARTE 03010

CLAVE DEL R.F.C. TABP750513B12

CLAVE ÚNICA DE REGISTRO DE POBLACIÓN **TABP750513MNEZNT06**

ADMINISTRACIÓN LOCAL DE RECAUDACIÓN **DEL ORIENTE DEL D.F.**

ACTIVIDAD **Comercio al por mayor de calzados**

SITUACIÓN DE REGISTRO **ACTIVO**

FECHA DE INSCRIPCIÓN **08-02-2007** FECHA DE INICIO DE OPERACIONES **05-02-2007**

OBLIGACIONES

DESCRIPCIÓN	FECHA ALTA
Presentar la declaración y pago provisional mensual del impuesto al Activo (IMPAC).	01-01-2010
Presentar la declaración anual del impuesto al Activo (IMPAC).	05-02-2007
Presentar la declaración anual de impuesto Sobre la Renta (ISR) donde se informe sobre los clientes y proveedores de bienes y servicios.	05-02-2007
Presentar la declaración anual de impuesto Sobre la Renta (ISR) de personas físicas.	05-02-2007
Proporcionar la información del impuesto al Valor Agregado (IVA) que se solicita en las declaraciones del impuesto Sobre la Renta (ISR).	05-02-2007
Presentar la declaración y pago provisional mensual de impuesto Sobre la Renta (ISR) por realizar actividades empresariales.	05-02-2007
Presentar la declaración mensual donde se informe sobre las operaciones con terceros para efectos de impuesto al Valor Agregado (IVA).	05-02-2007
Presentar la declaración y pago definitivo mensual de impuesto al Valor Agregado (IVA).	05-02-2007

TRÁMITES EFECTUADOS

TRÁMITES EFECTUADOS	FECHA DE PRESENTACIÓN	FOLIO DEL TRÁMITE
Reg. Federal Contribuyente / Inscripción / Inscripción de Persona Física	08-02-2007	RF2007280613

Fecha de Impresión: 08 de Febrero de 2007
TELEFONO DE ATENCION CIUDADANA (QUEJAS Y SUGERENCIAS) 01-800-728-2000

lpOOwKXSHUdeYVG3yU277g2+AH1SjYb6JefKsXnePFZce5QKdBRhSHUSC Mdeq4R4X1y9h6v1AiqMPOGatp+6vIB7x3MPbtvSpOyqulKHcLiqZU3NZ4A UGMSQRpDme413355MaXN4c5MCx5FajJachefstJDUAw

Próximamente podrá imprimir este documento desde internet, por lo que la impresión podrá ser a color o blanco y negro, dependiendo del tipo de impresora que tenga.

Figura 3.2.1 Cédula de Identificación Fiscal

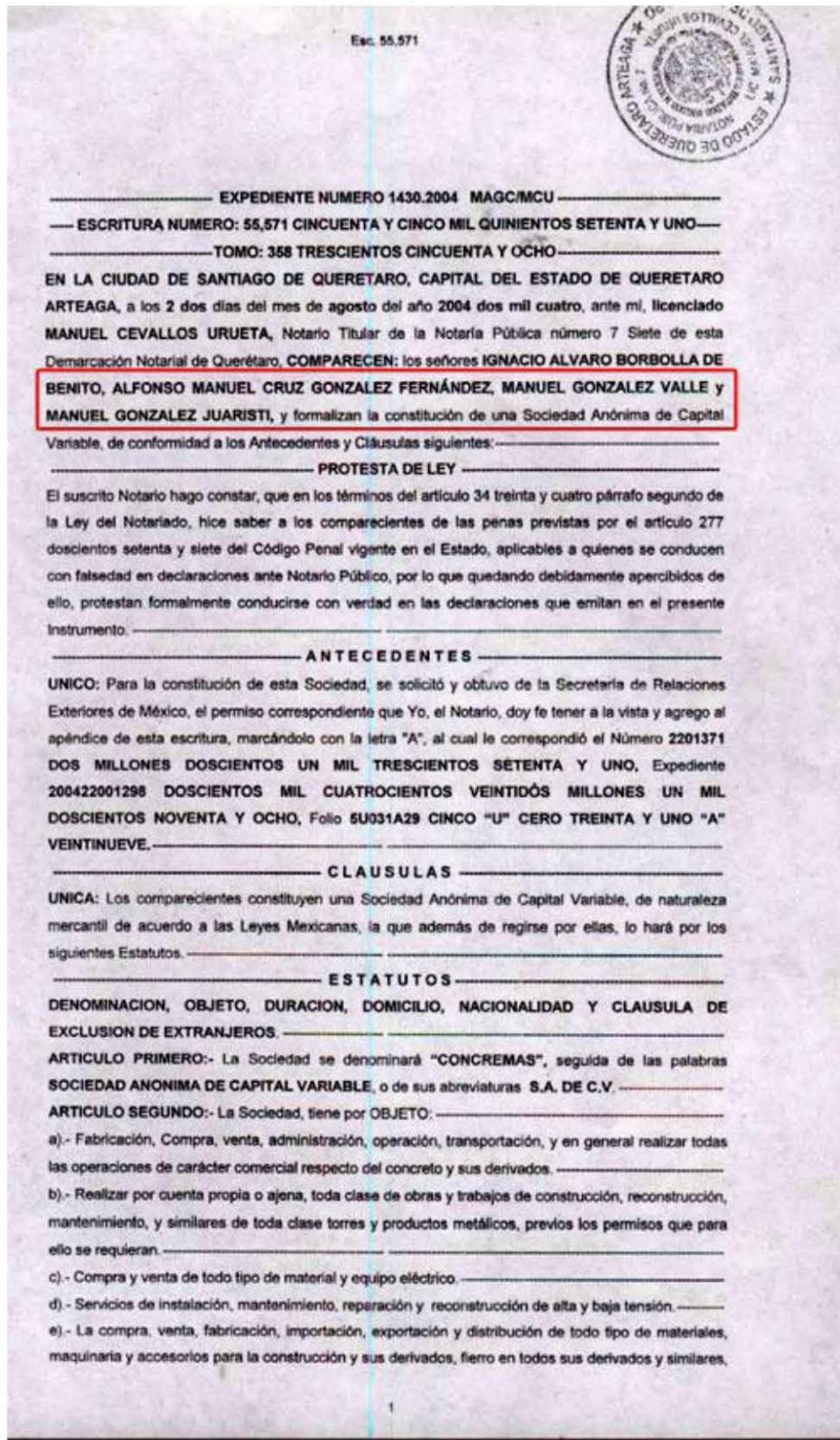


Figura 3.2.2 Acta constitutiva

Certificado de nombramiento de Director y de sus facultades

D., con D.N.I. núm., en su calidad de Secretario del Consejo Rector de, S. COOP., con domicilio en (provincia), C.I.F. e inscrita en el Registro de Cooperativas con el número

CERTIFICA

Que consta en el acta de la sesión del Consejo Rector, celebrada el de de, que fue debidamente convocada por el Sr. Presidente, y que se celebró a las horas, acudiendo personalmente los siguientes consejeros:

Presidente:

Secretario:

Vocal:

Vocal:

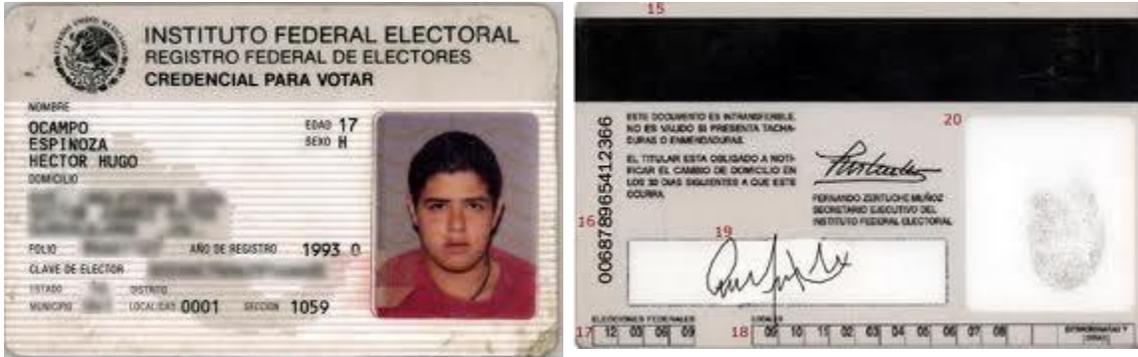
que en el punto 2º del Orden del Día, se adoptó por unanimidad el siguiente

ACUERDO: Designar como Director de, S. COOP. a D., provisto de D.N.I. nº, con domicilio en, de nacionalidad atribuyéndole las siguientes

FACULTADES

- 1.- Administrar los negocios sociales; nombrar y separar personal, corregirlo, fijar sus sueldos, remuneraciones y gratificaciones; distribuir las funciones y servicios, organizar el régimen de trabajo; designa y revocar mandatarios, corresponsales, representantes, asesores, Procuradores y Agentes, determinando sus facultades, atribuciones, obligaciones y retribuciones, y, en general, llevar la gestión interior y exterior de la sociedad.
- 2.- Concertar arrendamientos, convenir ejecuciones de obras, suministros, seguros, fletamentos y transportes; concurrir a subastas y concursos de toda clase, constituir y retirar fianzas o depósitos, formular las pertinentes propuestas y formalizar las adjudicaciones; liquidar, saldar y finiquitar

Figura 3.2.3 Acta designación de apoderados



FORMULARIO 3 - CONYUGE



AVAL DE CONYUGE

Señor Gerente de
FEDERACION PATRONAL SEGUROS S.A.
Presente

De mi consideración:

Tengo el agrado de dirigirme a usted con el objeto de hacerle saber que tengo pleno conocimiento de que mi cónyuge, es asegurado de esa Empresa y para ratificarle de acuerdo a las disposiciones legales vigentes, y en particular a lo determinado por los artículos 1.276 y 1.277 del Código Civil, que mi cónyuge ha sido y es el administrador de mis bienes propios y los de la sociedad conyugal, por lo cual puede suscribir todas las obligaciones que estime convenientes. La presente debe ser tomada como plena aceptación por mi parte de todas las responsabilidades que tenga contraídas o contraiga, y mantendrá su vigencia hasta que en forma expresa comunique lo contrario, dejando aclarado que, en este supuesto, mi responsabilidad subsistirá para todas las obligaciones existentes a la fecha de mi comunicación.

Atentamente.



FIRMA DEL CONYUGE CERTIFICADA POR ESCRIBANO

FIRMA

DOCUMENTO

Figura 3.2.4 .- Identificación Oficial o aval



TELEFONOS DE MEXICO S.A.B. de C.V.
Parque Vía 198, Col. Guauhtémoc
C.P. 06599 México D.F.
RFC: TME840315-KT6 11-AGO-2008 DV 8

TORIELLO SORIANO MARIA GUADALUPE

PRL DIVISION DEL NORTE

TLALPAN, DF
C.P. 14350-CR-14391

Pág 1 de 4

RESIDENCIAL

Total a Pagar: \$ 1,230.00

Pagar antes de: 03-SEP-2008

Mes de Facturación: Agosto

Teléfono:

Factura No.: 060108080180597

Importe enviado a cobro en su
Banco Banamex

Estado de Cuenta

Saldo Anterior		1,263.00
Su Pago Gracias	21-Jul-08	- 1,263.00
Saldo		0.00
Cargo por Redondeo		+ 0.84
Cargos del Mes		+ 1,069.56
IVA		+ 160.43
SubTotal		\$ 1,230.83
Crédito por Redondeo*		- 0.83
Total a Pagar		\$ 1,230.00

(mil doscientos treinta pesos 00/100 M.N.)

*La diferencia de Centavos aplicará en su próximo Estado de Cuenta.

Cargos del Mes

Planes y Paquetes	868.70
Tiendas Telmex	183.66
Celulares	17.20
SubTotal	\$ 1,069.56

Usted goza de los beneficios de estar con Telmex
Su saldo pendiente por compras a plazos es de: **\$ 8,072.12***
*IVA Incluido

Atención a Clientes: 01 (800) 123 0000

TORIELLO SORIANO MARIA GUADALUPE

Teléfono: Total a Pagar: \$ 1,230.00

Mes de Facturación: Agosto Pagar antes de: 03-SEP-2008

DV 8



55567858460001230006

Dale la vuelta al mundo
por **\$2.30** el minuto
LVA Incluido

Por eso ahora con tu paquete
TODO MÉXICO
SIN COSTOS

tarifa para que llames
a todo el mundo

telmex.com Tiendas TELMEX

*Excepto Cuba

Profesor 24 horas

Por sólo **\$120** anuales

La mejor herramienta para las tareas de tus hijos con **profesores en línea las 24 hrs.**

telmex.com Tiendas TELMEX

Figura 3.2.5 Comprobante Domicilio

CUENTAS	MAYO 200Y	MAYO 200X
ACTIVOS		
Activos Circulantes		
Efectivo y equivalentes	119,804	90,449
Cuentas por Cobrar, menos estimación por cuentas incobrables		
De \$. 14,288 y 10,624	521,588	400,877
Inventarios	586,594	309,476
Gastos pagados por anticipado	26,738	19,851
Otros activos circulantes	25,536	17,029
Activos Circulantes Totales	1'280,260	837,682
Activos Fijos (No Circulantes)		
Inmuebles, Planta y Equipo		
Al costo	397,601	238,461
Menos: Depreciación acumulada	105,138	78,797
Inmuebles, Planta y Equipo, netos	292,463	159,664
Plusvalía Comercial	114,710	81,021
Otros Activos	20,997	16,185
Activos fijos (no circulantes) totales	428,170	256,870
Activos Totales	1'708,430	1'094,552
PASIVO Y CAPITAL EN ACCIONES		
Pasivos Circulantes		
Documentos por Pagar	300,364	31,102
Cuentas por Pagar	165,912	107,423
Pasivos Acumulados	115,824	94,939
Impuesto sobre la renta por pagar	45,792	30,905
Porción circulante de la deuda a largo plazo	580	8,792
Pasivos Fijos (no circulantes) totales	628,472	273,161
Pasivos no circulantes		
Deuda a largo plazo	29,992	25,941
Pasivos no circulantes totales	46,869	36,872
Pasivos totales	675,341	310,033
Capital en acciones		
Acciones preferentes redimibles	300	300
Acciones comunes al valor declarado	2,876	2,874
Capital en exceso del valor declarado	84,681	78,582
Utilidades acumuladas	949,660	701,728
Ajuste por conversión de divisas	(4,428)	1,035
Capital en acciones total	1'033,089	784,519
Total de Pasivos y Capital en Acciones	1'708,430	1'094,552

Figura 3.2.6 Estados financieros anuales

REGISTRO FEDERAL DE CONTRIBUYENTES

3

CLAVE ÚNICA DE REGISTRO DE POBLACIÓN

DECLARASAT IMPRESIÓN DIDÁCTICA

VERIFICAR

DCSP3A09

DETERMINACIÓN DEL IMPUESTO AL ACTIVO (Ejercicio 2007 y anteriores) (Continuación)

<p>o. IMPUESTO SOBRE LA RENTA ACREDITADO <input type="text"/></p> <p>p. ISR ACREDITADO DE EJERCICIOS ANTERIORES <input type="text"/></p> <p>q. OTROS ACREDITAMIENTOS <input type="text"/></p> <p>r. ESTÍMULOS FISCALES APLICADOS DURANTE EL EJERCICIO <input type="text"/></p> <p>s. PAGOS PROVISIONALES EFECTIVAMENTE PAGADOS SIN ACREDITAMIENTO DEL ISR <input type="text"/></p> <p>t. DIFERENCIA A CARGO (n - (o + p + q + r + s) cuando n es mayor) <input type="text"/></p>	<p>u. SALDO A FAVOR ((o + p + q + r + s) - n cuando n es menor) <input type="text"/></p> <p>v. ISR PAGADO EN EXCESO APLICADO CONTRA EL IMPAC (campo CC de la página 1) <input type="text"/></p> <p>w. OTRAS CANTIDADES A CARGO DEL CONTRIBUYENTE <input type="text"/></p> <p>x. OTRAS CANTIDADES A FAVOR DEL CONTRIBUYENTE <input type="text"/></p> <p>y. IMPUESTO A CARGO (t + v + w + x) <input type="text"/></p>
--	--

DETERMINACIÓN DEL IMPUESTO EMPRESARIAL A TASA ÚNICA

<p>A. BASE GRAVABLE (Campo R de la página 18) <input type="text" value="214737"/></p> <p>B. IMPUESTO CAUSADO (Campo A por la tasa correspondiente) <input type="text" value="36505"/></p> <p>C. CRÉDITO FISCAL POR DEDUCCIONES MAYORES A LOS INGRESOS (Sin exceder de B) <input type="text"/></p> <p>D. ACREDITAMIENTO POR SUELDOS Y SALARIOS GRAVADOS (Sin exceder de B - C) <input type="text"/></p> <p>E. ACREDITAMIENTO POR APORTACIONES DE SEGURIDAD SOCIAL (Sin exceder de B - C - D) <input type="text"/></p> <p>F. CRÉDITO FISCAL POR INVERSIONES (1998 a 2007) (Sin exceder de B - C - D - E) <input type="text"/></p> <p>G. CRÉDITO FISCAL DE DEDUCCIÓN INMEDIATA / PÉRDIDAS FISCALES (Sin exceder de B - C - D - E - F) <input type="text"/></p> <p>H. CRÉDITO FISCAL SOBRE PÉRDIDAS FISCALES (RÉGIMEN SIMPLIFICADO) (Sin exceder de B - C - D - E - F - G) <input type="text"/></p> <p>I. CRÉDITO FISCAL POR ENAJENACIONES A PLAZO (Sin exceder de B - C - D - E - F - G - H) <input type="text"/></p> <p>J. ISR PROPIO DEL EJERCICIO (Sin exceder de B - C - D - E - F - G - H - I) <input type="text" value="16278"/></p> <p>K. ACREDITAMIENTO DEL ISR PROPIO PAGADO EN EL EXTRANJERO (Sin exceder de B - C - D - E - F - G - H - I - J) <input type="text"/></p>	<p>L. IMPUESTO A CARGO (1a DIFERENCIA) B - (C + D + E + F + G + H + I + J + K) cuando B es mayor o igual <input type="text" value="20227"/></p> <p>M. PAGOS PROVISIONALES DE IETU <input type="text" value="15031"/></p> <p>N. EXCEDENTES DE PAGOS PROVISIONALES DEL IETU ACREDITADOS CONTRA PAGOS PROVISIONALES DEL ISR DEL MISMO PERIODO <input type="text"/></p> <p>O. PAGOS PROVISIONALES DE IETU EFECTIVAMENTE PAGADOS <input type="text" value="15031"/></p> <p>P. DIFERENCIA A CARGO (L - O) cuando L es mayor <input type="text" value="5196"/></p> <p>Q. DIFERENCIA A FAVOR (O - L) cuando O es mayor <input type="text"/></p> <p>R. ISR PAGADO EN EXCESO APLICADO CONTRA IETU (campo CC1 de la página 1) <input type="text" value="5196"/></p> <p>S. OTRAS CANTIDADES A CARGO <input type="text"/></p> <p>T. OTRAS CANTIDADES A FAVOR <input type="text"/></p> <p>U. IMPUESTO A CARGO DEL EJERCICIO (P + S) - (R + T) cuando (P + S) es mayor <input type="text" value="0"/></p> <p>V. IMPUESTO A FAVOR DEL EJERCICIO (P + S) - (Q + T) cuando (P + S) es menor <input type="text"/></p>
--	---

DATOS DEL REPRESENTANTE LEGAL

REGISTRO FEDERAL DE CONTRIBUYENTES	<input type="text"/>
CLAVE ÚNICA DE REGISTRO DE POBLACIÓN	<input type="text"/>
APELLIDO PATERNO	<input type="text"/>
APELLIDO MATERNO	<input type="text"/>
NOMBRE(S)	<input type="text"/>

* Para recibir orientación sobre esta forma oficial, puede utilizar los siguientes servicios:
 Portal del SAT (www.sat.gob.mx). Ingresando a orientación en línea podrá obtener información en los siguientes medios: Orientación a través de su portal privado; Chat uno a uno; Atención telefónica; marque al 01 800 INFOSAT (463 36 728) de cualquier parte del país, sin costo. Denuncias sobre posibles actos de corrupción: 01 800 33 54 867 o bien a la dirección de correo electrónico: denuncias@sat.gob.mx o en su caso, previa cita, acudir a los módulos de las Administraciones Locales de Servicios al Contribuyente.

ESTE DOCUMENTO NO ES UNA FORMA OFICIAL Y NO ESTABLECE OBLIGACIONES NI CREA DERECHOS DISTINTOS DE LOS CONTENIDOS EN LAS DISPOSICIONES FISCALES VIGENTES

Figura 3.2.7 Declaración anual de impuestos

"SISTEMA DE SUPERVISIÓN DE CLIENTES CREDITICIOS PARA UNA INSTITUCIÓN FINANCIERA"

JUAN CAMILO MOURIÑO TERRAZO
TIPO DE DECLARACIÓN: INICIAL
FECHA DE LA DECLARACIÓN: 21/05/2004
DEPENDENCIA: SECRETARÍA DE ENERGÍA

DATOS GENERALES DEL SERVIDOR PÚBLICO

NOMBRE: MOURIÑO TERRAZO, JUAN CAMILO
 FECHA DE NACIMIENTO: 21/08/1971
 SEXO: HOMBRE
 ESTADO CIVIL: CASADO(A)
 PAÍS DONDE NACIÓ: ESPAÑA
 NACIONALIDAD: MEXICANA
 ENTIDAD DONDE NACIÓ: FUERZA DE TRÉSCUOS

DATOS DEL PUESTO O ENCARGO DEL SERVIDOR PÚBLICO

NOMBRE DEL ENCARGO O PUESTO: SUBSECRETARIO DE ELECTRICIDAD
 DEPENDENCIA O ENTIDAD: SECRETARÍA DE ENERGÍA
 DOMICILIO: CALLE INSURGENTES SUR, NUMERO EXTERIOR: 890, NUMERO INTERIOR: PISO 15, LOCALIDAD O COLONIA: DEL VALLE, CÓDIGO POSTAL: 03100, ENTIDAD FEDERATIVA: DISTRITO FEDERAL, MUNICIPIO O DELEGACIÓN: BENITO JUAREZ
 ÁREA DE ASIGNACIÓN: SUBSECRETARÍA DE ELECTRICIDAD
 FUNCIONES PRINCIPALES: MANEJO Y TRÁFICO DE BIENES MATERIALES, ÁREAS TÉCNICAS.
 TELÉFONO: 015543000120
 CORREO ELECTRÓNICO INSTITUCIONAL: jcamilo@semgta.gob.mx
 FECHA DE INICIO DEL ENCARGO: 14/05/2004
 ESTÁ CONTRATADO(A) POR HORARIOS?: NO
 CLAVE PRESUPUESTAL O EQUIVALENTE: 8007 200 01K001 000002

DATOS CURRICULARES DEL SERVIDOR PÚBLICO

ESCOLARIDAD

NIVEL	UBICACIÓN	NOMBRE DE LA INSTITUCIÓN	CARRERA O ÁREA DE CONOCIMIENTO	ESTATUS	PERIODOS CURSADOS	DOCUMENTO OBTENIDO
LECIENSIATURA	ESTADOS UNIDOS, FLORIDA TAMPA	UNIVERSITY OF TAMPA	ECONOMÍA	FINALIZADO		TÍTULO
MAESTRIA	ESTADOS UNIDOS, FLORIDA TAMPA	UNIVERSIDAD AUTÓNOMA DE CAMPECHE	FINANZAS	TRUNDO	2 AÑO	OTRO
OTRO DIPLOMADO	ESTADOS UNIDOS, FLORIDA TAMPA	UNIVERSITY OF TAMPA	MERCADOTECNIA	FINALIZADO		CARTA DE PASANTE CERTIFICADO

EXPERIENCIA LABORAL

SECTOR	PODER	AMBITO	INSTITUCIÓN O EMPRESA	UNIDAD ADMINISTRATIVA	PUESTO	FUNCIÓN PRINCIPAL	INGRESO - Egreso
PUBLICO	LEGISLATIVO	FEDERAL	H. CAMARA DE DIPUTADOS	COMISION DE ENERGIA	DIPUTADO FEDERAL	PRESIDENTE DE LA COMISION DE ENERGIA	09/2000 - 04/2002
PUBLICO	LEGISLATIVO	ESTATAL	H. CONGRESO DEL ESTADO DE CAMPECHE	CAMARA ESTATAL DE CAMPECHE	DIPUTADO V DISTRITO	DIPUTADO V DISTRITO	10/1997 - 09/2000
PRIVADO			GRUPO ENERGETICO DEL SURESTE S.A. DE C.V.	FINANZAS	DIRECTOR DE FINANZAS	OPTIMIZAR LOS RECURSOS MONETARIOS DE LA EMPRESA	01/1995 - 01/1997
PRIVADO			GRUPO ENERGETICO DEL SURESTE S.A. DE C.V.	ADMINISTRACION	GERENTE ADMINISTRATIVO	REVISION Y SUPERVISION DE LA INFORMACION FINANCIERA	08/1993 - 12/1994
PUBLICO	LEGISLATIVO	FEDERAL	SECRETARIA DE ENERGIA	SUBSECRETARIA DE POLITICA ENERGETICA Y DESARROLLO TEC.	DIRECTOR GENERAL	AREAS TECNICA	10/2003 - 03/2004

EXPERIENCIA ACADÉMICA
 EL SERVIDOR PÚBLICO NO PROPORCIONA INFORMACIÓN DE EXPERIENCIA ACADÉMICA.

LOGROS LABORALES O ACADÉMICOS A DESTACAR
 CONSEJERO DE DIVERSAS CAMARAS Y ASOCIACIONES

DECLARACION ANTERIOR

TIPO DE DECLARACION ANTERIOR: MODIFICACION PATRIMONIAL
 FECHA DE PRESENTACION DE LA DECLARACION ANTERIOR: 21/05/2004

EL SERVIDOR ACEPTA HACER PUBLICOS SUS DATOS PATRIMONIALES

DATOS PATRIMONIALES.- INGRESOS MENSUALES NETOS

POR CONCEPTO	VALOR
POR ACTIVIDAD INDUSTRIAL O COMERCIAL	0000
POR ACTIVIDAD FINANCIERA	000056000
POR SERVICIOS PROFESIONALES	
OTROS	
TOTAL	01037

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- SÓLO SE INCLUYEN LOS INGRESOS DEL SERVIDOR PÚBLICO
 NO SE INCLUYEN LOS DEL CÓNYUGE Y DEPENDIENTES ECONÓMICOS

DATOS PATRIMONIALES.- BIENES INMUEBLES

TIPO DE OPERACIÓN	TIPO DE BIEN	UBI. TERRENO EN ÚS	UBI. CONSTRUCCIÓN EN ÚS	FORMA DE OPERACIÓN	REGISTRO PÚBLICO DE LA PROPIEDAD	FECHA	VALOR	MONEDA
SIN CAMBIO	CASA	020002825	020002030	CONTADO	701235	01/04/1997	020484300	PESOS MEXICANOS
SIN CAMBIO	CASA	020001927	020002711	DONACION	00712	05/11/2003	002070280	PESOS MEXICANOS

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- SÓLO SE PROPORCIONAN LOS BIENES QUE REPORTÓ EL SERVIDOR PÚBLICO A NOMBRE DEL DECLARANTE O DEL CÓNYUGE
 NO SE INCLUYEN LOS BIENES DECLARADOS A NOMBRE DE SU CÓNYUGE, SUS DEPENDIENTES ECONÓMICOS O DE OTROS.

DATOS PATRIMONIALES.- VEHICULOS

TIPO DE OPERACIÓN	MARCA	TIPO	MODELO	FORMA DE OPERACIÓN	VALOR DE LA OPERACIÓN	MONEDA	FECHA
SIN CAMBIO	CHRYSLER	GRAND CHEROKEE	1999	CREDITO	030000	PESOS MEXICANOS	16/09/2000
INCORPORACION	BMW	RS	2004	CREDITO	020000000	PESOS MEXICANOS	16/12/2003

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- SÓLO SE PROPORCIONAN LOS VEHICULOS QUE REPORTÓ EL SERVIDOR PÚBLICO A NOMBRE DEL DECLARANTE O DEL DECLARANTE Y SU CÓNYUGE.
 NO SE INCLUYEN LOS BIENES DECLARADOS A NOMBRE DE SU CÓNYUGE, SUS DEPENDIENTES ECONÓMICOS O DE OTROS.

DATOS PATRIMONIALES.- BIENES MUEBLES

TIPO DE OPERACIÓN	TIPO DE BIEN	DESCRIPCIÓN DEL BIEN	FORMA DE OPERACIÓN	VALOR DE LA OPERACIÓN	MONEDA	FECHA
SIN CAMBIO	OTROS	MUEBLE PARA CASA HABITACION	PAGALOS RECIBIDOS	000000000	PESOS MEXICANOS	08/05/1999
SIN CAMBIO	OTROS	MUEBLE PARA CASA HABITACION	CONTADO	000000000	PESOS MEXICANOS	05/05/2000
SIN CAMBIO	JOYAS	JOYAS Y OBRAS DE ARTE	PAGALOS RECIBIDOS	000000000	PESOS MEXICANOS	08/05/1999
SIN CAMBIO	JOYAS	JOYAS Y OBRAS DE ARTE	CONTADO	000040000	PESOS MEXICANOS	01/12/2000
SIN CAMBIO	MEBLES DE CASA	MEBLES DE CASA	CONTADO	000120000	PESOS MEXICANOS	

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- SÓLO SE PROPORCIONAN LOS BIENES QUE REPORTÓ EL SERVIDOR PÚBLICO A NOMBRE DEL DECLARANTE O DEL DECLARANTE Y SU CÓNYUGE.
 NO SE INCLUYEN LOS BIENES DECLARADOS A NOMBRE DE SU CÓNYUGE, SUS DEPENDIENTES ECONÓMICOS O DE OTROS.

DATOS PATRIMONIALES.- INVERSIONES

TIPO DE OPERACIÓN	TIPO DE INVERSIÓN	VALOR	MONEDA
SALDO	BANCAARIA	000007183	PESOS MEXICANOS
SALDO	BANCAARIA	000100000	PESOS MEXICANOS
SALDO	BANCAARIA	000000049	PESOS MEXICANOS
SALDO	BANCAARIA	000000007	PESOS MEXICANOS
INCORPORACION	BANCAARIA	000040000	PESOS MEXICANOS
INCORPORACION	BANCAARIA	000000	EUROS
INCORPORACION	BANCAARIA	01	EUROS

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- LOS DATOS CORRESPONDEN A LAS CUENTAS REPORTADAS POR TIPO DE INVERSIÓN Y MONEDA.
 3.- SÓLO SE INCORPORA LA INFORMACIÓN REPORTADA DE CUENTAS E INVERSIONES A NOMBRE DEL DECLARANTE Y DEL DECLARANTE Y SU CÓNYUGE.
 NO SE INCLUYEN LAS QUE ESTÁN A NOMBRE DEL CÓNYUGE, DEPENDIENTES ECONÓMICOS O DE OTROS.

DATOS PATRIMONIALES.- ADEUDOS

TIPO DE OPERACIÓN	TIPO DE ADEUDO	VALOR	FECHA DE OTORGAMIENTO	MONTO ORIGINAL DEL ADEUDO	DESTINO
SALDO	PAGAR DE CREDITO	000000000 PESOS MEXICANOS			
INCORPORACION	COMPRINA DE VEHICULO	000044749 PESOS MEXICANOS	16/12/2003	030816856 PESOS MEXICANOS	

1.- LOS DATOS CORRESPONDEN A LA FECHA DE INICIO DEL ENCARGO
 2.- LA INFORMACIÓN CORRESPONDE A TODOS LOS ADEUDOS REPORTADOS A NOMBRE DEL DECLARANTE Y DEL DECLARANTE Y SU CÓNYUGE.
 NO SE INCLUYEN LOS QUE ESTÁN A NOMBRE DEL CÓNYUGE, DEPENDIENTES ECONÓMICOS O OTROS.
 *TODA LA INFORMACIÓN FUE CAPTURADA DIRECTAMENTE POR EL SERVIDOR PÚBLICO

Figura 3.2.8 Declaración Patrimonial



3.3 LEVANTAMIENTO DE LOS REQUERIMIENTOS GENERALES Y PARTICULARES QUE DEBE CUBRIR LA SOLUCIÓN TECNOLÓGICA A CONSTRUIR.

En la ingeniería de sistemas, un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. En la ingeniería clásica, los requerimientos se utilizan como datos de entrada en la etapa de diseño del producto, establece que debe hacer el sistema, pero no como hacerlo.

Se usa en un sentido formal en la ingeniería de sistemas pudiendo dividirse como:

Un requerimiento funcional puede ser una descripción de lo que un sistema tiene que hacer.

Un requerimiento no funcional: Se refiere a la parte tangible del sistema como lo es el rendimiento, la calidad, etc.

Una colección de requerimientos describe las características o atributos del sistema deseado.

Requerimientos Generales

El sistema tiene que ser:

Fiable.

Estable.

Accesible.

Desarrollado en un lenguaje de alto nivel que por definición son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje máquina.

Seguro.

Documentado.

Una solución a los problemas mediante el uso de EDD's o mejor conocidas como Estructuras Dinámicas de Datos, algo muy utilizado en todos los lenguajes de programación ya son estructuras que pueden cambiar de tamaño durante la ejecución del programa.

Extensible.

Una estructura de datos que se adapten a las necesidades reales de un programa.

Un programa que debe ser independiente de la arquitectura del ordenador.

Escrito en un lenguaje de alto nivel que debe ser portable.

Accesible desde diferentes plataformas como son vía web, computadora personal y dispositivos móviles.

Una opción para descargar la carga de trabajo del equipo cliente.

De alto rendimiento.

Recibir y responder peticiones desde cualquier parte del mundo.

Utilizar el poder de procesamiento de datos del servidor.

Tener un manejo de la información la cual debe quedar almacenada encontrándose disponible en cualquier momento para ser utilizada desde cualquier lugar.

Tener disponibilidad y comunicación permanente.

Tener compatibilidad con los diferentes browser existentes.

Testeado.

Contar con soporte.

Eficiente.

Contar con un sistema de seguridad.

Contar con una tabla de roles y permisos.

Contar con el hardware y software necesario para desempeñar su labor de manera eficiente.

Requerimientos Particulares.

El sistema deberá de:

Proporcionar el performance necesario para su desempeño.

Contener una interfaz de usuario que permita registrar vía web la información de la entrega de la documentación recibida y que deberá ser almacenada y registrada en el sistema de tal forma que deba tener una organización lógica y coherente además de garantizar su integridad .

Contar con la información registrada la cual estará almacenada de una manera lógica y transparente para él usuario de tal forma que pueda ser consulta desde cualquier punto en donde se cuente con una conexión a internet.

Utilizar el soporte de almacenamiento de ORACLE.

Permitir dar el seguimiento adecuado a la información de acuerdo al flujo establecido para la revisión de los créditos de tal forma que sea factible de acuerdo a parámetros establecidos y procedimientos fijos.

Ser capaz de actualizar, modificar e incluso dar de baja la información de la que sea responsable de acuerdo al nivel de seguridad establecido con su respectivo seguimiento sin perder la integridad de la misma.

Manejar la jerarquía de roles de una forma clara en función de los puestos y funciones desempeñados dentro del sistema.

Proporcionar las herramientas adecuadas para el seguimiento de la información de las garantías proporcionadas por el cliente las cuales se deberán de ir registrando, almacenando y darle seguimiento de acuerdo al proceso establecido por la institución.

Llevar un registro o bitácora de las visitas realizadas a manera de seguimiento en el cual se permita actualizar los status del crédito.

Almacenar un registro o bitácora que permita el agendar y dar seguimiento a visitas posteriores permitiendo llevar un control del número de vistas y los avances y observaciones de las mismas.

Permitir generar reportes basados en la información almacenada y de acuerdo a los estándares y necesidades propios del proceso.

Contar con los reportes generados en una alta disponibilidad de manera inmediata para su consulta y descarga. De tal forma que las personas puedan consultar en tiempo real los avances y atrasos de las visitas realizadas, así como de los status de cada uno de los casos.

Proporcionar los elementos necesarios a las personas responsables del proyecto para que les permita en tiempo real conocer los avances, status y atrasos de los mismos y en función de ellos determinar seguir con el crédito y/o proyecto o bien finalizar la relación crediticia con el cliente.

Almacenar la información de forma organizada y confiable y permitir guardar y consultar un historial que permita a la entidad financiera no solo ayudarlo a tomar una decisión en el momento si no también conocer de manera más amplia el comportamiento del cliente en función de la experiencia adquirida basándonos en su historial crediticio dentro de la institución bancaria que permita recomendar o levantar puntos de control o seguimiento a fin de tener un mejor panorama sobre el mismo.

Contar con una bitácora de uso que permita establecer los parámetros de rendimiento en función de las visitas y los recursos utilizados.

Consideraciones técnicas

Las aplicaciones web deben de funcionar de igual forma independientemente de la versión del sistema operativo instalado en el cliente. Esto es en vez de crear clientes para Windows, Mac OS X, GNU/Linux y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes, sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones estándar para navegadores web que pueden causar problemas en el desarrollo y soporte de estas aplicaciones, principalmente debido a la falta de adicción de los navegadores a dichos estándares web (especialmente versiones de Internet Explorer anteriores a la 7.0). Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar Java script) puede interferir con la consistencia de la aplicación web.

Otra aproximación es utilizar Adobe Flash Player o Java applets para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de plugins), las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones, debido a que no son estándares. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas "aplicaciones web"; un término alternativo es "Aplicación Enriquecida de Internet".

3.4 IDENTIFICACIÓN DE POSIBLES BLOQUES

El primer bloque deberá estar destinado a la autenticación del usuario, esta debe hacerse mediante un usuario previamente asignado y la contraseña correspondiente, la cual debe cumplir con los requisitos mínimos de seguridad establecidos. En caso de no dar los datos correctos se le enviará un mensaje de error, evitando que tengan acceso al sistema.

Dependiendo del rango de autoridad de cada usuario se podrá tener acceso a los diferentes Módulos del Sistema de Supervisión de Crédito.

Una vez introducidos los datos solicitados de Usuario y Contraseña, se presentará los bloques principales del Sistema de Supervisión de Crédito

Vea Figura 3.4.1

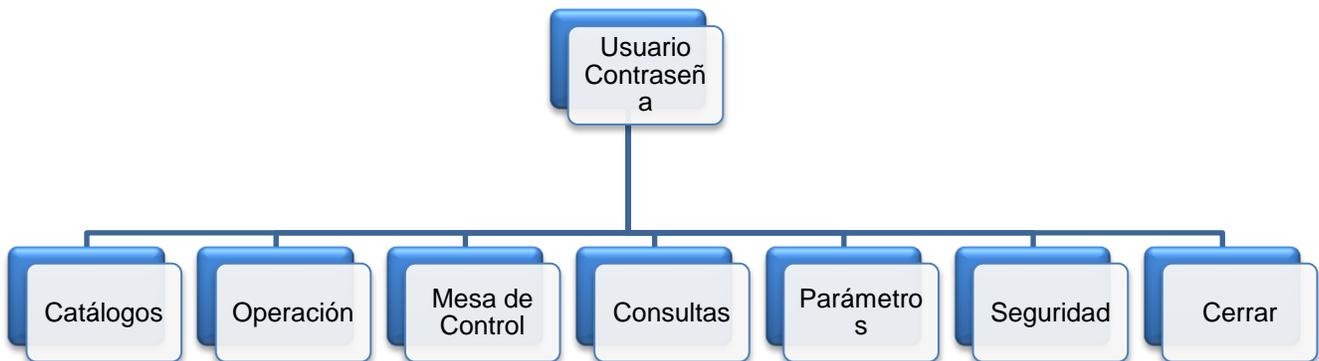


Figura 3.4.1 Bloques Principales del Sistema

Cada uno de los bloques principales contendrá un grupo de sub-bloques que se presentarán a continuación con una breve explicación de cada uno de ellos:

Bloque Catálogos



Figura 3.4.2 Bloque Catálogos

En el primer bloque principal “Catálogos” deberá contener toda la información necesaria de apoyo para los siguientes bloques.

El usuario de supervisión de Crédito podrá dar de alta, modificar y dar de baja la información de los diferentes catálogos que forma el Sistema de Supervisión de Crédito.

De acuerdo a la ubicación de la solicitud del crédito se llenarán los primeros 3 catálogos:

Las Direcciones Regionales serán identificadas por una clave numérica de 3 caracteres.

Las Áreas Responsables serán identificadas por una clave numérica de 2 caracteres.

Los Sectores serán identificados por una clave alfanumérica de 3 caracteres.

En el Catálogo de Tipo de empresa podremos identificar si se trata de Sector Público, Corporativo, Empresarial, etc.

En Tipos de Empresa encontraremos el giro de la empresa que solicita el crédito.

En el siguiente catálogo encontraremos los Tipos de garantía que el cliente deja a cambio del crédito.

En los siguientes dos catálogos encontraremos toda la información referente al cliente (Clave del número que le corresponde al cliente, Descripción, Descripción de la actividad del cliente, Dirección de la planta, Razón Social, Dirección de Oficina, Nombre, Teléfono, mails, Puesto, Año de Ingreso, Tipo de Cliente, Actividad Económica, Área responsable, Analista, Supervisor, Promotor, Tipo de Empresa, Sector).

En los últimos 2 catálogos encontraremos los tipos de requisitos como: pólizas, avalúos, estados financieros, etc.

Bloque de operación



Figura 3.4.3 Bloque Operación

En este Bloque de Operación en el Sistema de Supervisión de Crédito, se desplegarán los siguientes sub-bloques. Donde se podrá ver el estatus de Requisito Contractual mediante semáforos de color.

En el bloque de registro de garantías se podrá agregar observaciones a las garantías, Consultar y Modificar líneas de Crédito, Consultar, Agregar, Modificar y Eliminar Garantías.

Para consultar o modificar una o unas líneas de Crédito de un cliente, se requerirá la modificación de los siguientes datos: No. De Línea, Tipo de Línea, Moneda, Fecha Vigencia, Monto, Monto Vigente, Monto Vencido.

Se podrán agregar notas adicionales.

Los siguientes 3 sub-bloques tienen que ver con un plan de seguimiento a los clientes que solicitan los créditos como son visitas o supervisiones que son calendarizadas y periódicas para validar y cotejar la información proporcionada por el cliente.

Los últimos 3 sub-bloques se refieren al resultado del plan de supervisión el cuál se vacía en un informe que sirve de base para la toma de decisiones y plan de acciones a seguir.

Bloque Consultas

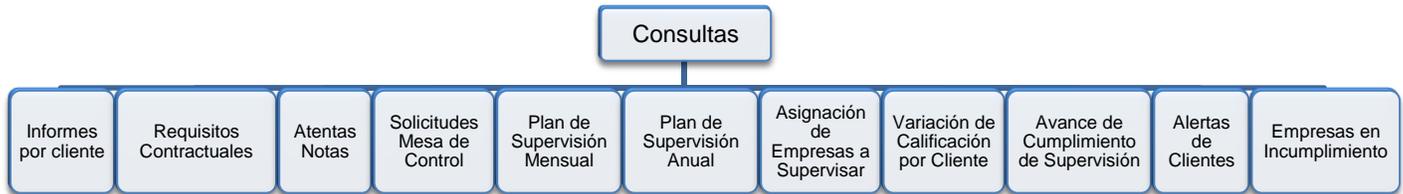


Figura 3.4.4 Bloque Consultas

En el primer sub-bloque se podrá consultar a los clientes con los siguientes datos: Cliente, Razón Social, Informes.

En el siguiente sub-bloque, se podrá consultar los Requisitos Contractuales de los clientes con los siguientes datos: Clave del Requisito y Razón Social.

Las Atentas Notas de los clientes se podrán consultar con los siguientes datos: Cliente, Razón Social, Notas.

Se podrá consultar las Solicitudes Mesa de Control de los clientes.

Se podrá consultar el Plan de Supervisión Mensual de los clientes con los siguientes datos: Cliente, Razón Social, Supervisor, Promotor, Analista, Monto, Calificación.

Se podrá consultar el Plan de Supervisión Anual de los clientes con los siguientes datos: Clave, Razón Social, E.Sup., E.Anual.

Se podrá consultar las diferentes Empresas a Supervisar durante un año y mes con los siguientes datos: Cliente, Razón Social, Saldo, Calificación, Tipo de Informe, Área Responsable, Analista, Promotor.

Se podrá consultar la Variación de Calificación por Cliente con los siguientes datos: Cliente, Razón Social o eligiendo mes y año a consultar.

Se podrá consultar el Avance de Cumplimiento de Supervisión de los clientes de acuerdo al mes y año seleccionado con los datos (Informe, Cliente, Razón Social, F. Visita, F.Cierre, Puntuación, Alerta Sup., Calificación, Monto, Supervisor, D. Regional, A. Responsable) en Informes Cerrados y Aperturados y los datos (Cliente, Razón Social) en Informes Programados.

Se podrá consultar las Alertas de Clientes con los siguientes datos: Clave, Razón Social, Calificación, Tipo de Alerta.

Y por último podremos consultar las Empresas en Incumplimiento con los siguientes datos: Clave, Razón Social.

Bloque Parámetros

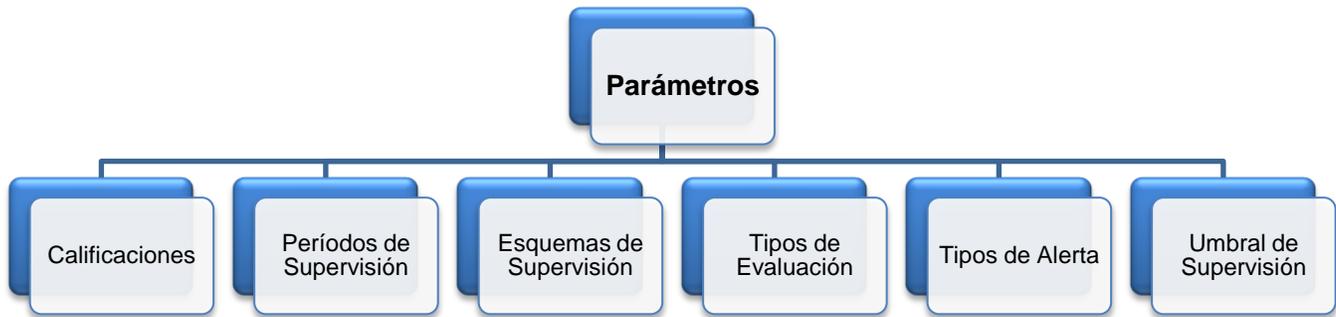


Figura 3.4.5 Bloque Parámetros

En los 3 primeros sub-bloques se podrán dar de alta, modificar y dar de baja una calificación (Clave, Nombre, Reserva de Inicio, Reserva Final), un período de supervisión (Clave, Nombre, Meses) y un esquema de supervisión (Clave y Nombre) como por ejemplo: “Esquema por encima del umbral” (umbral definido por la institución financiera) el cual contará con una calificación y un período para la supervisión a los clientes que solicitan los créditos.

En el sub-bloque Tipos de Evaluación conoceremos el status de la solicitud generada por el cliente.

Las alertas podrán ser generadas de acuerdo a los umbrales predefinidos (Máxima, Alta, Media y Mínima), para generar una nueva alerta usaremos los datos: Clave, Nombre, Alerta Superior, Alerta Inferior, para dar de alta un nuevo tipo de Alerta.

Los Umbrales de Supervisión estarán definidos por cantidades en Moneda Nacional en donde se les asignará un Esquema de Supervisión con todos los campos ya mencionados.

Bloque Seguridad

En este bloque se administraran (Alta, Modificación y Baja de usuarios) los usuarios que accederán al sistema de acuerdo al nivel de jerarquía con el que cuenten.

Los datos requeridos para los procesos mencionados serán: Clave Tipo de Usuario, Clave, Nombre, Nombre Corto, Email, Email2 y No. de empleado. El proceso se muestra en las figuras 3.4.6 y 3.4.7



Figura 3.4.6 Bloque Seguridad



Figura 3.4.7 Bloque Cerrar

3.5 JUSTIFICACIÓN DE LA METODOLOGÍA Y DEL SOFTWARE A UTILIZAR

Lo complicado de cada sistema manejador de base de datos provoca que no haya pruebas de comparación oficiales entre éstos, como por ejemplo

La complejidad en el manejo de las bases de datos.

Las herramientas sólo pueden ser juzgadas a través de la experiencia práctica.

Diferencias de requerimientos en la arquitectura.

Los lenguajes de programación son diferentes.

De acuerdo con la información recabada, se obtuvieron las tablas 3.5.1, 3.5.2, 3.5.3, 3.5.4 y 3.5.5 que comparan la información general y técnica para un número suficiente de sistemas de gestión de bases de datos.

Información general				
Fabricante	Manejador	Primera versión	Versión actual	Licencia
IBM	Informix	1980	11.70.xC1 2010-10	Propietario
Oracle Corporation	Oracle	1979	11g Release 2 2009-9	propietario
Adaptive Server Enterprise	Sybase	1987	15	propietario

Tabla 3.5.1 Información general de manejadores de bases de datos.

Plataformas						
Fabricante	Windows	Mac OS X	Linux	BSD	UNIX	z/OS
Informix	Sí	Sí	Sí	Sí	Sí	Sí
Oracle	Sí	Sí	Sí	No	Sí	Sí
Sybase	Sí	Sí	Sí	Sí	Sí	No

Tabla 3.5.2 Manejadores de bases de datos por plataforma de Sistema Operativo.

Fabricante	ACID ²	Integridad referencial	Transacciones	Unicode
Informix	Sí	Sí	Sí	Sí
Oracle	Sí	Sí	Sí	Sí
Sybase	Sí	Sí	Sí	Sí

Tabla 3.5.3 Características técnicas de manejadores de bases de datos.

² ACID: Atomicidad, Consistencia, Aislamiento, Durabilidad.

Fabricante	Unión	Combinación interna	Otras combinaciones	Inner selects	Blobs/ Clobs
Informix	?	?	?	?	?
Oracle	Sí	Sí	Sí	Sí	Sí
Sybase	?	?	?	?	?

Tabla 3.5.4 Blob (Objeto Binario Largo) y Clob (Objeto Carácter Largo).

	Data Domain	Cursor	Trigger (Disparador)	Función	Procedimiento
Informix	?	Sí	Sí	Sí	Sí
Oracle	Sí	Sí	Sí	Sí	Sí
Sybase	Sí	Sí	Sí	Sí	Sí

Tabla 3.5.5 Otros elementos en los manejadores de bases de datos.

Comportamiento de Oracle, Informix y Sybase.

Para Sybase el segmento Cliente es un ambiente amigable, ágil, y gráfico, en cuanto a la instalación, lo complicado es al momento de configurar los permisos, direccionar la base de datos y el manejo de la misma.

El segmento Servidor, se apoya en la reconocida calidad y prestaciones que ofrece SYBASE con su motor de base de datos relacional versión Adaptive Server Enterprise Express Edition (ASE) de uso libre.

Informix resulta mucho más complejo de instalar. Se puede obtener el archivo rpm de instalación aunque con éste no habremos completado el proceso de instalación. No se crea ninguna base de datos inicialmente, por lo que hay que adivinar cuál es el siguiente paso tras instalar el archivo rpm. Usando documentación, que no viene con la instalación, podemos descubrir cómo crear una base de datos y como configurarla. La documentación puede obtenerse en el portal de Informix.

Oracle resulta muy difícil de instalar. No se dispone de un archivo rpm de instalación. Es posible descargar un instalador particular del portal de Oracle. Incorpora documentación para poder ejecutar el instalador. Se trata de un rudimentario asistente que realiza preguntas al usuario o administrador y presenta una lista de componentes a elegir. Un novato podría perderse en esta parte.

Se mencionarán a continuación características específicas de los sistemas manejadores de Bases de Datos comparadas:

Transacciones: Todos permiten realizar operaciones Commit, Rollback y crear niveles de aislamiento en al menos lectura.

Bloqueo de filas: Oracle permite bloquear filas para actualizar sin bloquearlas para leerlas.

Restricciones: las claves primarias y foráneas que permiten eliminar o añadir registros vinculados están soportadas por completo en Oracle, Informix y Sybase.

Programabilidad: La posibilidad de crear triggers y extensiones procedurales en SQL está soportada en todos ellos.

Resistencia a fallos: Se prueba a borrar 100 registros de un total de 100000. Luego se invoca a commit y se apaga el servidor. Oracle supera la prueba sin pérdida de datos, Informix y Sybase no.

Respaldo en caliente: Sólo Oracle permite realizar un backup simultáneamente a la realización de transacciones.

Carga masiva: Se mide el tiempo en segundos que le lleva a un motor de base de datos cargar un millón de registros: Oracle 65 segundos, Sybase 120 segundos, Informix 170 segundos

Creación de registros: Desde PHP se generan 10000 registros. Oracle 31 segundos. Sybase 45 segundos, Informix no se ha podido medir por falta del Informix SDK

Consulta de registros: Se mide el tiempo en mostrar una suma de los 5000 registros en una tabla que contiene 1 millón. Se efectúa esta operación a través de una consulta JOIN de 2 tablas. Oracle 2 segundos, Sybase 5 segundos, Informix 10 segundos.

Conclusiones de los programas comparados.

Para el manejador de base de datos es necesario elegir que cumpla con las siguientes características:

Confiabilidad. Puesto que los sistemas de bases de datos en línea no deben fallar.

Disponibilidad. Por ser un sistema en línea debe estar actualizado correctamente todo el tiempo.

Tiempos de Respuesta. En sistemas de este tipo, el tiempo de respuesta de las transacciones no debe ser mayor de diez segundos.

Throughput. Los sistemas de base de datos en línea requieren procesar miles de transacciones por segundo.

Atomicidad. En el procesamiento de transacciones no se aceptan resultados parciales.

Permanencia. No se permite la eliminación en la base de datos de los efectos de una transacción que ha culminado con éxito.

Por lo tanto concluimos que: Oracle es el motor de base de datos con más funcionalidades. La seguridad de los datos que es un punto esencial solo se puede fiar en Oracle. Hay que observar que Oracle requiere de un parche para comportarse de forma realmente segura con los datos.

Una vez elegido el motor de base de datos, la selección del servidor de aplicación fue de manera natural el que mejor se adapta a dicho motor de base de datos, se seleccionó WebLogic por que proporciona las siguientes ventajas:

La línea de productos de servidores de aplicaciones Oracle WebLogic 11g es la plataforma más completa del mercado para desarrollar, implantar e integrar aplicaciones empresariales.

Proporciona la base para application grid, una arquitectura que permite a las empresas ofrecer un rendimiento superior al de la competencia pero que al mismo tiempo reduce los costes de explotación.

Sistemas rápidos y con capacidad de ampliación que hacen posibles los cambios informáticos a medida que crece la empresa

Mayor facilidad de uso gracias a la automatización, lo que reduce el tiempo y el coste que supone mantener los sistemas en funcionamiento

La línea de productos se integra perfectamente con el resto de la cartera Oracle, y ofrece una compatibilidad excelente para la interoperabilidad con otros sistemas

Fiabilidad sólida

Facilidad de uso inigualable

Capacidad de ampliación extrema

La más rápida en una infraestructura virtualizada

Rendimiento en tiempo real infalible

Plataforma líder para SOA

Seguridad de aplicaciones ágil

Base para application grid

Instalación automatizada en entornos virtualizados

Generador de reportes

iReport

Desarrollador Jinfonet

Versión actual 2010-07-15

Sistema operativo Plataforma-cruzada

Disponible Multilingüe

Tipo Reportes empresariales

Website <http://jinfonet.com/products/iReport-suite>

Existen varias formas de agregar capacidades de generar reportes a una aplicación. Para varios desarrolladores web, el crear un reporte significa solamente crear una página web, el cual produce buenos resultados en pantalla pero muy pobres cuando se imprimen en papel. y crear un reporte en PDF significa escribir una gran cantidad de código adicional, haciendo que los reportes sean tardados de escribir y difíciles de mantener. Cuando se trabaja con otras tecnologías el problema no cambia significativamente. Por ejemplo, Java provee de una extensa API para impresión, pero aún hay mucho trabajo para escribir el código específico para cada formato del documento.

Para resolver este problema, iReport usa la librería de reportes de Java "JasperReports" que es la librería de código abierto en Java más popular en el mundo, e iReport es una herramienta de diseño visual para JasperReports. La librería es un motor de reportes que puede ser integrada en una aplicación para generar los reportes diseñados con iReport, desplegarlos en pantalla o exportarlos en un formato final como puede ser PDF, OpenOffice, DOCX y muchos otros. Alternativamente se puede enviar el resultado a través de una aplicación web o enviar el documento final directamente a la impresora.

Crystal Reports

Desarrollador: Business Objects

Sistema operativo: Microsoft Windows

Disponible: Multilingüe

Tipo: Software para reportes

Licencia: Propietario

Website: <http://www.sap.com/solutions/sap-crystal-solutions/>

Es una aplicación de inteligencia de negocios usada para diseñar y generar reportes de una amplia gama de fuentes de datos. Permite al usuario diseñar una conexión a datos y la disposición del informe de manera gráfica. Como experto de bases de datos el usuario puede seleccionar y ligar tablas desde una amplia variedad de fuentes, incluyendo hojas de Microsoft Excel, Bases de datos Oracle, Objetos de negocios de vistas de negocios empresariales, e información del sistema de archivos local. Los campos de estas tablas pueden ser colocados en el reporte y puede incluso ser usado en formulas personalizadas, usando ya sea BASIC o la propia sintaxis de Crystal, que se pondrán después en el reporte. Las formulas pueden ser evaluadas en varias fases durante la generación del reporte como lo especifique el desarrollador.

Tanto los campos como las formulas tienen un amplio arreglo de opciones disponibles, que se pueden aplicar de manera absoluta o condicional. Los datos pueden ser agrupados en bandas, cada una de las cuales puede ser dividida aún más y de manera condicional suprimirla cuando sea necesario. Además soporta sub reportes.

BIRT Project

Desarrollador: Eclipse Foundation

Versión actual: 2010-06-24

Escrito en Java

Sistema operativo: Plataforma-cruzada

Disponible: Multilingüe

Tipo: Reportes empresariales / Inteligencia de negocios

Licencia: Eclipse Public License

Website: <http://www.eclipse.org/birt>

El proyecto BIRT (Business Intelligence and Reporting Tools) es un proyecto de software de código abierto que provee capacidades de reporte e inteligencia de negocios para clientes y aplicaciones web, especialmente aquellas basadas en Java y Java EE. BIRT es un sistema generador de reportes para aplicaciones web de código abierto basado en Eclipse, especialmente aquellos basados en Java y Java EE. Tiene dos componentes principales: un diseñador de reportes basado en Eclipse, y un

componente en tiempo de ejecución que se puede agregar al servidor de la aplicación. Es posible también cambiar el motor para agregar características a la aplicación. Los reportes BIRT consisten de cuatro partes principales:

Datos – Bases de datos, servicios web, objetos java pueden suministrar datos hacia el reporte. BIRT provee soporte para JDBC, XML, servicios web, y archivos planos además de soporte para usar códigos provenientes de otras fuentes; usa paquetes de trabajo ODA (Open Data Access) permitiendo a cualquiera construir una nueva interfaz de usuario y soporte en tiempo de ejecución para cualquier tipo de datos tabulares. Además un solo reporte puede incluir datos de cualquier suministro de datos para ser combinados usando joins internos y externos.

Transformación de datos – Los reportes presentan datos ordenados, resumizados filtrados y agrupados para cumplir con las necesidades del cliente. Mientras que la base de datos puede hacer algo de este trabajo, BIRT permite operaciones sofisticadas como son agrupar, sumar, porcentajes de los totales y más.

Lógica de negocios – Los datos del mundo real rara vez se estructuran exactamente como se requieren para un reporte. Muchos reportes requieren de una lógica de negocios específica para convertir los datos crudos en información aprovechable por el usuario. Si la lógica es sólo para el reporte, se puede hacer un script usando el soporte para scripts Java de BIRT. Si la aplicación ya contiene la lógica, se puede llamar el código Java existente.

Presentación – Una vez que los datos están listos, se dispone de un amplio rango de opciones para presentarlos al usuario. Tablas, cartas, texto, y más. Un simple conjunto de datos puede aparecer de múltiples maneras, y un reporte puede presentar datos de varios conjuntos de datos.

En conclusión se usará iReports por que cumple con todas las características que la aplicación requiere para la generación de reporte, y sobre todo por ser una herramienta gráfica que disminuirá en gran medida el tiempo de desarrollo y mantenimiento de los reportes.

Capítulo 4

“DISEÑO Y CONSTRUCCION DE LA APLICACIÓN”



4.1 DIAGRAMACIÓN

4.1.1. DIAGRAMA DE CASOS DE USO.

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la figura 4.1.1 se muestra el diagrama de casos de uso del “Sistema de supervisión de crédito”.



Figura 4.1.1.1 diagrama de casos de uso.

A continuación se muestran los diagramas a detalle de los principales casos de uso.

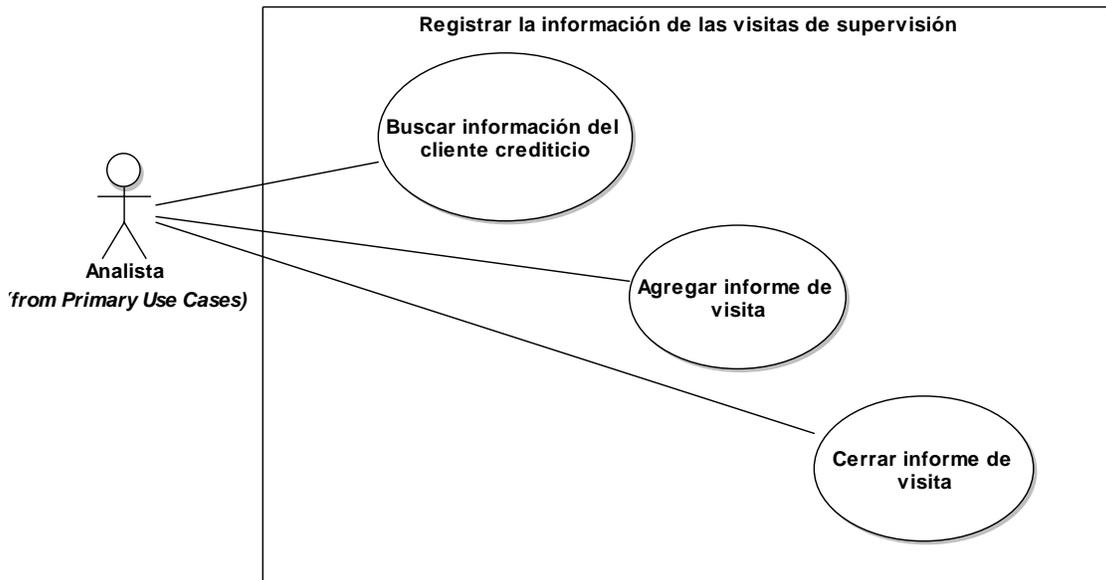


Figura 4.1.1.2 Detalle del caso de uso “Registrar la información de las visitas de supervisión”.

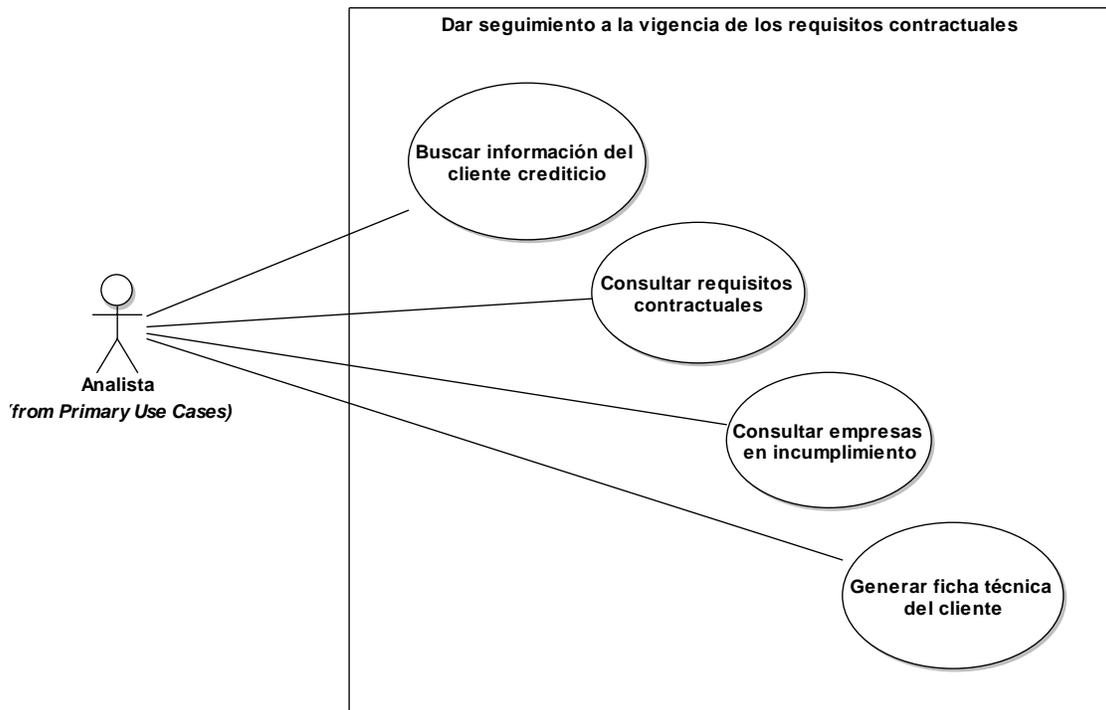


Figura 4.1.1.3 Detalle del caso de uso “Dar seguimiento a la vigencia de los requisitos contractuales”.

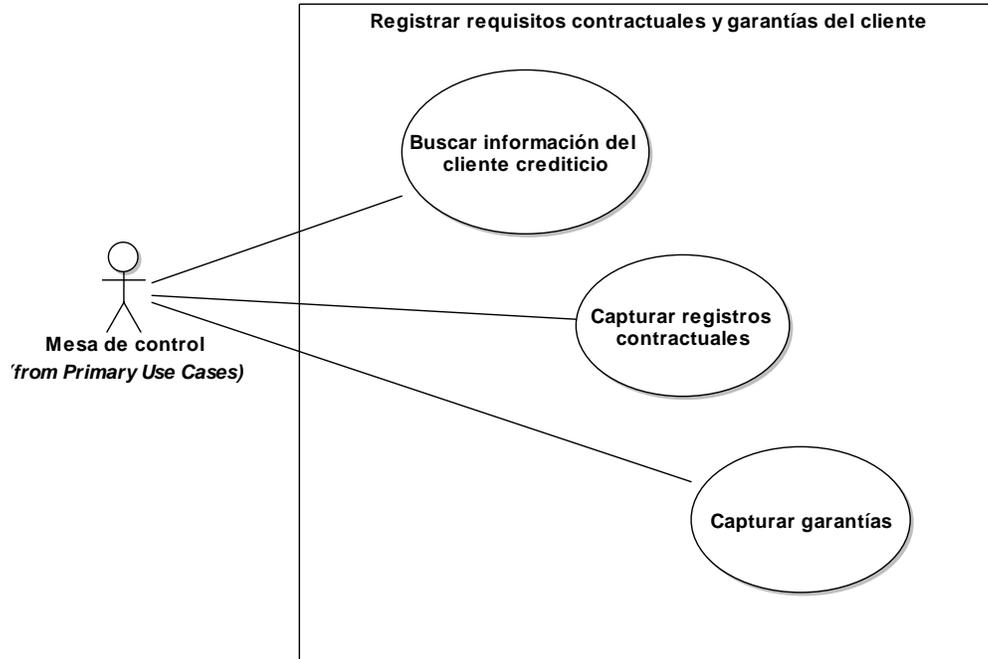


Figura 4.1.1.4 Detalle del caso de uso “Registrar requisitos contractuales y garantías del cliente”.

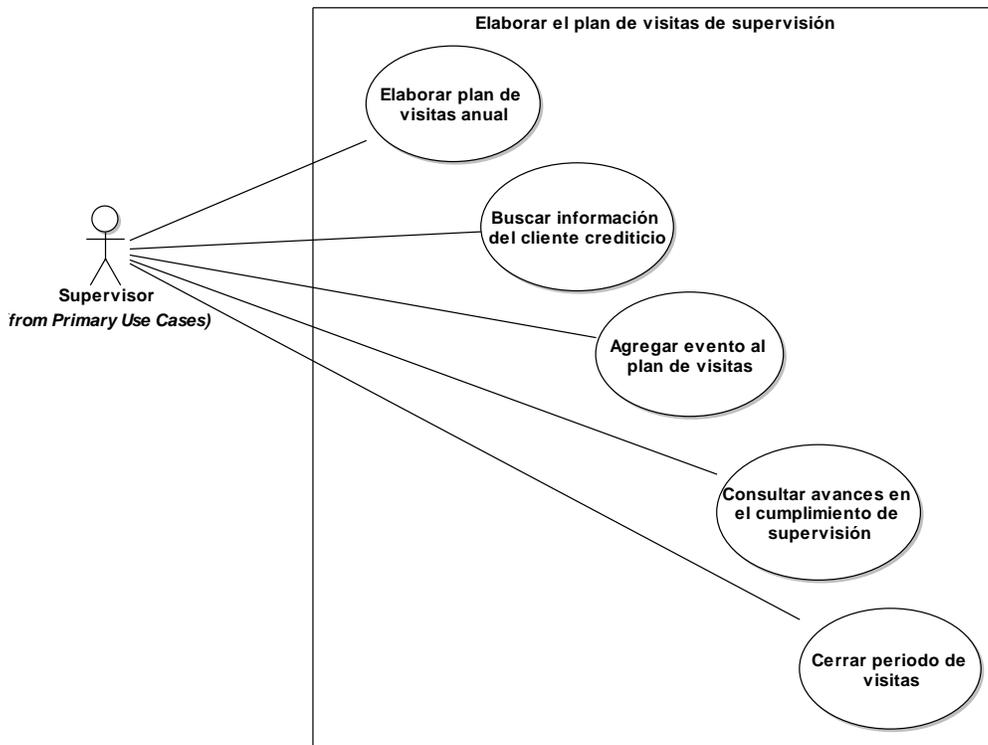


Figura 4.1.1.5 Detalle del caso de uso “Elaborar el plan de visitas de supervisión”.

4.1.2 DIAGRAMA DE MODELOS DE DATOS

Un diagrama de modelo de datos es un esquema que representa el diseño de una base de datos. Este modelo se basa en representaciones entre registros por medio de ligas, existen relaciones en las que participan solo dos entidades (binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación.

Un diagrama de modelo de datos, especifica la estructura lógica global de la base de datos; su representación gráfica se basa en el acomodo de los campos de un registro en un conjunto de celdas que se ligan con otro(s) registro(s)

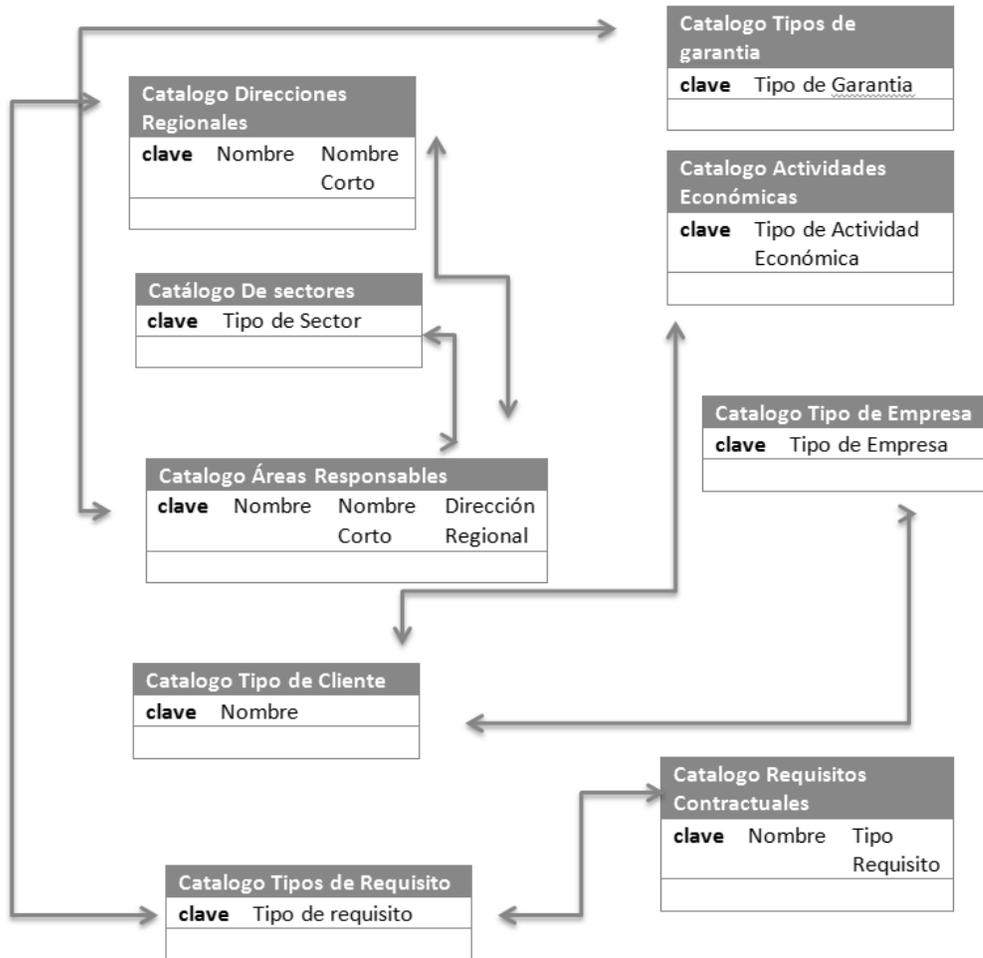


Figura 4.1.2.1 Diagrama de modelo de datos

4.1.3. DIAGRAMA ENTIDAD-RELACIÓN.

El modelo “Entidad-Relación”, es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del diagrama “Entidad-Relación”. Se puede definir como “Entidad” a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información. Los atributos son características o propiedades asociadas a la entidad que toman valor en una instancia particular. Se entiende por “Relación” a la asociación entre 2 o más entidades.

Relación “Uno a Uno”.

Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa. En este caso la llave foránea se ubica en alguna de las 2 tablas.

Relación “Uno a Muchos”.

Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria. En este caso la llave foránea se ubica en la tabla secundaria.

Relación “Muchos a Muchos”.

Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

A continuación, en la figura 4.1.3.1 se muestra el diagrama entidad-relación del “Sistema de supervisión de crédito”.

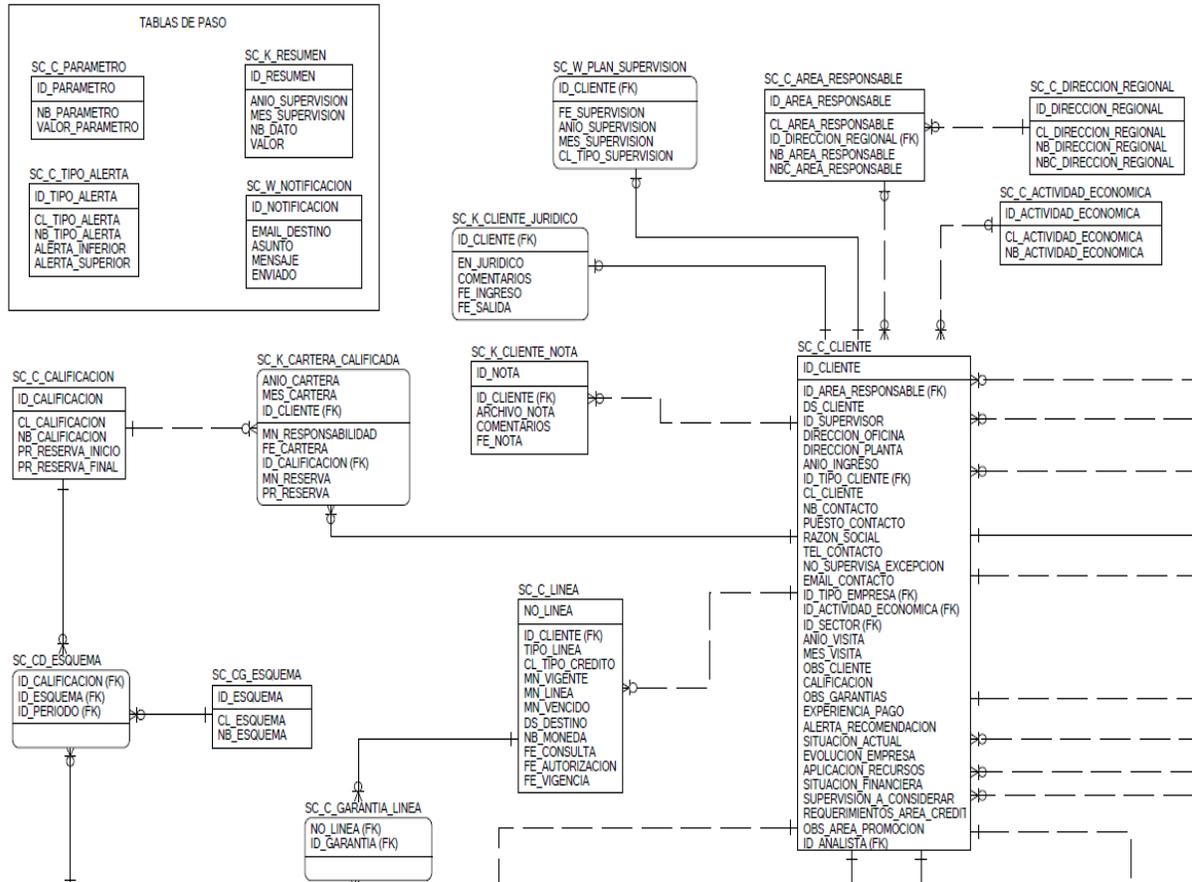


Figura 4.1.3.2 Diagrama ampliado 1

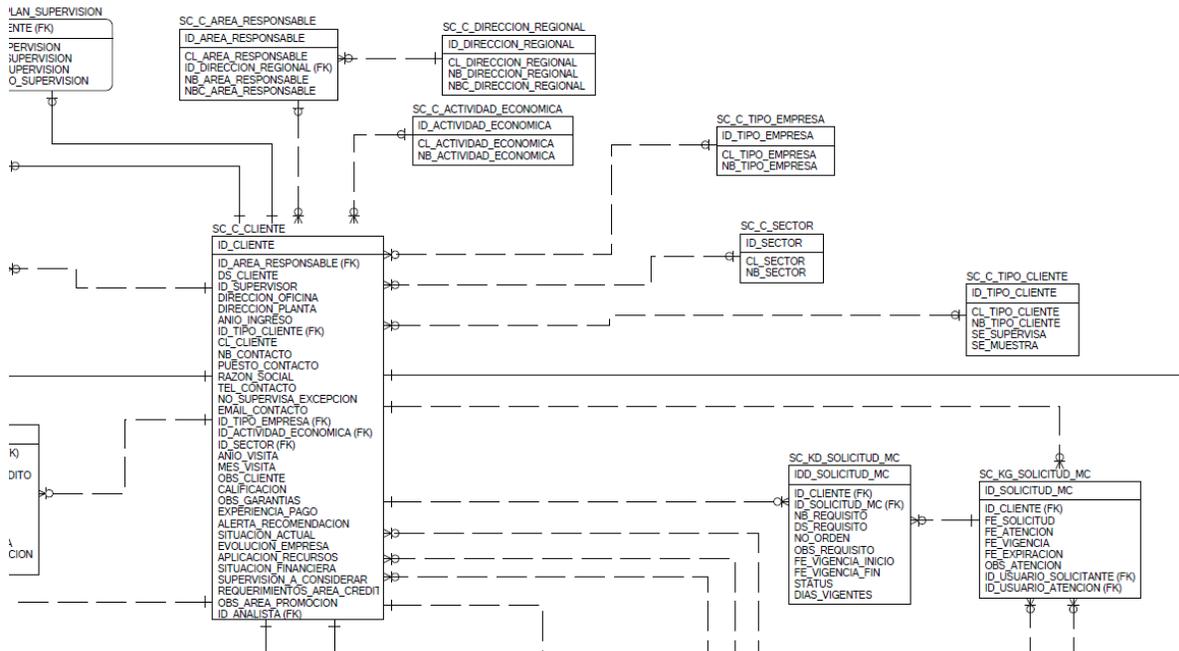


Figura 4.1.3.3 Diagrama ampliado 2

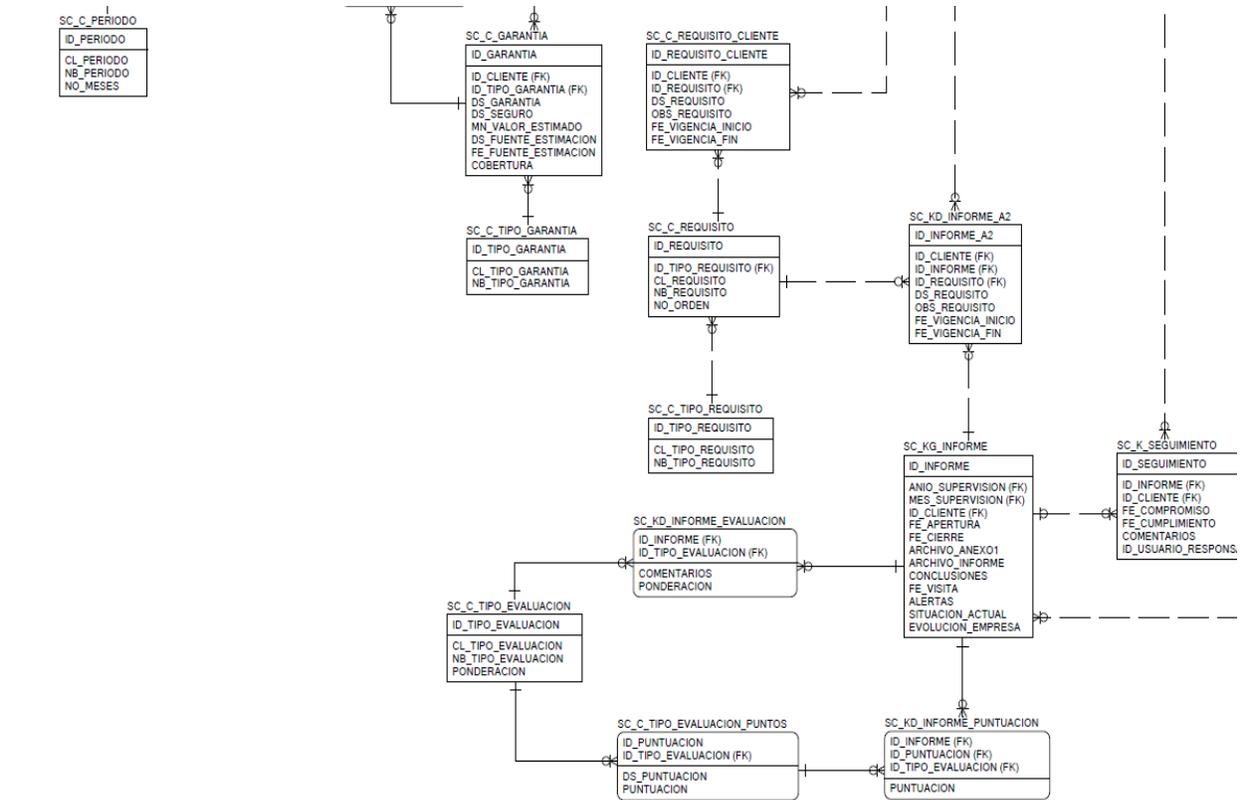


Figura 4.1.3.4 Diagrama ampliado 3

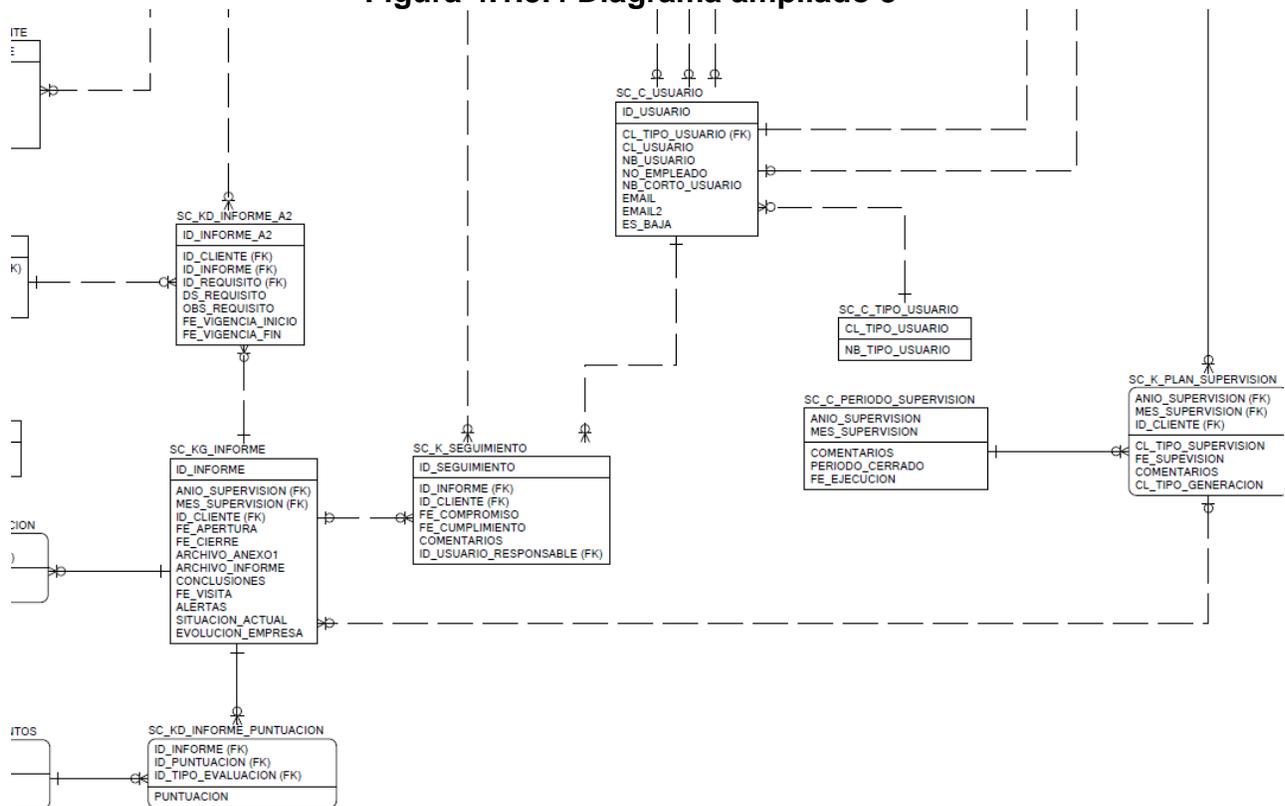


Figura 4.1.3.5 Diagrama ampliado 4

4.1.4 DIAGRAMA DE CLASES

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

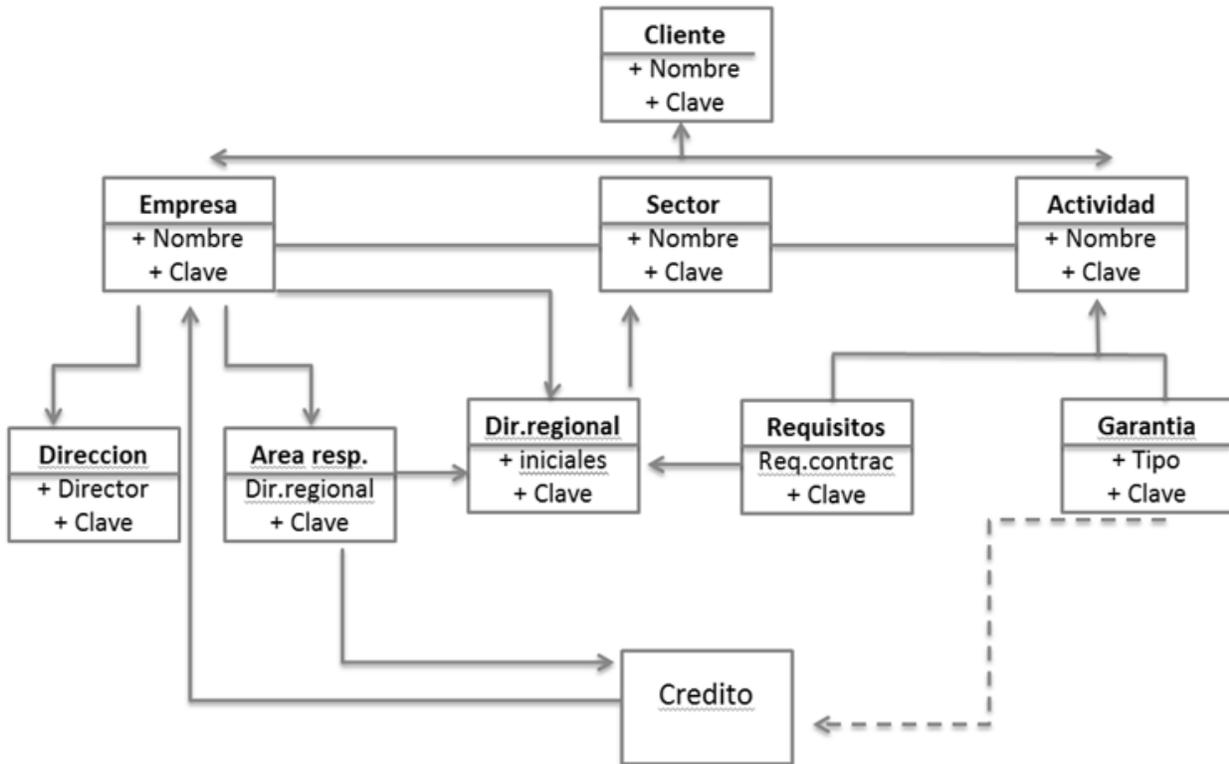


Figura 4.1.2.1 Diagrama de clases

4.1.5 DIAGRAMAS DE SECUENCIA

Este diagrama (también llamado diagrama de interacción) muestra las interacciones entre un conjunto de objetos (clases, actores), ordenadas según el tiempo en que tienen lugar. Es decir, muestra el orden de las llamadas en el sistema. Se utiliza un diagrama para cada llamada a representar. Es imposible representar en un solo diagrama la secuencia de todas las llamadas posibles del sistema, es por ello que se escoge un punto de partida. El diagrama se compone con los objetos que forman parte de la secuencia, estos se sitúan en la parte superior de la pantalla, normalmente a la izquierda se sitúa el que inicia la acción. De estos objetos sale una línea que indica su vida en el sistema. Esta línea simple se convierte en una línea gruesa cuando representa que el objeto tiene el foco del sistema, es decir cuando él está activo.

El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción.

En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos: el objeto origen solicita (llama a) una operación del objeto destino. El diagrama de secuencia puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o desde el de Casos de Usos.

Recopilación de Datos para el análisis crediticio del Cliente

En la figura 4.1.5.1 se muestra el diagrama de secuencia del proceso de recopilación de datos para realizar el análisis crediticio del Cliente.

El proceso consiste en que el Cliente entrega una solicitud de crédito al Promotor este a su vez revisa la solicitud y regresa al Cliente los requisitos de documentación necesaria para entregar el crédito, el Cliente entrega la documentación nuevamente al Promotor y la captura en el sistema de supervisión para que la información sea analizada por el Analista, el resultado del análisis es enviado de regreso al Promotor para que este a su vez se encargue de entregar el resultado al Cliente ya sea que se acepta o se declina la solicitud del crédito.

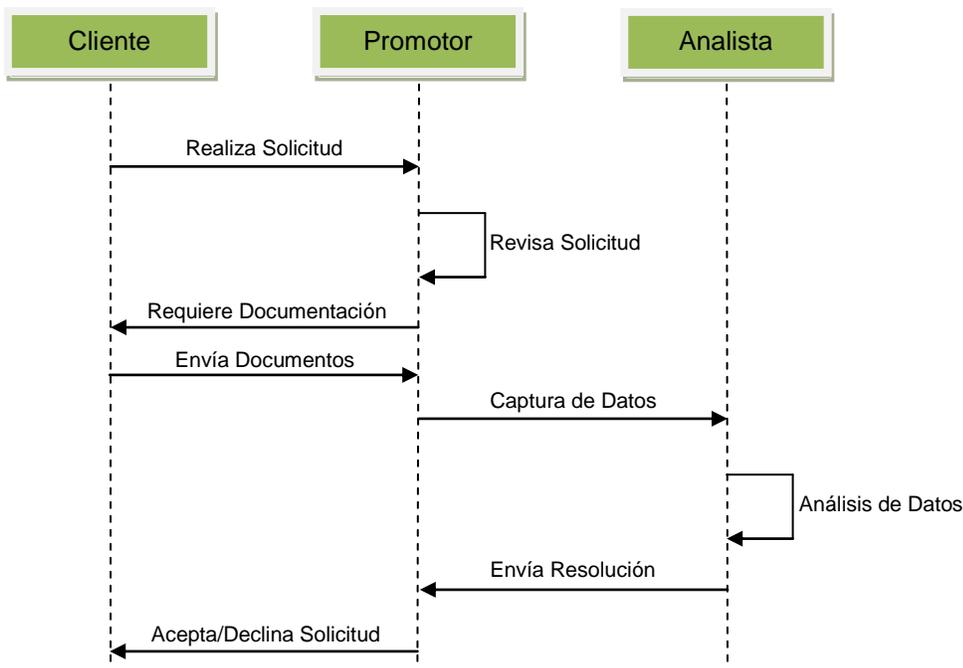


Figura 4.1.5.1 Diagrama de Secuencia del Proceso de Recopilación de Datos.

Registro y Seguimiento de Requisitos Contractuales y Garantías

En la figura 4.1.5.2 se muestra el diagrama de secuencia del proceso de registro y seguimiento de requisitos contractuales y garantías.

El proceso consiste en que una vez aceptada la solicitud del crédito, la Mesa de Control envía al Cliente una solicitud de requisitos contractuales y una solicitud de garantías, el cliente revisa las solicitudes y envía de regreso a la Mesa de Control dichos requerimientos y garantías, estos son registrados en el sistema de supervisión para que el Analista revise la información y le dé seguimiento a los requisitos, en base a esta información genera los reportes de cumplimiento y los envía a la Mesa de Control la cual se encarga de notificar al cliente.

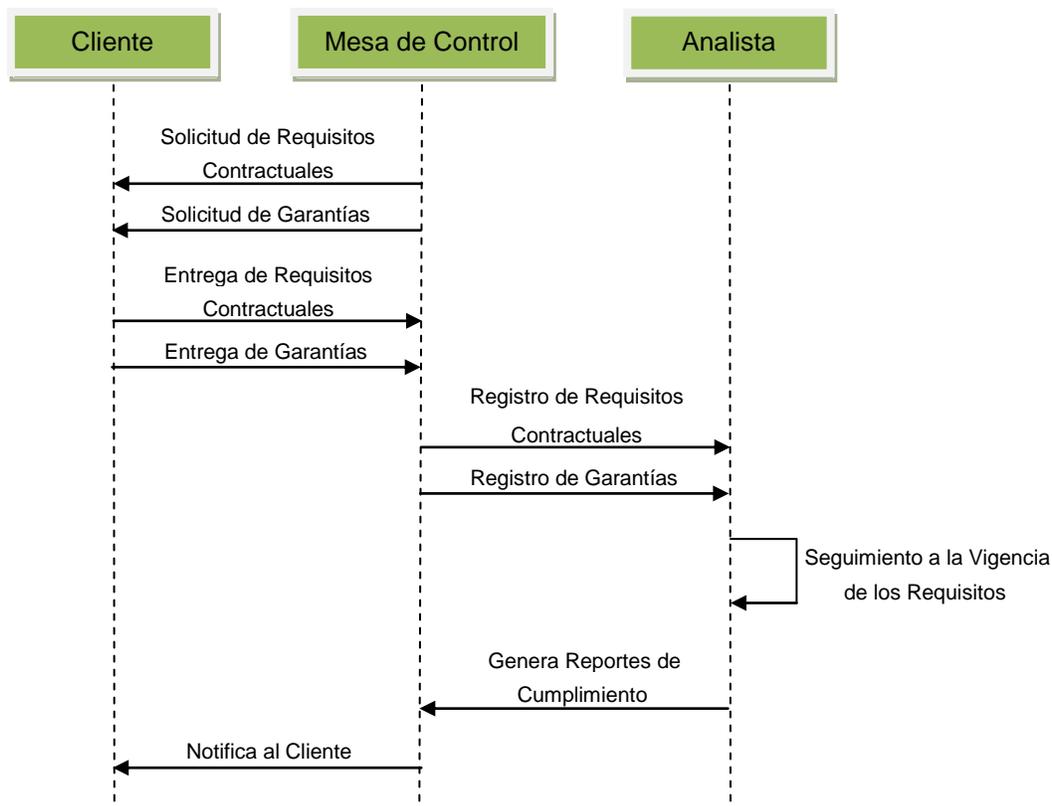


Figura 4.1.5.2 Diagrama de Secuencia del Proceso de Registro y Seguimiento de Requisitos Contractuales y Garantías.

Elaboración y Seguimiento del Plan de Visitas de Supervisión

En la figura 4.1.5.3 se muestra el diagrama de secuencia del proceso de Elaboración y Seguimiento del Plan de Visitas de Supervisión.

El proceso consiste en que una vez que se notifica al cliente sobre los reportes de cumplimiento, el Supervisor elabora un plan de visitas a las instalaciones del cliente para llevar a cabo una Supervisión en sitio, dicho plan es enviado al Comité de Aprobación el cual se encarga de validarlo y enviarlo de regreso al Supervisor quien se encarga de notificar al Cliente, realiza las visitas, , registra la información y genera los reportes e informes correspondientes los cuales envía nuevamente al Comité que se encarga de validar estos documentos y emite una acción correctiva, esta se envía al Supervisor quien es el encargado de tomar con el Cliente dichas acciones.

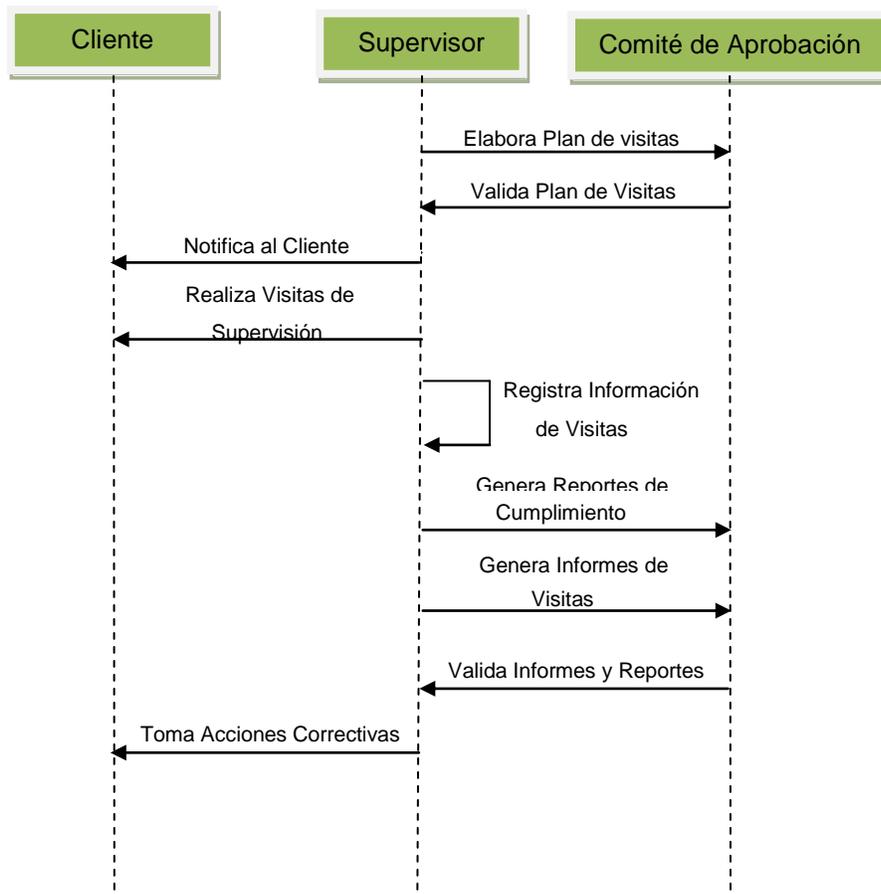


Figura 4.1.5.3 Diagrama de Secuencia del Proceso de Elaboración y Seguimiento del Plan de Visitas.

4.2 Diccionario de datos

Conjunto central de tablas y vistas de la base de datos maestra y de solo lectura

Un diccionario de datos proporciona la siguiente información:

Estructura lógica y física de la BD.

Nombre de los usuarios

Privilegios y roles que tienen los usuarios

Definiciones de los objetos de la BD (Tablas, Vistas, Índices, Sinónimos, Clúster, Secuencias, Procedimientos, Triggers, Tablespace, Funciones, etc.)

Auditoría de información, como los accesos a los objetos

Espacio ocupado por los objetos

Un diccionario de datos debe cumplir las siguientes características:

Debe soportar las descripciones de los modelos conceptual, lógico, interno y externo de la BD.

Debe estar integrado dentro del Sistema Gestor de BD

Debe apoyar la transferencia eficiente de información al Sistema Gestor de BD. La conexión entre los modelos interno y externo debe ser realizada en tiempo de ejecución.

Debe comenzar con la reorganización de versiones de producción de la BD. Además debe reflejar los cambios en la descripción de la BD. Cualquier cambio a la descripción de programas ha de ser reflejado automáticamente en la librería de descripción de programas con la ayuda del diccionario de datos.

Debe estar guardado en un medio de almacenamiento con acceso directo para la fácil recuperación de información.

A continuación se muestran las tablas del modelo Entidad Relación. Desde la 4.2.1 hasta la 4.2.15, que solo son una muestra del total de 39 tablas.

SC_C_ACTIVIDAD_ECONOMICA. Catalogo que contiene las actividades económicas registradas. Ver tabla 4.2.1.

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador de actividad económica	ID_ACTIVIDAD_ECONOMICA	Pk	N	NUMBER (6)	Número que representa el identificador de la actividad económica
Clave actividad económica	CL_ACTIVIDAD_ECONOMICA		N	VARCHAR2 (20 Byte)	Clave asignada a la actividad económica
Nombre actividad económica	NB_ACTIVIDAD_ECONOMICA		N	VARCHAR2 (50 Byte)	Nombre de la actividad económica

Tablas relacionadas: ID_CLIENTE

Tabla 4.2.1 SC_C_ACTIVIDAD_ECONOMICA.

SC_C_AREA_RESPONSABLE. Catalogo que contiene el área (región) responsable de llevar el crédito. Ver tabla 4.2.2.

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador del área responsable	ID_AREA_RESPONSABLE	Pk	N	NUMBER (6)	Número que identifica el área responsable de llevar el crédito
Clave del área responsable	CL_AREA_RESPONSABLE		N	NUMBER (4)	Número que representa la clave de identificación del área responsable
Identificador de dirección regional	ID_DIRECCION_REGIONAL	Fk	N	NUMBER (6)	Número que identifica la dirección regional responsable de llevar el crédito
Nombre del área responsable	NB_AREA_RESPONSABLE		N	VARCHAR2 (50 Byte)	Nombre del área responsable del crédito
Nombre clave del área responsable	NBC_AREA_RESPONSABLE		N	VARCHAR2 (15 Byte)	Nombre clave del área responsable del crédito

Tablas relacionadas: SC_C_CLIENTE, SC_C_DIRECCION_REGIONAL

Tabla 4.2.2 SC_C_AREA_RESPONSABLE.

SC_C_CALIFICACION. Catalogo de calificación asignada al cliente. Ver tabla 4.2.3.

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador de la calificación	ID_CALIFICACION	Pk	N	NUMBER (6)	Número que identifica cada calificación
Clave de la clasificación	CL_CALIFICACION		N	VARCHAR2 (20 Byte)	Clave asignada a cada clasificación
Nombre de la clasificación	NB_CALIFICACION		Y	VARCHAR2 (50 Byte)	Nombre de la clasificación
Porcentaje de reserva inicial	PR_RESERVA_INICIO		N	NUMBER (5,2)	Porcentaje de reserva inicial
Porcentaje de reserva final	PR_RESERVA_FINAL		Y	NUMBER (5,2)	Porcentaje de reserva final

Tablas relacionadas: SC_CD_ESQUEMA, SC_K_CARTERA_CALIFICADA

Tabla 4.2.3 SC_C_CALIFICACION.

SC_C_CLIENTE. Tabla de contacto con el cliente. Ver tabla 4.2.4

Nombre	Acrónimo	Tipo Llave	Acepta		Tipo Dato	Descripción
			Nulos			
Identificador del cliente	ID_CLIENTE	Pk	N		NUMBER (6)	Número que identifica cada cliente
Identificador del área responsable	ID_AREA_RESPONSABLE	Fk	N		NUMBER (6)	Número que identifica cada área responsable del crédito
	DS_CLIENTE		Y		CLOB	Yes
Identificador del supervisor	ID_SUPERVISOR			N	NUMBER (6)	Número que identifica a cada supervisor
Identificador de promotor	ID_PROMOTOR	Fk	N		NUMBER (6)	Número que identifica a cada promotor
Dirección de oficina	DIRECCION_OFICINA		Y		VARCHAR2 (512 Byte)	Contiene la dirección de la oficina del cliente
Identificador del analista	ID_ANALISTA	Fk	N		NUMBER (6)	
Dirección de la planta	DIRECCION_PLANTA		Y		VARCHAR2 (512 Byte)	Contiene la dirección de la planta o taller del cliente
Año de ingreso	ANIO_INGRESO			N	NUMBER (4)	Contiene el año de ingreso del crédito
Identificador de tipo de cliente	ID_TIPO_CLIENTE	Fk	N		NUMBER (6)	Número que identifica el tipo de cliente
Clave del cliente	CL_CLIENTE			N	NUMBER (7)	Clave que identifica a cada cliente
Nombre del contacto	NB_CONTACTO			N	VARCHAR2 (100 Byte)	Contiene el nombre de la persona a contactar
Puesto del contacto	PUESTO_CONTACTO			N	VARCHAR2 (100 Byte)	Contiene el puesto de la persona a contactar
Razón social	RAZON_SOCIAL			N	VARCHAR2 (100 Byte)	Contiene la razón social de la empresa
Teléfono del contacto	TEL_CONTACTO		Y		VARCHAR2 (100 Byte)	Contiene el teléfono de la persona a contactar
	NO_SUPERVISA_EXCEPCION		Y		NUMBER (1)	
E-mail del contacto	EMAIL_CONTACTO			N	VARCHAR2 (100 Byte)	Contiene el correo electrónico de la persona a contactar
Identificador del tipo de empresa	ID_TIPO_EMPRESA	Fk	N		NUMBER (6)	Número que identifica a cada tipo de empresa
Identificador de actividad económica	ID_ACTIVIDAD_ECONOMICA	Fk	N		NUMBER (6)	Número que identifica a cada actividad económica
Identificador de sector	ID_SECTOR	Fk	N		NUMBER (6)	Número que identifica a cada sector
Año de la visita	ANIO_VISITA			Y	NUMBER (4)	Contiene el año en que se agendó la visita
Mes de la visita	MES_VISITA			Y	NUMBER (2)	Contiene el mes del año en que se agendó la visita
Observaciones al cliente	OBS_CLIENTE		Y		CLOB	Contiene las observaciones del crédito
Calificación	CALIFICACION		Y		NUMBER (3)	Contiene la calificación asignada al cliente
Observaciones de garantías	OBS_GARANTIAS		Y		CLOB	Contiene las observaciones de las garantías del crédito
Experiencia de pago	EXPERIENCIA_PAGO		Y		CLOB	Experiencia de pago
Alerta de recomendación	ALERTA_RECOMENDACION		Y		CLOB	Alerta de recomendación
Situación actual	SITUACION_ACTUAL		Y		CLOB	Situación actual
Evolución de la empresa	EVOLUCION_EMPRESA		Y		CLOB	Evolución de la empresa
Aplicación de los recursos	APLICACION_RECURSOS		Y		CLOB	Aplicación de los recursos
Situación financiera	SITUACION_FINANCIERA		Y		CLOB	Situación financiera
Supervisión a considerar	SUPERVISION_A_CONSIDERAR		Y		CLOB	Supervisión a considerar
Requerimientos del área de crédito	REQUERIMIENTOS_AREA_CREDITO			N	CLOB	Requerimientos del área de crédito
Observaciones del área de promoción	OBS_AREA_PROMOCION		Y		CLOB	Observaciones del área de promoción
Identificador del promotor	ID_PROMOTOR2	Fk	N		NUMBER (6)	Número que identifica a cada promotor

Tablas relacionadas: SC_C_LINER, SC_R_CLIENTE_NOTA, SC_R_CLIENTE_JURIDICO, SC_W_PLAN_SUPERVISION, SC_C_AREA_RESPONSABLE, SC_C_DIRECCION_REGIONAL, SC_C_ACTIVIDAD_ECONOMICA, SC_C_TIPO_EMPRESA, SC_C_SECTOR, SC_C_TIPO_CLIENTE, SC_K_PLAN_SUPERVISION, SC_KG_SOLICITUD_MC, SC_KD_SOLICITUD_MC, SC_C_USUARIO, SC_K_SEGUIMIENTO, SC_KD_INFORME_A2, SC_C_REQUISITO_CLIENTE, SC_C_GARANTIA,

Tabla 4.2.4 SC_C_CLIENTE.

SC_CD_ESQUEMA. Ver tabla 4.2.5

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador de la calificación	ID_CALIFICACION	Pk	N	NUMBER (6)	Número que identifica cada calificación
Identificador del periodo	ID_PERIODO	Fk	N	NUMBER (6)	Número que identifica cada periodo
Identificador de esquema	ID_ESQUEMA	Fk	N	NUMBER (6)	Número que identifica cada periodo

Tablas relacionadas: SC_CG ESQUEMA, SC_C PERIODO

Tabla 4.2.5 SC_CD_ESQUEMA

SC_C_DIRECCION_REGIONAL. Ver tabla 4.2.6

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador de dirección regional	ID_DIRECCION_REGIONAL	Pk	N	NUMBER (6)	Número que identifica la dirección regional responsable de llevar el crédito
Clave Dirección regional	CL_DIRECCION_REGIONAL		N	NUMBER (4)	Número que identifica cada dirección regional
Nombre de la dirección regional	NB_DIRECCION_REGIONAL		N	VARCHAR2 (50 Byte)	Nombre de la dirección regional
Nombre clave de la dirección regional	NBC_DIRECCION_REGIONAL		N	VARCHAR2 (15 Byte)	Nombre clave de la dirección regional

Tablas relacionadas: SC_C AREA_RESPONSABLE

Tabla 4.2.6 SC_C_DIRECCION_REGIONAL.

SC_C_GARANTIA. Ver tabla 4.2.7

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador de garantía	ID_GARANTIA	Pk	N	NUMBER (10)	Número que identifica cada garantía
Identificador cliente	ID_CLIENTE	Fk	N	NUMBER (6)	Número que identifica a cada cliente
Identificador de tipo de garantía	ID_TIPO_GARANTIA	Fk	N	NUMBER (6)	Número que identifica cada tipo de garantía
Descripción de la garantía	DS_GARANTIA		Y	CLOB	Descripción de la garantía
Descripción del seguro	DS_SEGURO		Y	CLOB	Descripción del seguro
Monto del valor estimado	MN_VALOR_ESTIMADO		Y	NUMBER (19,6)	Monto del valor estimado
Descripción de la fuente de estimación	DS_FUENTE_ESTIMACION		Y	VARCHAR2 (100 Byte)	Descripción de la fuente de estimación
Fecha de la estimación	FE_FUENTE_ESTIMACION		Y	DATE	Fecha de la estimación
Cobertura	COBERTURA		Y	VARCHAR2 (10 Byte)	Cobertura

tablas relacionadas: SC_C TIPO_GARANTIA, SC_C GARANTIA LINEA, SC_C LINEA, SC_C CLIENTE

Tabla: 4.2.7 SC_C_GARANTIA.

SC_C_GARANTIA_LINEA. Catalogo número de línea de crédito y garantía Ver tabla 4.2.8

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Número de línea de crédito	NO_LINEA	Fk	N	NUMBER (10)	Número que representa línea de crédito
Identificador de garantía	ID_GARANTIA	Fk	N	NUMBER (10)	Número que identifica a cada garantía

Tablas relacionadas: SC_C LINEA, SC_C GARANTIA

Tabla 4.2.8 SC_C_GARANTIA_LINEA.

SC_CG_ESQUEMA. Catalogo de esquemas registrados. Ver tabla 4.2.9

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador del esquema	ID_ESQUEMA	Pk	N	NUMBER (6)	Número que identifica cada esquema
Clave del esquema	CL_ESQUEMA		N	VARCHAR2 (20 Byte)	Clave del esquema
Nombre del esquema	NB_ESQUEMA		N	VARCHAR2 (100 Byte)	Nombre del esquema

Tablas relacionadas: SC_CD_ESQUEMA

Tabla 4.2.9 SC_CG_ESQUEMA.

SC_C_INFORMES. Catalogo de informes programados Ver tabla 4.2.10

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Número de cliente	CLIENTE		Y	NUMBER (9)	Número de cliente
Tipo de informe	TIPO_INFORME		Y	CHAR (1 Byte)	Tipo de informe
Fecha del informe	FECHA_INFORME		Y	CHAR (4 Byte)	Fecha del informe
Fecha de la visita	FECHA_VISITA		Y	CHAR (4 Byte)	Fecha de la visita
Número consecutivo	CONSECUTIVO		Y	NUMBER (3)	Número consecutivo

Tablas relacionadas:

Tabla 4.2.10 SC_C_INFORMES.

SC_C_LINEA. Catalogo de líneas de crédito. Ver tabla 4.2.11

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Número de la línea	NO_LINEA	Pk	N	NUMBER (10)	Número que identifica cada línea
Identificador del cliente	ID_CLIENTE	Fk	N	NUMBER (6)	Número que identifica a cada cliente
Tipo de línea	TIPO_LINEA		Y	VARCHAR2 (1 Byte)	Tipo de línea
Clave del tipo de línea	CL_TIPO_CREDITO		N	VARCHAR2 (40 Byte)	Clave del tipo de línea
Monto vigente	MN_VIGENTE		Y	NUMBER (19,6)	Monto vigente
Monto de la línea	MN_LINEA		Y	NUMBER (19,6)	Monto de la línea
Monto vencido	MN_VENCIDO		Y	NUMBER (19,6)	Monto vencido
Descripción destino	DS_DESTINO		Y	CLOB	Descripción del uso del crédito
Nombre de la moneda	NB_MONEDA		N	VARCHAR2 (30 Byte)	Nombre de la moneda usada
Fecha de consulta	FE_CONSULTA		N	DATE	Fecha de consulta
Fecha de autorización	FE_AUTORIZACION		N	DATE	Fecha de autorización
Fecha de vigencia	FE_VIGENCIA		N	DATE	Fecha de vigencia

Tablas relacionadas: SC_C_GARANTIA_LINEA, SC_C_CLIENTE

Tabla 4.2.11 SC_C_LINEA.

SC_C_PARAMETRO. Catalogo de parámetros del crédito. Ver tabla 4.2.12

Nombre	Acrónimo	Tipo Llave	Acept a Nulos	Tipo Dato	Descripción
Identificador del parámetro	ID_PARAMETRO	Pk	N	NUMBER (1)	Número que identifica a cada parámetro
Nombre del parámetro	NB_PARAMETRO		Y	VARCHAR2 (20 Byte)	Nombre del parámetro
Valor del parámetro	VALOR_PARAMETRO		Y	VARCHAR2 (20 Byte)	Valor del parámetro

Tablas relacionadas:

Tabla 4.2.12 SC_C_PARAMETRO.

SC_C_PERIODO. Catalogo de periodos del crédito. Ver tabla 4.2.13

Nombre	Acrónimo	Tipo Llave	Accept a Nulos	Tipo Dato	Descripción
Identificador del período	ID_PERIODO	Pk	N	NUMBER (6)	Número que identifica a cada período
Calve del período	CL_PERIODO		N	VARCHAR2 (20 Byte)	Clave de cada período
Nombre del período	NB_PERIODO		Y	VARCHAR2 (50 Byte)	Nombre del período
Número de meses	NO_MESES		Y	NUMBER (2)	Número de meses
Tablas relacionadas: SC_CD_ESQUEMA					
Tabla 4.2.13 SC_C_PERIODO.					

SC_C_PERIODO_SUPERVISION. Catalogo de supervisiones. Ver tabla 4.2.14

Nombre	Acrónimo	Tipo Llave	Accept a Nulos	Tipo Dato	Descripción
Año de supervisión	ANIO_SUPERVISION	Pk	N	NUMBER (4)	Año de la supervisión
Mes de supervisión	MES_SUPERVISION	Pk	N	NUMBER (2)	Mes de la supervisión
Comentarios	COMENTARIOS		Y	CLOB	Comentarios de la supervisión
Período cerrado	PERIODO_CERRADO		Y	NUMBER (1)	Período que se cerro
Fecha de ejecución	FE_EJECUCION		Y	DATE	Fecha de ejecución
Tablas relacionadas: SC_K_PLAN_SUPERVISION					
Tabla 4.2.13 SC_C_PERIODO_SUPERVISION.					

SC_C_REQUISITO. Catalogo de requisitos cubiertos para el crédito. Ver Tabla 4.2.14

Nombre	Acrónimo	Tipo Llave	Accept a Nulos	Tipo Dato	Descripción
Identificador del requisito	ID_REQUISITO	Pk	N	NUMBER (6)	Número que identifica a cada requisito
Identificador del tipo de requisito	ID_TIPO_REQUISITO	Fk	N	NUMBER (6)	Número que identifica cada tipo de requisito
Clave del requisito	CL_REQUISITO		N	VARCHAR2 (20 Byte)	Clave del requisito
Nombre del requisito	NB_REQUISITO		N	VARCHAR2 (150 Byte)	Nombre del requisito
No de orden	NO_ORDEN		N	NUMBER (3)	Orden asignado a los requisitos
Tablas relacionadas: SC_C_REQUISITO_CLIENTE, SC_C_TIPO_REQUISITO, SC_KD_INFORME_A2					
Tabla 4.2.14 SC_C_REQUISITO.					

SC_C_REQUISITO_CLIENTE. Catalogo de requisitos del cliente. Ver tabla 4.2.15

Nombre	Acrónimo	Tipo Llave	Accept a Nulos	Tipo Dato	Descripción
Identificador del cliente	ID_CLIENTE	Pk	N	NUMBER (6)	Número que identifica a cada cliente
Identificador del requisito del cliente	ID_REQUISITO_CLIENTE	Fk	N	NUMBER (10)	Número que identifica a cada requisito del cliente
Identificador del requisito descripción del requisito	ID_REQUISITO_DS_REQUISITO	Fk	N	NUMBER (6)	Número que identifica a cada requisito
Observaciones al requisito	OBS_REQUISITO		Y	CLOB	Descripción del requisito
Fecha de inicio de vigencia	FE_VIGENCIA_INICIO		N	DATE	Observaciones al requisito
Fecha de fin fe vigencia	FE_VIGENCIA_FIN		N	DATE	Fecha de inicio de vigencia
Tablas relacionadas: SC_C_REQUISITO, SC_C_CLIENTE					
Tabla 4.2.15 SC_C_REQUISITO_CLIENTE.					

4.3 DISEÑO Y CONSTRUCCIÓN DEL BACK END.

El diseño del BackEnd para el sistema está compuesto por la construcción de la base de datos en el manejador Oracle en este caso utilizaremos ORACLE 10g Express. Esta tarea es realizada con la ayuda de dicho manejador propio de Oracle para ello lo invocaremos desde la siguiente ruta.

>> "Inicio" - "Programas" - "Oracle - OraHome92" - "Configuration and Migration Tools" - "Database Configuration Assistant", como es la siguiente figura 4.3.1

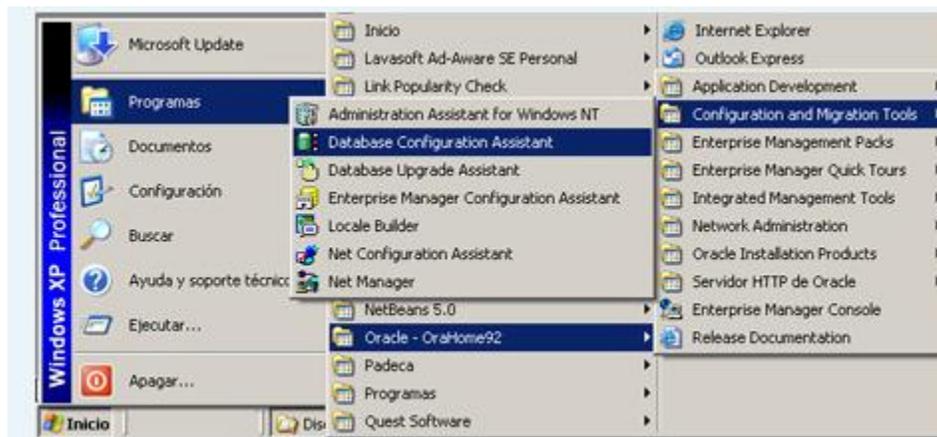


Figura 4.3.1. Ruta en Windows del Manejador de Bases de Datos de Oracle 10g

Nos mostrará el asistente de configuración de bases de datos, marcaremos la opción "Crear una base de datos" y pulsaremos "Siguiente", como lo muestra la figura 4.3.2



Figura 4.3.2 Asistente de configuración de Bases de Datos

En el siguiente paso especificaremos una plantilla con opciones preestablecidas, seleccionaremos la que más se ajuste a nuestras necesidades. Puesto que estamos creando una nueva base de datos, seleccionaremos "New Database" y pulsaremos "Siguiente", como se muestra en la figura 4.3.3



Figura 4.3.3 Creación de una base de datos por medio del asistente de Configuración de Oracle

En el siguiente paso especificaremos el nombre global de la base de datos y el SID:

Nombre de la Base de Datos Global: Las bases de datos Oracle10g se identifican únicamente mediante un Nombre de la Base de Datos Global, nombre completo de la base de datos que la identifica como única distinguiéndola de las demás bases de datos. El nombre de la base de datos global tiene el formato nombre_basedatos.dominio_basedatos". Por ejemplo: test.us.acme.com La parte del nombre de la base de datos, test, es el nombre sencillo que desea asignar a la base de datos. La parte del dominio de la base de datos, us.acme.com, especifica el dominio en el que se ubica la base de datos, con lo cual, la base de datos global tiene asignado un nombre único. Si es posible, Oracle recomienda que el dominio de la base de datos refleje el dominio de la red. La parte del nombre de la base de datos es una cadena de 8 caracteres como máximo que puede contener caracteres alfabéticos, numéricos y adicionales. La parte del dominio es una cadena de 128 caracteres como máximo que puede contener caracteres alfabéticos, numéricos, adicionales y el símbolo del punto (.).

SID: identifica la instancia específica del software Oracle9i en ejecución. Para cada base de datos, hay al menos una instancia que hace referencia a la base de datos. El SID no puede ser un nombre utilizado para otra instancia de esta computadora. Introduzca el identificador del sistema (SID) para la base de datos o acepte el SID por defecto. El SID es un identificador único de una instancia de base de datos Oracle que puede tener una longitud hasta ocho caracteres alfanuméricos. Cada instancia de base de datos se corresponde con un SID y una serie de archivos de base de datos. Por ejemplo, cuando se crea una base de datos con el SID XXX, también se crea la instancia de base de datos y sus archivos de base de datos (el archivo de parámetros de inicialización, el archivo de control, los archivos redo log y los archivos de datos).

Posterior a la creación de la base de datos se debe inicializarla o como comúnmente se le dice “levantar la base de datos”, para ello ingresaremos en la siguiente ruta:

>>Inicio >Todos los programas >Base de Datos Oracle 10g express Edition >Iniciar base de datos, como se muestra en la figura 4.3.6

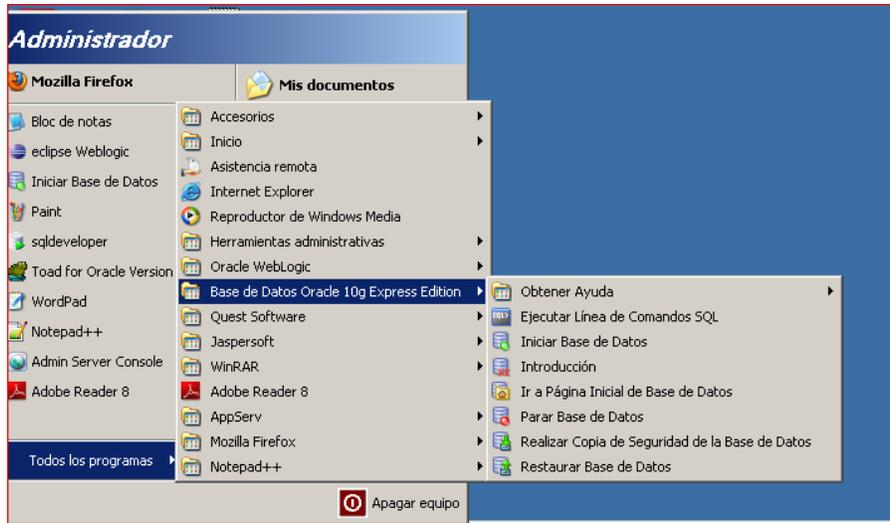


Figura 4.3.6 Inicializar la base de datos de Oracle

Posterior a ello veremos como el sistema ejecuta una serie de instrucciones internas como lo podemos ver en la figura 4.3.7

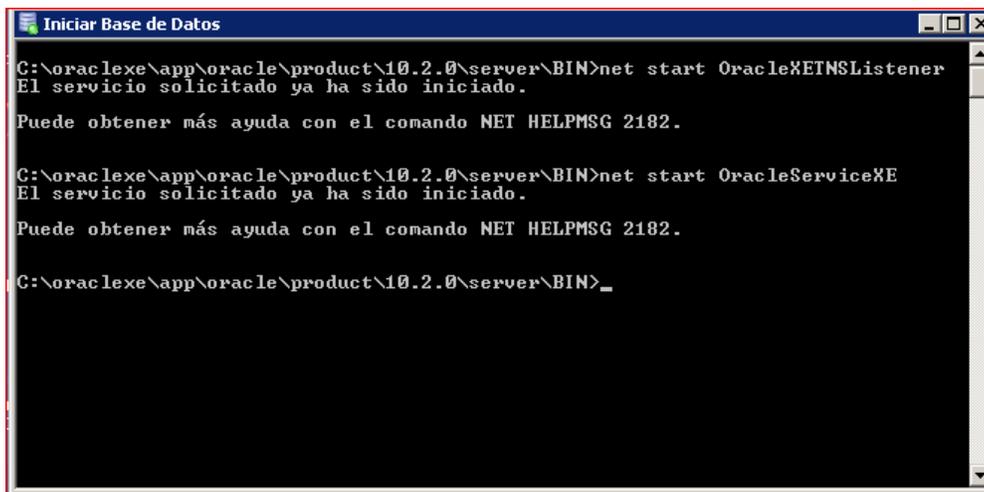


Figura 4.3.7 Scripts de inicialización de la base de datos

Una base de datos es un sistema de almacenamiento de datos basado en un conjunto de tablas unidas mediante relaciones. ORACLE SQL Developer nos permite generar estos elementos para tener la estructura necesaria y adecuada para nuestra aplicación.

Algunos de los componentes de nuestra estructura son:

Tablas, columnas y tipos de datos: las tablas son el mecanismo de almacenamiento de los datos en una base de datos Oracle. Contienen un conjunto fijo de columnas. Las columnas de una tabla describen los atributos de la entidad que se representa con la tabla. Cada columna tiene un nombre y unas características específicas: tipo de datos y

longitud, restricciones, etc. Para la creación de una tabla dentro de la herramienta de administración de Oracle 10g basta con dar botón derecho sobre el listado de tablas y seleccionar crear nueva tabla, tal como lo muestra la figura 4.3.8

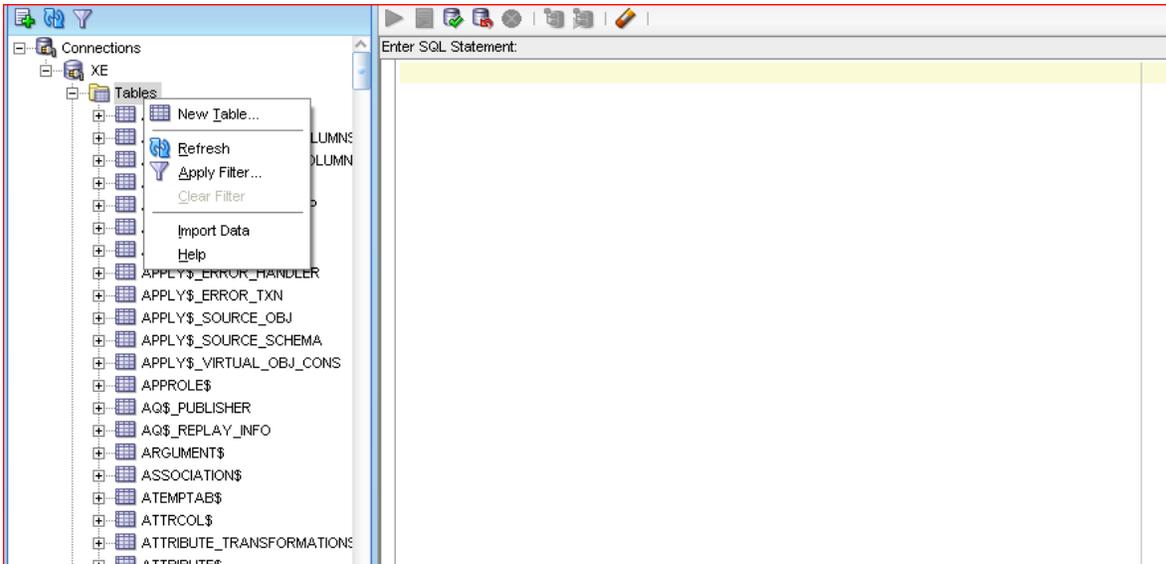


Figura 4.3.8 Creación de tablas

Una vez creada la tabla debemos de definir el contenido y para ello seleccionamos los tipos de datos que almacenara, tal como lo muestra la figura 4.3.9

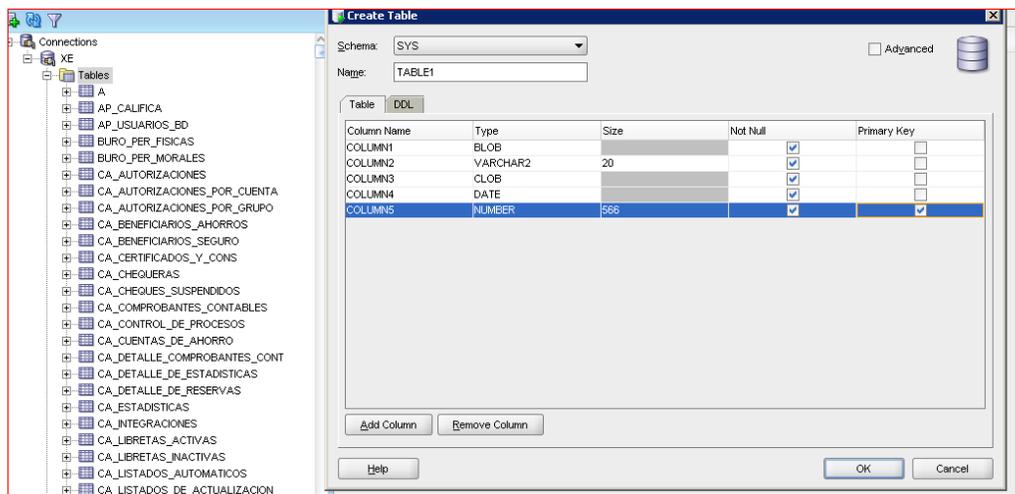


Figura 4.3.9 Definición de la tabla y datos de la misma

Tablas temporales: al igual que una tabla normal, una tabla temporal es un mecanismo de almacenamiento de datos que reside en una base de datos Oracle. Una tabla temporal está compuesta por columnas para las que se han definido tipos de datos y longitudes. A diferencia de una tabla normal, la configuración de una tabla temporal se

conserva, mientras que los datos insertados en la tabla permanecen en ella o bien durante una sesión, o bien, durante una transacción.

Restricciones de columnas: se puede crear restricciones en las columnas de una tabla, cuando esto ocurre, todas las filas de la tabla deben cumplir las condiciones que se especifiquen en la definición de la restricción.

PRIMARY KEY: la columna será clave primaria de la tabla (no podrá contener valores duplicados ni nulos). Se creará un índice automáticamente para esta columna.

NOT NULL: esta columna no podrá contener valores nulos.

DEFAULT VALOR: la columna, al crearse un nuevo registro, contendrá, por defecto, el valor especificado en VALOR.

CHECK (FORMULA): la columna deberá cumplir la fórmula especificada en FORMULA. Por ejemplo: *CHECK (salario > 1000)* en este caso, el valor de la columna *salario* deberá ser mayor que 1000.

UNIQUE: los valores de la columna a la que se le aplique esta restricción no podrán duplicarse (al igual que PRIMARY KEY). Se creará un índice automáticamente para esta columna.

FOREIGN KEY (NOMBRECAMPOTABLA ACTUAL) REFERENCES NOMBRETABLAFORANEA (NOMBRECAMPOTABLAFORANEA): este tipo de restricción se utiliza para especificar las relaciones entre tablas (clave externa o clave foránea). Una clave foránea de una tabla (en este caso NOMBRECAMPOTABLA ACTUAL) hace referencia a una clave primaria que se haya definido previamente en cualquier otra tabla (en este caso la tabla externa será NOMBRETABLAFORANEA y la clave primaria de esta tabla será NOMBRECAMPOTABLAFORANEA). Es importante mencionar que para crear una restricción de clave foránea en una tabla, previamente ha de existir la tabla y la columna de la tabla (como clave primaria) a la que se quiera referenciar, ver figura 4.3.10

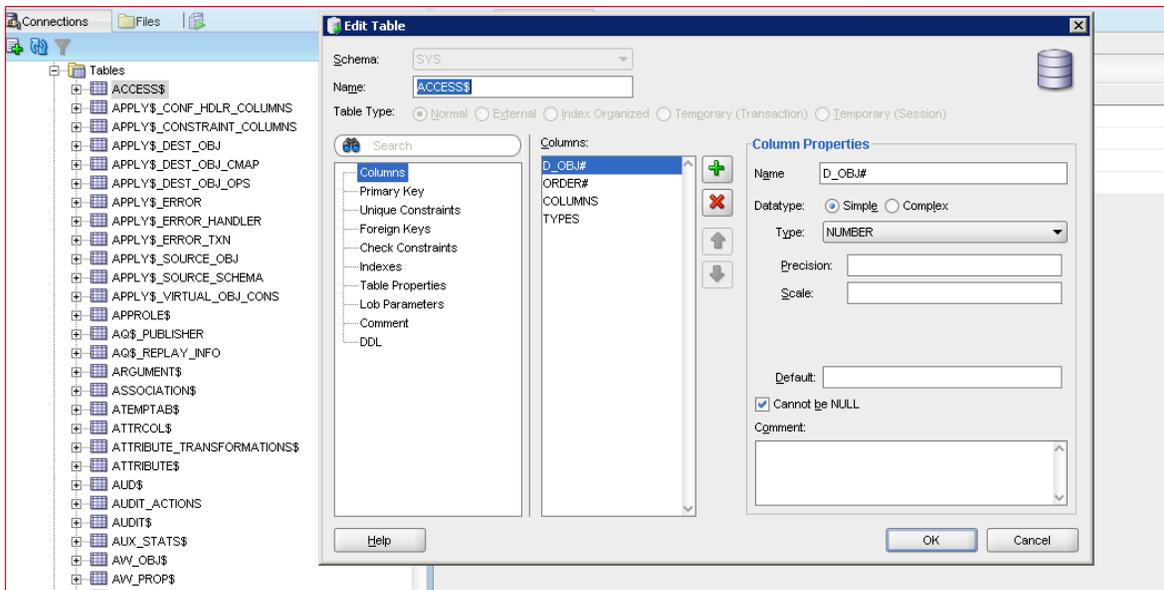


Figura 4.3.10 Componentes de una tabla

Tipos de datos de ORACLE más habituales

CHAR: Campo de caracteres de longitud fija, con un máximo de 2000.

VARCHAR2: campo de caracteres de longitud variable, con un máximo de 4000.

DATE: Almacena la fecha y la hora. En las consultas la fecha aparece en formato DD-MON-YY (por ejemplo: 18-APR-05), salvo que se modifique el parámetro NLS_DATE_FORMAT o se utilice alguna función especial.

NUMBER: Columna numérica de longitud variable. Se puede especificar la precisión entre paréntesis, por ejemplo: NUMBER (9,2) será un número de nueve dígitos de los cuales dos serán decimales, por ejemplo: 5235323,85.

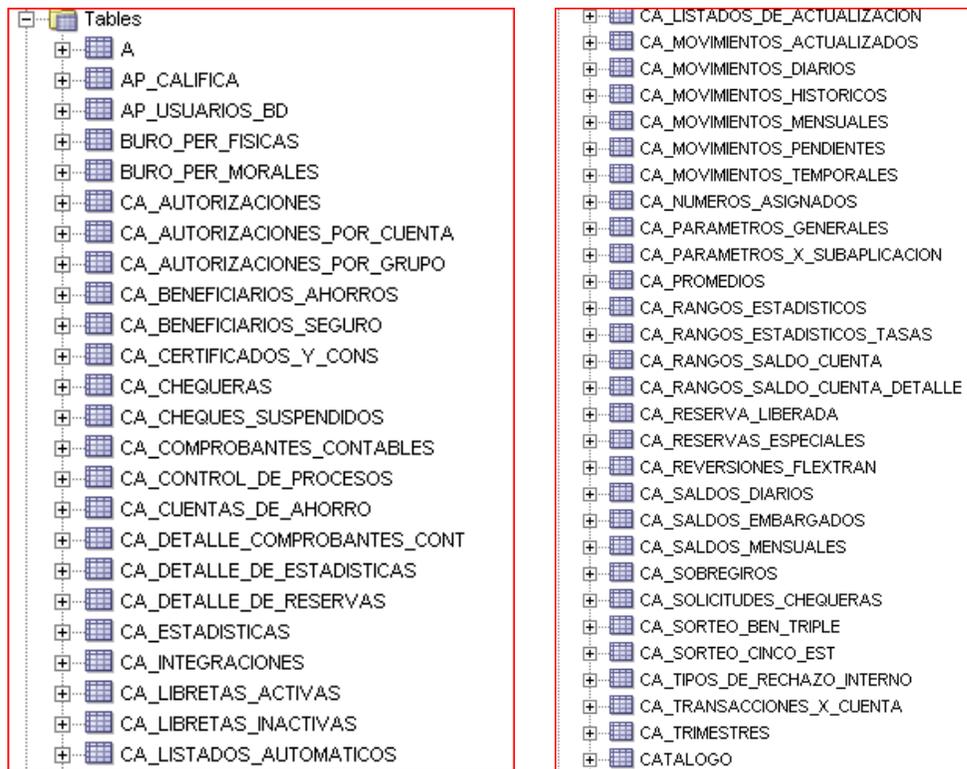
CLOB: Objeto de caracteres de gran tamaño, de hasta 4 GB de longitud. Es el equivalente al MEMO en Access y Paradox.

BLOB: Objeto binario de gran tamaño, de hasta 4 GB de longitud.

ROWID: Datos binarios que representan un identificador de fila.

Tipos abstractos de datos: A partir de Oracle8 el administrador puede definir sus propios tipos de datos. Por ejemplo se puede crear un tipo de datos que contenga todas las partes del nombre de una persona (nombre, segundo nombre, apellido y segundo apellido).

Podemos ver un detalle de las tablas usadas en el sistema a través del árbol de tablas en Oracle, como lo muestra la figura 4.3.11



<input type="checkbox"/> CC_AUTORIZACIONES	<input type="checkbox"/> CC_NUMEROS_ASSIGNADOS
<input type="checkbox"/> CC_AUTORIZACIONES_VENCIDAS	<input type="checkbox"/> CC_PAQUETES
<input type="checkbox"/> CC_AUTORIZACIONES_X_CUENTA	<input type="checkbox"/> CC_PARAMETROS_GENERALES
<input type="checkbox"/> CC_AUTORIZACIONES_X_GRUPO	<input type="checkbox"/> CC_PROMEDIOS
<input type="checkbox"/> CC_CHEQUERAS	<input type="checkbox"/> CC_RANGOS_CONSULTA
<input type="checkbox"/> CC_CHEQUERAS_MOORE	<input type="checkbox"/> CC_RANGOS_ESTADISTICOS_TASAS
<input type="checkbox"/> CC_CHEQUES_AJENOS	<input type="checkbox"/> CC_RANGOS_REPORTE
<input type="checkbox"/> CC_CHEQUES_CERTIFICADOS	<input type="checkbox"/> CC_RANGOS_SALDO_CUENTA
<input type="checkbox"/> CC_CHEQUES_CONSULTADOS	<input type="checkbox"/> CC_RANGOS_SALDO_CUENTA_DETALLE
<input type="checkbox"/> CC_CHEQUES_ENTREGADOS	<input type="checkbox"/> CC_RESERVA_LIBERADA
<input type="checkbox"/> CC_CHEQUES_RECHAZADOS	<input type="checkbox"/> CC_RESERVAS_ESPECIALES
<input type="checkbox"/> CC_CHEQUES_SUSPENDIDOS	<input type="checkbox"/> CC_REVERSIONES_FLEXTRAN
<input type="checkbox"/> CC_COMPROBANTES_CONTABLES	<input type="checkbox"/> CC_SALDOS_DIARIOS
<input type="checkbox"/> CC_CONTROL_DE_PROCESOS	<input type="checkbox"/> CC_SALDOS_DISPONIBLES
<input type="checkbox"/> CC_DEPOSITOS_MONETARIOS	<input type="checkbox"/> CC_SALDOS_EMBARGADOS
<input type="checkbox"/> CC_DETALLE_DE_ESTADISTICAS	<input type="checkbox"/> CC_SALDOS_MENSUALES
<input type="checkbox"/> CC_DETALLE_DE_RESERVAS	<input type="checkbox"/> CC_SOBREGIROS
<input type="checkbox"/> CC_ESTADISTICAS	<input type="checkbox"/> CC_SOLICITUDES_CHEQUERAS
<input type="checkbox"/> CC_GARANTIAS	<input type="checkbox"/> CC_TIPOS_DE_RECHAZO_INTERNO
<input type="checkbox"/> CC_INTEGRACIONES	<input type="checkbox"/> CC_TRANSACCIONES_X_CUENTA
<input type="checkbox"/> CC_LISTADOS_AUTOMATICOS	<input type="checkbox"/> CC_TRIMESTRES
<input type="checkbox"/> CC_LISTADOS_DE_ACTUALIZACION	<input type="checkbox"/> CD_BENEFICIARIOS
<input type="checkbox"/> CC_MICROFICHAS_TEMPORALES	<input type="checkbox"/> CD_CAMPOS_X_CREDITO
<input type="checkbox"/> CC_MOVIMIENTOS_ACTUALIZADOS	<input type="checkbox"/> CD_CODIGOS_ENMIENDA
<input type="checkbox"/> CC_MOVIMIENTOS_COMPENSACION	<input type="checkbox"/> CD_COMPROBANTES_CONTABLES
<input type="checkbox"/> CC_MOVIMIENTOS_DIARIOS	<input type="checkbox"/> CD_CONCEPTO_X_CREDITO_X_MONEDA
<input type="checkbox"/> CC_MOVIMIENTOS_HISTORICOS	<input type="checkbox"/> CD_CONCEPTOS_X_CREDITO
<input type="checkbox"/> CC_MOVIMIENTOS_MENSUALES	<input type="checkbox"/> CD_CONCEPTOS_X_TRANSACCION
<input type="checkbox"/> CC_MOVIMIENTOS_TEMPORALES	<input type="checkbox"/> CD_CREDITOS_DOCUMENTALES
<input type="checkbox"/> CC_NCR7770	<input type="checkbox"/> CD_CREDITOS_X_CLIENTES
<input type="checkbox"/> CD_DOCUMENTOS_X_IDIOMA	<input type="checkbox"/> CD_DETALLES_PAGO
<input type="checkbox"/> CD_EMBARQUES	<input type="checkbox"/> CD_DOCUMENTOS_X_CREDITO
<input type="checkbox"/> CD_ENMIENDAS	<input type="checkbox"/> CD_USUARIOS_X_TIPO_CREDITO
<input type="checkbox"/> CD_ENVIOS_PARCIALES	<input type="checkbox"/> CE_PTE_AJUSTE
<input type="checkbox"/> CD_ESTADISTICA_PAGO_COBRO_CORR	<input type="checkbox"/> CE_PTE_CONSULTA
<input type="checkbox"/> CD_ESTADISTICAS	<input type="checkbox"/> CE_PTE_CONSULTA_DET
<input type="checkbox"/> CD_ESTADISTICAS_PAGO_COBRO	<input type="checkbox"/> CE_PTE_CUOTA
<input type="checkbox"/> CD_ESTADISTICAS_X_TRANSACCION	<input type="checkbox"/> CE_PTE_PARAMETRO
<input type="checkbox"/> CD_GARANTIAS	<input type="checkbox"/> CE_PTE_PROVISION
<input type="checkbox"/> CD_IDIOMAS	<input type="checkbox"/> CE_PTE_SALDO
<input type="checkbox"/> CD_INSTRUCCIONES_X_CREDITO	<input type="checkbox"/> CE_PTE_TRASPASO
<input type="checkbox"/> CD_ITEMS	<input type="checkbox"/> CG_CHEQUERAS
<input type="checkbox"/> CD_LETRAS_CAMBIO	<input type="checkbox"/> CG_COMPROBANTES_CONTABLE
<input type="checkbox"/> CD_LINEAS_MODELO	<input type="checkbox"/> CG_DETALLE_GASTOS
<input type="checkbox"/> CD_LINEAS_X_IDIOMA	<input type="checkbox"/> CG_ESTADISTICAS
<input type="checkbox"/> CD_LINEAS_X_IDIOMA_X_MODELO	<input type="checkbox"/> CG_ESTADISTICAS_COMISION
<input type="checkbox"/> CD_MODELOS	<input type="checkbox"/> CG_ESTADISTICAS_DETALLE
<input type="checkbox"/> CD_MOVIMIENTOS	<input type="checkbox"/> CG_ESTADISTICOS_FORMA_CARGO
<input type="checkbox"/> CD_PARAMETROS_GENERALES	<input type="checkbox"/> CG_FORM_HELP
<input type="checkbox"/> CD_PARAMETROS_X_SUBAPLICACION	<input type="checkbox"/> CG_FORMAS_CARGO
<input type="checkbox"/> CD_POSICIONES_ARANSEL	<input type="checkbox"/> CG_GASTOS_X_SUBAPLICACION
<input type="checkbox"/> CD_TERMINOS	<input type="checkbox"/> CG_HISTORICO_DETALLES
<input type="checkbox"/> CD_TERMINOS_X_CARTA	<input type="checkbox"/> CG_HISTORICO_MOVIMIENTOS
<input type="checkbox"/> CD_TIPOS_CAMPO	<input type="checkbox"/> CG_MAESTROS
<input type="checkbox"/> CD_TIPOS_CAMPO_X_CREDITO	<input type="checkbox"/> CG_MOVIMIENTOS
<input type="checkbox"/> CD_TIPOS_CAMPO_X_LINEA	<input type="checkbox"/> CG_PARAMETROS
<input type="checkbox"/> CD_TIPOS_CONCEPTO	<input type="checkbox"/> CG_PARAMETROS_X_SUBAPLICACION
<input type="checkbox"/> CD_TIPOS_CONCEPTO_X_CREDITO	<input type="checkbox"/> CG_RANGOS_COMISION
<input type="checkbox"/> CD_TIPOS_CREDITO_DOCUMENTAL	<input type="checkbox"/> CG_REF_CODES
<input type="checkbox"/> CD_TIPOS_DOCUMENTO	<input type="checkbox"/> CG_TIPOS_BLOQUEO
<input type="checkbox"/> CD_TIPOS_DOCUMENTO_X_CREDITO	<input type="checkbox"/> CG_TIPOS_GASTO
<input type="checkbox"/> CD_USUARIOS_X_TIPO_CREDITO	<input type="checkbox"/> CJ_ABOGADOS
	<input type="checkbox"/> CJ_ABOGADOS_X_JUICIO
	<input type="checkbox"/> CJ_ANTECEDENTES

+	CJ_COBRO_ADMINISTRATIVO
+	CJ_COBRO_HONORARIO_ABOGADOS
+	CJ_COBRO_NOTARIO
+	CJ_COBROS_JUDICIALES
+	CJ_COBROS_X_LINEA
+	CJ_COMPROBANTES_CONTABLES
+	CJ_DACION_ADJUDICACION
+	CJ_DACION_ADJUDICACION_BIENES
+	CJ_DETALLE_CONTABLES
+	CJ_DETALLE_ESTADISTICAS
+	CJ_DETALLE_HONORARIO_ABOGADOS
+	CJ_EMBARGOS
+	CJ_EMBARGOS_BIENES
+	CJ_ESTADISTICAS
+	CJ_ETAPAS
+	CJ_ETAPAS_JUICIO
+	CJ_ETAPAS_JUICIO_COB_JUD
+	CJ_EXCEPCIONES
+	CJ_GRAVAMENES
+	CJ_GRAVAMENES_ACREEDORES
+	CJ_HONORARIO_COBRO_X_ETAPA
+	CJ_IVA_X_AGENCIA
+	CJ_JUICIOS_X_COBRO
+	CJ_JUICIOS_X_JUZGADOS
+	CJ_LOCALIZADOR
+	CJ_LOCALIZADORES_X_COBRO
+	CJ_MONTO_X_MONEDA
+	CJ_PAGOS
+	CJ_PARAMETROS_GENERALES
+	CJ_PRUEBAS
+	CJ_PRUEBAS_DESAHOGO
+	CJ_SENTENCIAS

+	CJ_SENTENCIAS_DATOS
+	CJ_SUBJUICIOS_X_JUICIO
+	CJ_TIPO_JUICIOS
+	CJ_TRANSAC_MOVIMIENTOS
+	CJ_TRANSAC_MOVIMIENTOS_B
+	CO_BITACORAS
+	CO_ETAPAS
+	CO_FIRMAS
+	CO_FORMATOS
+	CO_GASTOS_BITACORA
+	CO_GESTION_JUDICIAL
+	CO_GESTION_JUDICIAL_ERR
+	CO_GESTIONES
+	CO_GESTIONES_BITACORA
+	CO_PARRAFOS_FORMATO
+	CO_RECORDATORIOS
+	COPIA_OLD
+	CR_AJUSTES
+	CR_ATIENDE
+	CR_CAT_REQUERIMIENTOS
+	CR_CAUSAS
+	CR_DETALLE_REQUERIMIENTOS
+	CR_REQUERIMIENTOS
+	CR_SISTEMA
+	CUENTAS
+	CV_GARANTIAS_X_LINEA
+	CV_LINEAS_X_CLIENTE
+	CV_PRESTAMOS_X_CONTRATO
+	CV_SALDOS_X_CUOTA_PRESTAMOS
+	DD_CARGA_EXPORTA_FACIL
+	DD_CARGA_EXPORTA_FACIL_DETALLE
+	DD_CATALOGO COMPRADOR EXP F

+	DD_CATALOGO_COMPRADOR_EXP_F
+	DD_CATALOGO_EMPRESAS_COBRANZA
+	DD_CATALOGO_EXPORTADOR_EXP_F
+	DD_CATALOGO_PERSONA_EXP_F
+	DD_CATALOGO_SECTOR_EXP_F
+	DD_CATALOGO_TAMEMPRESA_EXP_F
+	DD_COMISIONES_FACTURA
+	DD_COMPROBANTES_CONTABLES
+	DD_CONCILIACION_EXPORTA_FACIL
+	DD_CONTRAPARTES
+	DD_DATOS_COMPRA_CONTRA
+	DD_DATOS_CONTRAPARTE
+	DD_DETALLE_COMPROBANTES
+	DD_DOCTOS_EXP_F
+	DD_DOCTOS_EXPORTADOR_EXP_F
+	DD_DOCTOS_FACTURAS_EXP_F
+	DD_DOCUMENTOS
+	DD_ENVIO_EMPRESAS_COBRANZA
+	DD_ESTADISTICOS
+	DD_ESTADOS
+	DD_ESTATUS_FACTURAS_EXP_F
+	DD_EXPORTADOR_CONTRAPARTE
+	DD_EXPORTAFACIL
+	DD_IMPORTADOR_CONTRAPARTE
+	DD_IMPORTADORES
+	DD_MOVIMIENTO_DIARIOS
+	DD_MOVIMIENTOS_DESGLOSE
+	DD_MOVIMIENTOS_FACTURA
+	DD_NEG_HISTORICA
+	DD_NEG_PARCIAL
+	DD_NEGOCIACIONES
+	DD_NEGOCIACIONES_NO_FINA

+	DD_PAGOS_DOCUMENTOS
+	DD_PARAM_CLIENTES_CADENAS
+	DD_PARAM_COMISIONES
+	DD_PARAMETROS_DE_COBRO
+	DD_PARAMETROS_GENERALES
+	DD_PASO_IMPORTADORES
+	DD_PLAN_PAGOS
+	DD_PROVEEDOR_PASO
+	DD_RELACION_COMERCIAL_EXP_F
+	DD_SEGUIMIENTO_COBRANZA
+	DD_TIPO_COMISION_CONTRA
+	DD_TIPO_DOCUMENTOS
+	DD_TMP_NAFIN_BNX
+	DD_TRANSFER_EXP
+	DD_USUARIO_IMPORTADOR
+	DIS_DOCS_
+	DIS_GRANTS_
+	DP_AUTORIZACIONES_X_CUENTA
+	DP_AUTORIZACIONES_X_GRUPO
+	DP_CASOS_PENDIENTES
+	DP_COMPROBANTES_CONTABLES
+	DP_CONTROL_DE_PROCESOS
+	DP_DEPOSITOS_A_PLAZOS
+	DP_DETALLE_DE_ESTADISTICAS
+	DP_DETALLE_DE_RESERVAS
+	DP_ESTADISTICAS
+	DP_FORMA_DE_PAGO
+	DP_INTEGRACIONES
+	DP_LISTADOS_AUTOMATICOS
+	DP_LISTADOS_DE_ACTUALIZACION
+	DP_MOVIMIENTOS_ACTUALIZADOS
+	DP MOVIMIENTOS DIARIOS

+	DP_MOVIMIENTOS_HISTORICOS
+	DP_MOVIMIENTOS_MENSUALES
+	DP_PARAMETROS_GENERALES
+	DP_RANGOS_CONSULTA
+	DP_RANGOS_ESTADISTICOS_TASAS
+	DP_RANGOS_REPORTE
+	DP_RANGOS_SALDO_CUENTA
+	DP_RANGOS_SALDO_CUENTA_DETALLE
+	DP_SALDOS_DIARIOS
+	DP_SALDOS_EMBARGADOS
+	DP_SALDOS_MENSUALES
+	DP_TASA_INTERES_DETALLE
+	DP_TASA_INTERES_ENCABEZADO
+	DP_TIPO_ASIGNACION
+	DP_TIPOS_DE_RECHAZO_INTERNO
+	DP_TRANSACCIONES_X_CUENTA
+	DP_TRIMESTRES
+	DW_CIFRAS_LINEAS
+	DW_CIFRAS_PRESTAMOS_DIARIO
+	DW_CIFRAS_PRESTAMOS_MENSUAL
+	DW_COMPARATIVO_SALDOS
+	DW_COMPARATIVO_SALDOS_DIARIO
+	DW_OTORGAMIENTOS
+	DW_PREPAGOS
+	DW_PROYECCION_SALDOS_CAPITAL
+	DW_PROYECCION_VENCIMIENTOS
+	DW_RECUPERACIONES
+	DW_SALDOS_CAPITAL_DIARIO
+	DW_SALDOS_CAPITAL_MENSUAL
+	DW_SALDOS_CAPITAL_TOTAL
+	DW_SALDOS_CAPITAL_TOTAL_D
+	DW SOLICITUDES

+	DW_SOLICITUDES
+	DW_VENCIMIENTO_X_PRESTAMO
+	DW_VENCIMIENTOS_MENSUAL
+	EGU_VALORES_TASAS
+	EGU_VALORES_TASAS_BANKS
+	ERR_MN
+	ERRORES
+	EUL_ACCESS_PRIVS
+	EUL_APP_PARAMS
+	EUL_BA_OBJ_LINKS
+	EUL_BATCH_PARAMS
+	EUL_BATCH_QUERIES
+	EUL_BATCH_REPORTS
+	EUL_BATCH_SHEETS
+	EUL_BQ_DEPS
+	EUL_BQ_TABLES
+	EUL_BR_RUNS
+	EUL_BUSINESS_AREAS
+	EUL_DBH_NODES
+	EUL_DOMAINS
+	EUL_EUL_USERS
+	EUL_EXP_DEPS
+	EUL_EXPRESSIONS
+	EUL_FREQ_UNITS
+	EUL_FUN_ARGUMENTS
+	EUL_FUN_CATEGORIES
+	EUL_FUN_FC_LINKS
+	EUL_FUNCTIONS
+	EUL_GATEWAYS
+	EUL_HI_NODES
+	EUL_HI_SEGMENTS
+	EUL_HIERARCHIES

+	EUL_JG_EXP_LINKS	+	GA_DECRETOS
+	EUL_JC_JOIN_LINKS	+	GA_DEPRECIACIONES
+	EUL_JOIN_COMBS	+	GA_DETALLE_BIEN_GTIA
+	EUL_KEY_CONS	+	GA_DETALLE_CONTABLES
+	EUL_NMSDO_JC_USGS	+	GA_DETALLE_HIPOTECARIA
+	EUL_OBJ_DEPS	+	GA_DETALLE_INSTRUMENTO
+	EUL_OBJ_JOIN_USGS	+	GA_ESTADISTICO_X_TRANSACCION
+	EUL_OBJJS	+	GA_ESTADISTICOS
+	EUL_PLAN_TABLE	+	GA_ESTADOS_FISICOS
+	EUL_QPP_STATISTICS	+	GA_GARANTIAS
+	EUL_SBO_DSGN_ELEMS	+	GA_GARANTIAS_BIEN
+	EUL_SEGMENTS	+	GA_GARANTIAS_FIDU
+	EUL_SEQUENCES	+	GA_GARANTIAS_HIPO
+	EUL_SQ_CRRLTNS	+	GA_GARANTIAS_PARTICIPACION
+	EUL_SUB_QUERIES	+	GA_GARANTIAS_TITU
+	EUL_SUM_BITMAPS	+	GA_GARANTIAS_X_CLIENTES
+	EUL_SUM_RFRSH_SETS	+	GA_GRADOS
+	EUL_SUMMARY_OBJJS	+	GA_GRADOS_PROMETIDOS
+	EUL_SUMO_ITEM_USGS	+	GA_HISTORICO_GRADOS
+	EUL_VERSIONS	+	GA_HISTORICO_REVAL
+	EXCEPTIONORA	+	GA_IMAGEN
+	EXCEPTIONS	+	GA_INSPECCIONES
+	EXCEPTIONSSUO	+	GA_INSPECTORES
+	GA_ACTUALIZACIONES	+	GA_INSTRUMENTOS_X_DOCUMENTO
+	GA_AUTORIZACIONES	+	GA_MANDATOS
+	GA_AVALUOS	+	GA_METODOS_DE_ACTUALIZACION
+	GA_COMPROBANTES_CONTABLES	+	GA_MODELOS_BIEN
+	GA_CONTROL_RUTAS	+	GA_MOVIMIENTOS
+	GA_CONTROL_RUTAS2	+	GA_MOVIMIENTOS_DESGLOSE
+	GA_CORREDOR_ADUANAS	+	GA_OBSOLENCIA
+	GA_CUENTAS_GARANTIAS	+	GA_OFICINAS_REGISTROS
+	GA_CUENTAS_PROMETIDAS	+	GA_OPERACIONES_REVALORIZACION
+	GA_PARAMETROS_CONTABLES	+	GL_SALDOS_PLAN_PAGO
+	GA_PERITOS	+	GL_STATUS
+	GA_RESULTADOS_REVALORIZACION	+	GL_UNIVERSAL_CONTRATOS
+	GA_REVALORIZACION	+	GM_AGENCIAS_REGIONALES_CONT
+	GA_SEGUROS	+	GM_ARCHIVOS_VALIDADOR
+	GA_TIPO_BIENES	+	GM_AREAS_CONTABLES
+	GA_TIPO_FORMULAS	+	GM_AREAS_CONTABLES_X_SUCURSAL
+	GA_TIPOS_DE_TASAS_DE_INTERES	+	GM_BALANCE_ACUMULADO
+	GA_TIPOS_DOCUMENTO	+	GM_BALANCE_CUENTAS
+	GA_TIPOS_SEGUROS	+	GM_BALANCE_DETALLE
+	GA_ZONA_INSPECCION	+	GM_BALANCE_EXTR_ACUMULADO
+	GC_AMORTIZACIONES	+	GM_BALANCE_EXTRANJERO
+	GC_CONTRATOS	+	GM_BALANCE_EXTRANJERO_DETALLE
+	GC_DOCS_GARANTIAS_SIGAR	+	GM_CAMBIOS_TIPO_CUENTA
+	GC_DOCS_SOPORTE	+	GM_CATALOGO_RELAC_EMPRESA
+	GC_FACTOR_IVA	+	GM_CATEGORIAS_CONTABLES
+	GC_FRACCIONES_PROHIBIDAS	+	GM_CATEGORIAS_CUENTAS
+	GC_GARANTIAS	+	GM_CIERRES_EMPRESAS
+	GC_GRUPO_EMPRESARIAL	+	GM_CODIGO_FUENTES_USOS
+	GC_GRUPOS	+	GM_COMPROBANTES
+	GC_GRUPOS_SECTOR	+	GM_COMPROBANTES_DETALLE
+	GC_PAISES_PROHIBIDOS	+	GM_COMPROBANTES_DETALLE_HIST
+	GC_PARAMETROS_CALIFICACION	+	GM_COMPROBANTES_HISTORICO
+	GC_SECTORES	+	GM_CONTROL_TRANSACCIONES
+	GC_TIPOS_GARANTIAS	+	GM_CUENTAS_DIARIO
+	GL_BITACORA_PROCESO	+	GM_CUENTAS_DISTRIBUCION
+	GL_CAT_COC	+	GM_CUENTAS_INTERSUCURSALES
+	GL_CAT_CTG	+	GM_CUENTAS_MANEJO_DIVISAS
+	GL_CONTRATOS	+	GM_CUENTAS_MONEDAS
+	GL_FRACCION_ARANCELARIA	+	GM_CUENTAS_RECIBE_DISTRIBUCION
+	GL_GARANTIA	+	GM_CUENTAS_RECIBE_PERIOD_DIST
+	GL_PLAN_PAGO	+	GM_CUENTAS_REFERENCIA_INT

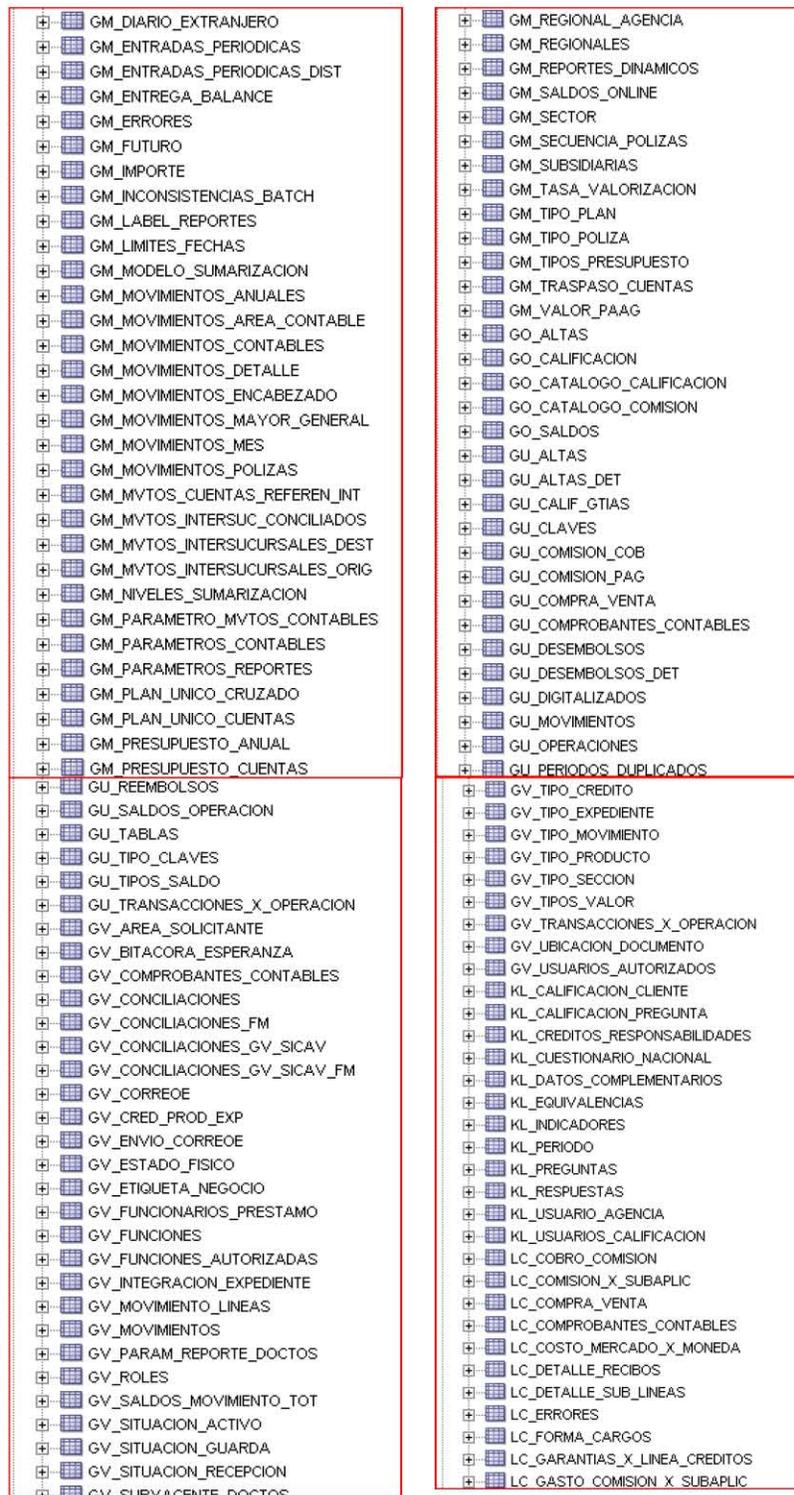


Figura 4.3.11 Estructura de árbol Tablas del sistema.

Siguiendo con los elementos indispensables en el manejo de una base de datos y su información son los procedimientos almacenados, los cuales son un bloque de

instrucciones PL/SQL que se almacena en el diccionario de datos y al que pueden llamar las aplicaciones. Un procedimiento almacenado es un programa dentro de la base de datos que ejecuta una acción o conjunto de acciones específicas. Los procedimientos permiten almacenar dentro de la base de datos la lógica de las aplicaciones que se emplea con más frecuencia. Cuando se ejecuta un procedimiento, sus instrucciones se ejecutan como una unidad.

Para genera un procedimiento dentro del manejador hay que dar botón derecho y seleccionar Nuevo procedimiento almacenado como lo muestra la figura 4.3.12

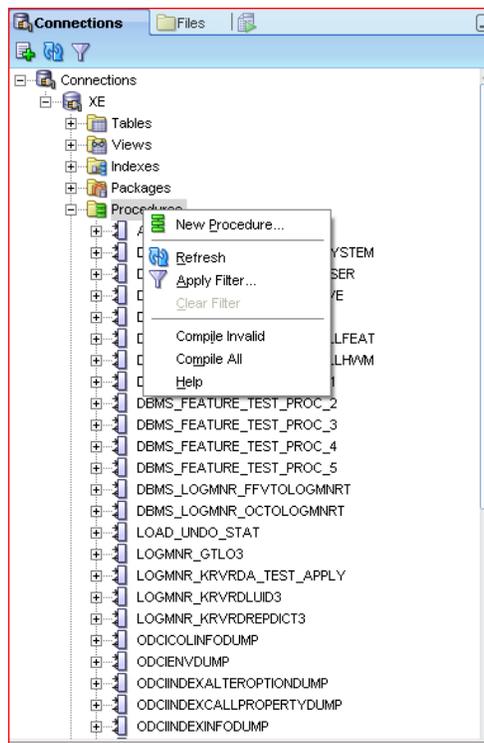


Figura 4.3.12 Creación de un Procedimiento Almacenado

En la parte de la derecha nos habilitara un espacio donde generaremos las instrucciones de SQL que nos permitirán indicar las instrucciones de lo que debe hacer dicho procedimiento almacenado como podemos ver en la figura 4.3.13

```

create or replace
PROCEDURE aw_drop_proc
(obj_type IN VARCHAR2, obj_name IN VARCHAR2, obj_owner IN VARCHAR2)
AS
db_name VARCHAR2(50);
db_ammn NUMBER;
db_exist NUMBER;
del_stat1 VARCHAR2(100) := 'DELETE FROM ps$ WHERE awseq#=1';
del_stat2 VARCHAR2(100) := 'DELETE FROM aw$ WHERE awseq#=1';
del_stat3 VARCHAR2(100) := 'DELETE FROM aw_prop$ WHERE awseq#=1';
del_stat4 VARCHAR2(100) := 'DELETE FROM aw_obj$ WHERE awseq#=1';
del_stat5 VARCHAR2(100) := 'DELETE FROM noexp$ WHERE owner=:1 AND name=:2';
del_stat6 VARCHAR2(200) := 'DELETE FROM expdepact$ WHERE obj#=(SELECT object_id FROM all_objects WHERE owner=:1 A
BEGIN
-- Check if we are deleting a table prefixed with 'AW$'.
IF obj_type = 'TABLE' AND obj_name like 'AW$%' THEN
-- Count on AW$ and PS$ tables both being there.
db_name := SUBSTR(obj_name, 4, LENGTH(obj_name));
SELECT a.awseq# INTO db_ammn FROM aw$ a, user$ u
WHERE a.awname = db_name AND
a.owner# = u.user# and u.name = obj_owner;
EXECUTE IMMEDIATE del_stat1 USING db_ammn;
EXECUTE IMMEDIATE del_stat2 USING db_ammn;
EXECUTE IMMEDIATE del_stat3 USING db_ammn;
EXECUTE IMMEDIATE del_stat4 USING db_ammn;
EXECUTE IMMEDIATE del_stat5 USING obj_owner, obj_name;
EXECUTE IMMEDIATE del_stat6 USING obj_owner, obj_name, obj_owner;
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
NULL; -- aw$ table may already be empty.
WHEN OTHERS THEN
NULL;

```

Figura 4.3.13 Instrucciones SQL del Procedimiento almacenado generado

Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código como se muestra en la figura 4.3.14

```

create or replace PROCEDURE carga_datos
IS
BEGIN
declare
cursor cl is
select * from mg_clientes_sf;
cliente%rowtype;
begin
null;
END;

```

Figura 4.3.14 Ejemplo sencillo de un procedimiento almacenado

A continuación se muestra el árbol de los procedimientos almacenados utilizados en el sistema, como se puede ver en la figura 4.3.15



Figura 4.3.15 Árbol de Procedimientos almacenados utilizados en el sistema

Otro elemento importante dentro de la generación del sistema son los disparadores (triggers), estos son procedimientos que se ejecutan cuando se produce un suceso de base de datos determinado en una tabla específica. Pueden utilizarse para aumentar la integridad referencial de los datos, imponer requisitos de seguridad adicionales o mejorar la auditoría. Existen dos tipos de disparadores:

Disparadores de instrucción: se activan una vez por cada instrucción de disparo

Disparadores de fila: se activan una vez por cada fila de una tabla afectada por las instrucciones.

Por ejemplo, un disparador de instrucción se activa una sola vez para un comando *delete* que elimina 1.000 filas. En cambio un disparador de fila se activará 1.000 veces para esa misma instrucción.

Para cada tipo de disparador puede crearse un disparador BEFORE (antes) y otro AFTER (después) para cada tipo de suceso de disparo. Entre los sucesos de disparo se encuentran las operaciones de *insert* (insertar), *update* (actualizar) y *delete* (eliminar).

Para crear un trigger en Oracle SQL Developer se deberá seleccionar la parte reservada para estos tal, posteriormente dar botón derecho y seleccionar la opción de nuevo trigger tal como lo muestra la figura 4.3.16

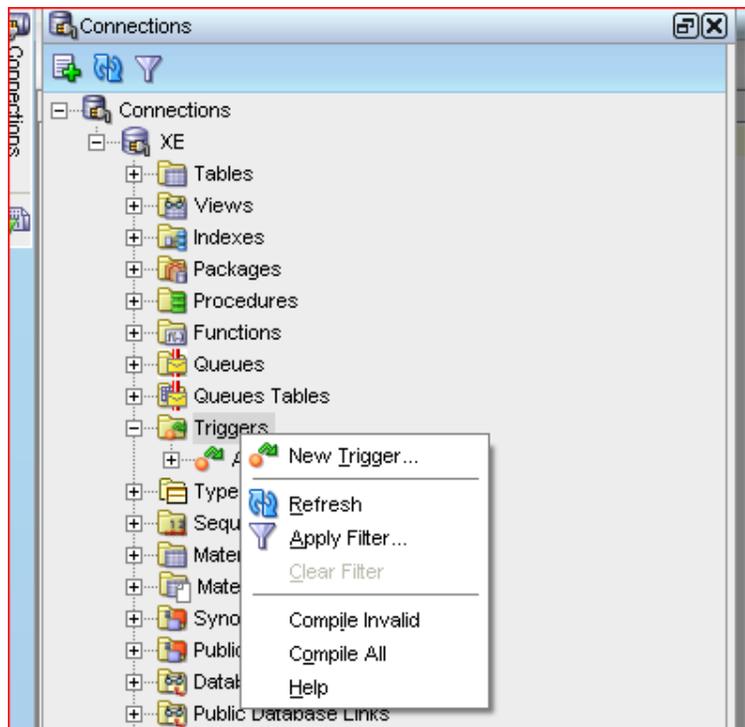


Figura 4.3.16 Creación de triggers

Posteriormente se activara la región de trabajo en donde se deberán ingresar las instrucciones SQL que determinaran la acción a tomar del trigger como se muestra en la figura 4.3.17

```

create or replace TRIGGER TRG_AU_08
BEFORE UPDATE ON Pr_Saldos_Fin_Mes_Detalle
REFERENCING OLD AS OLD NEW AS NEW
FOR EACH ROW
BEGIN
INSERT INTO PR_AU_08
VALUES
(:old.Anio_Mes,
:old.Numero_Prestamo,
:old.Codigo_Empresa,
:old.Codigo_Agencia,
:old.Codigo_Sub_Aplicacion,
:old.Codigo_Tipo_Saldo,
:old.Valor,
USER,
SYSDATE,
'A');

INSERT INTO PR_AU_08
VALUES
(:new.Anio_Mes,
:new.Numero_Prestamo,
:new.Codigo_Empresa,
:new.Codigo_Agencia,
:new.Codigo_Sub_Aplicacion,
:new.Codigo_Tipo_Saldo,
:new.Valor,
USER,
SYSDATE,
'D');
END TRG_AU_08;
    
```

Figura 4.3.17 Generación de las instrucciones del trigger

En la figura 4.3.18 podemos observar el árbol de triggers utilizados en la aplicación.



Figura 4.3.18 Árbol de triggers en la aplicación

Otro elemento importante en la generación de la estructura de la aplicación son las vistas. Una vista de base de datos es un resultado de una consulta SQL de una o varias tablas; también se le puede considerar una tabla virtual.

Las vistas tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos. Los datos que se recuperan mediante una consulta a una vista se presentarán igual que los de una tabla. De hecho, si no se sabe que se está trabajando con una vista, nada hace suponer que es así. Al igual que sucede con una tabla, se pueden insertar, actualizar, borrar y seleccionar datos en una vista. Aunque siempre es posible seleccionar datos de una vista, en algunas condiciones existen restricciones para realizar el resto de las operaciones sobre vistas, la creación de una vista la podemos observar en la figura 4.3.19

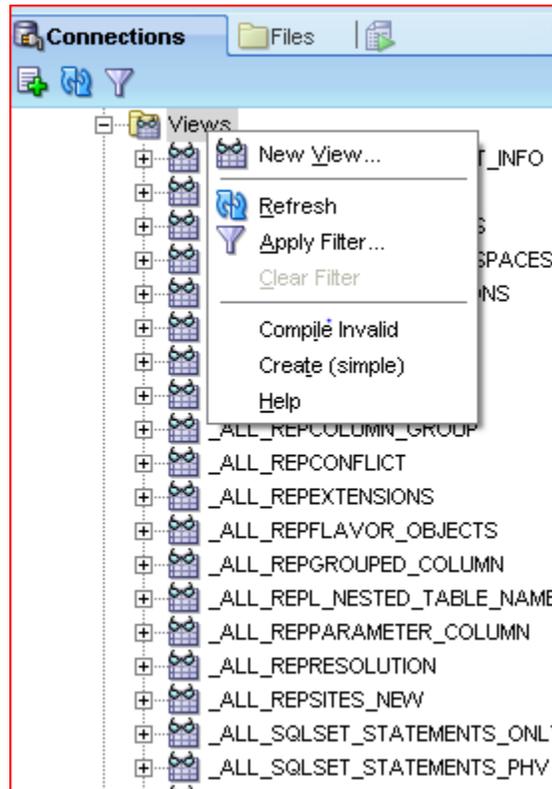


Figura 4.3.19 Creación de una Vista

Una vista se especifica a través de una expresión de consulta (una sentencia SELECT) que la calcula y que puede realizarse sobre una o más tablas. Sobre un conjunto de tablas relacionales se puede trabajar con un número cualquiera de vistas como podemos observar en la figura 4.3.20

Se seleccionan los campos y tablas de las cuales queremos obtener la información combinada y se agregan las instrucciones que definirán la información que contendrá dicha vista.

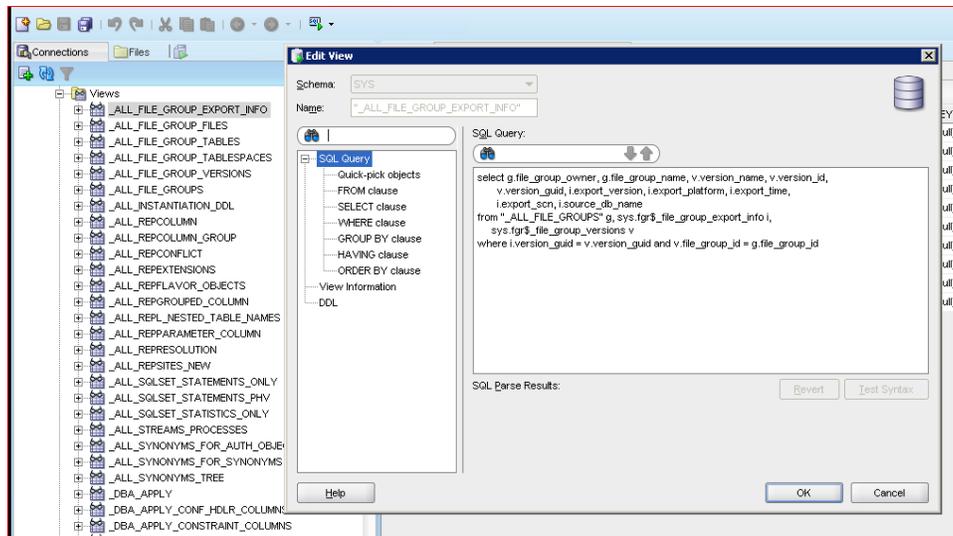


Figura 4.3.20 Estructura de una vista

4.4. DISEÑO Y CONSTRUCCIÓN DEL SITIO WEB.

Instalación de WebLogic Server 10.

Ejecutamos el instalador y nos aparecerán las pantallas de las figuras 4.4.1 y 4.4.2, damos click en **Next** (Siguiete) en cada una de ellas.



Figura 4.4.1 Pantalla de carga del programa de instalación

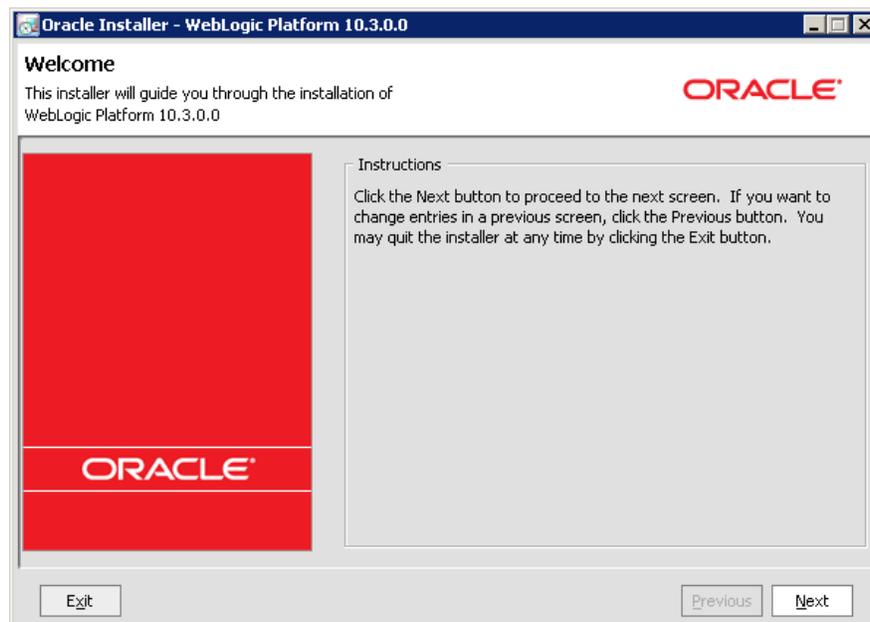


Figura 4.4.2 Pantalla de inicio de instalación

En el instalador podemos ver todos los “BEA Home Directory” que están previamente instalados, para este caso se creará uno nuevo llamado “**bea**”, vea la figura 4.4.3.

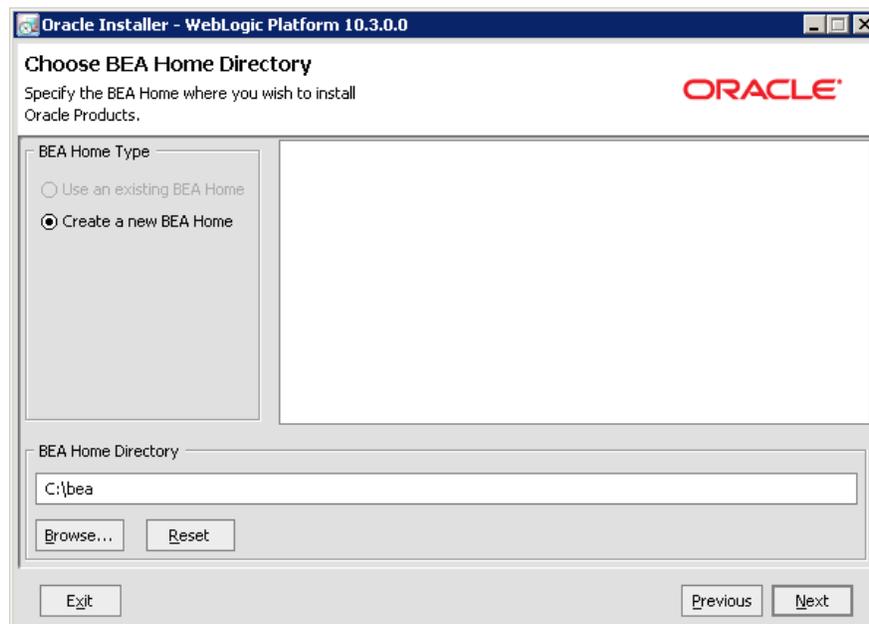


Figura 4.4.3 Definición del directorio de instalación

Luego nos pedirá elegir el tipo de instalación “Completa” o “Personalizada”, figura 4.4.4

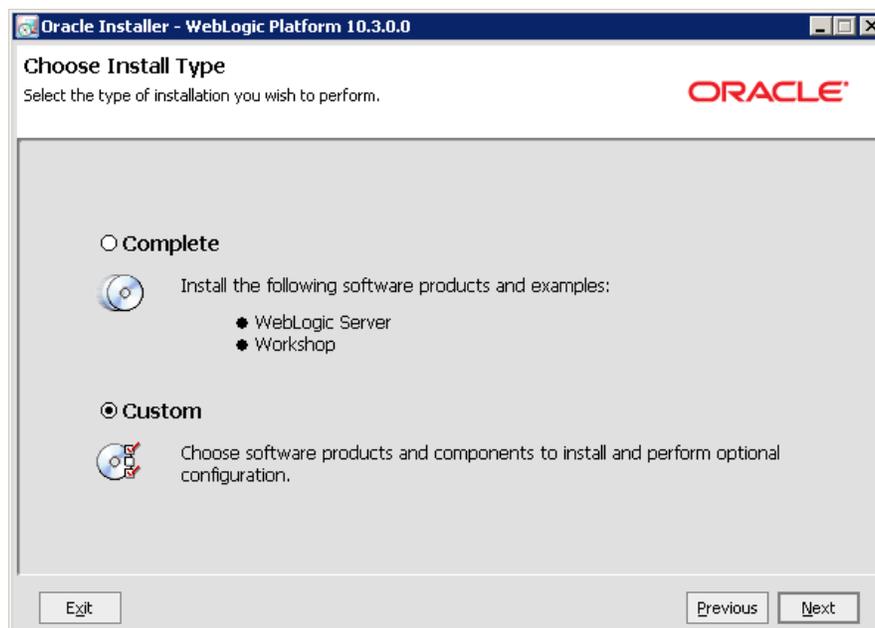


Figura 4.4.4 Selección del tipo de instalación

Elegimos el tipo de instalación “**Custom**” y damos click en Next

Instalamos solamente en WebLogic Server los siguientes componentes, vea la figura 4.4.5.

Core Application Server

Administration Console

Configuration Wizard and Update Framework

WebLogic JDBC Drivers

Thrid Party JDBC Drivers

WebLogic Server Client

WebLogic Web Server Plugins

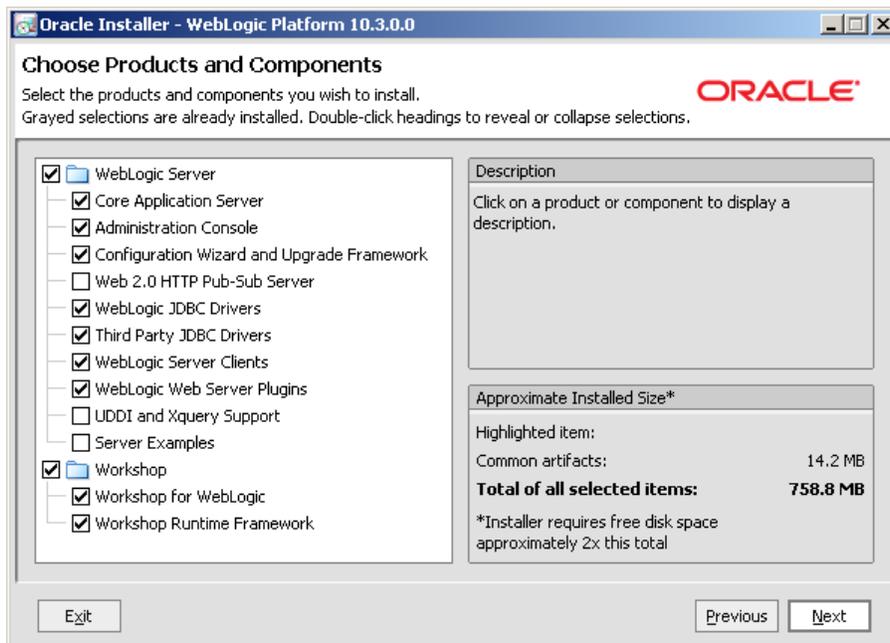


Figura 4.4.5 Componentes para instalar

En la siguiente pantalla seleccionamos que JDKs deseamos instalar, Sun SDK es mejor para el Desarrollo y BEA JRockit SDK es mejor para producción, podemos instalar solo el que ocupemos o ambos, para efectos de este caso se instalarán ambos, como se observa en la pantalla de la figura 4.4.6.

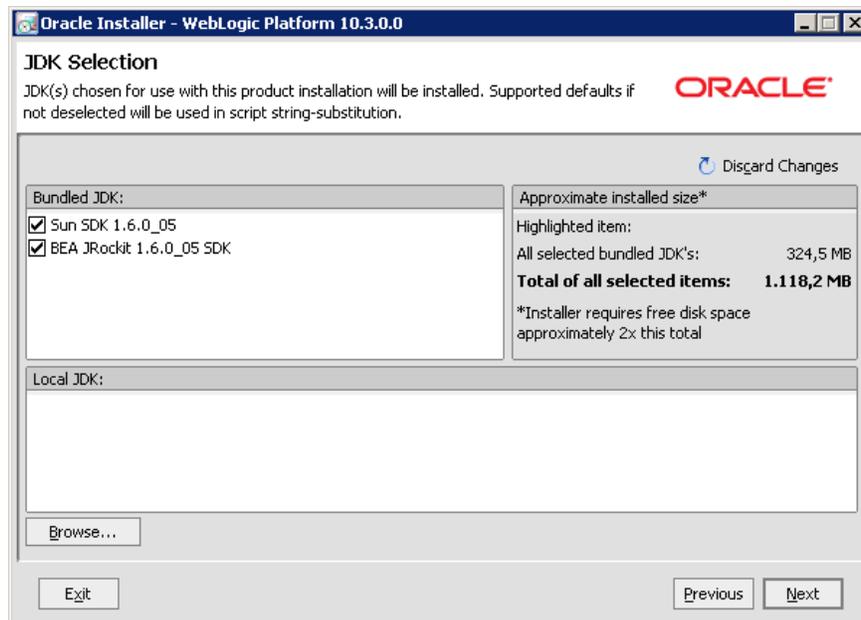


Figura 4.4.6 Opciones de JDKs a instalar

La siguiente sección se confirmará los directorios donde se instalarán los productos que seleccionamos anteriormente, damos click en Next. Vea las figuras 4.4.7 y 4.4.8.

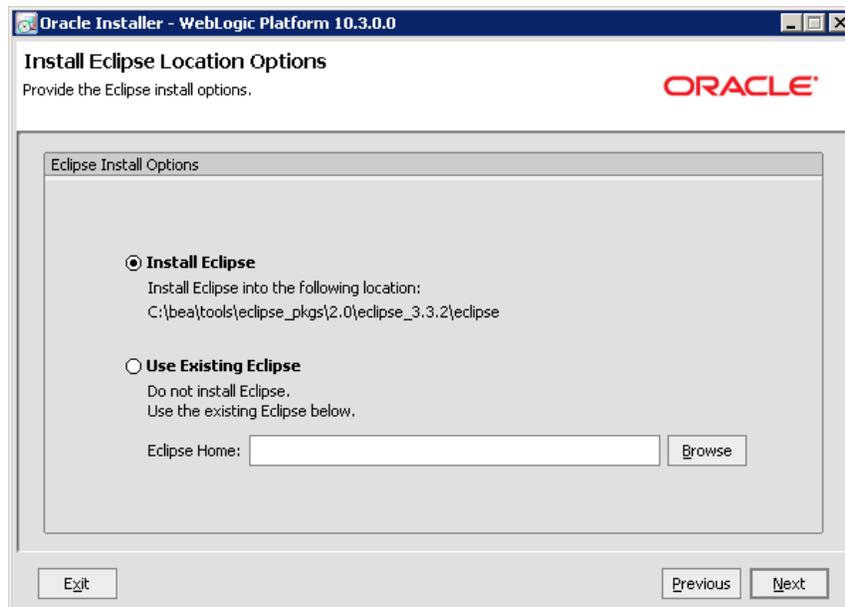


Figura 4.4.7 Instalación de Eclipse



Figura 4.4.8 Directorios de instalación

De aquí en adelante solo es dar click en el botón de Next, ya que las configuraciones que trae por defecto son las más recomendadas.

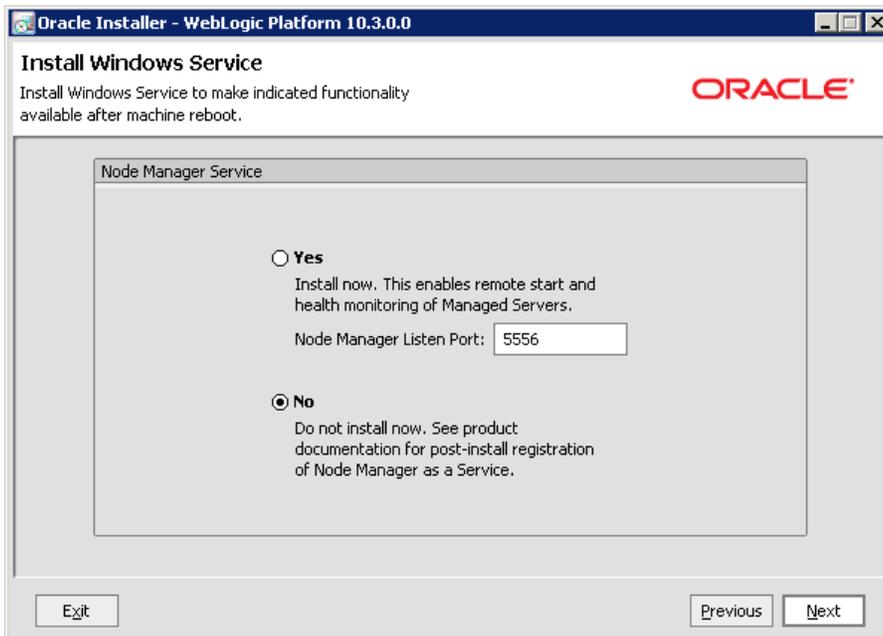


Figura 4.4.9 Servicio de administración para Windows

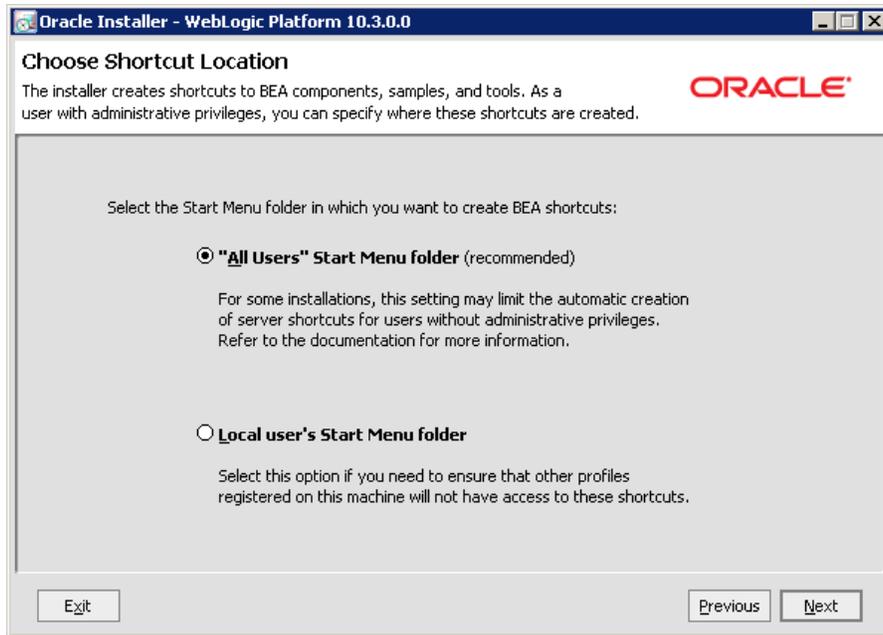


Figura 4.4.10 Configuración para el menú de inicio

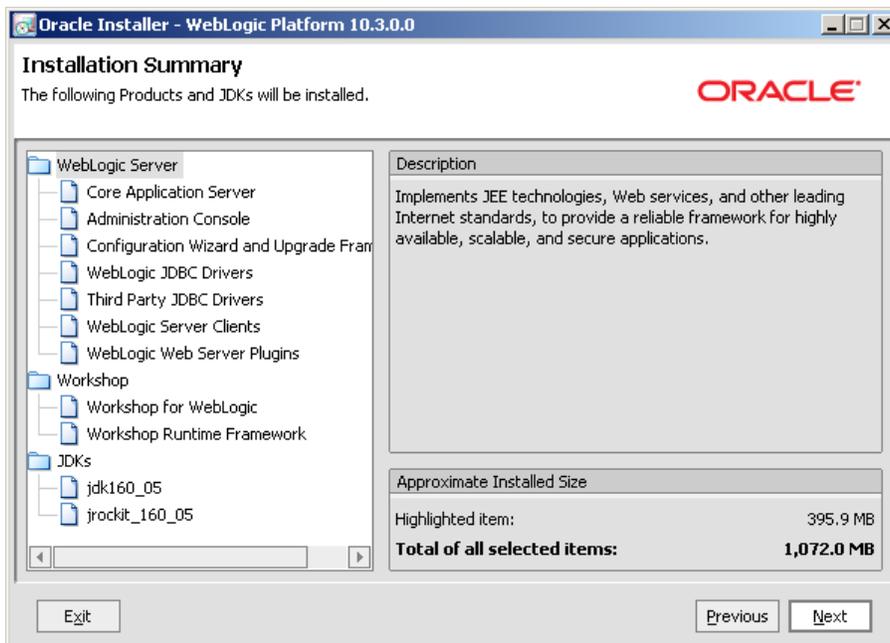


Figura 4.4.11 Resumen final de la instalación

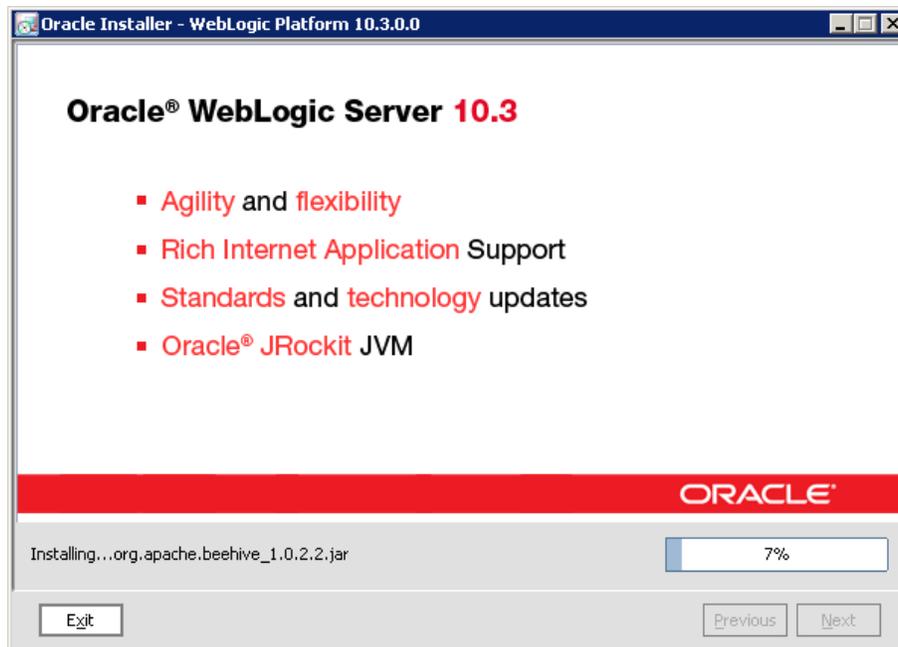


Figura 4.4.12 Proceso de instalación en curso

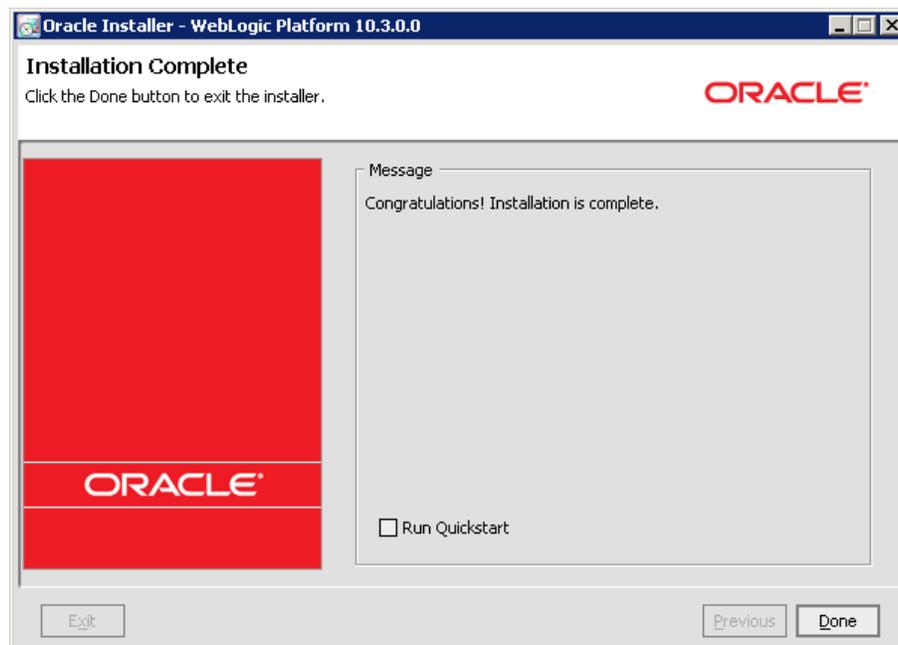


Figura 4.4.13 Pantalla final del proceso de instalación

Creación de un nuevo dominio WebLogic.

Un conjunto de recursos de WebLogic Server inter-relacionados manejados como una unidad se llama un dominio. Un dominio incluye uno o más WebLogic Servers, y podría incluir clusters WebLogic Server. En una configuración con varios WebLogic Servers, sólo uno es el Administration Server (servidor de administración); a los otros servidores se les llama Controlled Servers (servidores controlados).

Cuando un clúster está contenido en un dominio, todos sus servidores deben ser parte de ese dominio también.

Un dominio es una unidad administrativa autocontenida. Si una aplicación se despliega en un dominio, los componentes de esa aplicación no pueden desplegarse en servidores que no sean parte de ese dominio.

A continuación se describe el proceso de creación de un dominio WebLogic.

Vamos al menú de Inicio de Windows y llegamos hasta “**Configuration Wizard**” y lo ejecutamos, vea la figura 4.4.14

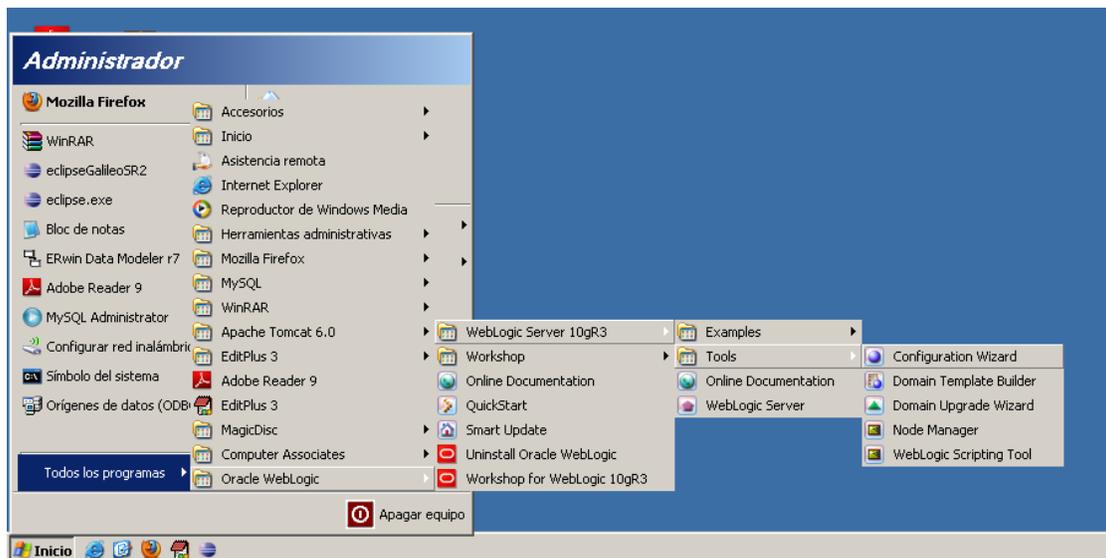


Figura 4.4.14 Menú de Windows para la configuración

Aparece la pantalla de bienvenida como se muestra en la de la figura 4.4.15



Figura 4.4.15 Opciones de creación del dominio

Seleccionamos “**Create a new WebLogic domain**” y damos click en Next

En la siguiente sección, vea la figura 4.4.15, seleccionamos el tipo de origen del dominio a crear y damos click en Next.

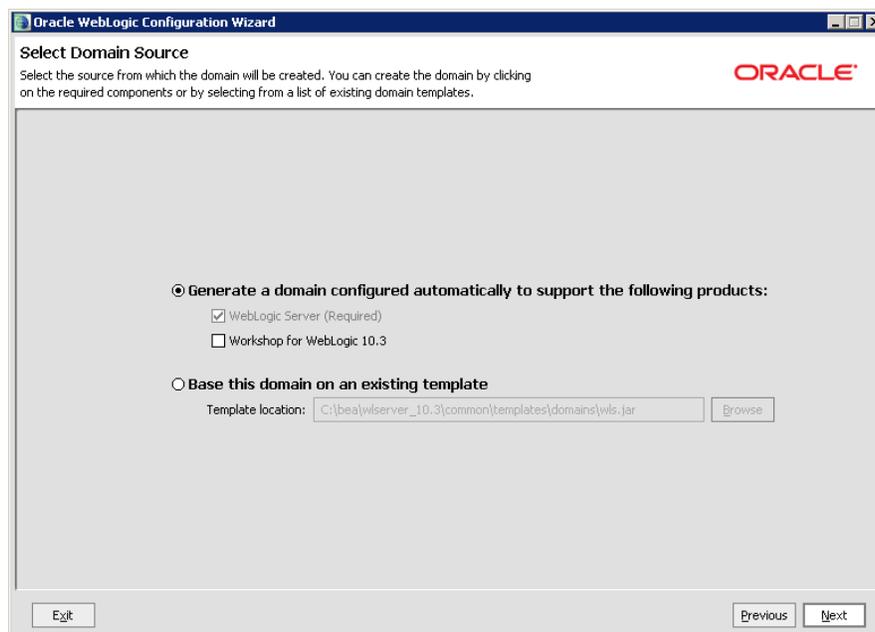


Figura 4.4.16 Selección del tipo de origen de dominio

Luego creamos al usuario administrador y le asignamos una contraseña, ésta debe de ser de un mínimo de 8 caracteres y contener minúsculas, mayúsculas, números o signos, por ejemplo para el usuario va a ser “**weblogic**” y la contraseña “**weblogic**”, vea la figura 4.4.16

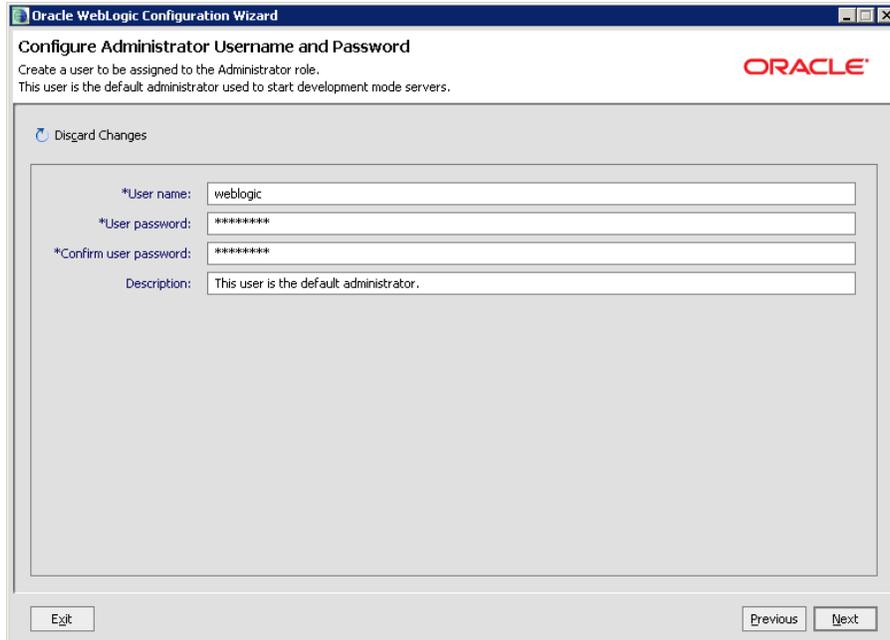


Figura 4.4.16 Datos del administrador

En la siguiente sección nos pregunta por la configuración del SDK, para el desarrollo lo recomendable es seleccionar “**Development Mode**” y el JDK “**Sun SDK**”, para producción se debe seleccionar “**Production Mode**” y el JDK “**JRockit SDK**”, vea la pantalla mostrada en la figura 4.4.17

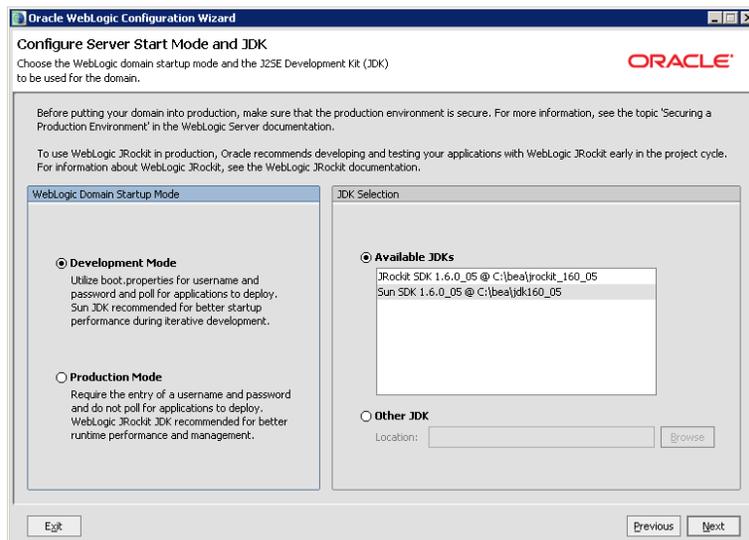


Figura 4.4.17 Configuración del SDK

Posteriormente no presenta la pantalla de configuración del entorno de ejecución y de los servicios Weblogic, vea la figura 4.4.18, elegimos “No” y damos click en Next.

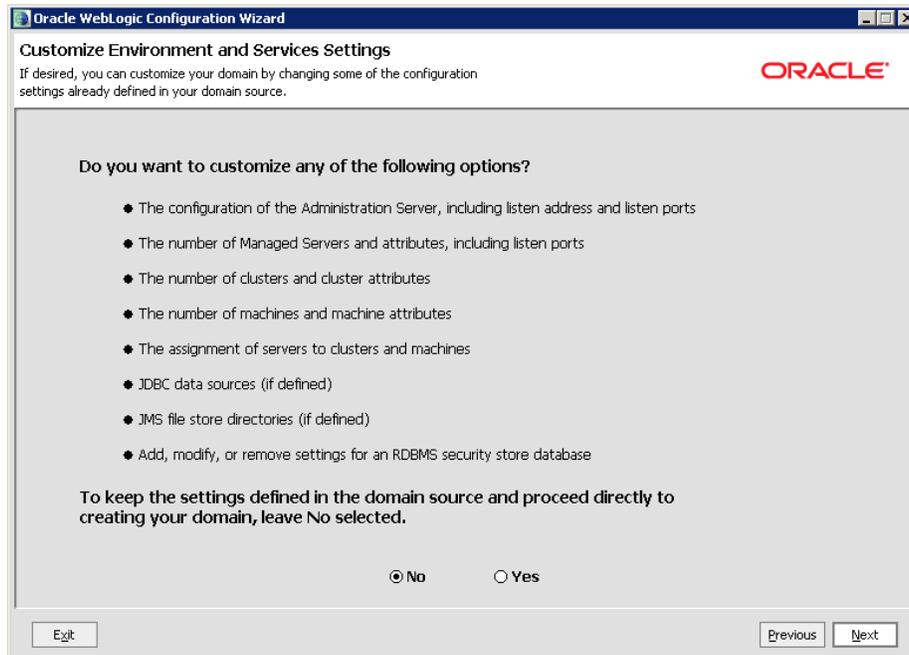


Figura 4.4.18 Configuración del entorno y de los servicios

Lo siguiente es seleccionar un nombre para nuestro nuevo dominio, por ejemplo: **sisc_domain**, damos click en Create, vea la pantalla mostrada en la figura 4.4.19

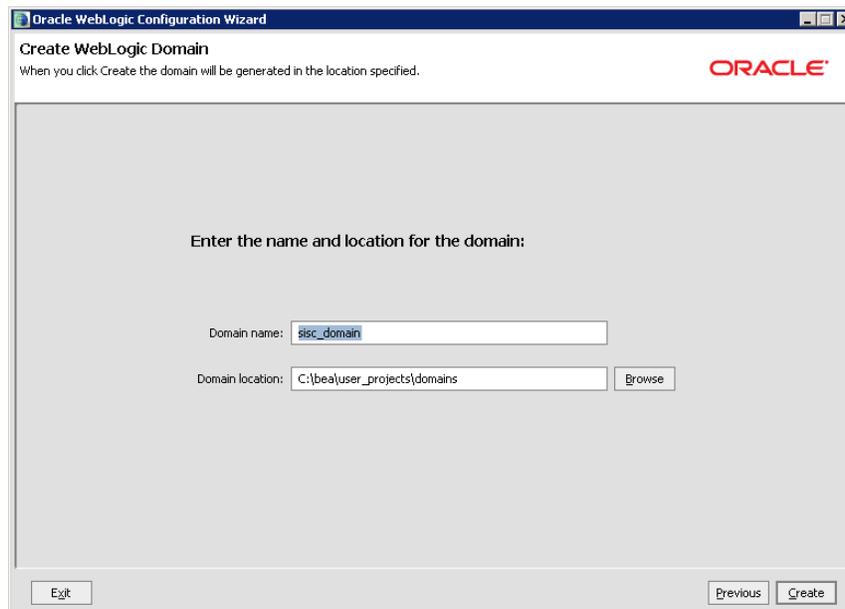


Figura 4.4.19 Selección del tipo de origen de dominio

Inicia el proceso de creación y después de algunos minutos concluye, vea la pantalla en la figura 4.4.20

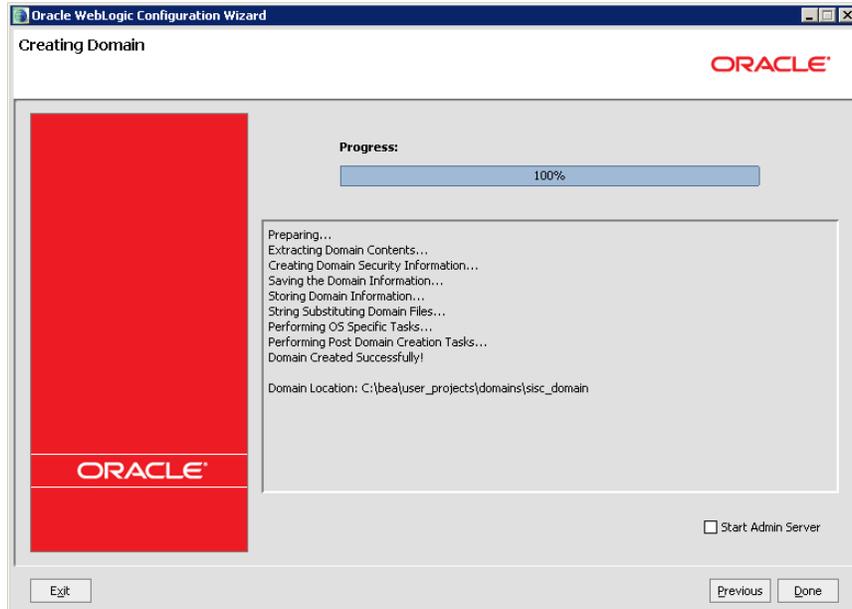


Figura 4.4.20 Creación del dominio

Lo siguiente es levantar el dominio, para esto vamos al menú “User Projects”, dentro de este podemos encontrar los dominios que hayamos instalado, vamos a “sisc_domain” y ejecutamos “Start Admin Server for Weblogic Server Domain”, vea la figura 4.4.21

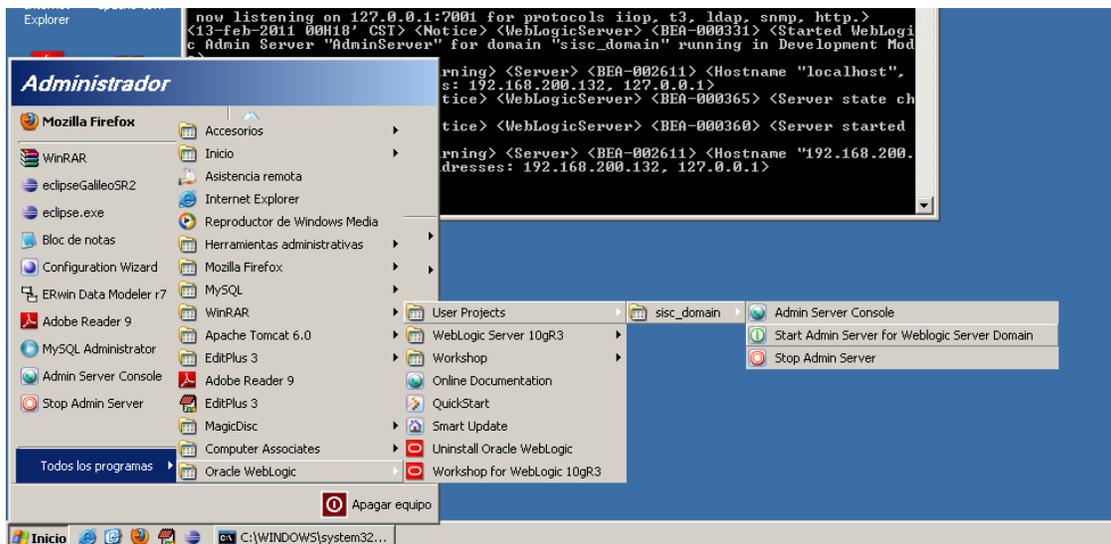


Figura 4.4.21 Menú de opciones administrativas del dominio

Se ejecutará una consola o terminal y esperaremos hasta que aparezca **<Server started in RUNNING mode>**, vea la figura 4.4.22

```

Start Admin Server for Weblogic Server Domain - C:\bea\user_projects\domains\sisc_domain\bin\st...
* To start WebLogic Server, use a username and *
* password assigned to an admin-level user. For *
* server administration, use the WebLogic Server *
* console at http://hostname:port/console *
*****
starting weblogic with Java version:
java version "1.6.0_05"
Java(TM) SE Runtime Environment (build 1.6.0_05-b13)
Java HotSpot(TM) Client VM (build 10.0-b19, mixed mode)
Starting WLS with line:
C:\bea\jdk160\1\bin\java -client -Xms256m -Xmx512m -XX:CompileThreshold=8000 -
XX:PermSize=48m -XX:MaxPermSize=128m -Xverify:none -da -Dplatform.home=C:\bea
\WLSERU~1.3 -Dwls.home=C:\bea\WLSERU~1.3\server -Dweblogic.home=C:\bea\WLSERU~1.
3\server -Dweblogic.management.discover=true -Dwls.iterativeDev=-Dwls.testCo
nsole=-Dwls.logErrorsToConsole=-Dweblogic.ext.dirs=C:\bea\patch_wlw1030\profil
es\default\sysext_manifest_classpath;C:\bea\patch_wls1030\profiles\default\syse
xt_manifest_classpath;C:\bea\patch_cie660\profiles\default\sysext_manifest_class
path -Dweblogic.Name=AdminServer -Djava.security.policy=C:\bea\WLSERU~1.3\server\
lib\weblogic.policy weblogic.Server
<13-feb-2011 00H25' CST> <Info> <WebLogicServer> <BEA-000377> <Starting WebLogic
Server with Java HotSpot(TM) Client VM Version 10.0-b19 from Sun Microsystems I
nc.>
<13-feb-2011 00H25' CST> <Info> <Management> <BEA-141107> <Version: WebLogic Ser
ver 10.3 Fri Jul 25 16:30:05 EDT 2008 1137967 >
    
```

Figura 4.4.22 Proceso de arranque del servidor

Regresamos al menú y ejecutamos **“Admin Server Console”**, vea la figura 4.4.23

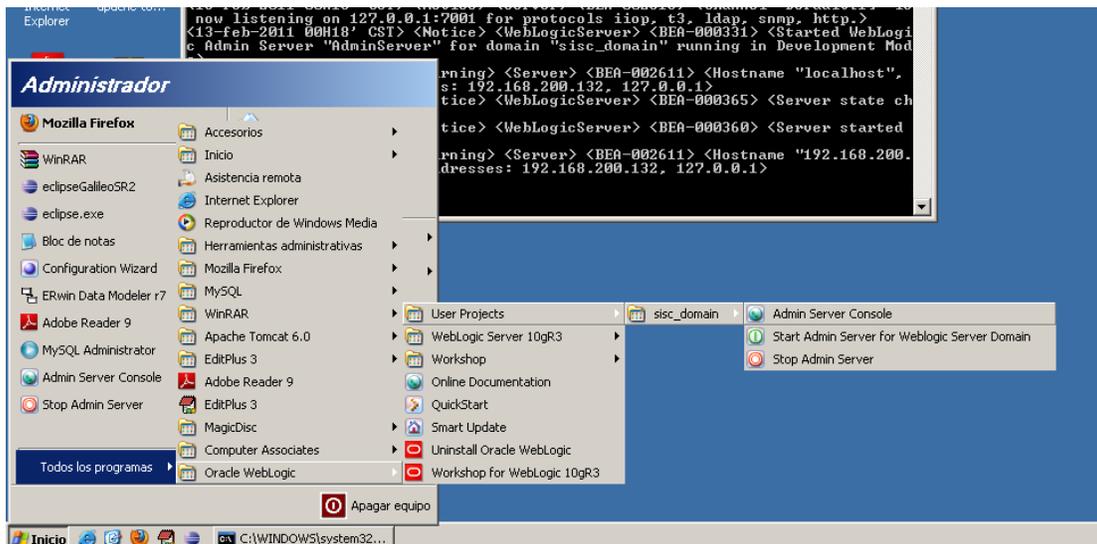


Figura 4.4.23 Menú de opciones administrativas del dominio

Esto abrirá nuestro navegador en la página de inicio de sesión de la consola de administración, ingresamos nuestro usuario y contraseña definidos durante el proceso de creación del dominio. Al entrar, si todo está bien, debemos ver la pantalla principal de la consola con todas las opciones de administración de Weblogic Server, vea las pantallas mostradas en las figuras 4.4.24 y 4.4.25

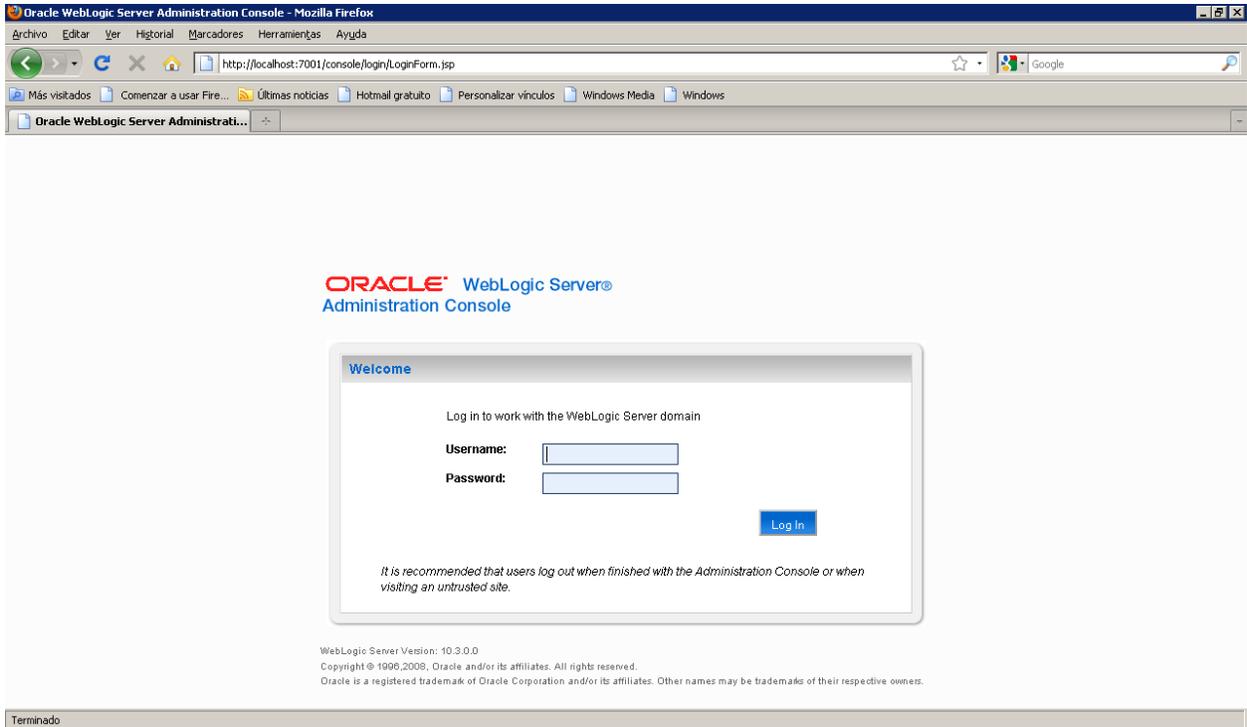


Figura 4.4.24 Pantalla de inicio de sesión en la consola de administración

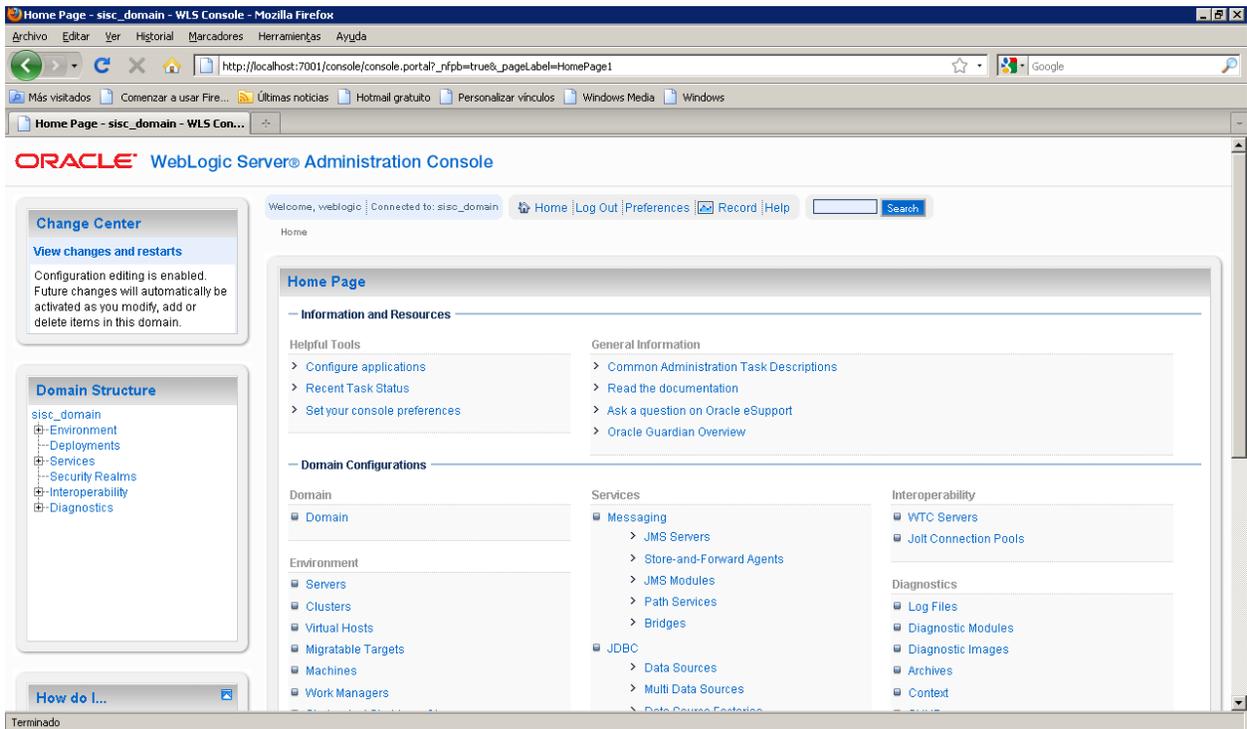


Figura 4.4.25 Pantalla principal de la consola de administración

4.5 DISEÑO Y CONSTRUCCIÓN DEL FRONT END

El sitio fue diseñado con Eclipse Weblogic, en él se encuentran diversos elementos que le ayudaran a navegar y a utilizar el portal, se verán los más comunes que se encuentran en el sitio, los menús que se utilizan son pull-down. Con ellos se tienen las siguientes ventajas:

Los nombres de los menús son visibles textualmente sobre la barra de menús, de tal forma que el usuario puede ver de un solo vistazo las funciones principales de la aplicación.

Los títulos de los menús y los comandos del menú no deberán ser ambiguos, esto nos lleva a poder crear botones en una forma rápida y clara

La barra de menús está en la parte más alta ocupando poco espacio en la imagen.

Los menús son accesados por medio de una combinación de teclas (manteniendo presionada la tecla Alt y la letra que esta subrayada en el nombre del menú.

Las funciones son agrupadas para formar un conjunto de comandos que están relacionados, esto ayuda a los usuarios al buscar una función específica (saben dónde encontrarla).

Se han realizado esfuerzos para estandarizar los nombres y el orden de los menús, tal que los paquetes sean fáciles de aprender y usar.

Así cada uno de los menús está diseñado para contener opciones relacionadas con el título del mismo, así como de forma secuencias en la mayoría de los menús se encontraran opciones para dar de alta nuevos datos con botones y opciones bastante intuitivos y que son del dominio de cualquier usuario que esté relacionada con estos trámites, así mismo en la mayoría de las opciones se encontraran opciones para dar de baja datos como alternativa de que se hayan ingresado mal o que desea eliminar algún dato incorrecto. Aunque también se encuentra la opción de modificar por si no es necesario eliminar todos los datos ya introducidos.

Ejecutamos la aplicación de eclipse weblogic con el icono siguiente de la figura 4.5.1

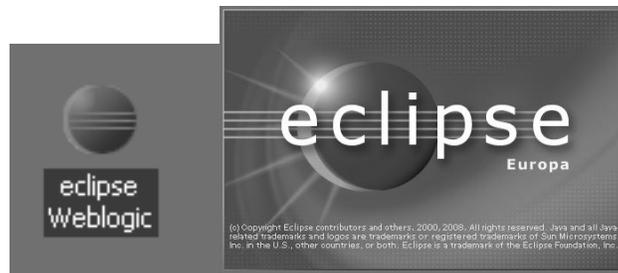


Figura 4.5.1 Icono de la aplicación

La aplicación nos pedirá un lugar de trabajo en donde se almacenara la información del sitio, tal como lo muestra la figura 4.5.2



Figura 4.5.2 Ejecución del programa

Menú file / new / project como lo muestra la figura 4.5.3

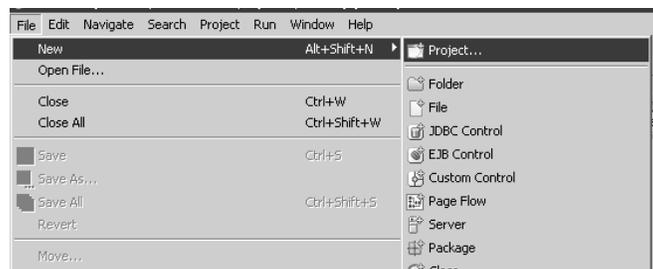


Figura 4.5.3 Barra de herramientas weblogic

Seleccionamos proyecto dinámico como lo muestra la figura 4.5.4

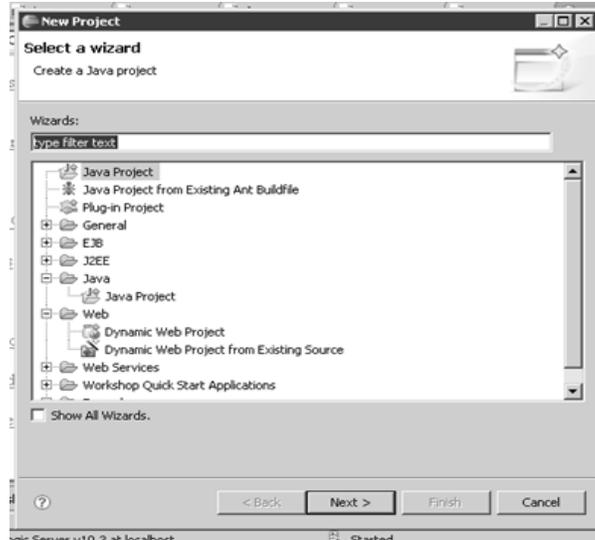


Figura 4.5.4 Interfaz para selección de tipo de proyecto

Asignamos el nombre al proyecto y finish tal como lo muestra la figura 4.5.5

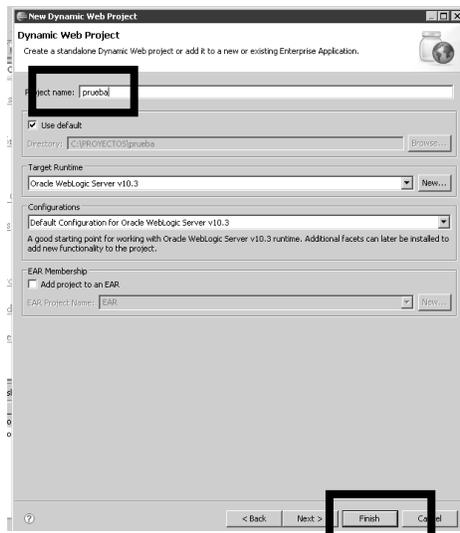


Figura 4.5.5 Ventana para asignar nombre al proyecto

Nos creara la estructura de carpetas en la parte izquierda de la pantalla como lo muestra la figura 4.5.6

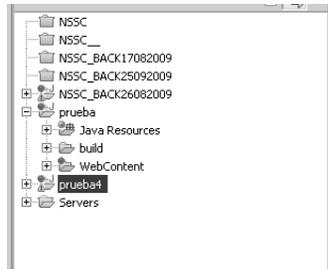


Figura 4.5.6 Árbol de proyectos en weblogic

Es aquí donde vamos agregando archivos .jsp para agregar el código para que se visualice en el navegador.

Click derecho sobre webcontent / new / jsp tal como lo muestra la figura 4.5.7

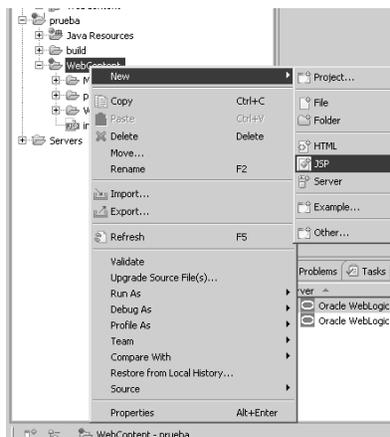


Figura 4.5.7 Menú contextual para agregar nuevos archivos

Le damos nombre a nuestro contenido en este caso la del inicio y finish como se puede apreciar en la figura 4.5.8

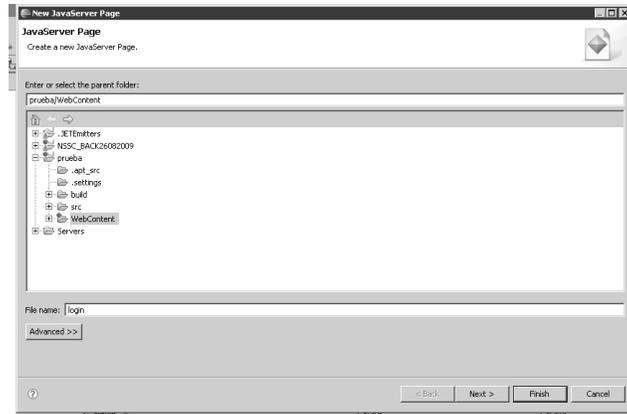


Figura 4.5.8 Asignar nombre al nuevo archivo

Seleccionamos el elemento e insertamos el código que presentara la pantalla de inicio, yal y como lo muestra la figura 4.5.9

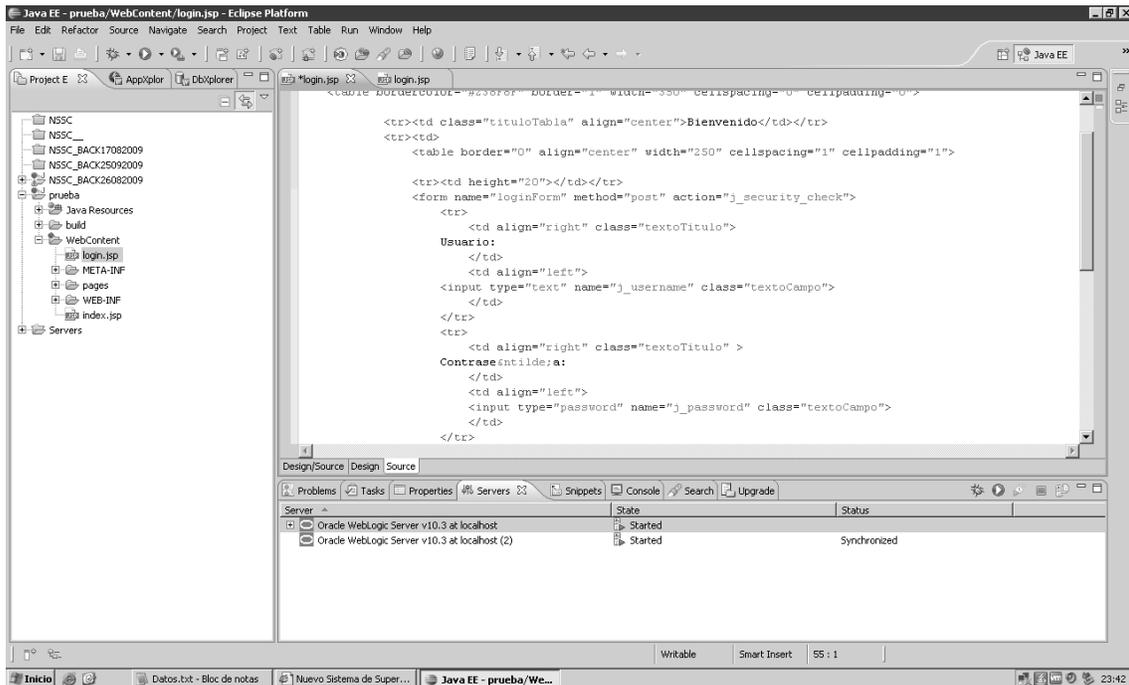


Figura 4.5.9 Editor donde se incluye el código

Para los menús desplegables se utiliza otro archivo .jsp y de nuevo ingresamos el código correspondiente como se muestra en la figura 4.5.10

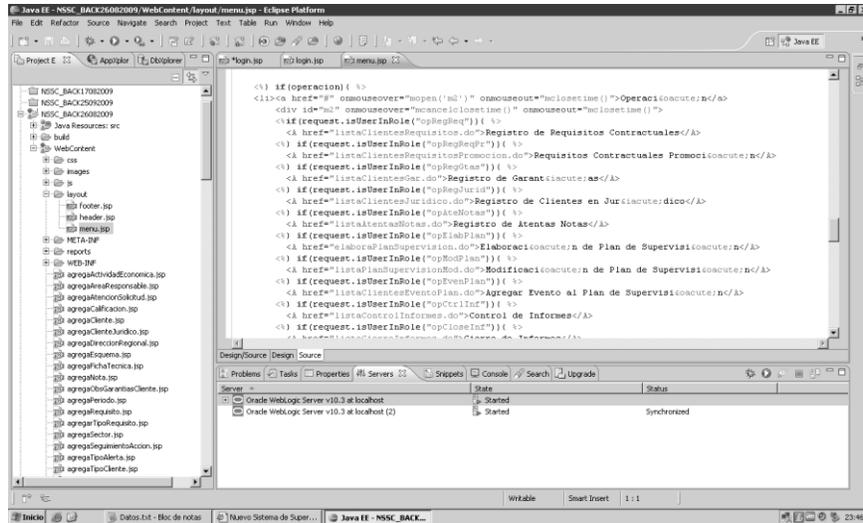


Figura 4.5.10 Editor donde se incluye el código de los menús desplegables

En este caso el menú operación se muestra en la siguiente figura 4.5.11

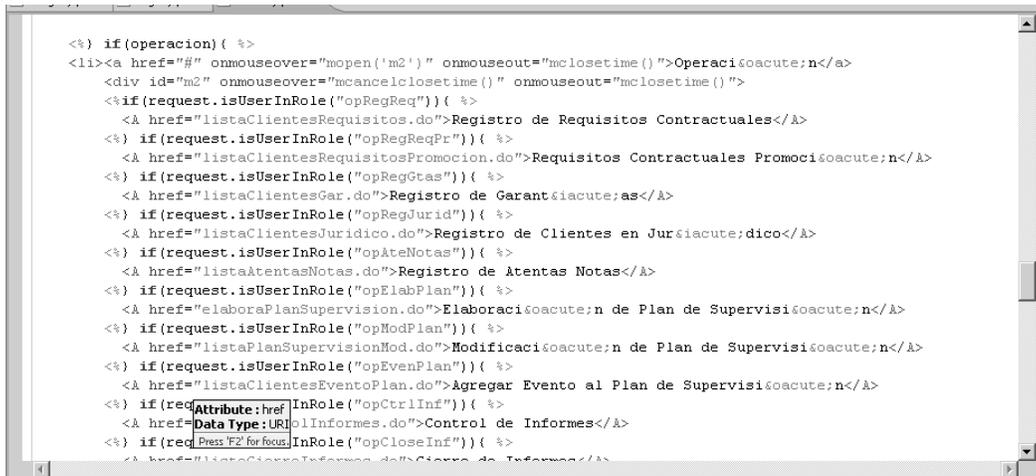


Figura 4.5.11 Editor donde se incluye el código del menú de operación

De esta forma el resultado de seguir los pasos anteriores se muestra en la figura 4.5.12

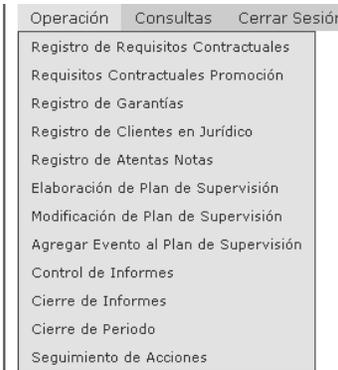


Figura 4.5.12 Resultado de la ejecución de los códigos anteriores

Para la opción del de dar de alta una dirección regional por ejemplo se crea la misma estructura con su correspondiente código, tal como se muestra en la figura 4.5.13 a continuación:

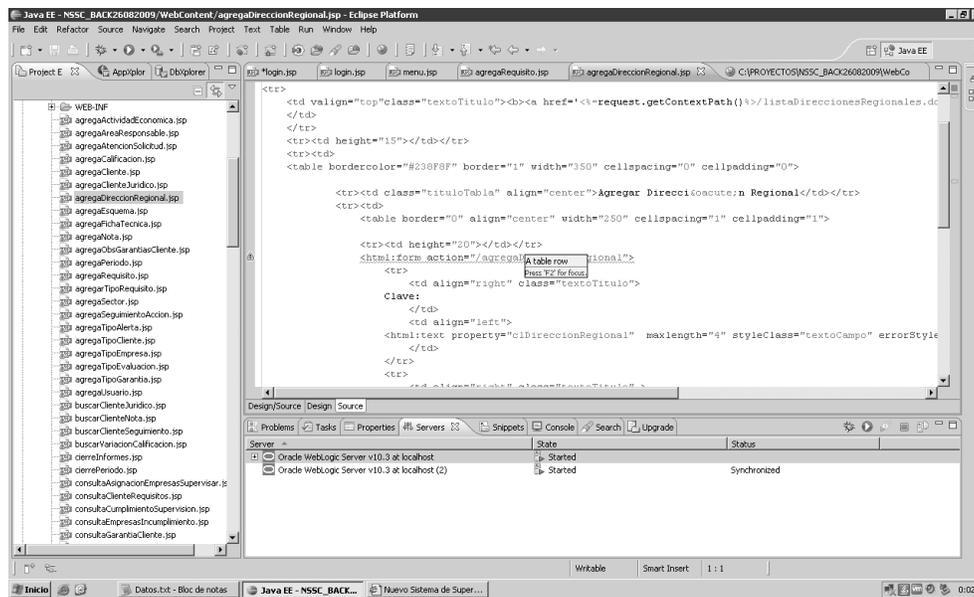


Figura 4.5.13 Editor en donde incluimos el código



```
<tr>
<td valign="top" class="textoTitulo"><b><a href='<%=request.getContextPath()%>/listaDireccionesRegionales.do
</td>
</tr>
<tr><td height="15"></td></tr>
<tr><td>
<table bordercolor="#238F8F" border="1" width="350" cellspacing="0" cellpadding="0">
<tr><td class="tituloTabla" align="center">Agregar Dirección Regional</td></tr>
<tr><td>
<table border="0" align="center" width="250" cellspacing="1" cellpadding="1">
<tr><td height="20"></td></tr>
<tr><td align="center"><html:form action="/agregarDireccionRegional">
<tr>
<td align="right" class="textoTitulo">
Clave:
<td align="left">
<html:text property="clDireccionRegional" maxlength="4" styleClass="textoCampo" errorStyle
</td>
</tr>
<tr>
<td align="right" class="textoTitulo">
Nombre:
<td align="left">
<html:text property="nombre" styleClass="textoCampo" errorStyle
</td>
</tr>
<tr>
<td align="right" class="textoTitulo">
Nombre Corto:
<td align="left">
<html:text property="nombreCorto" styleClass="textoCampo" errorStyle
</td>
</tr>
</tr>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

Figura 4.5.14 Muestra del código

El resultado de lo anterior se muestra en la figura 4.5.15 a continuación:



Catálogo de Direcciones Regionales >> Agregar

Agregar Dirección Regional

Clave:

Nombre:

Nombre Corto:

Aceptar Cancelar

Figura 4.5.15 Resultado de la ejecución del código

Para dar de alta un nuevo cliente se debe agregar el archivo .jsp e ingresar el código correspondiente como se muestra en la figura 4.5.16

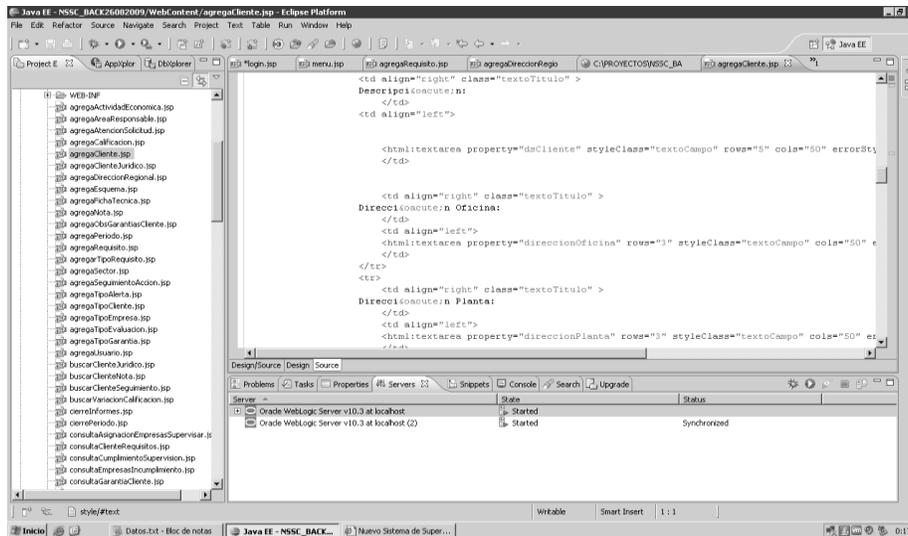


Fig. 4.5.16 Editor en donde incluimos el código

La descripción del código a detalle se muestra en la siguiente figura 4.5.17



Fig. 4.5.17 Muestra del código

El resultado de todo lo anterior lo podemos ver reflejado en la figura 4.5.18 que se muestra a continuación:

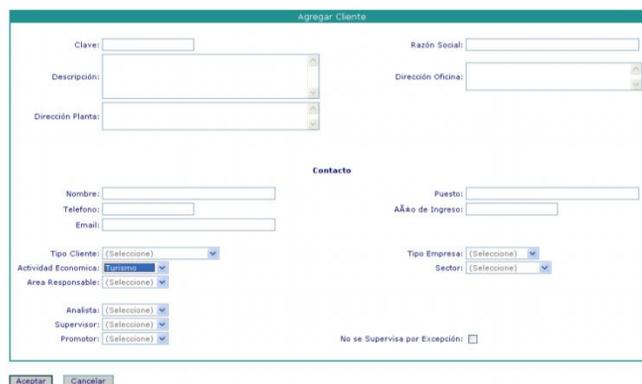


Fig. 4.5.18 Resultado de la ejecución del código

4.6 INTEGRACIÓN, PRUEBAS Y MANTENIMIENTO.

La fase de pruebas de un sistema añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre; ese es el valor que suministra. El objetivo específico de la fase de pruebas es encontrar cuantos más errores, mejor.

Como introducción a esta sección de pruebas y mantenimiento se presentan aquí algunas definiciones con la intención de proporcionar la teoría de las pruebas y aseguramiento de calidad, así como para homologar los conceptos necesarios utilizados durante esta fase:

Caja blanca

También llamadas:

Pruebas estructurales.

Pruebas de caja transparente.

Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que se comporta adecuadamente ante determinadas situaciones.

En estas pruebas estamos siempre observando el código. Esta noción de prueba total se formaliza en lo que se llama "cobertura" y no es sino una medida porcentual de ¿cuánto código hemos cubierto?

Ejemplo de Caja Blanca

Agregar un Nuevo Cliente

A continuación con fines ilustrativos, en la figura 4.6.1 se muestra una porción de código fuente del sistema, particularmente al Agregar a un Nuevo Cliente Crediticio.

En la figura 4.6.2 se muestra el formulario donde se dan de alta los datos del Nuevo Cliente, al oprimir el botón "Aceptar" el sistema guarda la información y el registro se almacena en la Base de Datos.

```

dao/ClienteDao.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
dao/ClienteDao.java
public void agrega(ScCCliente cliente) throws HibernateException {
    session.beginTransaction();
    session.save(cliente);
    session.getTransaction().commit();
    /* Close session */
    session.close();
}

public List ListaClientes(String criterio) throws HibernateException {
    ArrayList clientes = new ArrayList();
    try {
        Connection cn = session.connection();
        String QUERY = "SELECT C.ID_CLIENTE, C.CL_CLIENTE, C.RAZON_SOCIAL, "
            + "TC.ID_TIPO_CLIENTE, "
            + "TC.NB_TIPO_CLIENTE, "
            + "SC.FN_TIENE_REQ_VENCIDO(C.ID_CLIENTE) TIENE_REQ_VENCIDO "
            + "FROM SC_C_CLIENTE C, "
            + "SC_C_TIPO_CLIENTE TC "
            + "WHERE TC.ID_TIPO_CLIENTE(+) = C.ID_TIPO_CLIENTE ";

        if (criterio != null && criterio.length() > 0) {
            QUERY += "AND ( TO_CHAR(C.CL_CLIENTE) LIKE ? "+
                "OR UPPER(C.RAZON_SOCIAL) LIKE ? "+
                "OR UPPER(TC.NB_TIPO_CLIENTE) LIKE ? ) ";
        }

        QUERY += "ORDER BY C.RAZON_SOCIAL";
        PreparedStatement ps = cn.prepareStatement(QUERY);
        if (criterio != null && criterio.length() > 0) {
            ps.setString(1, "%" + criterio.toUpperCase() + "%");
            ps.setString(2, "%" + criterio.toUpperCase() + "%");
            ps.setString(3, "%" + criterio.toUpperCase() + "%");
        }
        ResultSet rs = ps.executeQuery();
        ScCCliente cliente = null;
    }
}

```

Figura 4.6.1 Código para agregar un Cliente Crediticio

Nuevo Sistema de Supervisión de Crédito - Internet explorer proporcionado por GP

Archivo Edición Ver Eavoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://localhost:7001/SI5C/consultaCliente.do?idCliente=586 Ir Vínculos

Dirección Planta: Año de Ingreso:

Datos del Contacto

Nombre: raul perez lopez * Puesto: Gerente general *

Teléfono(s): 31842061 * e-mail: raul@gmail.com *

Tipo Cliente: Zo Piso Bancos Tipo de Empresa: Empresarial

Actividad Económica: Sin Actividad Económica Sector: Sector Privado *

Área Responsable: METRO

Analista: Enrique Bermudez Mendez

Supervisor: J. Gabriel García Velasco

Promotor: Felipe Ortiz Flores No se Supervisa por Excepción: NO

Promotor Adicional: (Seleccione) Incluir en Watch List: NO

Calificación (puntos):

Datos de Visita

Año de Visita (aaaa): 2011 Mes de Visita (mm): 5

Observaciones Asociadas a la Visita:

Aceptar Cancelar

Listo Intranet local

Figura 4.6.2 Formulario para dar de alta un Nuevo Cliente Crediticio

Caja negra

También llamadas:

Pruebas de caja opaca.

Pruebas funcionales.

Pruebas de entrada/salida.

Pruebas inducidas por los datos.

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc.)

Prueba de Caja Negra

Acceso al Sistema

En la figura 4.6.3 se muestra la pantalla de acceso al sistema. Si introducimos valores incorrectos para el usuario o el password el sistema le niega el acceso al usuario y vacía los datos del formulario (cajas de texto), indicando además el error del que se trata.

La otra alternativa en esta misma prueba es cuando el usuario y el password proporcionados en el formulario son válidos, esto nos llevara a la pantalla de inicio (principal) del Sistema, la cual se muestra en la figura 4.6.4

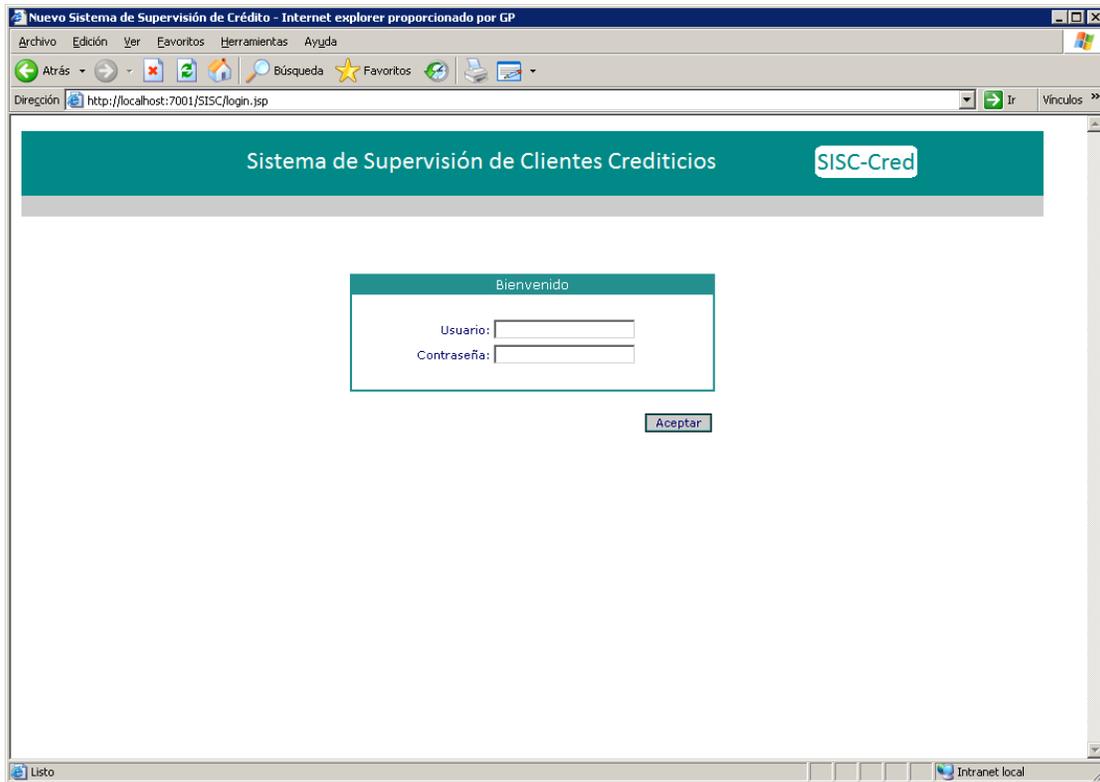


Figura 4.6.3 Pantalla de Acceso al Sistema

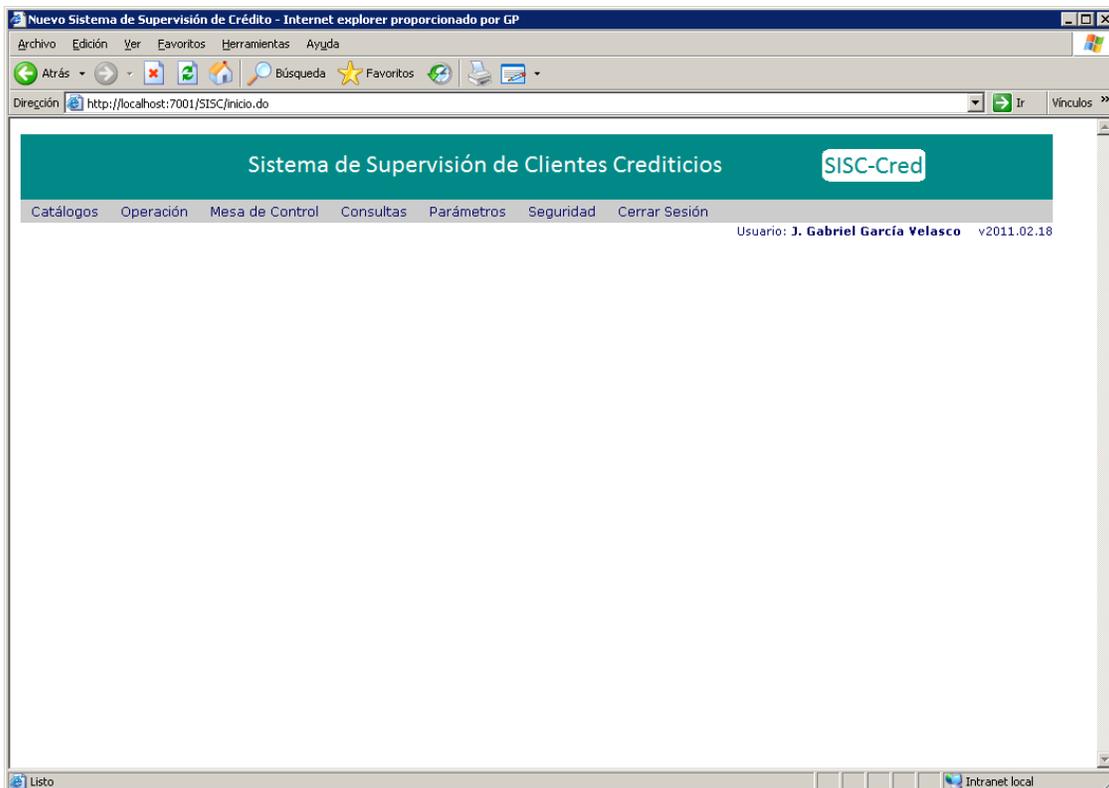


Figura 4.6.4 Pantalla Principal del Sistema

Pruebas de Integración

Estas pruebas se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.

Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Las pruebas estructurales de integración son similares a las de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirnos a las sentencias del lenguaje, nos referimos a llamadas entre módulos.

Pruebas de Carga

Se refieren al comportamiento de una aplicación ante una carga determinada: Esta carga puede ser el número de usuarios esperado ejecutando o un número de transacciones durante un tiempo predeterminado.

Pruebas de Regresión

Son aquellas pruebas que permiten validar las correcciones realizadas a las fallas detectadas en las iteraciones.

Pruebas de Estrés

Son utilizadas normalmente para someter a la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios superior al esperado.

Pruebas de Resistencia

Son las que determinan si la aplicación puede mantener la carga esperada de manera continua y durante un largo tiempo.

Pruebas de Picos

Se realizan insertando la carga en el sistema en forma de “picos” que se irán lanzando en distintos momentos de la prueba y que permitirán comprender el comportamiento de la aplicación ante cambios bruscos de carga.

Pruebas de Aceptación

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual de usuario. La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que solo aparecen cuando el cliente lo usa.

Mantenimientos

Mantenimiento Preventivo

La finalidad del mantenimiento preventivo es: Encontrar y corregir problemas menores antes de que estos provoquen fallas.

Bajo esa premisa se diseña el programa de revisiones periódicas al sistema, con el fin de determinar la efectividad del mismo y se optimiza en función de una constante evaluación.

Mantenimiento Correctivo

Es cuando se debe corregir errores de funcionamiento del sistema, “bugs”, o cuando aparecen situaciones que no se tuvieron en cuenta o se mal interpretaron en el relevamiento.

Ejemplos: totaliza mal un listado, no filtra un informe por zona, no redondea un total.

Mantenimiento Adaptativo

Cuando el objetivo es adaptar el sistema o partes de él a nuevas situaciones que generan nuevos requerimientos, por la dinámica evolución de las empresas y los negocios.

Ejemplos: se comienza a consignar mercadería, se abre un punto de venta, se vende en el interior, se aplica un nuevo impuesto, se define una nueva política de descuentos.

Mantenimiento perfectivo

Modificación de un producto software después de la entrega para mejorar el rendimiento.

No se considera mantenimiento a los cambios introducidos para incluir nuevos requisitos funcionales. No obstante, no hay un consenso unánime en este sentido, y de hecho, el concepto de evolución del software, amplía el espectro del mantenimiento a cambios en un sentido amplio. De hecho, hay autores que consideran que el mantenimiento perfectivo sí incluye cambios en la funcionalidad.

El estándar ISO/IEC 14764 clasifica las categorías comentadas hasta ahora según la siguiente Tabla 4.6.5, que nos puede ayudar a ver sus diferencias.

	Corrección	Mejora
Proactiva	Preventivo	Perfectivo
Reactiva	Correctivo	Adaptativo

Tabla 4.6.5 Clasificación de Mantenimientos

Una visión más general de los tipos de mantenimiento, se puede observar en la figura 4.6.6 siguiente, ya que se distinguen los diferentes tipos de mantenimiento según cambios de software, cambios de código fuente o cambios de funcionalidad.

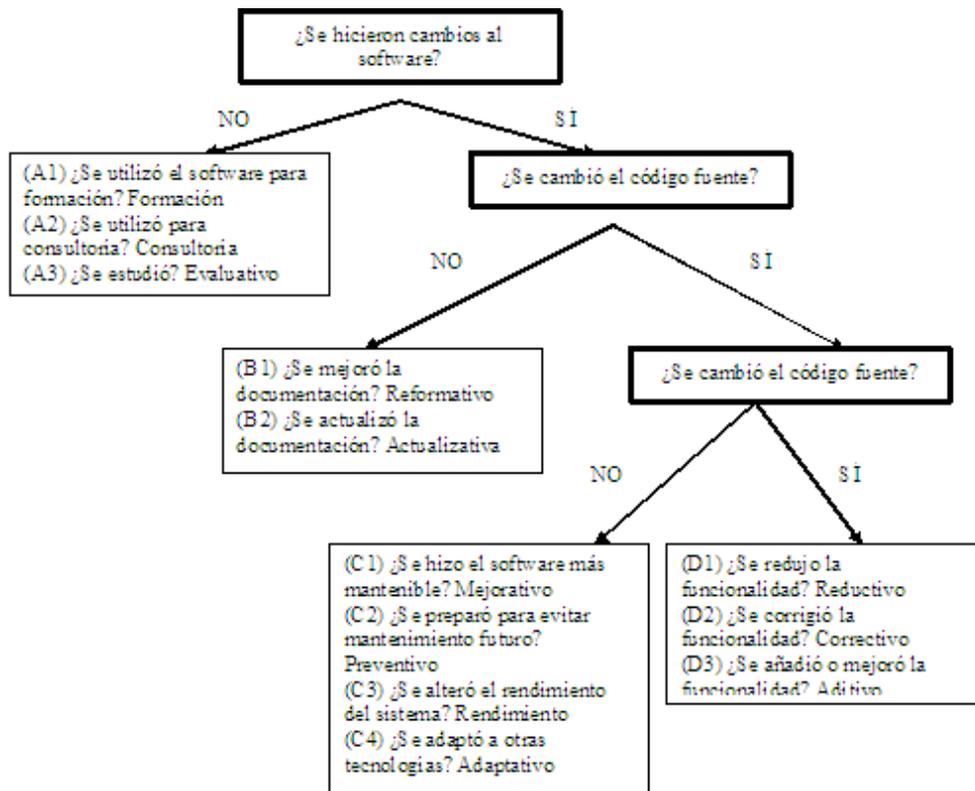


Figura 4.6.6 Tipos de Mantenimiento según cambios en el Software

4.7 GENERADOR DE REPORTES.

El sistema cuenta con la opción de generar los reportes siguientes: Informes por cliente, Requisitos Contractuales, Atentas Notas, Solicitudes Mesa de Control, Plan de Supervisión Mensual, Plan de Supervisión Anual, Asignación de Empresas a Supervisar, Variación de Calificación por Cliente, Avance de Cumplimiento de Supervisión, Alertas de Clientes, Empresas en Incumplimiento.

Debido a lo extenso de alguno de estos reportes, se presentan a continuación algunas capturas de pantalla de los más útiles para la supervisión del crédito.

Requisitos Contractuales.

Es un reporte detallado de los documentos contractuales del crédito, está organizado en forma tabular donde cada renglón contiene un documento como puede ser: Dividendos, Capitalización, Póliza de Seguro de los Bienes otorgados en Garantía, Avalúos Bancarios, Buró de Crédito, Documentación Comprobatoria del Uso de los Recursos, Demandas, Información para seguimiento y supervisión del Crédito, Estados Financieros Dictaminados o Certificados, Estados Financieros Internos, etc. Y cada columna contiene los siguientes campos para cada documento: Descripción, donde se describe de manera detallada el contenido del documento; Comentarios, que ayudan a contextualizar cada documento dentro del crédito y a darle seguimiento; Periodo de vigencia, especifica las fechas de inicio y fin de la vigencia del documento; Status, especifica si el documento está vigente o vencido; No. Días vigentes; pone en números positivos la cantidad de días que le quedan de vigencia a un documento vigente, y en el caso de los documentos vencidos, el número de días que lleva vencido. Este reporte es de suma importancia para la institución, pues en este se especifican las reglas para la supervisión del crédito Ver Figura 4.7.1

DIRECCION DE SEGUIMIENTO Y RECUPERACION							Página 1 de 3
REPORTE DE CUMPLIMIENTO DE REQUISITOS DURANTE LA VIGENCIA DE LAS LINEAS DE CREDITO							
IFG- FOMENTO - PYMEX							
Cumplimiento de Requisitos Durante la Vigencia de la Línea de Crédito							
Documento	Descripción	Comentarios	Periodo de Vigencia del	Vigencia al	Status	No. días Vigentes	
Ciente: 288552 11 DE DICIEMBRE DE 1996, S.P.R. DE R.I.							
1	Dividendos	Podrá decretar el pago de dividendos, siempre y cuando cuente con la autorización previa y por escrito de Bancomext, quien se reserva el derecho de en caso de aprobarlos, establecer condiciones tales como la realización de pagos anticipados al crédito, así como los términos bajo los cuales se aplicarán tales pagos anticipados.	De acuerdo a los estados financieros dictaminados al 31 de diciembre del 2007 se aprecia una reducción de \$ 1'055,000.00 en el rubro de utilidades acumuladas, por lo que se presume que se repartieron dividendos. Con fecha 24/07/2009 El Comité Central se dio por enterado del reparto de dividendos en el ejercicio del 2007 y expresó su conformidad para que por esta ocasión, no sea causa de vencimiento anticipado.	30/04/2008	30/11/2009	VIGENTE 13	
2	Capitalización	Dentro de un plazo de 3 meses contados a partir de la formalización del presente crédito, deberá aumentar su capital social hasta por un importe equivalente a USD 1'891,079.97.	En Informe de Supervisión del 11-jul-05 se validó el cumplimiento de este requisito.	30/04/2009	30/11/2009	VIGENTE 13	
3	Póliza de Seguro de los Bienes otorgados en Garantía	j) Contratar un seguro que cubra los bienes que sirven de garantía, incluyendo, en su caso, el riesgo que se llegaría a generar por su transporte y por la actividad de la empresa a la que están destinados, de por lo menos el 100% del monto adeudado y del valor de cobertura de los bienes otorgados en garantía. En cualquiera de las pólizas, de estar asegurados, endosar la póliza correspondiente, designando en ambos casos a Bancomext como beneficiaria preferencial. La acreditada se obliga a entregar a Bancomext, copia de la póliza y el original respectivo designando a Bancomext como beneficiario preferencial, así como copias del pago de las primas. Dicha póliza deberá mantenerse vigente mientras permanezca insóluto el monto adeudado. Si no cumple con esta obligación, Bancomext podrá contratar el seguro respectivo, pagando por cuenta de la acreditada su importe, devengándose a una tasa de interés del 50% anual. La acreditada deberá reembolsar a Bancomext dicho importe con accesorios dentro de los 3 días naturales que sigan a la fecha del aviso que Bancomext de la acreditada, conforme a lo pactado en el presente convenio.	HIPOTECA CIVIL: Casa habitación propiedad de Egrén Eduardo Encinas Yepiz Se cuenta con póliza de seguro GNP Nacional Provincial No. 21740691, con vigencia del 18 de febrero del 2009 al 18 de febrero del 2010, con suma asegurada de M.N. \$ 500,000 daños al inmueble y M.N. \$ 100,000 daños a los contenidos, con recibo de pago de prima anual que cubre hasta el 18 de febrero del 2010 y copia del endoso a favor de BANCOMEXT. El valor de avalúo es de M.N. \$ 453,000. La suma asegurada cubre el valor real. HIPOTECA CIVIL: Casa habitación propiedad de Griselda Gómez Villanueva Se cuenta con póliza de seguro GNP Nacional Provincial No. 21740394, con vigencia del 18 de febrero del 2009 al 18 de febrero del 2010, con suma asegurada de M.N. \$ 450,000 daños al inmueble y M.N. \$ 225,000 daños a los contenidos, con recibo de pago de prima anual que cubre hasta el 18 de febrero del 2010 y copia del endoso a favor de BANCOMEXT. El valor de avalúo es de M.N. \$ 518,000. La suma asegurada cubre el valor real. HIPOTECA CIVIL: Casa habitación propiedad de Miguel A. González Hernández Se cuenta con póliza de seguro GNP Nacional Provincial No 21740592, con vigencia del 19 de febrero del 2009 al 19 de febrero del 2010, con suma asegurada de M.N. \$ 445,000 daños al inmueble y M.N. \$ 222,500 daños a los contenidos, con recibo de pago de prima anual que cubre hasta el 19 de febrero del 2010 y copia del endoso a favor de BANCOMEXT. El valor de avalúo es de M.N. \$ 596,000. La suma asegurada cubre el valor real.	31/01/2009	30/11/2009	VIGENTE 13	

Figura 4.7.1 Reporte De Requisitos Contractuales

Cargas de trabajo. Es un reporte que presenta en forma tabular las cargas de trabajo de cada supervisor y el avance en la supervisión de los créditos que se le asignaron. La primera columna contiene el supervisor, la siguiente sección "Empresa asignada" se subdivide en 7 clasificaciones de empresas, y un total que es la suma de las anteriores. Le sigue el programa de supervisión que se subdivide en: supervisiones programadas, supervisiones realizadas, y porcentaje de avance. Por último, presenta el total de solicitudes atendidas. Este reporte puede servir a la institución para asignar de manera racional la carga de trabajo a los supervisores, llevar un seguimiento del avance de la supervisión, y como indicador de la eficiencia del supervisor y de la calidad del servicio. Ver Figura 4.7.2



BANCO NACIONAL DE COMERCIO EXTERIOR, S.N.C.
DIRECCIÓN DE SEGUIMIENTO Y RECUPERACIÓN
GERENCIA DE SUPERVISIÓN Y SEGUIMIENTO
CARGAS DE TRABAJO
CIFRAS AL: 1 / 2008

Página 1 de 1

Supervisor	Empresas Asignadas								Programa de Supervisión			Mesa de Control
	< 900 mil Udi's	> 900 mil Udi's	Juridico	Banco	IFNB	Cont.	CCI	Total	Programadas	Realizadas	Avance	Sol. Atendidas
Abenamar Albores Cruz	0	0	0	0	0	0	0	0	0	0	0.00	0
Angelica Leon Chavez	0	0	0	0	0	0	0	0	38	1	2.63	0
Carlos Campos Martinez	0	0	0	0	0	0	0	0	43	0	0.00	0
Carlos Lemus Romero	0	0	0	0	0	0	0	0	46	0	0.00	0
Cesar Augusto Aureoles Garcia	0	0	0	0	0	0	0	0	34	0	0.00	0
Gitzy Rizo Prieto	0	0	0	0	0	0	0	0	3	0	0.00	0
Guadalupe Garcia Bustos	0	0	0	0	0	0	0	0	36	0	0.00	0
Lucia Gomez Ramirez	0	0	0	0	0	0	0	0	49	0	0.00	0
Maria Elena Guerrero Herrera	0	0	0	0	0	0	0	0	24	0	0.00	0
Maria Guadalupe Roldan Assad	0	0	0	0	0	0	0	0	2	0	0.00	0
Rafael Jimenez Trejo	0	0	0	0	0	0	0	0	29	0	0.00	0
Sergio Moreno Cahue	0	0	0	0	0	0	0	0	53	0	0.00	0
SIN SUPERVISOR	0	0	0	0	0	0	0	0	4	0	0.00	0
Yazmin Becerra Kahuum	0	0	0	0	0	0	0	0	0	0	0.00	0
Total	0	0	0	0	0	0	0	0	361	1	0.28	0

Figura 4.7.2 Reporte de Cargas de trabajo

Cumplimientos de Supervisión.

Presenta los avances del cumplimiento de la Supervisión del crédito tanto mensual, como acumulada. Este reporte le sirve a la institución para llevar un seguimiento del avance de la Supervisión pues se especifican las supervisiones que se programaron, el avance de las mismas y el porcentaje de cumplimiento. También puede servir como indicador de la calidad del servicio que presta la institución. Ver Figura 4.7.3

Página 1 de 1

DIRECCIÓN DE SEGUIMIENTO Y RECUPERACIÓN
GERENCIA DE SUPERVISIÓN Y SEGUIMIENTO
PROGRAMA DE SUPERVISIÓN 2008
CUMPLIMIENTO: 1 / 2008

RESULTADOS DEL MES

Tipo de Informe	Supervisiones Programadas	Avance		% Cumplimiento
		Informes Realizados		
Supervisión	4	0		0.00
Seguimiento	0	0		0.00
Subtotal	4	0		0.00

RESULTADOS ACUMULADOS

Tipo de Informe	Supervisiones Programadas	Avance		% Cumplimiento
		Informes Realizados		
Supervisión	4	0		0.00
Seguimiento	0	0		0.00
Subtotal	4	0		0.00

Figura 4.7.3 Reporte de Cumplimientos de Supervisión

CONCLUSIONES.

Partiendo de los lineamientos y premisas establecidas a lo largo de este trabajo de tesis podemos concluir lo siguiente:

Los requerimientos generales y particulares del sistema se cumplieron.

La implantación de un sistema de seguimiento crediticio es una tarea compleja que requiere de personal.

La arquitectura del sistema implica tener conocimientos en áreas tales como redes, sistemas operativos, manejadores de bases de datos y programadores en ambientes web de las diversas plataformas utilizadas para este fin.

El diseño de un proceso de negocio implica adoptar un modelo para representar el proceso mismo, tener el conocimiento de la reglas del negocio ayuda a la mejor implementación del modelo y a la mejor gestión del mismo.

El uso adecuado del sistema permite realizar un adecuado manejo de datos.

Posibilita a la organización a actuar y decidir sobre la base de la información de una manera confiable, precisa y oportuna.

Permite realizar análisis comparativos, pronósticos y simulaciones, reduciendo así, en un gran porcentaje la incertidumbre y muchos de los riesgos que se enfrentan.

El sistema ayuda a la empresa en las buenas prácticas por lo que aumenta la eficiencia y la eficacia de los procesos.

Mejora la productividad reduciendo la dependencia de personas.

Mejora la comunicación e integración entre las distintas áreas que conforman la organización.

Los flujos de información son de mejor calidad, se mejora el control interno y se produce un aumento del conocimiento.

Se pueden obtener reportes específicos y generales de acuerdo a las necesidades.

El seguimiento del proceso puede ser rastreado en línea a través de internet.

La administración de usuarios es simple y confiable.

Se puede monitorear y dar seguimiento a los procesos de validación en tiempo real.

Se puede monitorear y dar seguimiento a los procesos de expedición y/o certificación de créditos en tiempo real.

Los beneficios del sistema se podrán visualizar a corto plazo.

El mayor beneficio se tiene cuando ya se encuentra funcionando en su plenitud con toda la información disponible

Se garantiza el manejo de la información de una manera confiable ya que es capaz de asegurar en un alto grado la consistencia, disponibilidad, integridad y todas las características deseables que debe tener la información, para ser considerada un activo que aumente el valor de la empresa.

La formación recibida en la Facultad de Ingeniería de la UNAM cumple ampliamente con las expectativas para la generación de cualquier sistema de información que se requiera desarrollar o implementar.

El trabajo en equipo genera una lluvia de ideas que ayuda a la mejor visualización del problema y da una más amplia perspectiva del objetivo y sus posibles soluciones.

El proyecto ha resultado exitoso ya que se cumplieron con los requerimientos generales y particulares de los cuales partió este trabajo de tesis.

BIBLIOGRAFÍA.

- UML 2.0 in Action. A Project-Based Tutorial.
Autores: Patrick Grässle, Henriette Baumann, Philippe Baumann.
Editorial Packt Publishing Ltd., September 2005.
- Mastering Oracle SQL, 2nd Edition.
Autores: Alan Beaulieu, Sanjay Mishra.
Editorial: O'Reilly, June 2004.
- An Introduction to Database Systems.
Autor: DATE, C. J.
Editorial: Addison Wesley, 2003.
- Fundamentos de Sistemas de Bases de datos
Autores: ELMASRI RAMEZ A., NAVATHE SHAMKANT B.
Editorial: Pearson Prentice Hall, 2003.
- Professional Oracle WebLogic Server.
Autores: Robert Patrick, Gregory Nyberg, Philip Aston.
Editorial: Wiley Publishing, Inc., 2010.
- Redes de Computadoras 3er Edición
Autor: Andrew S. Tanenbaum
Editorial: Prentice Hall
- Computer Networks, Protocols, standards, and Interfaces
Autor: Uyles Black
Editorial: Prentice Hall
- Sistemas de Información Gerencial, Administración de la Empresa Digital
Autores Kenneth C. Laudon, Jane P. Laudon
Editorial: Prentice Hall, Pearson

DIRECCIONES ELECTRÓNICAS CONSULTADAS.

- Oracle
http://download.oracle.com/docs/cd/E13222_01/wls/docs103/index.html
http://download.oracle.com/docs/cd/E12840_01/common/docs103/install/index.html
http://download.oracle.com/docs/cd/E12840_01/wls/docs103/domain_config/understand_domains.html
- Connexions.
Tipos de Mantenimientos:
<http://cnx.org/content/m17408/latest/>
- Universidad de Valencia
Mantenimientos:
<http://informatica.uv.es/iiguia/2000/IPI/material/tema7.pdf>
- Universidad Simón Bolívar
Paginas del personal académico.
[http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/Teor%EDa%20PS6117%20Mantenimiento%20y%20Herramientas%20CASE.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/Teor%EDa%20PS6117%20Mantenimiento%20y%20Herramientas%20CASE.pdf)
- Todo expertos
Ventajas y Desventajas de Oracle
<http://www.todoexpertos.com/categorias/tecnologia-e-internet/bases-de-datos/oracle/respuestas/14706/vetajas-y-desventajas>
- Colección de Tesis Digitales UDLAP
Pruebas de sistema.
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/moreno_a_jl/capitulo5.pdf
- Diario digital de Sta. Fe
Analizaron un protocolo de seguridad para bancos
http://www.notife.com/noticia/articulo/1011426/zona/4/Analizaron_un_protocolo_de_seguridad_para_bancos.html

- Seguridad en la RED
Fraudes Online: Normas de Seguridad para una clave perfecta en Internet
<http://www.seguridadenlared.org/63.html>
- Wikilearning
Protocolos de seguridad
http://www.wikilearning.com/curso_gratis/curso_de_criptografia_basica_para_principiantes-protocolos_de_seguridad/4306-9
- Emagister
Cuál es el mejor método para administrar una cartera sana
http://grupos.emagister.com/debate/cual_es_el_mejor_metodo_para_administrar_una_cartera_sana/6489-99598
- Centro de Estudios Monetarios Latinoamericanos
Principios Básicos para una Supervisión Bancaria Efectiva
<http://www.cemla.org/pdf/pub-di-var-pbse.PDF>
- Abanlex Abogados
Seguridad de la Información en Bancos y Cajas de Ahorros
<http://www.abanlex.com/2009/09/seguridad-de-la-informacion-en-bancos-y-cajas-de-ahorros/>
- Levante – EMV

Medidas de disponibilidad de tesorería

http://www.levante-mv.com/secciones/noticia.jsp?pRef=2008052100_22_449352__Medidas-disponibilidad-tesoreria

- Sparx Systems
Un Diagrama de Modelado de Datos
<http://www.sparxsystems.com.ar/download/ayuda/index.html?adatamodeldiagram.htm>

Tesis Consultadas

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN
Tesis: Sistemas para el otorgamiento de Créditos Hipotecarios.
Autor: Montaña Arreguin Sergio
DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA
FES ARAGÓN MARZO 2010

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE COMERCIO Y ADMINISTRACIÓN SANTO TOMÁS
SEMINARIO: El crédito comercial y su importancia en los negocios internacionales,
"Análisis de los estados financieros de Brokers & Partners para un financiamiento"
TRABAJO FINAL
AUTOR: MIGUEL ÁNGEL APARICIO HERNÁNDEZ
ANA KARINA ARCE GONZÁLEZ
LUZ MAIELLA MORA MALDONADO
JAZMIN ALEJANDRA RODRÍGUEZ ROJAS
CONDUCTOR: L.C.I ANTONIO LEMUS PALACIOS
MÉXICO, D.F. ENERO 2010

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN
TESIS: Sistema Automático para el control del área administración.
AUTOR: JOSÉ FELIPE AVELAR GARCÍA
DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE BIOLOGIA
DEPARTAMENTO DE MATEMÁTICAS APLICADAS (BIOMATEMÁTICA)
TESIS DOCTORAL: Nuevas Técnicas de Modelado Orientado a Objetos e
Implementación de un Generador de Sistemas Basados en conocimiento.
AUTOR: Ma. DE LA ALMUDENA BAILADOR FERRERAS
MADRID, 2000