

ESCUELA POLITECNICA NACIONAL

ESCUELA DE FORMACION TECNOLÓGICA

**DESARROLLO DE UNA APLICACIÓN WEB DE INFORMACIÓN
FINANCIERA PARA LA CORPORACIÓN EDUCATIVA Y
CULTURAL 4 DE MARZO.**

**PROYECTO PREVIO A LA OBTENCION DEL TITULO DE TECNOLOGO EN
ANALISIS DE SISTEMAS INFORMATICOS**

**MARCIA VERONICA DIAZ BRAVO
MAYRITA MARIBEL GUASTAY LEMA**

DIRECTORA: ING. ROSA NAVARRETE

Quito, septiembre 2006

DECLARACIÓN

Yo, Mayrita Maribel Guastay Lema, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

**Mayrita Maribel
Guastay Lema**

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Marcia Verónica Díaz Bravo y Mayrita Maribel Guastay Lema, bajo mi supervisión.

Ing. Rosa Navarrete
DIRECTORA DE PROYECTO

AGRADECIMIENTO

*A mis padres quienes me
infundieron humildad y ética que
guían mi transitar por la vida.*

A mis hermanos por confiar en mí.

*También agradezco a Dios por
haberme puesto en un hogar
maravilloso al nacer.*

Mayrita

DEDICATORIA

*A mi hermano José Luis, que me
apoyo en cada momento y me supo
estimular para el éxito de mi
carrera profesional.*

Mayrita

TABLA DE CONTENIDO

CAPÍTULO I: INTRODUCCIÓN	8
CAPITULO I	8
INTRODUCCION	8
1.1. ÁMBITO	8
1.2. DESCRIPCIÓN DEL PROBLEMA	8
1.3. OBJETIVO GENERAL	10
1.4. OBJETIVOS ESPECÍFICOS.....	10
1.5. ALCANCE Y LIMITACIONES	11
1.5.1. ALCANCE.....	11
1.5.2. LIMITACIONES.....	11
1.6. ANÁLISIS DE COSTOS.....	12
1.6.1. RECURSOS HUMANOS	12
1.6.2. COSTOS DE HARDWARE	12
1.6.3. COSTOS DE SOFTWARE	12
1.6.4. COSTOS INDIRECTOS.....	13
1.6.5. COSTO TOTAL DE LA APLICACIÓN WEB.....	13
CAPÍTULO II	14
MARCO TEÓRICO	14
2.1. INGENIERIA WEB	14
2.2. ARQUITECTURA WEB	15
2.2.1. PHP	17
2.2.2. JAVASCRIPT.....	18
2.2.2.1. Características De Javascript	18
2.2.3. XAMPP	19
2.2.4. MYSQL	19
2.3. HERRAMIENTAS DE APOYO.....	20
2.3.1. MACROMEDIA DREAMWEAVER.....	20
2.3.2. MACROMEDIA FLASH.....	20
2.3.3. MACROMEDIA FIREWORKS.....	21
2.4. SUGERENCIAS DE DISEÑO WEB.....	21
CAPÍTULO III	23
ASPECTOS METODOLÓGICOS	23
3.1. ASPECTOS TEÓRICOS.....	23
3.1.1. PARADIGMA EN ESPIRAL ORIENTADO A LA WEB.....	23

3.1.2. METODOLOGÍA OOHDM	25
3.1.2.1. Etapa 1: Esquema Conceptual	26
3.1.2.2. Etapa 2: Diseño Navegacional.....	26
3.1.2.3. Etapa 3: Diseño de Interfaces Abstractas.....	27
3.1.2.4. Implementación.....	27
3.2. UML (Unified Modeling Language)	27
3.2.1. UTILIDAD DE UML	28
3.2.2. DIAGRAMAS PARA MODELAMIENTO.....	29
3.2.2.1. Diagramas de Caso de Uso	29
3.2.2.2. Diagrama de Clases	31
3.2.2.3. Diagramas de Interacción	35
3.2.2.4. Diagrama de Estados	38
3.2.2.5. Diagrama de Actividades	41
CAPITULO IV	45
CONCLUSIONES	45
RECOMENDACIONES.....	456
BIBLIOGRAFÍA	477
ANEXOS.....	
478	
MANUAL TÉCNICO.....	
479	
MANUAL DE USUARIO.....	
1047	
GUIA DE INSTALACIÓN.....	137

CAPITULO I INTRODUCCION

1.1. ÁMBITO

La Corporación Educativa y Cultural 4 de Marzo, ubicada en Av. Ladrón de Guevara E11-253, Escuela Politécnica Nacional, Escuela de Formación Tecnológica es una entidad privada conformada por 12 socios, la cual ofrece servicios financieros y administrativos basados en los principios fundamentales del cooperativismo, para impulsar el crecimiento socio-económico de sus clientes.

1.2. DESCRIPCIÓN DEL PROBLEMA

La Corporación Educativa y Cultural 4 de Marzo realiza el control de transacciones apoyadas en herramientas informáticas de escritorio.

Con la Herramienta de Microsoft Office Excel, ingresa la información de sus clientes, define cuentas, calcula saldos de cada cuenta por medio de formulas matemáticas, dependiendo de las transacciones realizadas. Al no tener buen control administrativo, se ha visto en la obligación de abrir una cuenta en el Banco del Pichincha en la cual los clientes depositan el dinero y el administrador registra su transacción con el comprobante de depósito del Banco del Pichincha.

Los clientes que acceden a los créditos no obtienen una información detallada de sus cuotas, pero se le establece una cuota fija y un plazo previamente definido, el cual debe cumplir con los requisitos anteriormente mencionados para comprobar su cuota cancelada.

Si no cancela a tiempo se le cobrar una multa por los días de atraso y se le aplica al total de la cuota pendiente.

El cliente puede renovar su préstamo si ha cancelado el 50% de su crédito, si accede a una renovación debe esperar un tiempo estipulado para realizar un análisis de todas sus transacciones.

Los problemas que se derivan de esta forma de operación son, entre otros:

- Dado que la información está en Excel no tiene control para accesos autorizados.
- Los cálculos matemáticos que se están realizando no son exactos, ya que se los efectúa de forma manual por lo que el proceso no es eficiente.
- Se detecta pérdida de información.
- No se cuenta con reportes de forma clara y precisa.
- No se dispone de información actualizada de los estados financieros a los clientes de manera rápida.
- Comprobantes de pago de cuotas mensuales no son oportunos ni precisos.

1.3. OBJETIVO GENERAL

Mejorar la atención y la calidad de los servicios financieros que presta la Corporación Educativa y Cultural 4 de marzo, a través del desarrollo y posterior implementación de una aplicación web.

1.4. OBJETIVOS ESPECÍFICOS

- a. Desarrollar un módulo de CMS (Content Management System), para administrar la interfaz.
- b. Establecer perfiles de acceso a la aplicación.
- c. Parametrizar la información sujeta a cambios como (tasa de interés, plazo, etc.)
- d. Administrar información personalizadas de cada de los usuarios y clientes que participarán en nuestra aplicación.
- e. Administrar de forma detallada los movimientos transaccionales realizadas por los clientes como depósitos, retiros, intereses ganados, etc.
- f. Permitir la administración de préstamos y soportar el análisis de plan de pagos de cliente que deseen acceder a este crédito.
- g. Proporcionar al administrador una interfaz amigable, el cual facilite la información necesaria para realizar de cálculos de amortizaciones para posibles créditos, e información general de saldos de cuenta.
- h. Permitir reportes de los estados financieros de cada uno de los clientes.

1.5. ALCANCE Y LIMITACIONES

1.5.1. ALCANCE

La aplicación Web a desarrollarse permitirá:

- Actualizar información personal de los clientes o usuarios.
- Controlar el acceso a la aplicación a través de perfiles de usuarios que establecerá el administrador.
- Compartir detalles de informes en diferentes sucursales ya que al ser una aplicación publicada en la intranet puede ser utilizado por diferentes usuarios.
- Controlar los movimientos transaccionales de cada uno de los clientes.
- Generar reportes o consultas de los datos almacenados que requiera el usuario de forma ágil.

1.5.2. LIMITACIONES

No hay interfaz para Contabilidad.

1.6. ANÁLISIS DE COSTOS

1.6.1. RECURSOS HUMANOS

RECURSOS HUMANOS	TIEMPO	HORAS	VALOR HORA	VALOR TOTAL
Análisis	2 meses	194	\$ 5.00	\$ 970
Desarrolladores	4 meses	318	\$ 3.50	\$ 1113

1.6.2. COSTOS DE HARDWARE

HADWARE	VALOR TOTAL
Computador CPU Intel Pentium IV 2.8 Ghz Disco duro 80Gb Memoria 256 Mb Procesador 2.8G CD-writer 52x24x52 Teclado, mouse Ps2 Monitor 15" Impresora Kyocera Mita 1020 DFX	\$ 750

1.6.3. COSTOS DE SOFTWARE

SOFTWARE	VALOR TOTAL
Rational Rose	\$ 620
Macromedia Dreamweaver	\$ 400
Macromedia Fireworks MX 2004	\$ 300
Macromedia Flash MX 2004	\$ 400
MySQL	SIN COSTO
PHP	SIN COSTO

Servidor XAMPP	SIN COSTO
Internet	\$ 50

1.6.4. COSTOS INDIRECTOS

GASTOS VARIOS	VALOR TOTAL
Hojas de papel bond	
Diskettes y Cds	
Tonner de impresoras	
Servicios Básicos (agua, luz, teléfono)	
Investigación (Internet. Libros, copias)	
Transporte	\$ 300

1.6.5. COSTO TOTAL DE LA APLICACIÓN WEB

Detalle	Costo
Recursos Humanos	\$ 2083
Costos de Hardware	\$ 750
Costos de Software	\$ 1770
Costos Indirectos	\$ 300
Total :	\$ 4093

CAPÍTULO II MARCO TEÓRICO

2.1. INGENIERIA WEB

La ingeniería web esta relacionada con el establecimiento y utilización de principios científicos, de ingeniería y de gestión, y con enfoques sistemáticos y disciplinarios del éxito del desarrollo, empleo, mantenimiento de sistemas y aplicaciones basados en web de alta calidad. (Pressman, 522).

Los beneficios del uso de nuestra **Ingeniería Web** son enormes por que se asegura la correcta ejecución y terminación de cada proyecto. Usted se asegura que sus requerimientos están plasmados en el proyecto por que los analiza conjuntamente con nosotros así como los objetivos a alcanzar, desde el mensaje a transmitir en el sitio web hasta las posiciones a alcanzar en los motores de búsqueda.

El procedimiento de **Ingeniería Web** contempla 6 fases diferentes en el desarrollo de un proyecto: Objetivos, Planificación, Análisis, diseño de la solución, programación web y por último test y aceptación final.

- En la fase de Objetivos, se establecen las metas a alcanzar y que espera obtener. En Planificación se establece el calendario de trabajo, la evaluación de riesgos y costes, y las fechas de entrega.
- El análisis estudia los requerimientos.
- En la fase de diseño, se trabaja por un lado en los contenidos que van a formar parte y por otro en la arquitectura y navegabilidad de la web y como punto importante el diseño de la interfaz de usuario. Este es uno de los puntos más importantes del procedimiento de **Ingeniería Web**, conseguir una buena navegabilidad o usabilidad para los usuarios es sinónimo de aceptación de la web.
- Por último los test servirán para detectar errores cometidos en contenido, funcionamiento y rendimiento.

2.2. ARQUITECTURA WEB

Una arquitectura de dos capas de una aplicación cliente/servidor consiste en capa de lógica, presentación y otra capa de base de datos.

La primera tiene que ver con presentar al usuario conjuntos y objetos visuales y llevar a cabo el procesamiento que requieren los datos producidos por el usuario y los devueltos por el servidor.

Los datos se pueden almacenar en una base de datos convencional, en un archivo simple o pueden ser incluso los datos que están en la memoria.

Esta capa reside en el servidor.

La arquitectura de dos capas se utiliza cuando requieren mucho procesamiento de datos. La arquitectura del servidor de navegador web es un buen ejemplo de una arquitectura de dos capas ya que el navegador del cliente reside en la capa lógica y presentación mientras que los datos del servidor web, las páginas web residen en la capa de la base de datos.

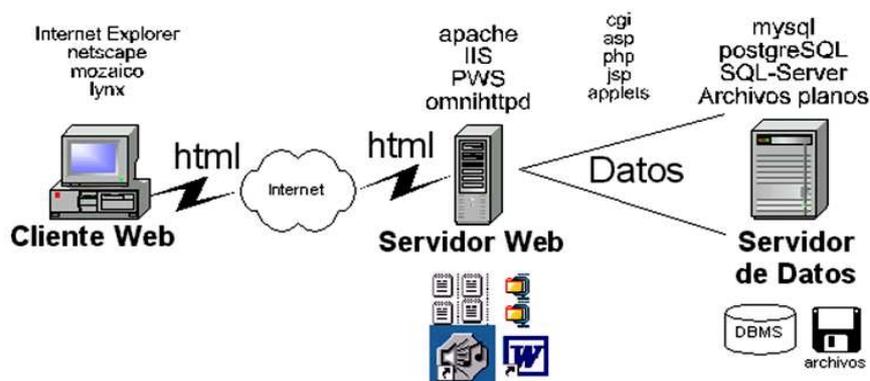


Figura 2.1: Arquitectura Web de tres niveles.

Una arquitectura de tres capas figura 2.1 se compone de una capa de presentación, una capa de procesamiento y una capa de base de datos.

La capa de presentación es la responsable de la presentación visual de la aplicación, la capa de la base de datos contiene los datos de la aplicación y la capa de procesamiento es la responsable del procesamiento que tiene lugar en la aplicación.

La capa de la base de datos esta compuesta de los archivos que contienen los datos de la aplicación.

La capa intermedia es la conlleva capacidad de mantenimiento y de reutilización. Contendrá objetos definidos por clases reutilizables que se pueden utilizar una y otra vez en otras aplicaciones.

Estos objetos se suelen llamar objetos de negocio y son los que contienen la gama normal de constructores, métodos para establecer y obtener variables, métodos que llevan a cabo cálculos y métodos, normalmente privados, en comunicación con la capa de la base de datos. (Pressman, 497).

Nivel de Aplicación (CLIENTE)

El cliente web ejecutará un navegador que interactúe con el sistema y que principalmente lleve a cabo funciones de navegados.

Nivel de Dominio de Aplicación (SERVIDOR)

Este servidor contendrá todas las páginas web a las que el cliente ira accediendo y se comunicara con el resto del sistema para proporcionar información tal como la disponibilidad. (Pressman, 501)

Nivel de Repositorio (SERVIDOR DE DATOS)

En realidad este nivel no ha cambiado para nada y sigue siendo la capa en donde se almacenan los datos y toda la información.

(http://www.webtaller.com/maletin/articulos/las_tecnologias_de_las_paginas_web.php)

Software para el desarrollo de la aplicación:

En el siguiente cuadro se describe el lenguaje de aplicación utilizado en cada nivel dentro de la arquitectura:

Nivel		Lenguaje
Cliente	Aplicación Cliente	HTML JavaScript
Servidor	Servidor Web	XAMPP
	Lenguaje Servidor	PHP
Servidor de Datos	Base de Datos	MySQL

2.2.1. PHP

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.... No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.



Figura 2.1.1. Proceso de ejecución de una página PHP

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

<http://www.webestilo.com/php/php00.phtml>

2.2.2. JAVASCRIPT

JavaScript es un lenguaje interpretado, al igual que VisualBasic, Perl, TCL... sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que utilizamos para viajar por ella los que interpretan (y por tanto ejecutan) los programas escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que incorporan el código fuente de un programa, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas. Los programas en JavaScript no son la primera forma que conoce la Web para transformar información, dado que el uso de CGIs está ampliamente difundido. La diferencia básica que existe entre un programa CGI y uno escrito en JavaScript es que el CGI se ejecuta en el servidor de páginas Web mientras que el programa en Javascript se ejecuta en el cliente (es decir, en el navegador). Por regla general, el CGI necesita unos datos de entrada (que normalmente se proporcionan mediante un formulario), los procesa y emite un resultado en forma de documento HTML. Esto implica tres transacciones en la red. La primera carga la página del formulario, la segunda envía los datos al servidor, y la tercera recibe la nueva página que ha generado el CGI. Por el contrario, los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados.

http://geneura.ugr.es/~victor/cursillos/curso_javascript_basico/jsb_intro.html

2.2.2.1. Características De Javascript

- Las dos principales características de JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones),
- y por otro JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas X-Windows). Esto implica que gran parte de la programación en JavaScript

se centra en describir objetos (con sus variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.

2.2.3. XAMPP

XAMPP es un paquete de desarrollo en php para windows y linux. Por defecto este paquete viene preparado para trabajar con mysql.

<http://www.codigophp.com/index.php/page/2/>

Una de las formas más fáciles y rápidas de tener Apache, MySQL, PHP y phpMyAdmin en su máquina es, sin lugar a duda, pasando por un paquete de instalación cómodo y automático como XAMPP. Ofrece un pack de instalación automática con lo que podrás alojar y servir tus páginas web desde tu máquina en local.

Además, XAMPP ofrece una colección de librerías y otras aplicaciones de gran utilidad para el manejo y administración de una página web, junto a todas las dependencias que resultan imprescindibles para ello.

Un servidor web, una base de datos MySQL, PHP, un servidor de correo electrónico, Perl y un servidor FTP son los elementos claves de esta distribución. Además incluye Apache 2 y las últimas versiones de MySQL y PHP; Apache y MySQL se instalarán como servicios. Y todo esto gracias a un asistente que automatizará todo el proceso para que sea lo más leve y rápido posible.

2.2.4. MYSQL

Mysql es un programa interactivo que permite conectarnos a un servidor MySQL, ejecutar algunas consultas, y ver los resultados. mysql puede ser usado también en modo batch: es decir, se pueden colocar toda una serie de consultas en un archivo, y posteriormente decirle a mysql que ejecute dichas consultas.

Para conectarse al servidor, usualmente necesitamos de un nombre de usuario (login) y de una contraseña (password), y si el servidor al que nos deseamos

conectar está en una máquina diferente de la nuestra, también necesitamos indicar el nombre o la dirección IP de dicho servidor. Una vez que conocemos estos tres valores, podemos conectarnos.

<http://www.programatium.com/tutoriales/cursos/mysql/mysql10.htm>

2.3. HERRAMIENTAS DE APOYO

2.3.1. MACROMEDIA DREAMWEAVER

Dreamweaver es **la herramienta de diseño de páginas web más avanzada**, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobretodo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar **páginas con aspecto profesional**, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Javascript para crear efectos e interactividades
- Inserción de archivos multimedia...

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas.

En resumen, el programa es realmente satisfactorio, incluso el **código generado es de buena calidad**. La única pega consiste en que al ser tan avanzado, puede resultar un poco difícil su manejo para personas menos experimentadas en el diseño de webs.

<http://www.desarrolloweb.com/articulos/332.php>

2.3.2. MACROMEDIA FLASH

Flash es el software de creación más avanzado para crear animación interactiva escalable para el Web Tanto si crea logotipos animados, controles de navegación

de sitios Web, animaciones de gran formato o sitios Web completos de Flash, descubrirá que la capacidad y flexibilidad de Flash es el medio ideal para desarrollar su propia creatividad.

Flash ofrece nuevas funciones para la creación de sitios Web interactivos interesantes y atractivos. Entre las nuevas funciones de reproducción se incluyen: flujos de sonido MP3, campos de texto para que el usuario pueda incluir texto editable en una película y mejoras en las acciones de Flash que permiten crear juegos, formularios y cuestionarios. Entre las nuevas funciones de creación se incluyen: un proceso de publicación fluido, una ventana de biblioteca con diseño nuevo, varios inspectores nuevos y una interfaz gráfica mejorada.

http://www.macromedia.com/go/thirdparty_es

2.3.3. MACROMEDIA FIREWORKS

El programa Fireworks MX, software que pertenece a la casa Macromedia, sirve para realizar aplicaciones gráficas de todo tipo, que potencien y hagan más atractivas nuestras páginas Web. MX identifica a la última versión de este producto, y si no tienes una copia, puedes descargar una versión de prueba que dura sólo 30 días, pero que te servirá para entender las nociones básicas del software.

Fireworks, a diferencia de otros productos gráficos como Photoshop, tiene la particularidad de trabajar con vectores y mapas de bit, combinación que lo hace indicado para realizar diseños Web. Esto se potencia gracias a que trae incorporado un sinnúmero de opciones que nos permiten crear aplicaciones bastante interesantes, como gifs animados, banners y menús desplegados, entre otras imágenes.

http://www.macromedia.com/go/thirdparty_es

2.4. SUGERENCIAS DE DISEÑO WEB

Este es el concepto de **Diseño para Todos**, que en América se conoce como Universal Design. Según esta filosofía los equipos y servicios deben ser

diseñados para satisfacer las necesidades de todos los usuarios; tanto el usuario medio como usuarios con distintos perfiles funcionales deben ser capaces de usar el producto en la medida de lo posible y con el máximo de prestaciones, sin necesidad de adaptación o diseño especializado adicional.

Los 7 principios que debe cumplir un diseño para ser un verdadero diseño para todos son los siguientes:

1. El diseño debe ser igualmente **utilizable** por cualquier tipo de usuario.
2. El diseño debe tener la suficiente **flexibilidad** para acomodar el mayor rango de preferencias o capacidades individuales.
3. Debe ser **simple e intuitivo**, fácil de comprender independientemente de la experiencia, conocimientos, idioma, o nivel de concentración del usuario,
4. El diseño **proporciona la información necesaria** para su uso al usuario independientemente de las condiciones ambientales y sus capacidades sensoriales.
5. El diseño es **resistente a los errores**. Minimiza los riesgos y las consecuencias adversas de los errores accidentales e inintencionados.
6. **Bajo esfuerzo físico**. El diseño puede ser usado eficientemente y confortablemente con un mínimo de fatiga.
7. **Ergonomía**. El entorno proporciona un espacio y condiciones adecuados para su uso con independencia del tamaño corporal, postura y movilidad del usuario.

<http://acceso.uv.es/accesibilidad/estudio/7recomendaciones.htm>

CAPÍTULO III ASPECTOS METODOLÓGICOS

3.1. ASPECTOS TEÓRICOS

3.1.1. PARADIGMA EN ESPIRAL ORIENTADO A LA WEB

Es un modelo de proceso de software evolutivo. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones. La versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez mas completas de ingeniería del sistema.

3.1.1.1. Características:

- Es también al igual que el anterior un modelo evolutivo que combina el modelo clásico con el diseño de prototipos.
- Incluye la etapa de análisis de riesgos, no incluida anteriormente.
- Es ideal para crear productos con diferentes versiones mejoradas como se hace con el software moderno de microcomputadoras.
- La ingeniería puede desarrollarse a través del ciclo de vida clásico o el de construcción de prototipos.
- Este es el enfoque más realista actualmente.

El modelo en espiral se divide en un número de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.

Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.

Planificación: las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.

Análisis de riesgos: las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.

Ingeniería: las tareas requeridas para construir una o más representaciones de la aplicación.

Construcción y adaptación: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

Evaluación el cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.

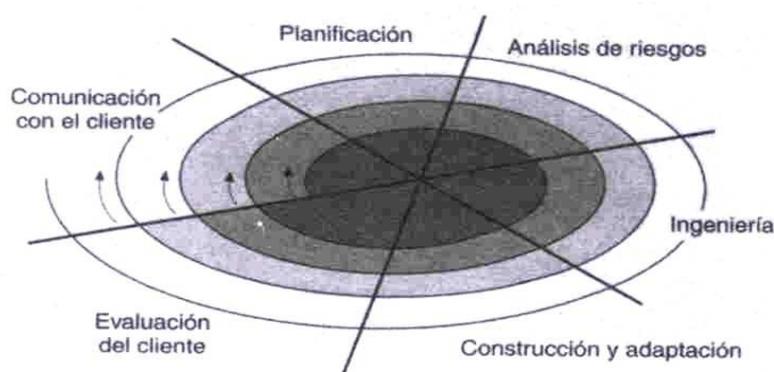


Figura. 20.1. El modelo de proceso OO

VENTAJAS:

- El modelado en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora, no termina cuando se entrega el software.
- Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- Demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto.
- Reduce los riesgos antes de que se conviertan en problemáticos.

3.1.2. METODOLOGÍA OOHDM

Paradigma	Espiral	Metodología	Métodos
Orientado a la Web		OOHDM	
Formulación de requerimientos			Especificación de actores Diagramas de Casos de Uso Diccionario de Casos de Uso
Planificación			
Análisis		Modelo Estático	Diagrama de Clases Diccionario de Clases
		Modelo Dinámico	Diagrama de Interacción - Diagrama de Secuencia - Diagrama de Colaboración Diagrama de Estados Diagrama de Actividades
		Modelo Arquitectónico.	Diagrama Arquitectónico Diagrama Navegacional
Ingeniería			Código Fuente

El modelo OOHDM u *Object Oriented Hypermedia Design Methodology*, para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa y es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web.

Al igual que RMM, este método se inspira en el modelo HDM, pero lo que le distingue claramente del primero es el proceso de concepción orientado a objetos.

OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 4 etapas:

- Diseño Conceptual
- Diseño Navegacional
- Diseño de Interfaces Abstractas
- Implementación

Cada etapa de la concepción define un esquema objeto específico en el que se introducen nuevos elementos (clases).

3.1.2.1. Etapa 1: Esquema Conceptual

En la **primera etapa** se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). El modelo OOHDM propone como esquema conceptual basado en **clases, relaciones y subsistemas**.

3.1.2.2. Etapa 2: Diseño Navegacional

Una vez identificados los componentes (páginas, guiones, applets y otras funciones de proceso) de la arquitectura, el diseñador deberá definir las rutas de navegación que permita al usuario acceder al contenido y a los servicios de la WebApp.

Para que el diseñador pueda llevar acabo debe identificar una semántica de la navegación para diferentes usuarios del sitio, y definir la mecánica (sintaxis) para lograr la navegación.

Generalmente una WebApp grande tendrá una variedad de roles de usuarios diferentes. Por ejemplo, los roles podrían ser el de visitante, cliente registrado o cliente privilegiado. Cada uno de estos roles se pueden asociar a diferentes contenidos y de servicios diferentes. (Pressman, 530).

3.1.2.3. Etapa 3: Diseño de Interfaces Abstractas

Los conceptos, principios y métodos de diseños de interfaz son todos aplicables al diseño de interfaz de usuario. Sin embargo, las características especiales de los sistemas de aplicaciones web requieren otras consideraciones adicionales.

La interfaz de usuario es la primera impresión. Independientemente del valor del contenido, la sofisticación de las capacidades, los servicios de procesamiento y el beneficio global, una interfaz con un diseño pobre decepcionara al usuario potencial.

Debemos tomar en cuenta las siguientes características básicas para el diseño de una interfaz en la web.

- La velocidad de lectura del monitor de una computadora es aproximadamente un 25 por 100 más lento que leer una copia impresa.
- Evite los símbolos <<bajo construcción>> levantar expectación y provocan un enlace innecesario que seguramente sea decepcionante.
- Los usuarios prefieren no tener que correr la pantalla.
- Los menús de navegación y las barras de cabecera se deberán diseñar consecuentemente y deberán estar disponibles en todas las páginas a las que el usuario tenga acceso.
- La estética nunca deberá sustituir la funcionalidad.

(Pressman, 531).

3.1.2.4. Implementación

Por fin, la **cuarta etapa**, dedicada a la puesta en práctica, es donde se hacen corresponder los objetos de interfaz con los objetos de implementación.

3.2. UML (Unified Modeling Language)

Al final de la pasada década, Grady Booch, James Rumbaugh e Ivar Jacobson empezaron a colaborar para recopilar las mejores características de cada uno de

los métodos de diseño y análisis orientado a objetos en un método unificado. El resultado denominado Lenguaje de Modelo Unificado (UML), se ha convertido en el método más utilizado.

UML permite a un ingeniero del software expresar un modelo de análisis utilizando una notación de modelado con unas reglas sintácticas, semánticas y prácticas. (Pressman, 363).

Entre los diagramas de UML podemos detallar los siguientes:

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

3.2.1. UTILIDAD DE UML

UML reescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros.

3.2.2. DIAGRAMAS PARA MODELAMIENTO

3.2.2.1. Diagramas de Caso de Uso

Introducción

Los casos de uso modelan el sistema desde el punto de vista del usuario. Creados durante la obtención de requisitos, los casos de uso deben cumplir los siguientes objetivos.

- Definir los requisitos funcionales y operativos del sistema, diseñando un escenario de uso acordado por el usuario final, y el equipo de desarrollo; proporcionar una descripción clara y sin ambigüedad de cómo el usuario final interactúa con el sistema y viceversa, y proporcionar una base para la validación de las pruebas.

Utilizando UML se puede crear una presentación visual de los casos de uso llamada diagrama de casos de usos. Como otros elementos del análisis, los casos de uso pueden representarse a diferentes niveles de abstracción.

Los diagramas de casos de uso contienen casos de uso y actores, siendo estos últimos las entidades que interactúan con el sistema. Pueden ser humanos u otras máquinas o sistemas que tengan definidas interfaces con nuestro sistema. (Pressman, 367).

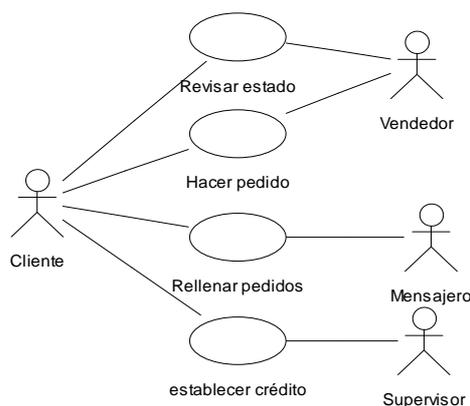


Figura 21.2. Diagramas de casos de uso
(Pressman, 368)

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

Elementos

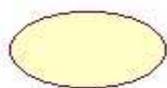
Actor:



Una definición previa, es que un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe de Local.

Caso de Uso:



Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Relaciones:

Asociación

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

Dependencia o Instanciación

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de **usar** o **heredar**.

3.2.2.2. Diagrama de Clases

El Diagrama de Clases es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema el primer

elemento de un diagrama de clases es una descripción de un diagrama de clases individual. (Pressman, 393)

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones. El mundo real puede ser visto desde abstracciones diferentes (subjectividad)

Mecanismos de abstracción:

1. Clasificación / Instanciación
2. Composición / Descomposición
3. Agrupación / Individualización
4. Especialización / Generalización

La clasificación es uno de los mecanismos de abstracción más utilizados. La clase define el ámbito de definición de un conjunto de objetos, y cada objeto pertenece a una clase, Los objetos se crean por instanciación de las clases.

Cada clase se representa en un rectángulo con tres compartimientos:

- nombre de la clase
- atributos de la clase
- operaciones de la clase

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para los elementos que son:

- (-) Privado : es el más fuerte. Esta parte es totalmente invisible (excepto para clases friends en terminología C++)
- (#) Los atributos/operaciones protegidos están visibles para las clases friends y para las clases derivadas de la original.
- (+) Los atributos/operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)

Relaciones entre clases:

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

- Asociación y Agregación (vista como un caso particular de asociación)
- Generalización/Especialización.

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Asociación:

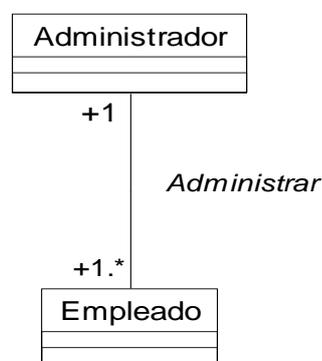


Figura 22.15. Un ejemplo de asociación

(Pressman, 395)

La asociación expresa una conexión bidireccional entre objetos. Una asociación es una abstracción de la relación existente en los enlaces entre los objetos. Puede determinarse por la especificación de multiplicidad (mínima...máxima)

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- Cero o muchos
- 0..* Cero o muchos
- 1..* Uno o muchos (al menos uno)

Agregación:

La agregación representa una relación parte de entre objetos. En UML se proporciona una escasa caracterización de la agregación. Esta relación puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes.

Una agregación se podría caracterizar según:

Puede el objeto parte comunicarse directamente con objetos externos al objeto agregado?

No => inclusiva

Si => no inclusiva

Puede cambiar La composición del objeto agregado?

Si => dinámica

No => estática

Diagrama de Clases y Diagramas de Objetos pertenecen a dos vistas complementarias del modelo. Un Diagrama de Clases muestra la abstracción de una parte del dominio. Un Diagrama de Objetos representa una situación concreta del dominio. Las clases abstractas no son instanciadas.

Generalización:

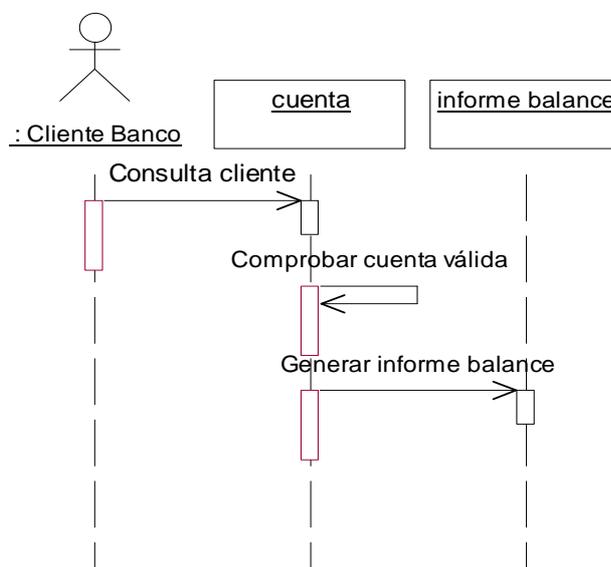
Permite gestionar la complejidad mediante un ordenamiento taxonómico de clases, se obtiene usando los mecanismos de abstracción de Generalización y/o Especialización. La Generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclases heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas. La Generalización y Especialización son equivalentes en cuanto al resultado: la jerarquía y herencia establecidas. Generalización y Especialización no son operaciones reflexivas ni simétricas pero

sí transitivas. La especialización es una técnica muy eficaz para la extensión y reutilización.

La noción de clase está próxima a la de conjunto. Dada una clase, podemos ver el conjunto relativo a las instancias que posee o bien relativo a las propiedades de la clase. Generalización y especialización expresan relaciones de inclusión entre conjuntos.

3.2.2.3. Diagramas de Interacción

3.2.2.3.1. Diagramas de Secuencia



Ejemplo 22.23. Diagrama de secuencia

(Pressman, 397)

La vista de interacción describe secuencias de intercambios de mensajes entre los roles que implementan el comportamiento de un sistema. Un rol clasificador, o simplemente "un rol", es la descripción de un objeto, que desempeña un determinado papel dentro de una interacción, distinto de los otros objetos de la misma clase. Esta visión proporciona una vista integral del comportamiento del sistema, es decir, muestra el flujo de control a través de muchos objetos. La vista de interacción se exhibe en dos diagramas centrados en distintos aspectos pero

complementarios: centrados en los objetos individuales y centrados en objetos cooperantes.

Los objetos interactúan para realizar colectivamente los servicios ofrecidos por las aplicaciones. Los diagramas de interacción muestran cómo se comunican los objetos en una interacción. Existen dos tipos de diagramas de interacción: el Diagrama de Colaboración y el Diagrama de Secuencia. El Diagrama de Secuencia es más adecuado para observar la perspectiva cronológica de las interacciones, muestra la secuencia explícita de mensajes y son mejores para especificaciones de tiempo real y para escenarios complejos.

3.2.2.3.2. Diagramas de Colaboración

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

Un diagrama de colaboración es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de sólo clasificadores y asociaciones. Los roles de clasificador y los de asociación describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la colaboración. Cuando se instancia una colaboración, los objetos están ligados a los roles de clasificador y los enlaces a los roles de asociación. El rol de asociación puede ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimiento o variables locales del procedimiento. Los símbolos de enlace pueden llevar estereotipos para indicar enlaces temporales.

Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. Cuando se implementa el comportamiento, la secuencia de los mensajes corresponde a la estructura de

llamadas anidadas y el paso de señales del programa. Un diagrama de secuencia muestra secuencias en el tiempo como dimensión geométrica, pero las relaciones son implícitas. Un diagrama de colaboración muestra relaciones entre roles geoméricamente y relaciona los mensajes con las relaciones, pero las secuencias temporales están menos claras.

Es útil marcar los objetos en cuatro grupos: los que existen con la interacción entera; los creados durante la interacción (restricción {new}); los destruidos durante la interacción (restricción {destroyed}); y los que se crean y se destruyen durante la interacción (restricción {transient}).

Aunque las colaboraciones muestran directamente la implementación de una operación, pueden también mostrar la realización de una clase entera. En este uso, muestran el contexto necesario para implementar todas las operaciones de una clase. Esto permite que el modelador vea los roles múltiples que los objetos pueden desempeñar en varias operaciones.

Mensajes

Los mensajes se muestran como flechas etiquetadas unidas a los enlaces. Cada mensaje tiene un número de secuencia, una lista opcional de mensajes precedentes, una condición opcional de guarda, un nombre y una lista de argumentos y un nombre de valor de retorno opcional. El nombre de serie incluye el nombre (opcional) de un hilo. Todos los mensajes del mismo hilo se ordenan secuencialmente. Los mensajes de diversos hilos son concurrentes a menos que haya una dependencia secuencial explícita.

Flujos

Generalmente, un diagrama de colaboración contiene un símbolo para un objeto durante una operación completa. Sin embargo, a veces, un objeto contiene diferentes estados que se deban hacer explícitos. Por ejemplo, un objeto pudo cambiar de localización o sus asociaciones pudieron diferenciarse. Los diferentes símbolos de objeto que representan un objeto se pueden conectar usando flujos "become" o "conversión". Un flujo "become" es una transición, a

partir de un estado de un objeto a otro. Se dibuja como una flecha de línea discontinua con el estereotipo "become" o "conversión" y puede ser etiquetado con un número de serie para mostrar cuando ocurre. Un flujo de conversión también se utiliza para mostrar la migración de un objeto a partir de una localización a otra distinta.

3.2.2.4. Diagrama de Estados

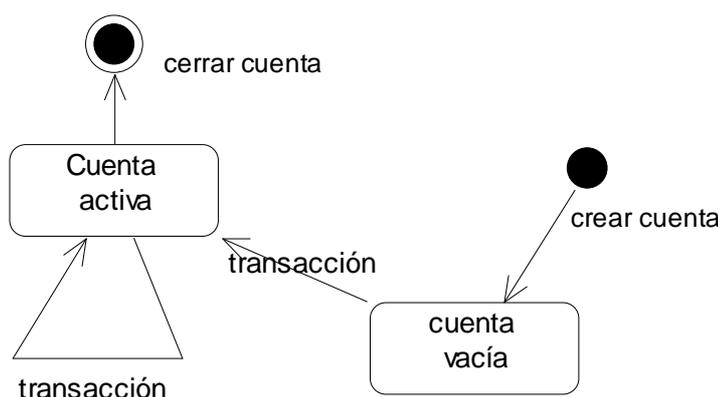


Figura 22.24. Diagrama de estados

(Pressman, 397)

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Los Diagramas de Estado representan autómatas de estados finitos, desde el p.d.v. de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El estado está caracterizado parcialmente por los valores algunos de los atributos del objeto. El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase. Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos, son grafos dirigidos y deterministas. La transición entre estados es instantánea y se debe a la ocurrencia de un evento.

Estado

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente).

Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de verdadero o falso
- Recepción de una señal de otro objeto en el modelo
- Recepción de un mensaje
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

Envío de mensajes

Además de mostrar y transición de estados por medio de eventos, puede representarse el momento en el cual se envían mensajes a otros objetos. Esto se realiza mediante una línea punteada dirigida al diagrama de estados del objeto receptor del mensaje.

Transición simple

Una transición simple es una relación entre dos estados que indica que un objeto en el primer estado puede entrar al segundo estado y ejecutar ciertas operaciones, cuando un evento ocurre y si ciertas condiciones son satisfechas. Se

representa como una línea sólida entre dos estados, que puede venir acompañada de un texto con el siguiente formato:

event-signature "[" **guard-condition**] "/" **action-expression** "^"**send-clause**

event-signature es la descripción del evento que da lugar la transición, **guard-condition** son las condiciones adicionales al evento necesarias para que la transición ocurra, **action-expression** es un mensaje al objeto o a otro objeto que se ejecuta como resultado de la transición y el cambio de estado y **send-clause** son acciones adicionales que se ejecutan con el cambio de estado, por ejemplo, el envío de eventos a otros paquetes o clases.

Transición interna

Es una transición que permanece en el mismo estado, en vez de involucrar dos estados distintos. Representa un evento que no causa cambio de estado. Se denota como una cadena adicional en el compartimiento de acciones del estado.

Acciones:

Podemos especificar la solicitud de un servicio a otro objeto como consecuencia de la transición. Se puede especificar el ejecutar una acción como consecuencia de entrar, salir, estar en un estado, o por la ocurrencia de un evento.

Subestados

Un estado puede descomponerse en subestados, con transiciones entre ellos y conexiones al nivel superior. Las conexiones se ven al nivel inferior como estados de inicio o fin, los cuales se suponen conectados a las entradas y salidas del nivel inmediatamente superior.

Transacción Compleja

Una transición compleja relaciona tres o más estados en una transición de múltiples fuentes y/o múltiples destinos. Representa la subdivisión en threads del

control del objeto o una sincronización. Se representa como una línea vertical de la cual salen o entran varias líneas de transición de estado.

Transición a estados anidados

Una transición de hacia un estado complejo (descrito mediante estados anidados) significa la entrada al estado inicial del subdiagrama. Las transiciones que salen del estado complejo se entienden como transiciones desde cada uno de los subestados hacia afuera (a cualquier nivel de profundidad).

Transiciones temporizadas

- Las esperas son actividades que tienen asociada cierta duración.
- La actividad de espera se interrumpe cuando el evento esperado tiene lugar.
- Este evento desencadena una transición que permite salir del estado que alberga la actividad de espera. El flujo de control se transmite entonces a otro estado.

3.2.2.5. Diagrama de Actividades

El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:

- Un método
- Un caso de uso
- Un proceso de negocio (Workflow)

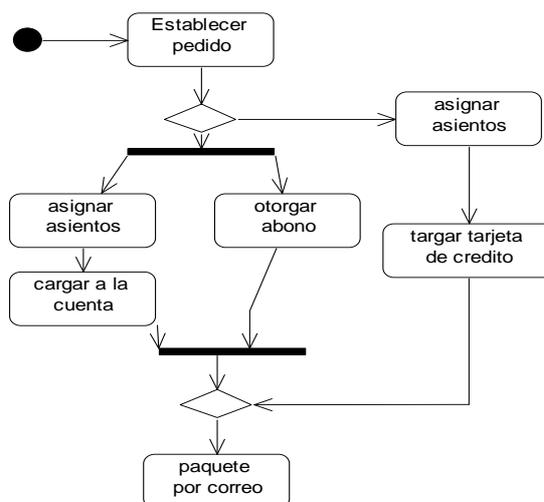


Figura 7.1 Diagrama de Actividades

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad.

Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

Un grafo de actividades contiene estados de actividad que representa la ejecución de una secuencia en un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En vez de esperar un evento, como en un estado de espera normal, un estado de actividad espera la terminación de su cómputo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de actividad dentro del diagrama. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente. Los estados de

actividad no tienen transiciones con eventos explícitos, pero pueden ser abortados por transiciones en estados que los incluyen.

Un grafo de actividades puede contener también estados de acción, que son similares a los de actividad pero son atómicos y no permiten transiciones mientras están activos. Los estados de acción se deben utilizar para las operaciones cortas de mantenimiento.

Un diagrama de actividades puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas. La concurrencia se representa a partir de la agregación, en la cual cada objeto tiene su propio hilo. Las actividades concurrentes se pueden realizar simultáneamente o en cualquier orden. Un diagrama de actividades es como un organigrama tradicional, excepto que permite el control de concurrencia además del control secuencial.

Notación

Un estado de actividad se representa como una caja con los extremos redondeados que contiene una descripción de actividad. Las transacciones simples de terminación se muestran como flechas. Las ramas se muestran como condiciones de guarda en transiciones o como diamantes con múltiples flechas de salida etiquetadas. Una división o una unión de control se representan con múltiples flechas que entran o salen de la barra gruesa de sincronización. Cuando es necesario incluir eventos externos, la recepción de un evento se puede mostrar como un disparador en una transición, o como un símbolo especial que denota la espera de una señal.

A menudo es útil organizar las actividades en un modelo según su responsabilidad. Esta clase de asignación puede mostrarse organizando las actividades en regiones distintas separados por líneas en el diagrama. Debido a su aspecto, esto es conocido como **Calles**.

Un diagrama de actividades puede mostrar el flujo de objetos como valores. Para un valor de salida, se dibuja una flecha con línea discontinua desde la actividad al objeto. Para un valor de entrada, se dibuja una flecha con línea discontinua desde el objeto a una actividad.

CAPITULO IV CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Esta aplicación Web es un software libre por lo que no necesita comprar licencias ya que esta es una gran inversión para la institución que utilizará la aplicación.

- Debido a las ventajas que presenta las herramientas de Macromedia: Dreamweaver, Flash, Fireworks, ha facilitado el diseño para una interfaz amigable.

- El servidor Xampp, ayudado a interactuar con MySQL base de datos y el CMS de manera rápida y sencilla.

4.2. RECOMENDACIONES

- La persona que administre la aplicación debe estar actualizado con las tasas de interés, ya que estas variables cambian constantemente, de acuerdo al reglamento de Corporaciones que establece el Banco Central junto con la Superintendencia de Bancos.

- El administrador debe tener cuidado al momento de administrar los movimientos transaccionales tanto de cuentas como préstamos, ya que maneja el crecimiento socio-económico de sus clientes.

BIBLIOGRAFÍA

LIBROS

Pressman, Roger S, Ingeniería de Software, Quinta Edición, Año 1998

INTERNET

1. (http://www.webtaller.com/maletin/articulos/las_tecnologias_de_las_paginas_web.php)
2. <http://www.webestilo.com/php/php00.phtml>
3. http://geneura.ugr.es/~victor/cursillos/curso_javascript_basico/jsb_intro.html
4. <http://www.codigophp.com/index.php/page/2/>
5. <http://www.programatium.com/tutoriales/cursos/mysql/mysql10.htm>
6. http://www.macromedia.com/go/thirdparty_es
7. <http://www.desarrolloweb.com/articulos/332.php>
8. <http://acceso.uv.es/accesibilidad/estudio/7recomendaciones.htm>

ANEXOS

MANUAL TECNICO

CONTENIDO

1. ANÁLISIS Y REQUERIMIENTOS

1.1. MODELO ESTÁTICO

1.1.1. IDENTIFICACIÓN DE ACTORES

1.1.2. DIAGRAMAS DE CASOS DE USO

1.1.3. DICCIONARIO DE CASOS DE USO

1.1.4. DIAGRAMAS DE CLASES

1.1.5. DIAGRAMAS DE OBJETOS

2. DISEÑO

2.1. MODELO DINÁMICO

2.1.1. DIAGRAMAS DE ITERACIÓN

2.2. MODELO FUNCIONAL

2.2.1. DIAGRAMA DE ESTADOS

2.2.2. DIAGRAMA DE ACTIVIDADES

2.3. MODELO ARQUITECTÓNICO

2.3.1. DIAGRAMA ARQUITECTÓNICO

2.3.2. DIAGRAMA NAVEGACIONAL

ANALISIS Y REQUERIMIENTOS

MODELO ESTÁTICO

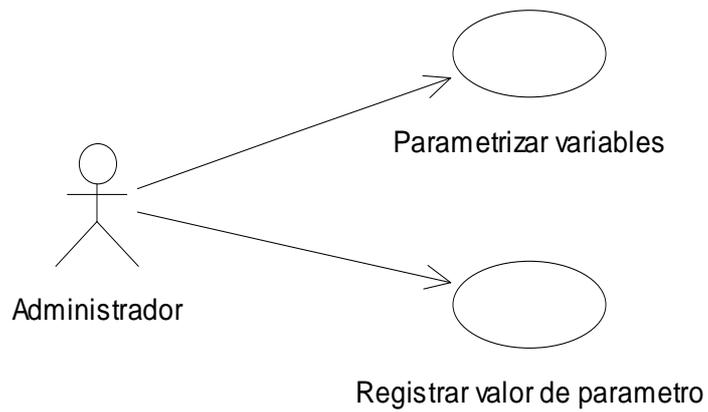
1.1.1. IDENTIFICACIÓN DE ACTORES

ACTORES	FUNCIÓN
 <p data-bbox="320 725 520 757">Administrador</p>	<ul style="list-style-type: none"> <li data-bbox="676 506 1267 539">➤ Ingresar datos personales de clientes. <li data-bbox="676 562 1182 595">➤ Administrar perfiles de usuarios. <li data-bbox="676 618 1054 651">➤ Administrar sucursales. <li data-bbox="676 674 1326 707">➤ Registrar movimientos (depósitos, retiros). <li data-bbox="676 730 1023 763">➤ Registrar préstamos. <li data-bbox="676 786 1091 819">➤ Registrar cuotas de cobro <li data-bbox="676 842 1050 875">➤ Parametrizar variables. <li data-bbox="676 898 1134 931">➤ Emitir reportes de préstamos <li data-bbox="676 954 1187 987">➤ Emitir reportes de cliente activos <li data-bbox="676 1010 1182 1043">➤ Emitir reportes de transacciones
 <p data-bbox="357 1330 456 1361">Cajero</p>	<ul style="list-style-type: none"> <li data-bbox="676 1167 1267 1200">➤ Ingresar datos personales de clientes. <li data-bbox="676 1223 1091 1256">➤ Administrar transacciones <li data-bbox="676 1279 1078 1312">➤ Consultar plan de pagos. <li data-bbox="676 1335 1046 1368">➤ Generar reporte diario.
 <p data-bbox="368 1563 467 1594">Cliente</p>	<ul style="list-style-type: none"> <li data-bbox="676 1480 1107 1514">➤ Entregar datos personales. <li data-bbox="676 1536 1283 1570">➤ Realiza depósitos de cuenta de ahorro. <li data-bbox="676 1592 1126 1626">➤ Entrega pago de préstamos.

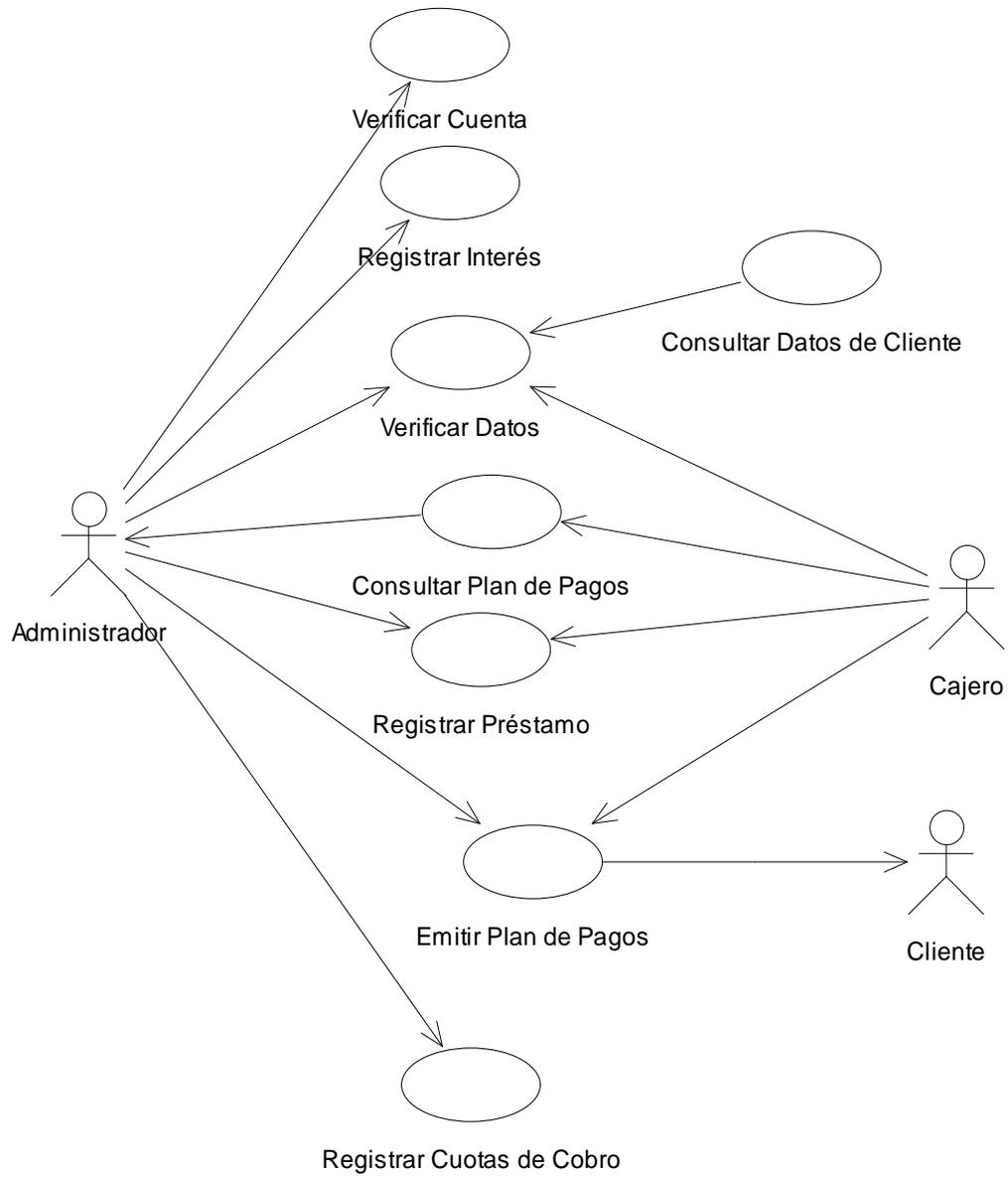
1.1.2. DIAGRAMAS DE CASOS DE USO

DIAGRAMAS DE CASOS DE USO POR ACTIVIDADES

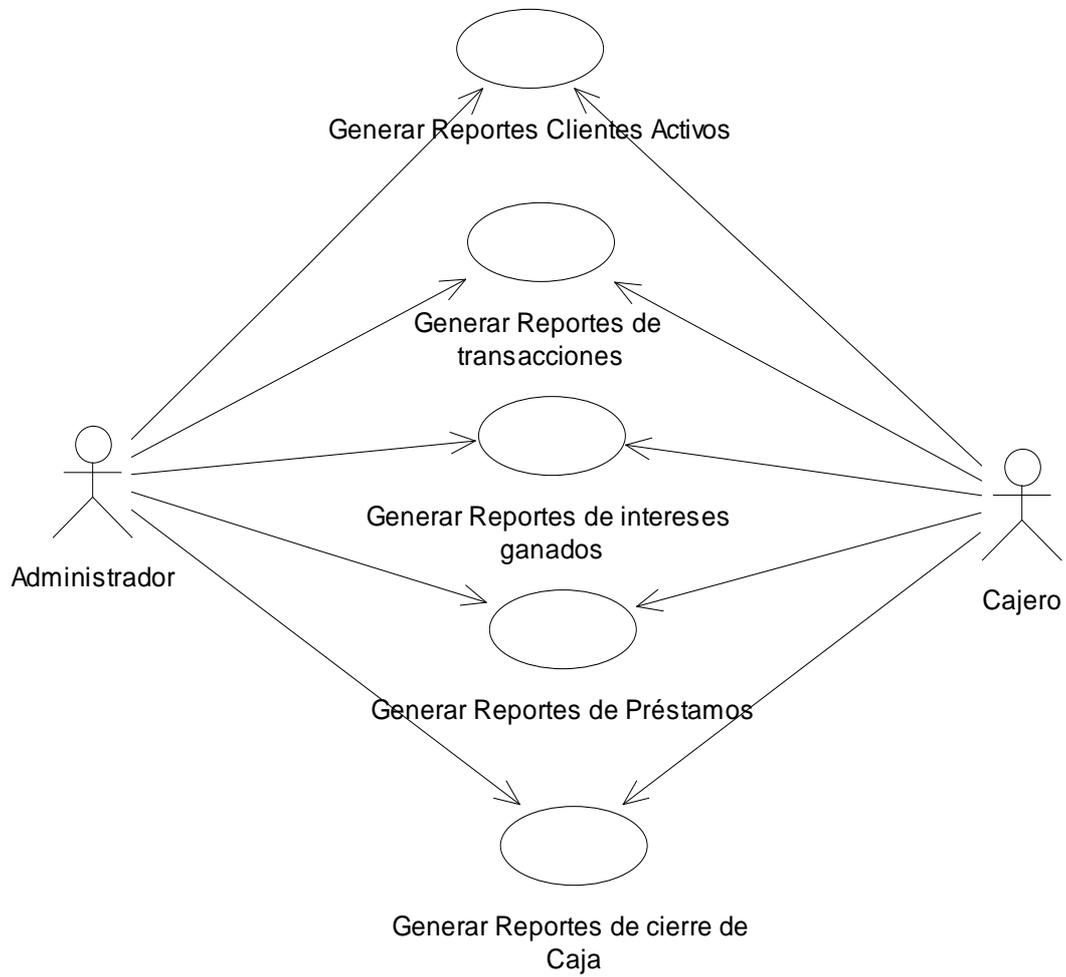
1. REGISTRAR PARAMETROS



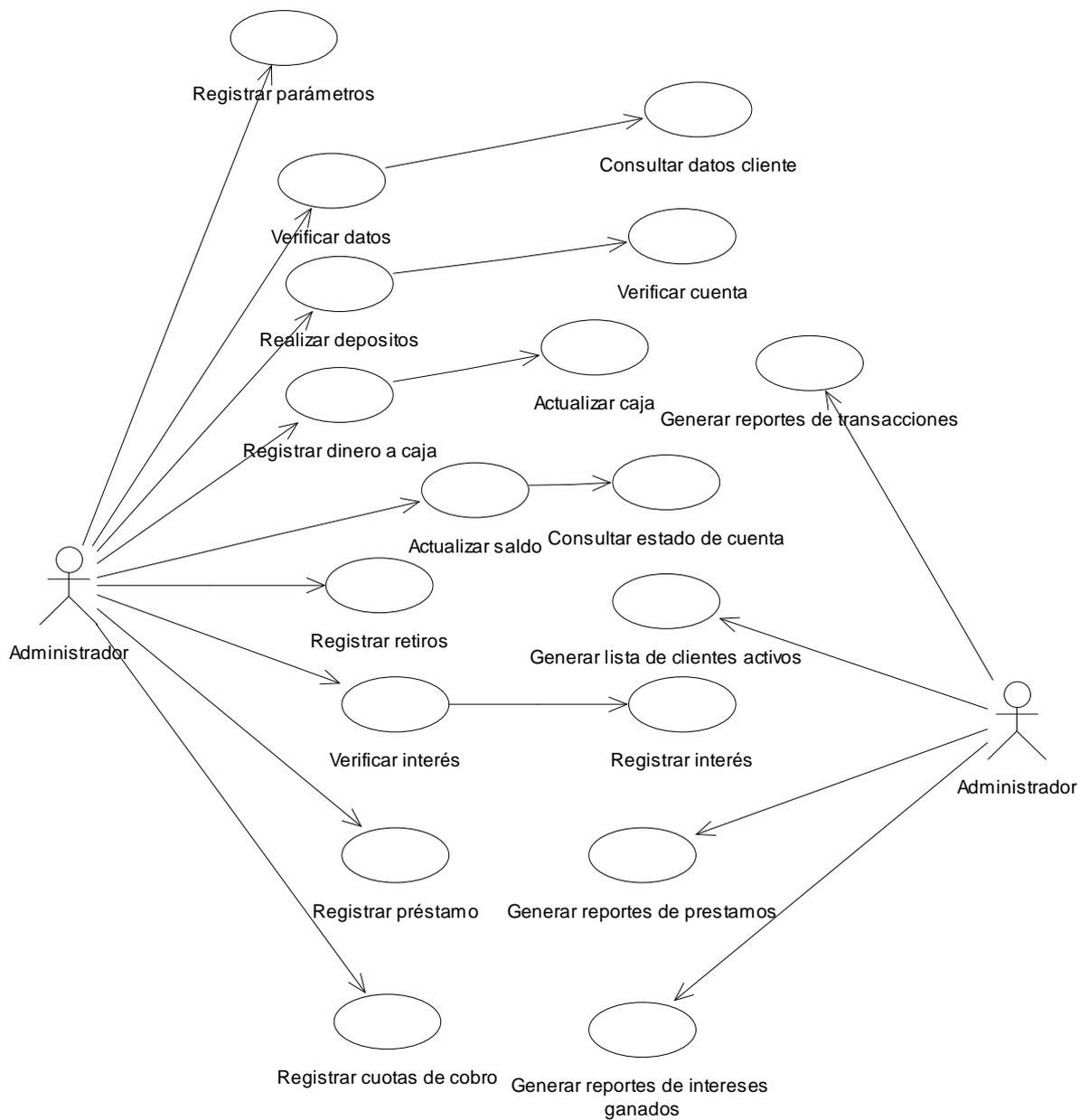
3. ADMINISTRAR PRESTAMOS

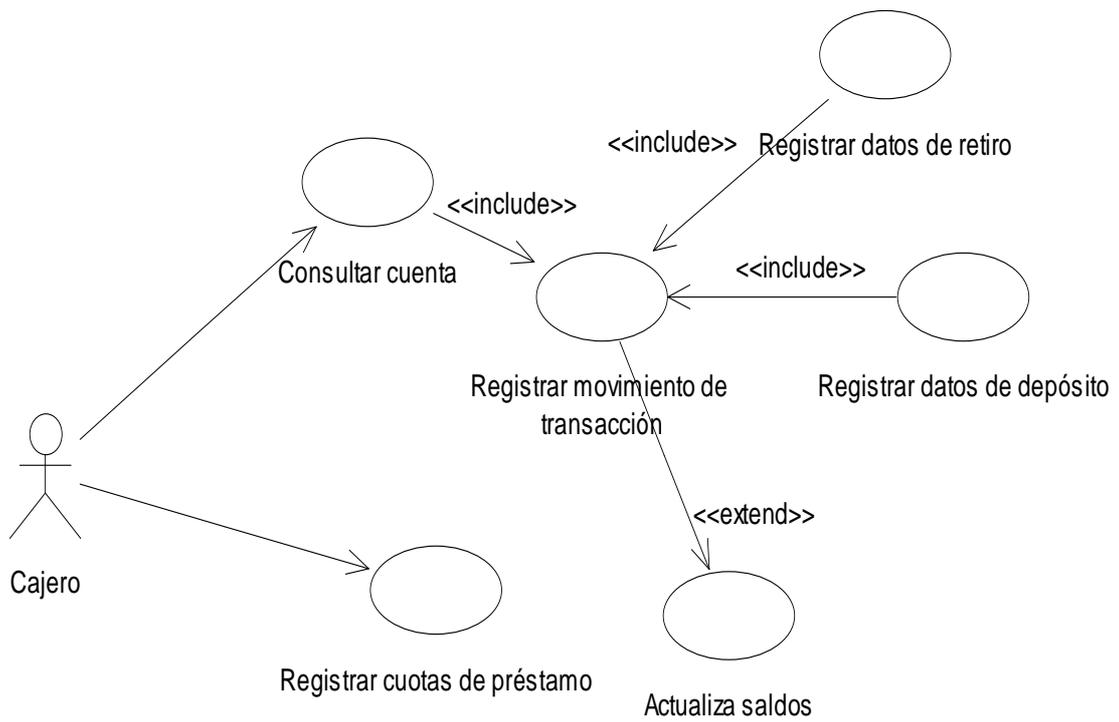
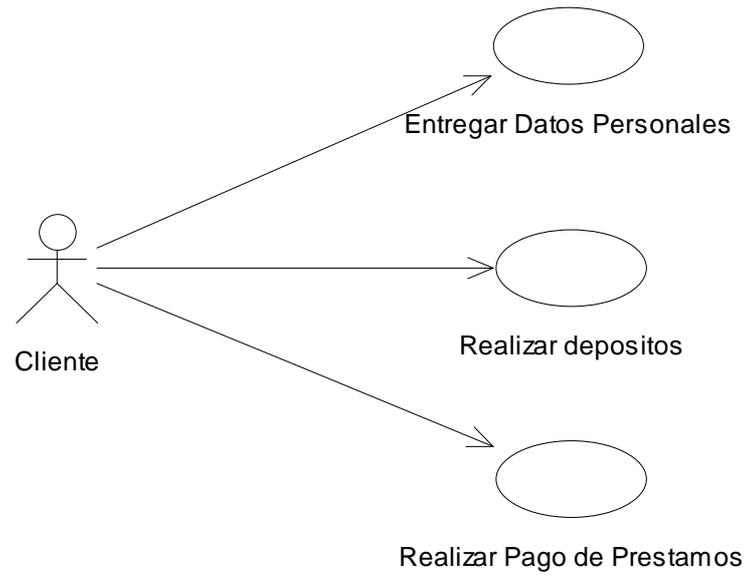


4. GENERAR REPORTES



DIAGRAMAS DE CASOS DE USO POR ACTORES





1.1.3. DICCIONARIO DE CASOS DE USO

Registrar parámetros

Parametrización inicial

ACCION

- Registrar variables a parametrizar.
- Ingresar valor inicial de parámetros

REACCION

- Ejecuta: Procedimiento registrar parámetro
- Ejecuta: Procedimiento ingresar parámetro

Datos Corporación

ACCION

- Registrar datos de la Corporación
- Guardar datos Corporación

REACCION

- Ejecuta: Buscar datos en la clase Corporación. si no existe ejecuta procedimiento registrar datos Corporación.
- Ejecuta: Actualizar datos Corporación

Registrar transacciones

Datos

ACCION

- Registrar datos
- Consultar dato cliente

REACCION

- Ejecuta: Buscar datos en la clase cliente si no existe ejecuta procedimiento registrar datos de cliente.
- Ejecuta: Buscar datos de cliente y desplegar los datos.

Cuenta

ACCION

- Registrar cuenta
- Actualizar saldos
- Consultar estados de cuenta

REACCION

- Ejecuta: Registrar código de cuenta
- Ejecuta: Modificar los saldos de cada cuenta después de una transacción.
- Ejecuta: Buscar saldos de cuentas para listar todas las transacciones

Depósito**ACCION**

- Registrar depósito
- Actualizar depósitos

REACCION

- Ejecuta: Ingresar el valor a depositar ya sea en efectivo como cheque
- Ejecuta: Modificar si el valor de un depósito fue incorrecto

Retiro**ACCION**

- Registrar retiro
- Actualizar retiro

REACCION

- Ejecuta: Ingresar el valor a retirar en efectivo.
- Ejecuta: Modificar si el valor de un retiro fue incorrecto

Interés**ACCION**

- Registrar interés
- Verificar interés

REACCION

- Ejecuta: Ingresar el valor de interés ganado o por pagar.
- Ejecuta: Buscar tipo de interés por mora, cuenta, préstamo

Registrar Préstamo**Cuenta****ACCION**

- Verificar cuenta
- Consultar saldos

REACCION

- Ejecuta: Buscar cuenta
- Ejecuta: Buscar saldos de cuentas para del cliente.

Interés**ACCION**

- Registrar interés
- Verificar interés

REACCION

- Ejecuta: Ingresar el valor de interés de acuerdo al monto solicitado.
- Ejecuta: Buscar interés por mora, y préstamo ejecuta procedimiento guardar interés

Plan Pagos**ACCION**

- Registrar plan de pagos

REACCION

- Ejecuta: Ingresar monto de préstamo, interés, y tiempo, ejecuta procedimiento guardar plan de pagos y actualizar cuentas

- Consultar plan de pagos

por cobrar

- Ejecuta: Consultar el valor a pagar en caso de realizar préstamos

Datos

ACCION

- Registrar datos
- Consultar dato cliente

REACCION

- Ejecuta: Buscar datos en la clase cliente si no existe ejecuta procedimiento registrar datos de cliente.
- Ejecuta: Buscar datos de cliente y desplegar los datos.

Préstamo

ACCION

- Registrar préstamos
- Actualizar préstamo

REACCION

- Ejecuta: Ingresar el valor del préstamo del cliente.
- Ejecuta: Consultar cuentas por pagar de la persona que adquirió el préstamo, ejecutar procedimiento registrar nuevo préstamo

Registrar cuotas de préstamo

ACCION

- Registrar cuotas de préstamo
- Actualizar cuotas de préstamo

REACCION

- Ejecuta: Buscar el monto del préstamo, ejecuta procedimiento cobro de cuotas de préstamo del cliente.
- Ejecuta: Verificar la cantidad a pagar del préstamo ejecuta procedimiento actualizar cuotas.

Generar Reportes

Tipo reporte

ACCION

- Seleccionar Tipo reporte
- Generar Reporte

REACCION

- Ejecuta: Buscar los diferentes tipos de reportes, ejecutar procedimiento listar clientes activos, prestamos, cierre de caja, intereses ganados.
- Ejecuta: procedimiento imprimir reporte seleccionado.

1.1.4. DIAGRAMA DE CLASES

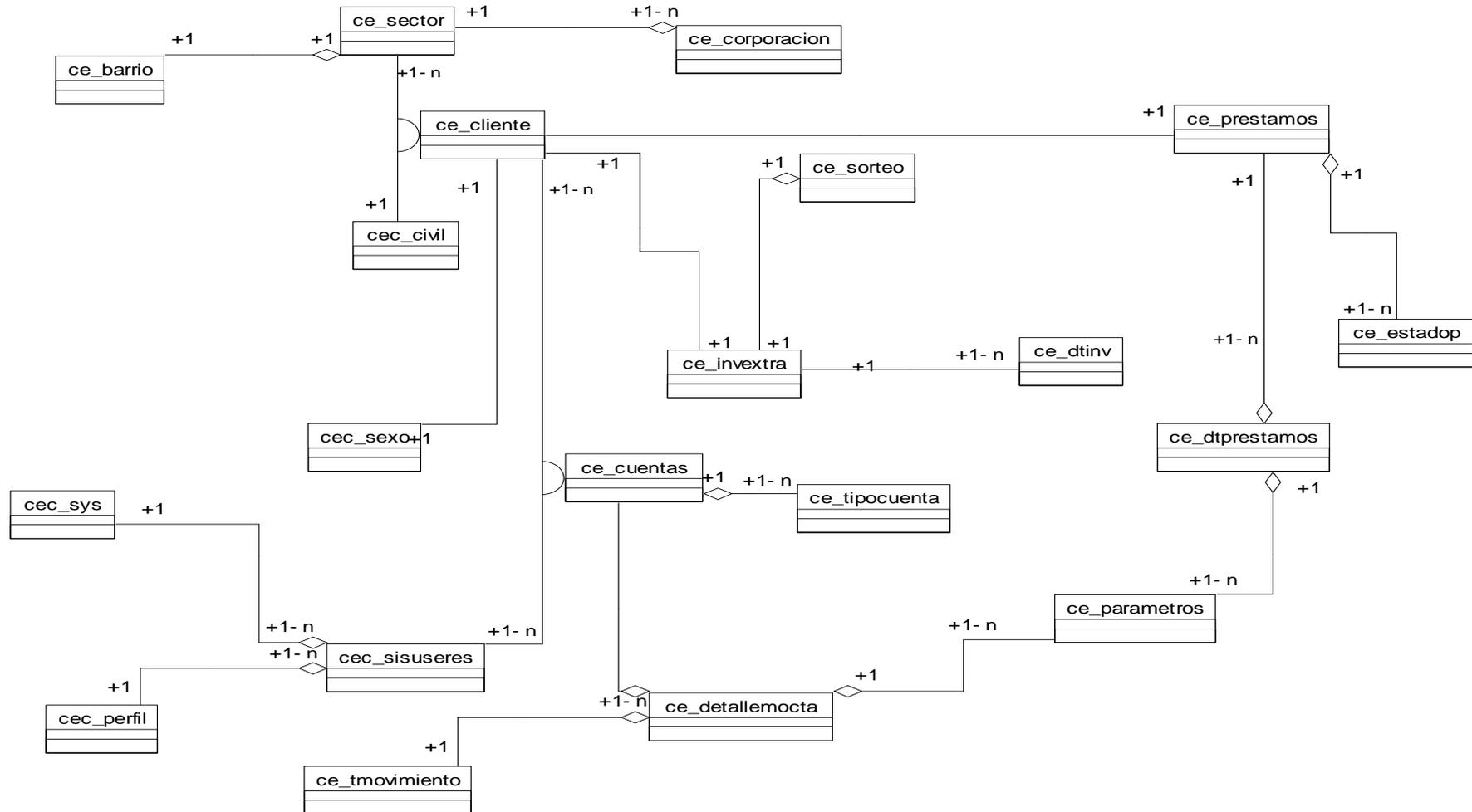
1.1.5. DICCIONARIO DE CLASES

Clase	Atributos	Descripción
ce_barrio	ba_id ba_valor ba_etiqueta	Permite almacenar la descripción de cada barrio
ce_sector	secod seubicacion senombre	Permite almacenar la información de los sectores y la relación con barrios
ce_corporación	ccod cruc cnombre secod cbarrio cdirec ctelef cfax cemail	Permite almacenar la descripción de nuestras sucursales
ce_civil	civ_id civ_etiqueta	Permite almacenar la descripción del estado civil con relación a usuarios y clientes
cecsexo	sex_id sex_value sex_etiqueta	Permite almacenar la descripción del genero con relación a usuarios y clientes
ce_clientes	<u>clnum</u> <u>clced</u> <u>clnom</u> <u>clapel</u> <u>sex_id</u> <u>secod</u> <u>ba_id</u> <u>cldirec</u> <u>cltelef</u> <u>cltelef_ofic</u> <u>clfechanac</u> <u>clst_civil</u> <u>clnom_rep</u> <u>clcargas</u> <u>clsario</u> <u>sisu_usu</u> <u>fechacambio</u>	Permite almacenar la información personal de clientes
cec_sys	sys_id sys_titulo sys_detalle	Permite almacenar la información de nuestro sistema financiero
cec_perfil	per_id per_nombre	Permite almacenar la información para dar

	per_detalle	permisos de usuarios
cec_sisusers	sisu_id sisu_usu sisu_pwd sisu_name sisu_telef secod ba_id sisu_direc sisu_email sisu_email sys_id per_id	Permite almacenar la información de usuarios
ce_cuentas	cunum clnum cutitulo cufecha_apertura ccod cutipo cusaldo_contable cusaldo_efectivo cuestado sisu_usu fehacambio	Permite almacenar la información de las cuentas con relación a los clientes
ce_tmovimiento	tmv_id tmv_etiqueta	Permite almacenar la información de que tipo de movimiento bancarios va a realizar con relación a las cuentas
ce_tipocuenta	cutido cu_etiqueta	Permite almacenar la información de los tipos de cuentas con relación a las cuentas
ce_detallemovcta	id_detallecta cunum ch_banco ch_codigo ch_valor dmcfhamov dmctipomov dmcvalor	Permite almacenar la información de los detalle de las cuentas con relación a los clientes y cuentas
ce_ctacont	ctacont_id ctacont_codigo ctacont_detalle	Permite almacenar la información de las cuentas contables ya establecidas
ce_tipocont	contipo_id contipo_etiqueta	Permite almacenar la clasificación de tipos de cuentas contables
ce_clascont	contclas_id contclas_etiqueta	Permite almacenar la clasificación de la

		descripción de las cuentas contables
ce_sorteo	scod snum_ganador sfecha svalor	Permite almacenar la información de los sorteos realizados con relación a la inversión de clientes
ce_inv	incod inum infehca insaldo	Permite almacenar la información de las inversiones extras con relación a los clientes
ce_dtinv	di_id di_valor di_fecha	Permite almacenar la información de los detalles de las inversiones con relación a las inversiones
ce_estadop	ep_id ep_etiqueta	Permite almacenar la información de los estados en que puede encontrarse los préstamos
ce_dtprestamos	dprnum dprfecha dprvalor dprestado	Permite almacenar las cuotas de cada crédito que se deben cancelar.
ce_prestamos	prnum prmonto prfechasolicitud prfeh_sproba prestado prdividendos prsaldo	Permite almacenar la información de los préstamos.
ce_parametros	id_param param_tipo par_etiqueta param_valor	Permite almacenar el valor de la tasa de interés, la fecha y el tipo de crédito al que pertenece.

1.1.6. DIAGRAMAS DE OBJETOS



**DISEÑO
MODELO DINÁMICO**

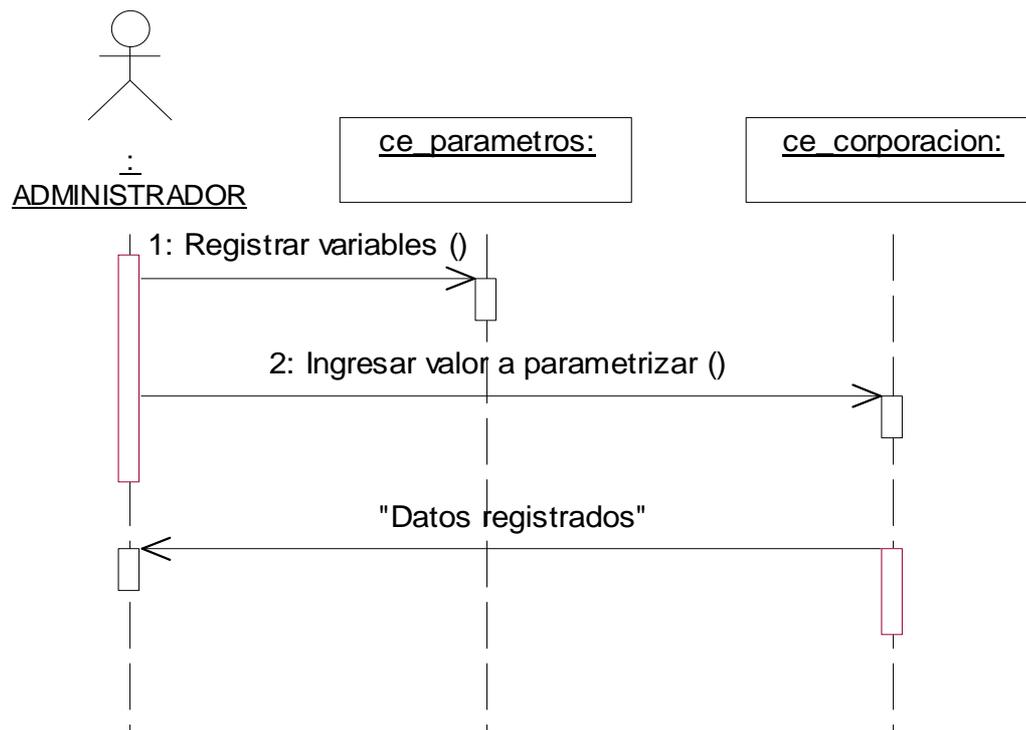
2. DISEÑO

2.1. MODELO DINÁMICO

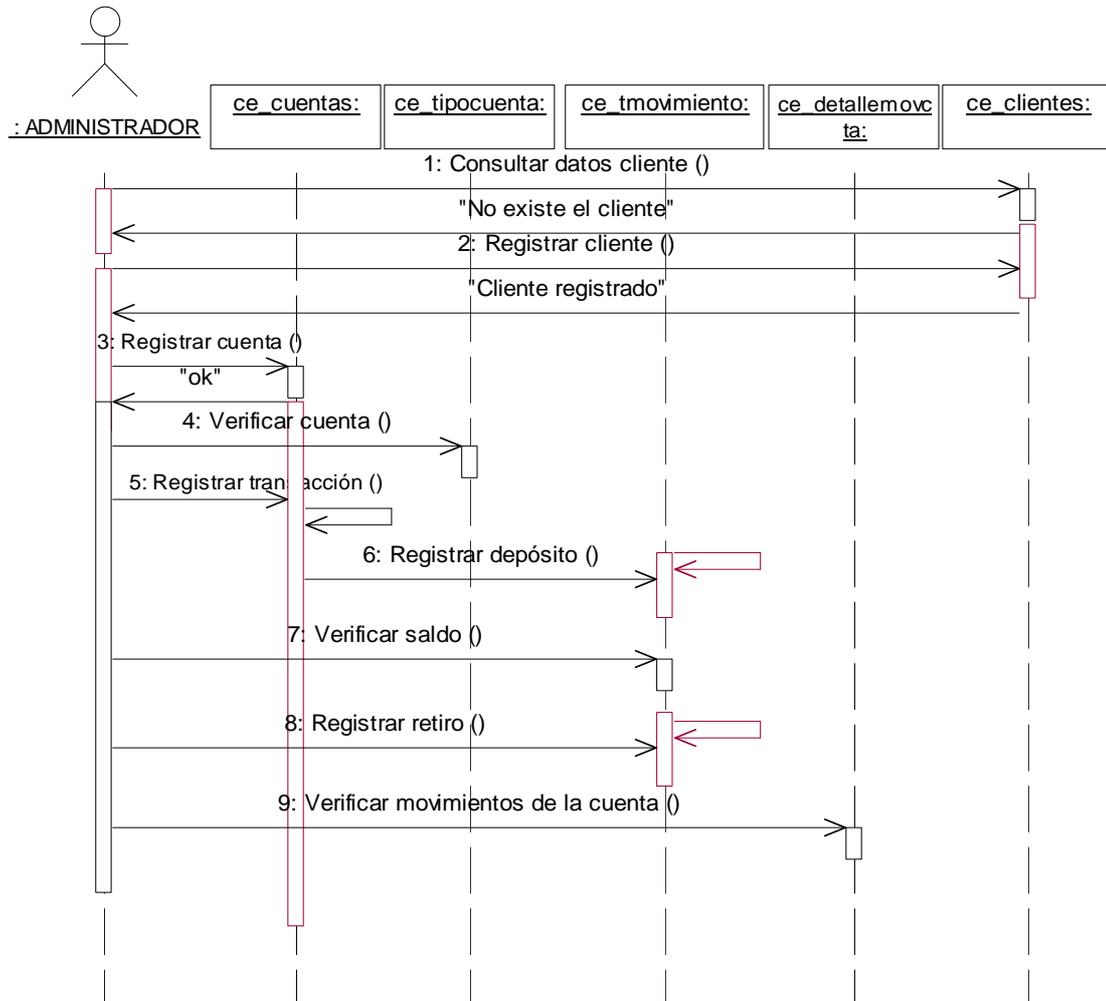
2.1.1. DIAGRAMAS DE ITERACION

DIAGRAMAS DE SECUENCIA

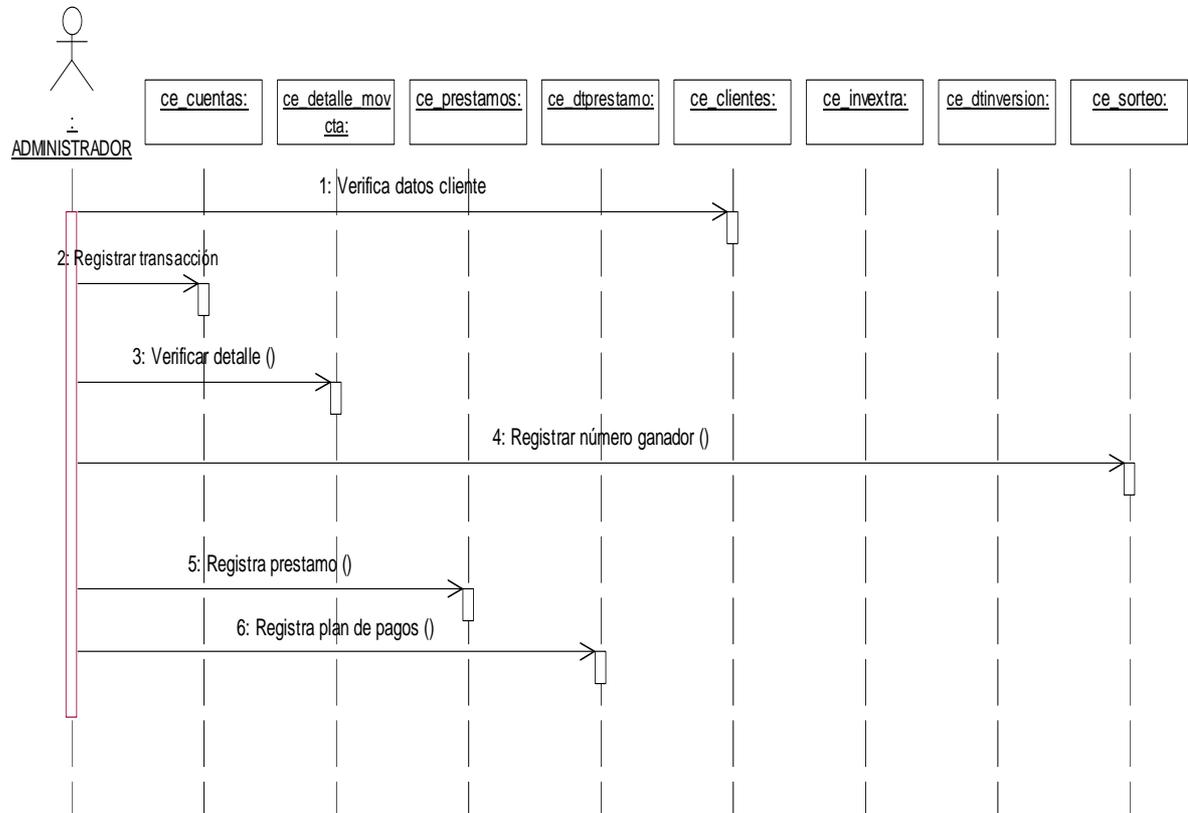
REGISTRAR PARAMETROS



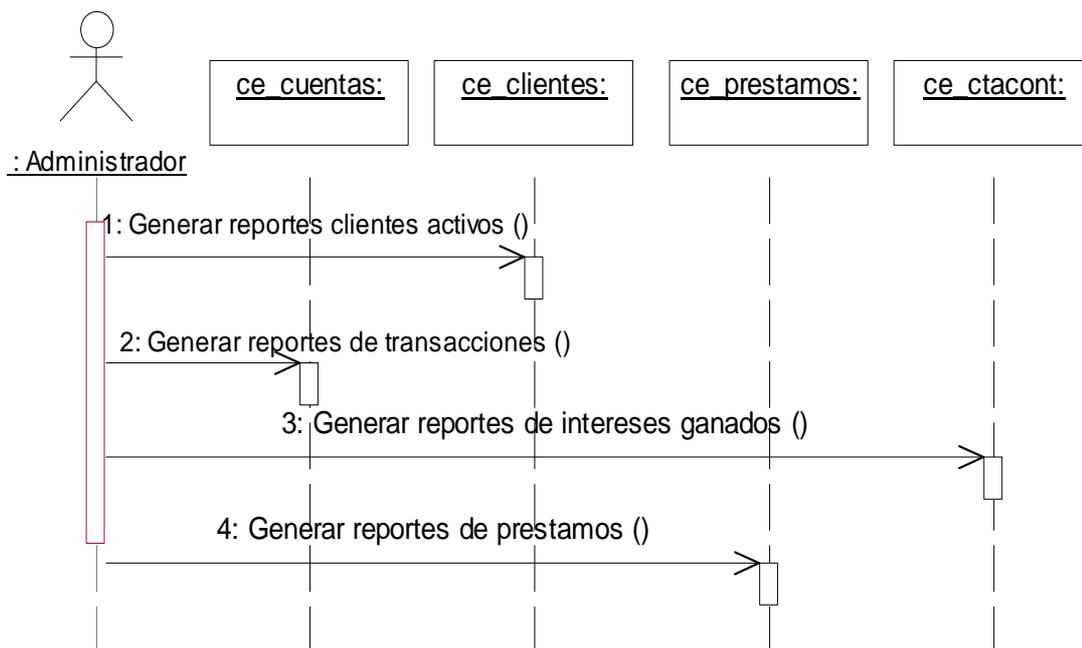
REGISTRAR TRANSACCION



REGISTRAR PRÉSTAMOS

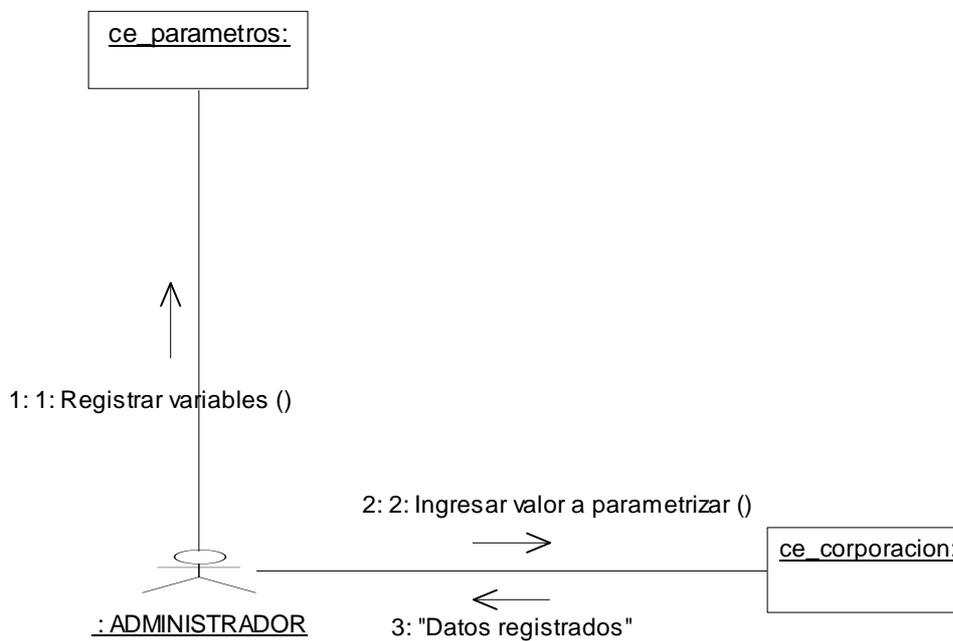


GENERAR REPORTES

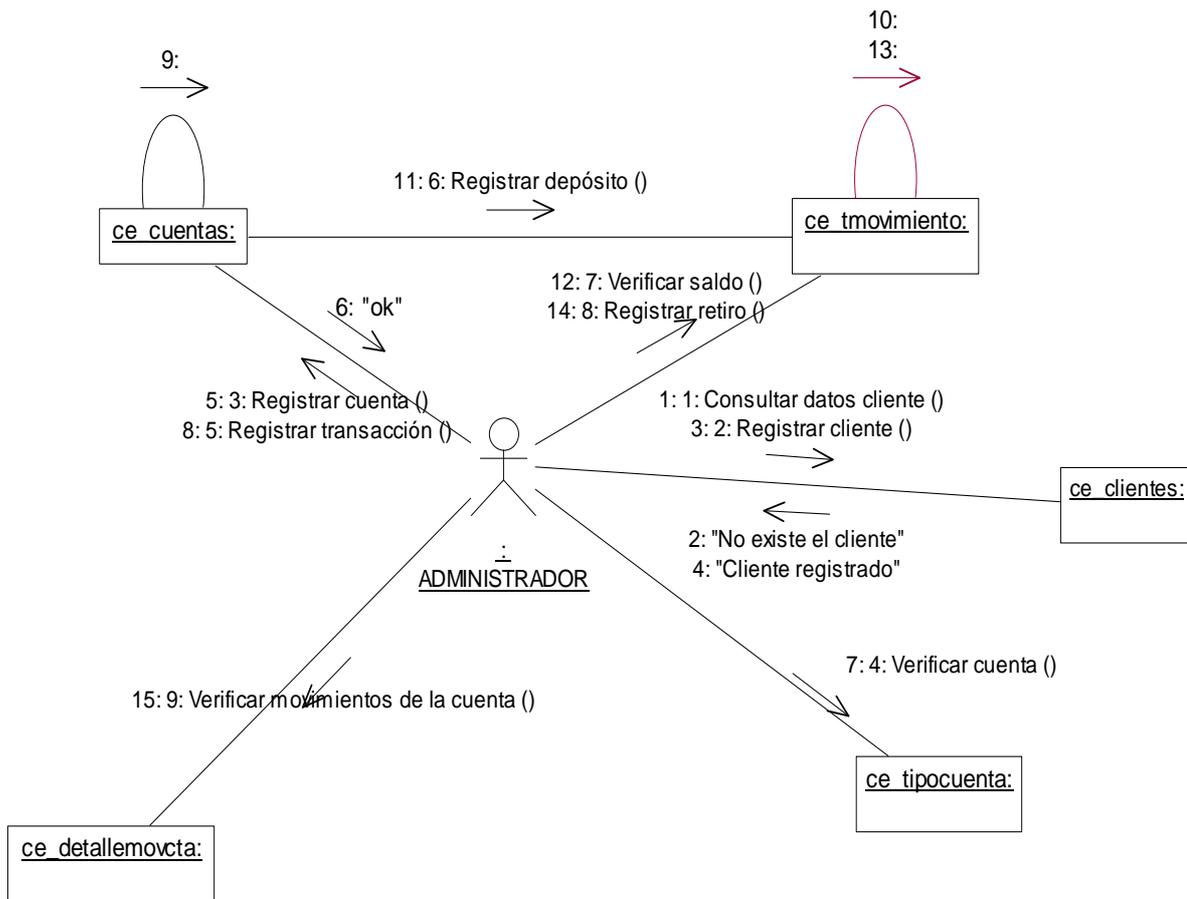


DIAGRAMAS DE COLABORACION

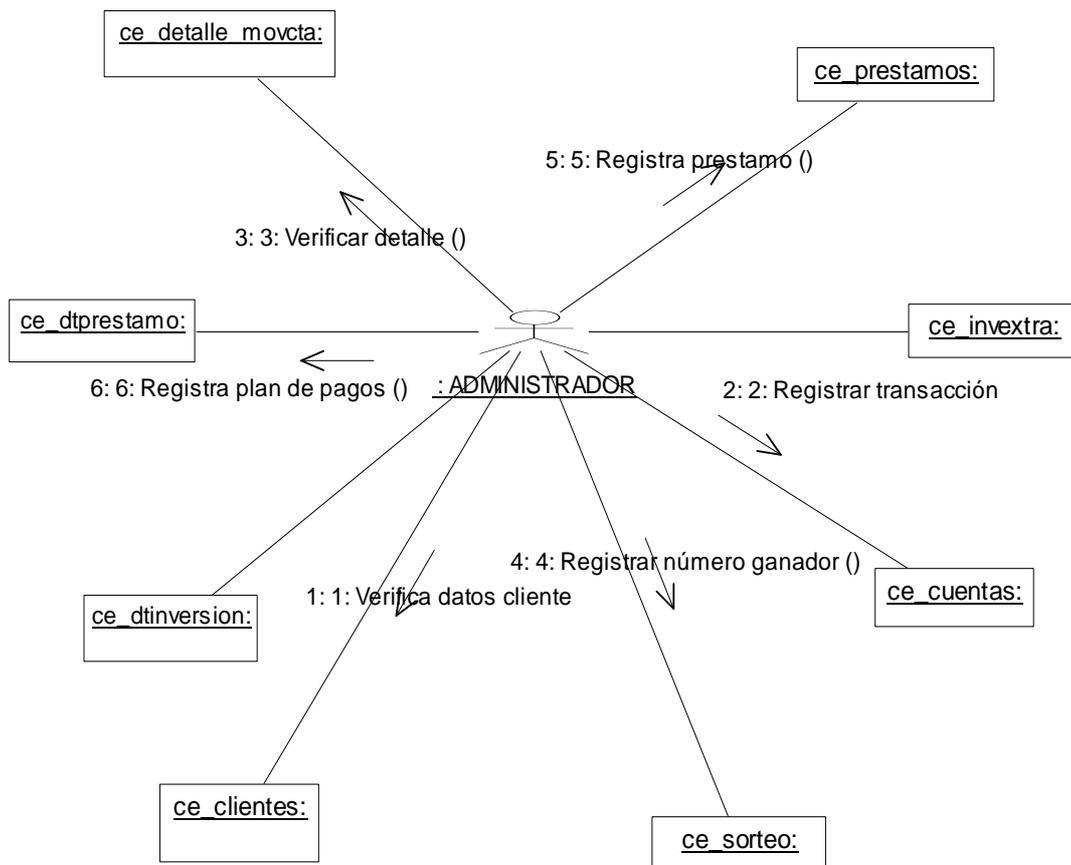
REGISTRAR PARAMETOS



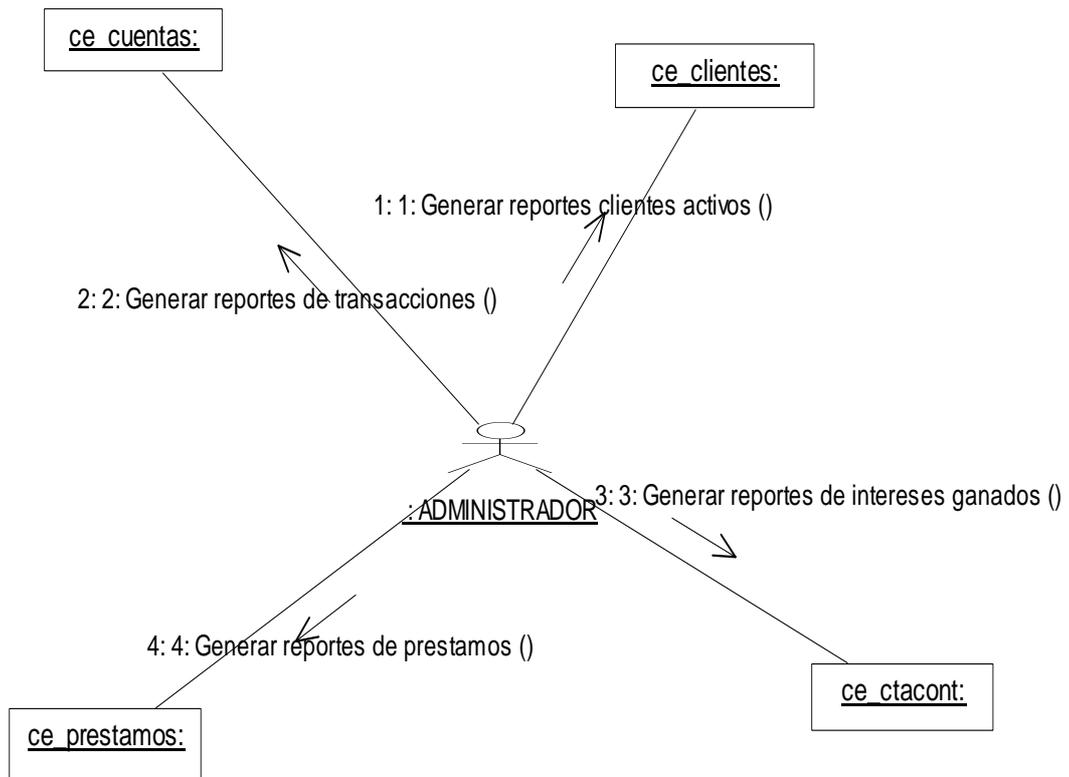
REGISTRAR TRANSACCIONES



REGISTRAR PRESTAMO



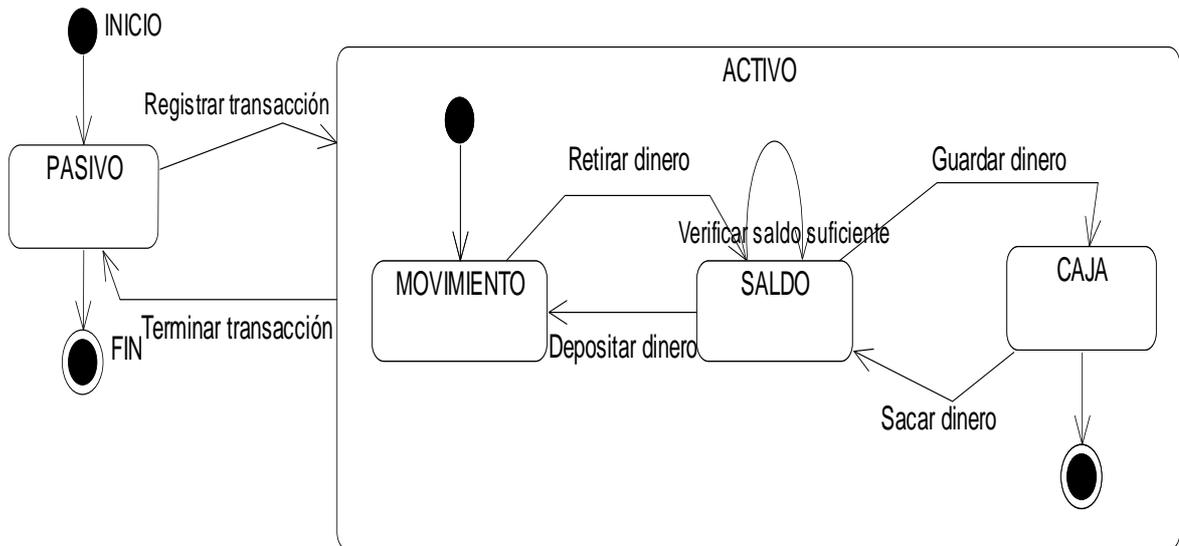
GENERAR REPORTES



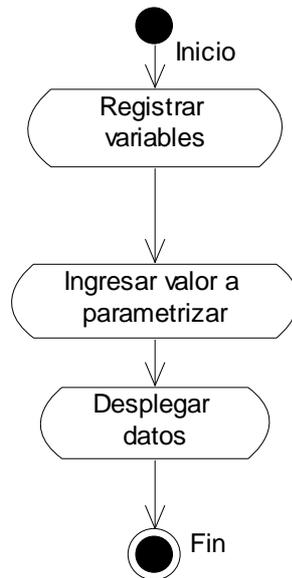
2.2. MODELO FUNCIONAL

2.2.1. DIAGRAMA DE ESTADOS

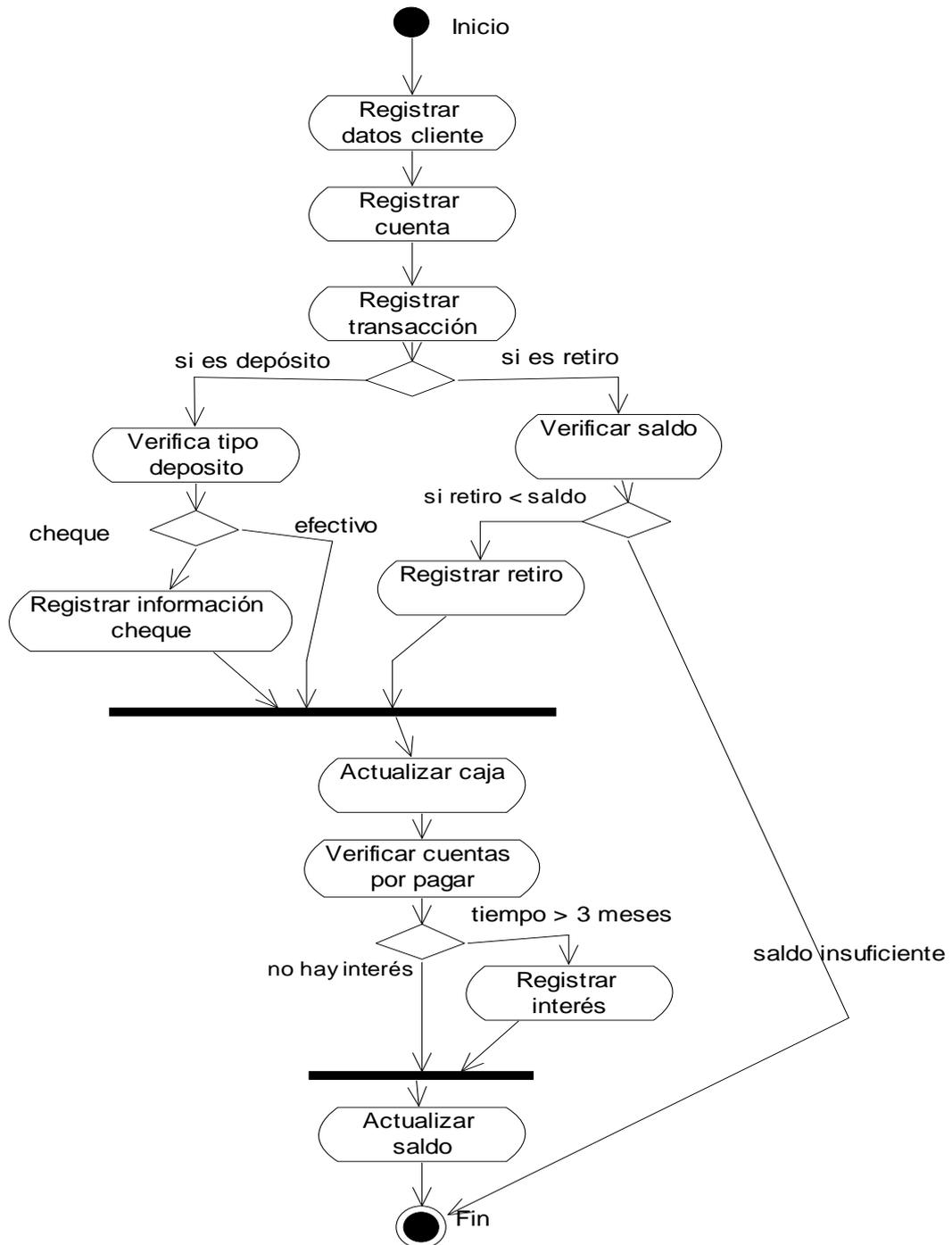
DIAGRAMA DE ESTADO CUENTA



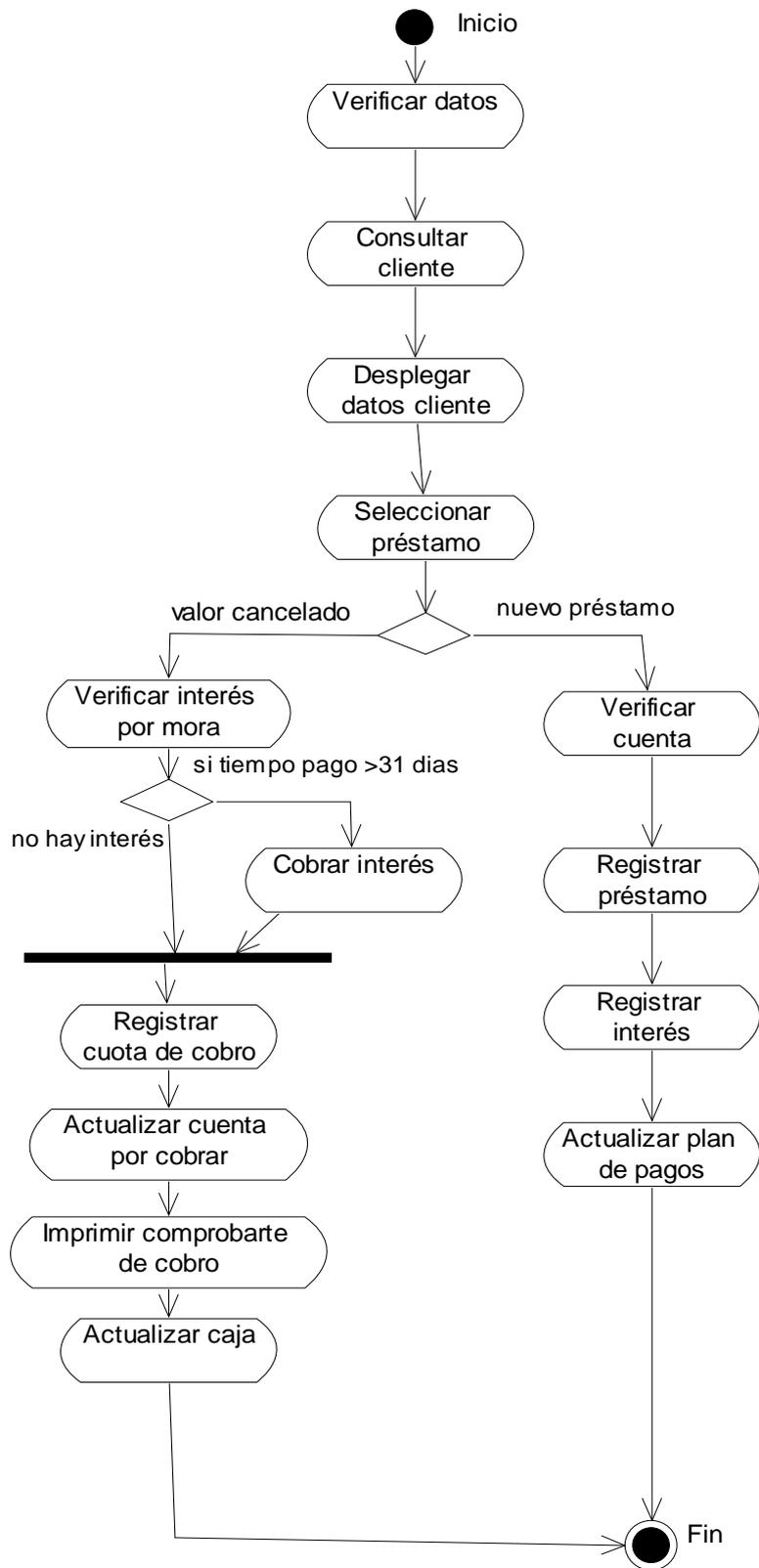
2.2.2. DIAGRAMA DE ACTIVIDADES REGISTRAR PARAMETROS



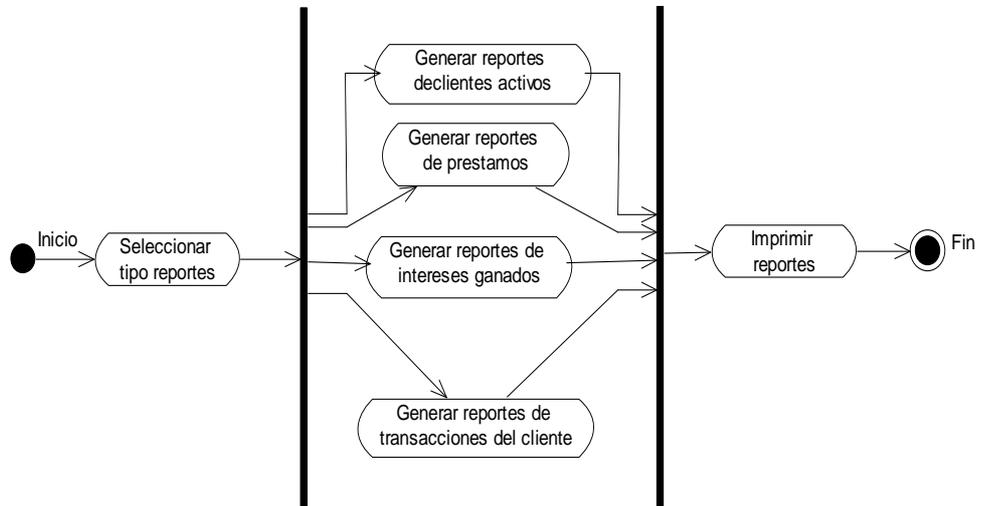
REGISTRAR TRANSACCIONES



REGISTRAR PRÉSTAMOS



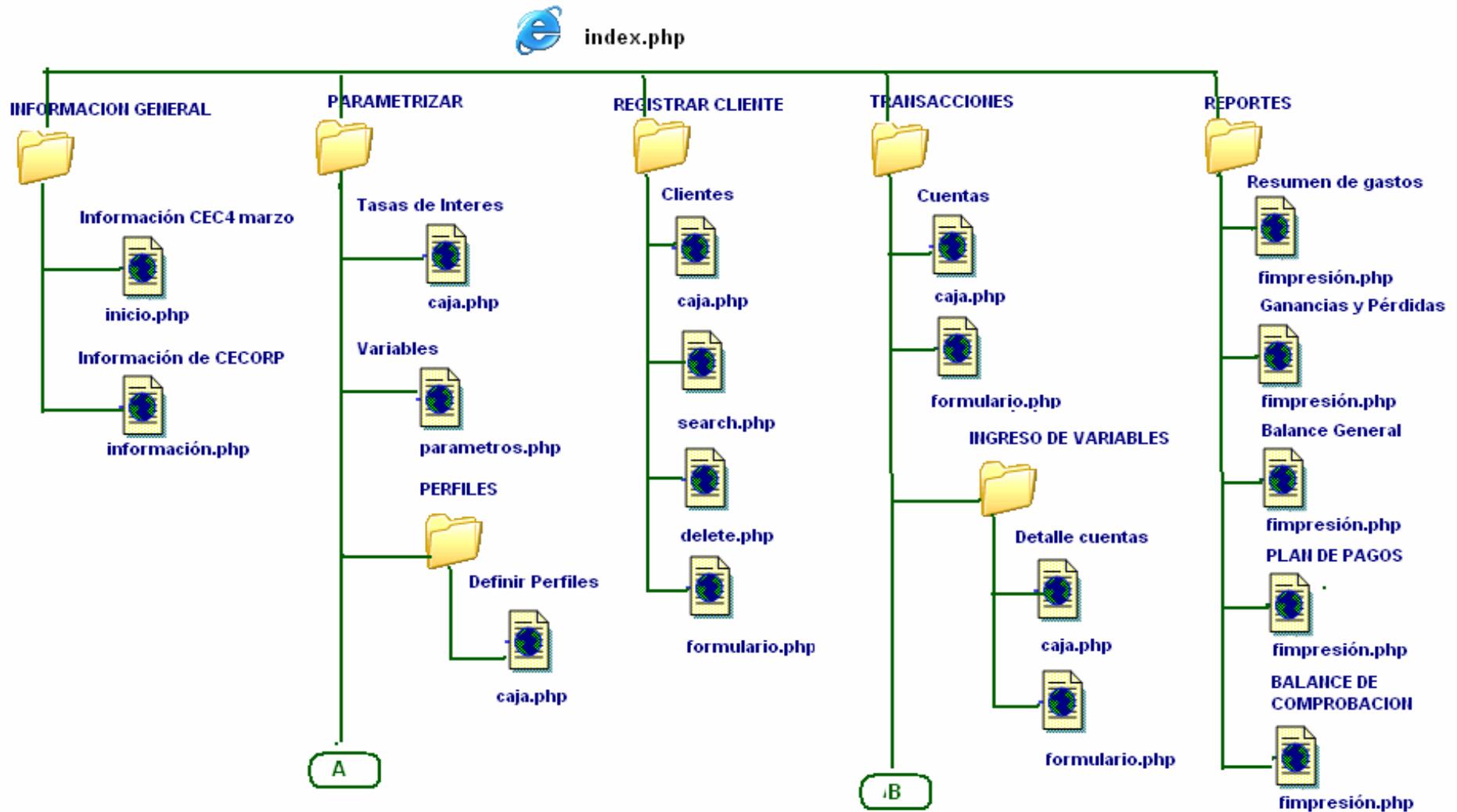
GENERAR REPORTES

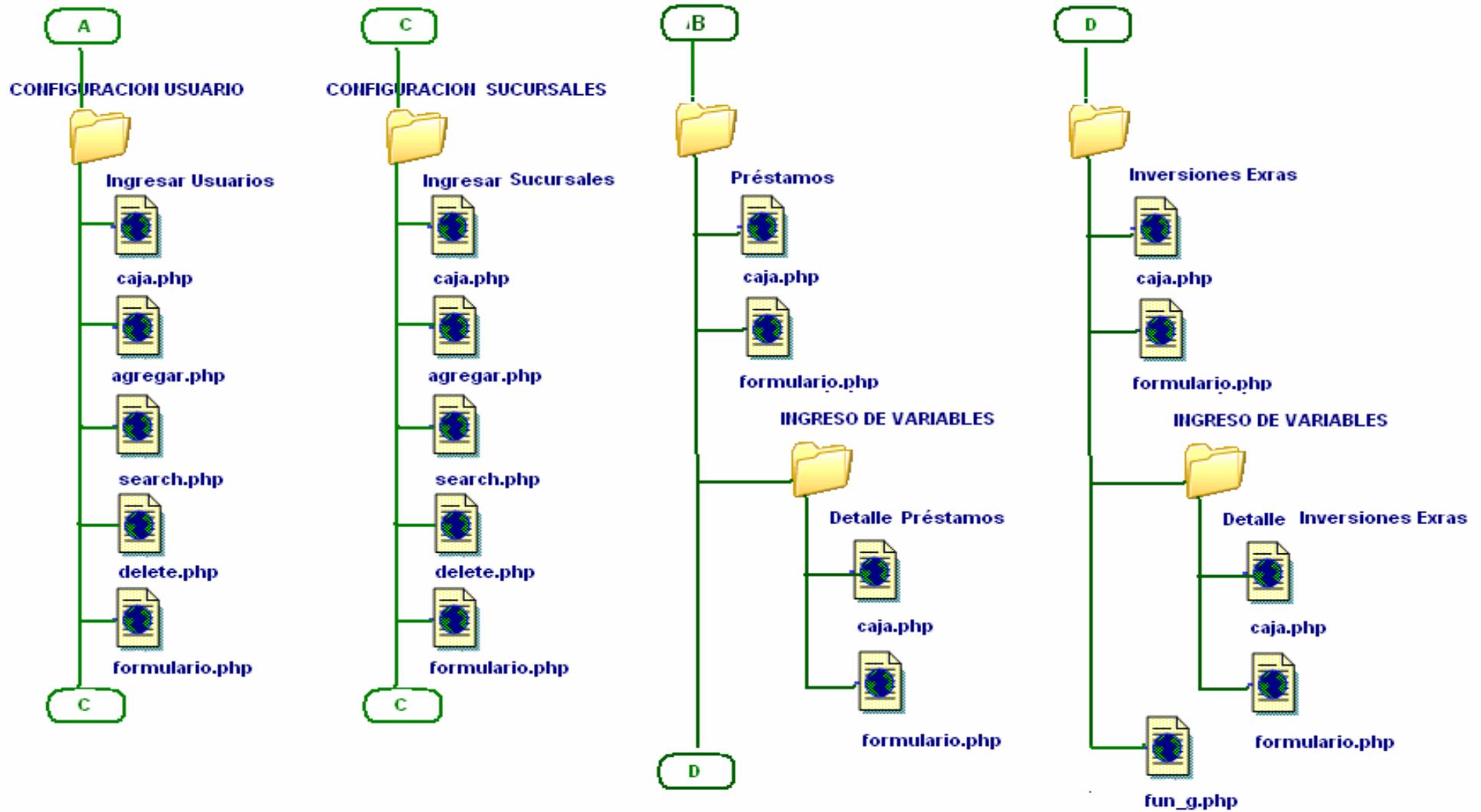


MODELO ARQUITECTÓNICO

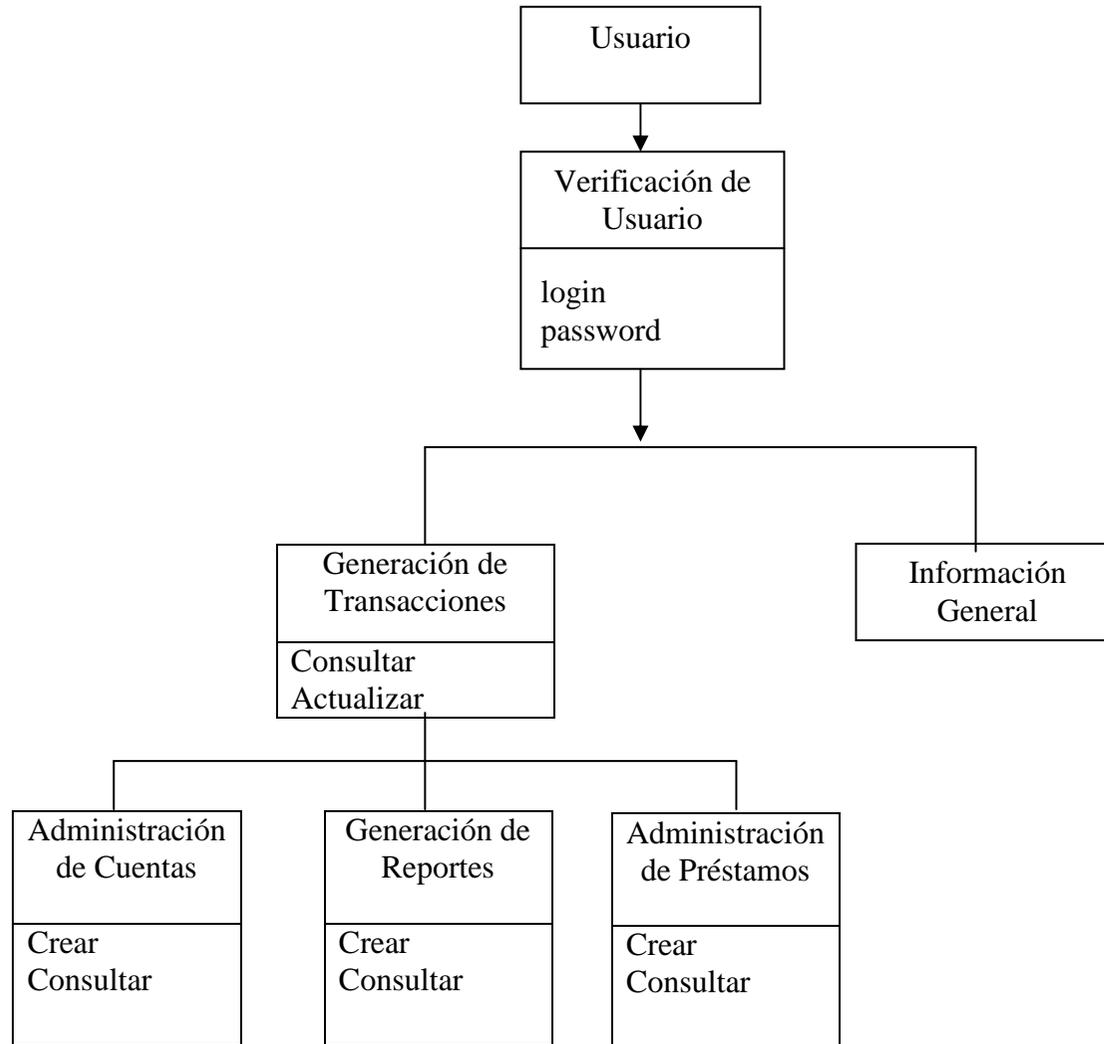
2.3. MODELO ARQUITECTÓNICO

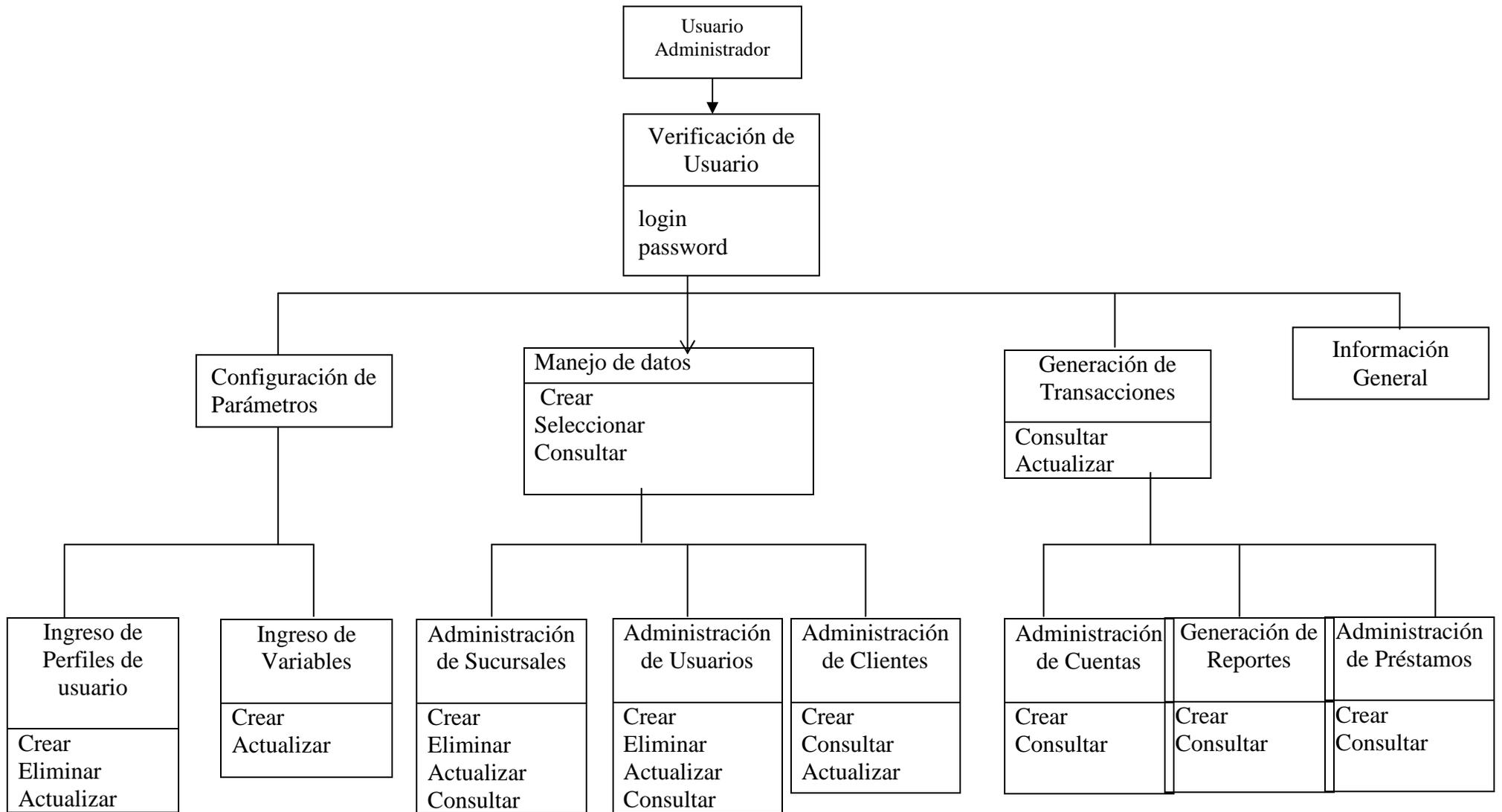
2.3.1. DIAGRAMA ARQUITECTÓNICO





2.3.2. DIAGRAMA NAVEGACIONAL





INGENIERÍA

CÓDIGO FUENTE**Clases.php**

```

<?php

if ($director)
{
    $director=$director."/";
}
else
{
    $director="";
}
//Clases para el sistema
include($director."libreria/dbcc.php");
include($director."libreria/formulario.php");
include($director."libreria/menu.php");
include($director."libreria/acces.php");
include($director."libreria/session.php");
include($director."libreria/varsend.php");
include($director."libreria/class.paginaZ.php");
?>

```

Libreria/Acces.Php

```

<?php
class acceso_system{
var $resultado;
var $user;
var $pass;
var $name;
var $perid;
function acept_session($sisusu,$pwdusu)
{
    $selecmenu="select      *      from      cec_sisusers,cec_perfil      where
cec_sisusers.per_id=cec_perfil.per_id and sisu_usu like '$sisusu' and  sisu_pwd
like '$pwdusu'";

    $resultado = mysql_query($selecmenu);
    while($row = mysql_fetch_array($resultado))
    {
        $this->user=$row[sisu_usu];
        $this->pass=$row[sisu_pwd];
        $this->name=$row[sisu_name];
        $this->perid=$row[per_id];
    }
}
}
?>

```

Libreria/Class.Paginaz.Php

```

<?
/*****
set_color_tabla($color);
set_tabla_transparente()
set_color_texto($color)
set_color_enlaces($color_inactivo,$color_activo)
/*****/
class sistema_paginacion{

    /* Colores - Configurable a gusto del consumidor :)/

var $color_link_inactivo="BLACK"; /* Color del enlace cuando el ratón no esta por encima */
var $color_link_activo="BLUE"; /* Color del enlace al pasar el ratón por encima */
var $color_texto="BLACK"; /* Color del texto normal (Que no es enlace) */
var $color_tablas="#A9BFFC"; /* Color de fondo de la tabla que contiene los enlaces */

    /* Datos para la creación de las paginas */
var $numero_por_pagina; /* Número de registros a mostrar por pagina */
var $numero_paginas;
var $total;
var $condiciones; /* Condiciones para realizar el query ..p.e. where visible=1.. */
var $id_inicio=0;

    /* Datos para la conexion a la base de datos */

var $nombre_tabla; /* Esta variable se establece al crear la clase */
var $campo_ordenacion; /* Campo de la tabla por el que se ordenan los resultados */

    /* Error */

var $error=0;
var $estilos_creados=0;
var $url;

/***** METODOS DE LA CLASE *****/

// Constructor -> establece el nombre de tabla donde consultar
function sistema_paginacion($tabla)
{
    global $id_inicio;

    $this->nombre_tabla=$tabla;
    // $this->conectar();

    if(isset($id_inicio))

```

```

        $this->id_inicio=$id_inicio;
    else
        $this->id_inicio=0;
    }

function crear_estilos()
{
    echo "<style type='text/css'>
        A.paginaZ
        {
            text-decoration:none;
            color: $this->color_link_inactivo;
            font-size: 11px;
            font-family: Verdana, Arial,
Helvetica, sans-serif;
        }
        A.paginaZ:hover
        {
            text-decoration:none;
            color: $this->color_link_activo;
            font-size: 11px;
            font-family: Verdana, Arial,
Helvetica, sans-serif;
        }
    </style>";
}

// Constructor -> establece el nombre de tabla y las condiciones para la
consulta
function set_condicion($cond)
{
    $this->condiciones=$cond;
}

// Obtiene la pagina en la que estamos a partir del id_inicio
function obtener_pagina_actual()
{
    $pagina_actual=($this->id_inicio/$this->numero_por_pagina)+1;
    return($pagina_actual);
}

/* Establece el campo por el que se realizara la ordenacion de registros */
function ordenar_por($campo)
{
    $this->campo_ordenacion=$campo;
}

```

```

/* Obtiene el numero total de registros a paginar (No modificar!) */
function obtener_total()
{
    $query="SELECT count(*) as total from $this->nombre_tabla ";

    if ($this->condiciones!="")
    {
        $query.=" $this->condiciones";
    }

    mysql_select_db($this->sql_db);
    $result=mysql_query($query);
    $row=mysql_fetch_object($result);
    $this->total=$row->total;
}
// Obtiene el numero de paginas a crear
function obtener_numero_paginas()
{
    $this->obtener_total();

    $this->numero_paginas=$this->total/$this->numero_por_pagina;

    // Si hay alguna pagina con menos del maximo tb se añade
    if (($this->total%$this->numero_por_pagina)>0)
        $this->numero_paginas++;

    $this->numero_paginas=floor($this->numero_paginas);
}
//Establece un numero maximo de elementos por pagina
function set_limite_pagina($num)
{
    $this->numero_por_pagina=$num;
    $this->obtener_numero_paginas();
}
/* Obtiene la url donde enlazar (NO MODIFICAR!!!!) */
function obtener_url()
{
    global $HTTP_GET_VARS;
    global $HTTP_POST_VARS;
    while (list ($clave, $val) = each ($HTTP_GET_VARS)) {
        if($clave!="id_inicio")
            $variables .= $clave."=".$val."&";
            if($clave=="table")
                $bandet=1;
            if($clave=="sessid")
                $bandes=1;
    }
}

```

```

    if(strpos($variables, "&")) $pag_devuelta =
$HTTP_REFERER."?". $variables;
    else
        $pag_devuelta = $HTTP_REFERER."?";

    if (!$bandet)
        $vari1="table=".$HTTP_POST_VARS['table'];
        if (!$bandes)
            $vari1.="&sessid=".$HTTP_POST_VARS['sessid']."&";

    $this->url=$pag_devuelta.$vari1;
}
// Devuelve una variable $result con los resultados de la consulta
function obtener_consulta()
{
    mysql_select_db($this->sql_db);
    $query="SELECT * from $this->nombre_tabla ";
    if ($this->condiciones!="") $query.=" $this->condiciones";
    if($this->campo_ordenacion!="") $query.=" order by $this-
>campo_ordenacion ";
    $query.=" limit $this->id_inicio,$this->numero_por_pagina";
    $result=mysql_query($query);
    return($result);
}
function error($num)
{
    $this->error=$num;
    switch($num)
    {
        case 1:echo "Error al conectar a la BD";break;
        case 2:break;
        case 3:break;
        case 4:break;
    }
}
function set_color_tabla($color)
{
    $this->color_tablas=$color;
}
function set_tabla_transparente()
{
    $this->color_tablas="NO";
}
function set_color_texto($color)
{
    $this->color_texto=$color;
}
function set_color_enlaces($color_inactivo,$color_activo)
{
    $this->color_link_inactivo=$color_inactivo;
}

```



```

{
    $numero_inicial=$i*$this->numero_por_pagina;
    $numero_final=$numero_inicial+$numero_por_pagina;
    $enlace_num=$this->url."id_inicio=$numero_inicial";
    $pagina=$i+1;
    echo " - ";
    if($pagina_actual!=$pagina)
        echo " <a class='paginaZ' href='$enlace_num'
class='paginaZ'>$pagina</a> ";
    else
        echo "<u>$pagina</u>";
}

/* Mostramos el enlace de >>Siguiete si es necesario */

$numero_siguiete=$this->id_inicio+$this->numero_por_pagina;
$enlace_sig=$this->url."id_inicio=$numero_siguiete";

if(isset($this->id_inicio)&&($this->id_inicio+$this-
>numero_por_pagina<$this->total))
    echo "&nbsp; <a class='paginaZ' href='$enlace_sig'
class='paginaZ'>&gt;&gt; Siguiete </a>";
echo "&nbsp; &nbsp; </td>";
</table>";
}
}
?>

```

INTERFAZ CLIENTE



Figura 1. cliente/caja.php

CODIGO FUENTE

```

<?php
if ($csearch)
{
    $sqlnom = "select * from ce_clientes where cnum = $csearch";
    $resultnom = mysql_query($sqlnom);
    if ($resultnom)
    {
        while($rownom = mysql_fetch_array($resultnom))
        {
            $cnumc=$rownom['cnum'];
        }
    }
    mysql_free_result($resultnom);
}

```

```

}
?>
<SCRIPT LANGUAGE=javascript>
<!--
function cedula1(ced)
{
    sp = 0
    si = 0
    for (x=1;x<10;x++)
    {
        car = ced.charAt(x-1)*1
        num = car * 2
        if (num > 9 )
        {
            num = num - 9
        }
        si= si * 1
        si = si + num
        x++
    }
    for (x=2;x<9;x=x+2)
    {
        car = ced.charAt(x-1)*1
        num = car
        sp= sp * 1
        sp = sp + num
    }
    st = sp + si
    ds = (Math.floor((st / 10)) + 1) * 10
    ud = ced.charAt(9)*1
    dv = ds - st
    if (dv == ud)
    {
        return 1
    }
    else
    {
        return 0
    }
}
//-->
</SCRIPT>
<style type="text/css">
<!--
.Estilo1 {
    font-size: 11px;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #003399;
    font-weight: bold;
}

```

```

}
.Estilo11 {font-weight: bold; color: #006699; font-family: Geneva, Arial, Helvetica,
sans-serif;}
.Estilo9 {font-size: 11px}
-->
</style>
<table width="764" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td bgcolor="#FFFFFF">
      <br>
      <table width="57%" border="1"
align="center" cellpadding="0" cellspacing="0">
        <tr>
          <td bgcolor="#CDE3F1">&nbsp;  </td>
          <td bgcolor="#CDE3F1"><span class="Estilo9"><span class="Estilo11">
            <?php
if ($csearch)
{
  printf("<a
href='index.php?table=ce_cuentas&sessid=%s&listab=%s&campo=clnum&obp='
class='Estilo11'>Cuentas</a>", $sessid, $clnumc);
}
?>
          </span></span></td>
          <td bgcolor="#CDE3F1"> <span class="Estilo9"><span class="Estilo11">
            <?php
if ($csearch)
{
  printf("<a
href='index.php?table=ce_prestamos&sessid=%s&listab=%s&campo=clnum&obp
=' class='Estilo11'>Prestamos</a>", $sessid, $clnumc);
}
?>
          </span></span></td>
        </tr>
        <tr>
          <td colspan="3" bgcolor="#CDE3F1"><div align="center"><span
class="Estilo1">ADMINISTRACION DE CLIENTES</span></div></td>
        </tr>
        <tr>
          <td colspan="3" bgcolor="#CDE3F1"> <table border="0" align="center"
cellpadding="0" cellspacing="0">
            <tr>
              <td><?php
$objformulario-
>generar_formulario($submit,$table,$atributos,$ancho,$varsend,$sessid,1);
?></td>
            </tr>
          </table></td>

```

```

        </tr>
    </table>
</td>
</tr>
</table>
<?php printf("<input name='valores' type='hidden' value=''><input name='csearch'
type='hidden' value=''><input name='idab' type='hidden' value=''><input
name='opcion' type='hidden' value='guardar'><input name='table' type='hidden'
value='%s'><input name='sessid' type='hidden' value='%s'>", $table, $sessid); ?>

```

INTERFAZ DETALLE MOVIMIENTO CUENTA

The screenshot displays two main sections of a web application interface:

- INFORMACION GENERAL:** A form with fields for:
 - Nombres y Apellido:
 - Número de cuenta:
 - Fecha de apertura:
 - Fecha última transacción:
 - Interés Ganado:
 - Total en cheques:
 - Saldo disponible:
 - Saldo de la cuenta:
- MOVIMIENTOS TRANSACCIONALES:** A section with a table structure. The table has columns for 'Cuenta', 'Fecha', 'Descripción', and 'Total'. A tooltip message reads: "Presione Control y haga clic para seleccionar celdas".

At the bottom left of the interface, there is a set of navigation icons (back, forward, search, etc.).

Figura 2. detcuentas/caja.php

CODIGO FUENTE

```

<?php include("modules/menuback2.php"); ?>
<style type="text/css">
<!--
.Estilo12 { font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
color: #003399;
font-weight: bold;
}
.Estilo15 {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; }
.Estilo17 {
font-family: "Times New Roman", Times, serif;
font-size: 10px;
color: #333333;
font-weight: lighter;
text-decoration: none;
}

```

```
.Estilo19 {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
```

```
-->
```

```
</style>
```

```
<SCRIPT LANGUAGE=javascript>
```

```
<!--
```

```
function sumatotal()
```

```
{
```

```
sumat= parseFloat(fa.ch_valor.value) + parseFloat(fa.dmcvalor.value);
```

```
fa.suma.value=parseFloat(sumat);
```

```
}
```

```
-->
```

```
</SCRIPT>
```

```
<link href="<?php echo $pathc.$objtableform->path_templateform
?>formatoper.css" rel="stylesheet" type="text/css">
```

```
<?php
```

```
    $sqlsal = "select * from ce_cuentas,ce_detallemovcta where
ce_cuentas.cunum=ce_detallemovcta.cunum and
ce_detallemovcta.cunum=$listab order by dmcfechamov desc";
```

```
    //echo $sqlsal;
```

```
    $resultsal = mysql_query($sqlsal);
```

```
    $i1=1;
```

```
    $j1=0;
```

```
    if ($resultsal)
```

```
    {
```

```
        while($rowsal = mysql_fetch_array($resultsal))
```

```
        {
```

```
            if ($i1==1)
```

```
            {
```

```
                $ultran=$rowsal["dmcfechamov"];
            }
```

```
            }
```

```
            $i1++;
```

```
            //Fecha de movimiento
```

```
            //echo $rowsal["dmcfechamov"]."<br>";
```

```
            $days= floor((time() - strtotime($rowsal["dmcfechamov"])))/86400);
```

```
        //Depositos en cheque y efectivo
```

```
        if ($rowsal["dmctipomov"]=="Deposito")
```

```
        {
```

```
            $sumadepef = $sumadepef + $rowsal["dmcvalor"];
            $intdepef = $intdepef + $rowsal["dmcvalor"] * ($tasaig *
```

```
($days/365));
```

```
        }
```

```
        if ($rowsal["dmctipomov"]=="Deposito")
```

```
        {
```

```

        $sumadepch = $sumadepch + $rowsal["ch_valor"];
        $intdepch = $intdepch + $rowsal["ch_valor"] * ($tasaig *
($days/365));
        if ($rowsal["ch_valor"])
        {
            $j1++;
        }
    }

//Retiros en efectivo y cheque

if ($rowsal["dmctipomov"]=="Retiro")
{
    $sumareef = $sumareef + $rowsal["dmcvalor"];
    $intreef = $intreef + $rowsal["dmcvalor"] * ($tasaig *
($days/365));
}

if ($rowsal["dmctipomov"]=="Retiro")
{
    $sumarech = $sumarech + $rowsal["ch_valor"];
    $intrech = $intrech + $rowsal["ch_valor"] * ($tasaig *
($days/365));
}

}

$Saldo= $sumadepef + $sumadepch - $sumareef - $sumarech;
$interesg = $intdepef + intdepch - $intreef - $intrech;
$cheques = $j1;

$sqlnom = "select * from ce_clientes,ce_cuentas where
ce_clientes.clnum=ce_cuentas.clnum and ce_cuentas.cunum=$listab";

$resultnom = mysql_query($sqlnom);
if ($resultnom)
{
    while($rownom = mysql_fetch_array($resultnom))
    {
        //fecha actual
        $dob=$rownom[clfechanac];

        $dia=date(j);
        $mes=date(n);
        $ano=date(Y);
    }
}

```

```
//fecha de nacimiento
list($y,$m,$d)=explode("-", $dob);
$dianaz=$d;
$mesnaz=$m;
$anonaz=$y;
```

//si el mes es el mismo pero el dia inferior aun no ha cumplido años, le quitaremos un año al actual

```
if (($mesnaz == $mes) && ($dianaz > $dia)) {
    $ano=($ano-1); }
```

//si el mes es superior al actual tampoco abra cumplido años, por eso le quitamos un año al actual

```
if ($mesnaz > $mes) {
    $ano=($ano-1);}
```

//ya no habria mas condiciones, ahora simplemente restamos los años y mostramos el resultado como su edad

```
$edad=($ano-$anonaz);
```

```
if ($dob)
{
    $edadnac= $edad;
}
```

?>

```
<table width="512" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="516" height="399" bgcolor="#FFFFFF"><table width="501"
align="center" cellpadding="0" cellspacing="0" bordercolor="#005BB7">
<tr bgcolor="#CDE3F1" class="Estilo17">
<td height="24" colspan="4"><div align="center"><span class="Estilo12">INFORMACION GENERAL
</span></div></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
<td width="39%" bordercolor="#F4F8FF" class="Estilo17"><span
class="Estilo15"><strong>Nombres y Apellido</strong>: </span></td>
<td height="20" colspan="3" bordercolor="#F4F8FF"><span
class="Estilo15"><?php echo $rownom[clapell]." ".$rownom[clnom];
?></span><span class="Estilo19"></span><span class="Estilo19"></span></td>
</tr>
```

```

<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF" class="Estilo17"><span
class="Estilo19">N&uacute;mero de cuenta: </span></td>
  <td colspan="3" bordercolor="#F4F8FF"><span class="Estilo15"><?php echo
$rownom[cunum]; ?></span><span class="Estilo19"></span><span
class="Estilo19"></span></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF" class="Estilo17"><span
class="Estilo15"><strong>Fecha de apertura </strong>: </span></td>
  <td colspan="3" bordercolor="#F4F8FF"><span class="Estilo15">
<?php
          echo $rownom[cufecha_apertura];
          ?>
</span><span class="Estilo15"> </span><span class="Estilo15"> </span></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF" class="Estilo17"><span
class="Estilo15"><strong>Fecha ultima transacci&oacute;n </strong>:
</span></td>
  <td colspan="3" bordercolor="#F4F8FF"><span class="Estilo15">
<?php
          echo $ultran;
          ?>
</span><span class="Estilo15"> </span><span class="Estilo15"> </span></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF" class="Estilo17"><span
class="Estilo15"><strong>Interes Ganado </strong>: </span></td>
  <td width="18%" bordercolor="#F4F8FF"><span class="Estilo15">
<?php
          echo number_format($interesg,3);
          ?>
</span></td>
  <td width="25%"><span class="Estilo15"><strong>Total en cheques </strong>:
</span></td>
  <td width="18%"><span class="Estilo15">
<?php
          echo $ncheques;
          ?>
</span></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF">&nbsp;</td>
  <td bordercolor="#F4F8FF">&nbsp;</td>
  <td><span class="Estilo15"><strong>Saldo disponible</strong>: </span></td>

```

```

<td bgcolor="#CDE3F1"><span class="Estilo15">
  <?php
      echo $Saldo;
  ?>
</span></td>
</tr>
<tr bordercolor="#FFFFFF" bgcolor="#FFFFFF" class="Estilo17">
  <td bordercolor="#F4F8FF" bgcolor="#FFFFFF">&nbsp;</td>
  <td bordercolor="#F4F8FF"><span class="Estilo15"> </span></td>
  <td bgcolor="#FFFFFF"><span class="Estilo15"><strong>Saldo de la cuenta
</strong>: </span></td>
  <td bgcolor="#CDE3F1"><span class="Estilo15">
  <?php
      echo $Saldo;
  ?>
  </span></td>
</tr>
</table> <?php
}
}
?>
<p>&nbsp;</p>
<table width="501" border="0" align="center" cellpadding="0" cellspacing="0"
bordercolor="#005BB7">
  <tr>
    <td bgcolor="#CDE3F1"><p align="center" class="Estilo12">MOVIMIENTOS
TRANSACCIONALES</p>
    </td>
  </tr>
  <tr>
    <td bgcolor="#CDE3F1"><div align="center"></div>
    <table width="100%" border="0" cellpadding="0" cellspacing="0">
      <tr bgcolor="#F4F8FF">
        <td width="40%" rowspan="6" valign="top" bgcolor="#F4F8FF"><?php
$objformulario->sendvar["cunumx"]=$listab;
$objformulario-
>generar_formulario($submit,$table,$atributos,$ancho,$varsend,$sessid,1);
$objformulario->sendvar["fechax"]=date("Y-m-d");
$objformulario-
>generar_formulario($submit,$table,$atributos,$ancho,$varsend,$sessid,4);
?></td>
        <td bgcolor="#F4F8FF">&nbsp;</td>
      </tr>
      <tr bgcolor="#F4F8FF">
        <td width="60%" bgcolor="#CDE3F1"><span
class="Estilo17">Cheque</span></td>

```

```

        </tr>
        <tr>
            <td bgcolor="#F4F8FF"><span class="Estilo15">
                <?php
$Objformulario-
>generar_formulario($submit,$table,$atributos,$ancho,$varsend,$sessid,2);
?>
            </span>
            <hr size="1">                </td>
        </tr>
        <tr>
            <td bgcolor="#CDE3F1"><span class="Estilo17">Efectivo</span></td>
        </tr>
        <tr>
            <td bgcolor="#F4F8FF"><span class="Estilo15">
</span>
        <table width="77%" height="20" border="0" cellpadding="0"
cellspacing="0">
            <tr>
                <td><span class="Estilo15">
                    <?php
$Objformulario-
>generar_formulario($submit,$table,$atributos,$ancho,$varsend,$sessid,3);
?>
                    </span></td>
            </tr>
        </table>
        <hr size="1">                </td>
        </tr>
        <tr>
            <td bgcolor="#CDE3F1"><span class="Estilo17">Total:
                <input name="suma" type="text" id="suma" value="<?php echo
$Objformulario->contenid["ch_valor"] + $Objformulario->contenid["dmcvalor"] ?>"
size="10" >
            </span></td>
        </tr>
        </table></td>
        </tr>
        </table></td>
        </tr>
        </table>
        <input name="listab" type="hidden" id="listab" value="<?php echo $listab;
?>">
        <input name="obp" type="hidden" id="obp" value="">
        <input name="campo" type="hidden" id="campo" value="cunum">
            <input name="listabant" type="hidden" id="listabant"
value="<?php echo $listabant; ?>">
        <input name="obpant" type="hidden" id="obpant" value="">
        <input name="campoant" type="hidden" id="campoant" value="cnum">

```

```
<?php    printf("<input name='valores' type='hidden' value=' '><input
name='csearch' type='hidden' value=' '><input name='idab' type='hidden'
value=' '><input name='opcion' type='hidden' value='guardar'><input name='table'
type='hidden' value='%s'><input name='sessid' type='hidden'
value='%s'>", $table, $sessid);
```

```
$ordername="dmcfechamov";
?>
```

MANUAL DE USUARIO

CONTENIDO

1. INTRODUCCIÓN
2. PARTES DE LA INTERFAZ
3. ACCESO A LA APLICACIÓN
4. ADMINISTRACIÓN DE PERFILES
5. ADMINISTRACIÓN DE USUARIOS
6. ADMINISTRAR PARÁMETROS
7. ADMINISTRAR SUCURSALES
8. INFORMACIÓN GENERAL
9. ADMINISTRAR CLIENTES
10. ADMINISTRAR CUENTAS
11. ADMINISTRAR PRÉSTAMOS
12. ACCESO A PRÉSTAMOS
13. CALCULAR PRESTAMO
14. CONSULTAR PANEL
15. MODULO DE SALDOS
16. MODULO BALANCE
17. SALIR

MANUAL DE USUARIO

APLICACIÓN WEB “CECORP”

1. INTRODUCCIÓN

La aplicación Web llamada CECORP aplicada a la Corporación Educativa y Cultural 4 de Marzo presenta las siguientes pantallas de presentación.

Para ingresar debemos acceder al Microsoft Internet Explorer y digitar en la Barra de Direcciones <http://127.0.0.1/cecorp/>.



Figura 1.1 Presentación

Y a continuación se desplegará la siguiente interfaz para acceder a la aplicación de la Corporación Educativa y Cultural 4 de Marzo.

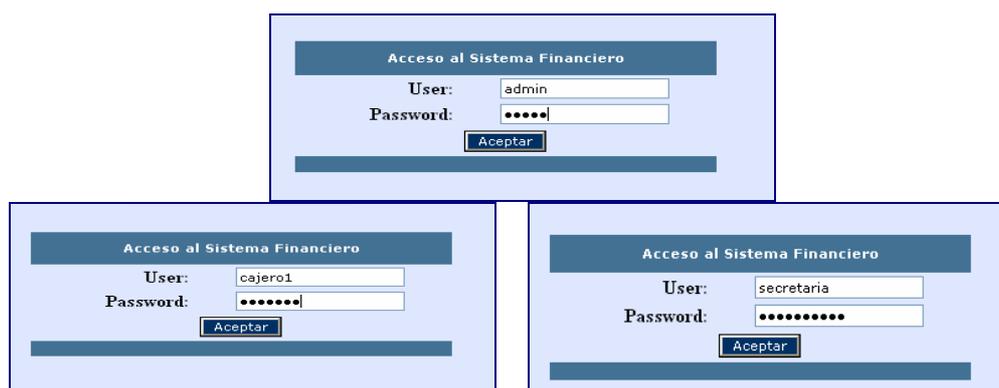


Figura 1.2 Accesos

Cada uno de los usuarios se les asigna un perfil el cual restringirá sus condiciones o funciones del manejo de la aplicación. Existen tres tipos de perfiles: el Administrador, cajero, secretaria.

Cada de los usuarios presentarán un menú cuyas opciones dependerán de sus funciones ya establecidas por la Corporación.

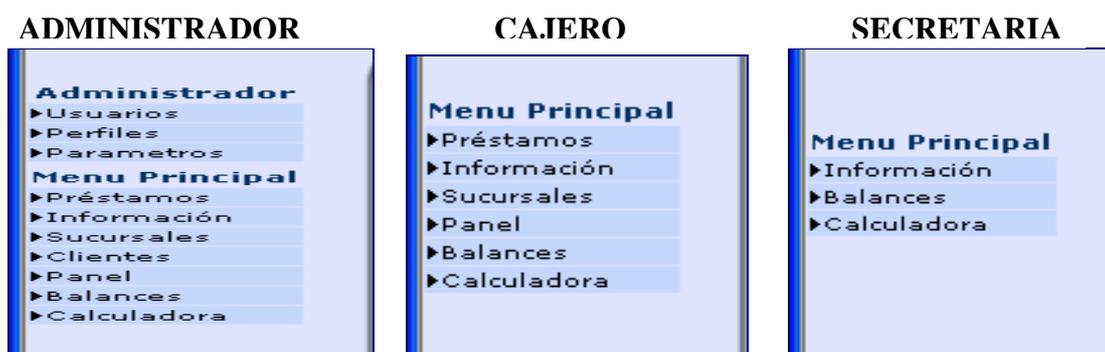


Figura 1.3 Menús

2. PARTES DE LA INTERFAZ

MENU (1)._ En la parte derecha se encuentra el panel de menú en el cual nos presenta los módulos o interfaces que interactúan con la aplicación.

BARRA SUPERIOR (2)._ Se encuentran en la parte superior de la interfaz y nos permite mostrar información de la aplicación entre otras opciones.

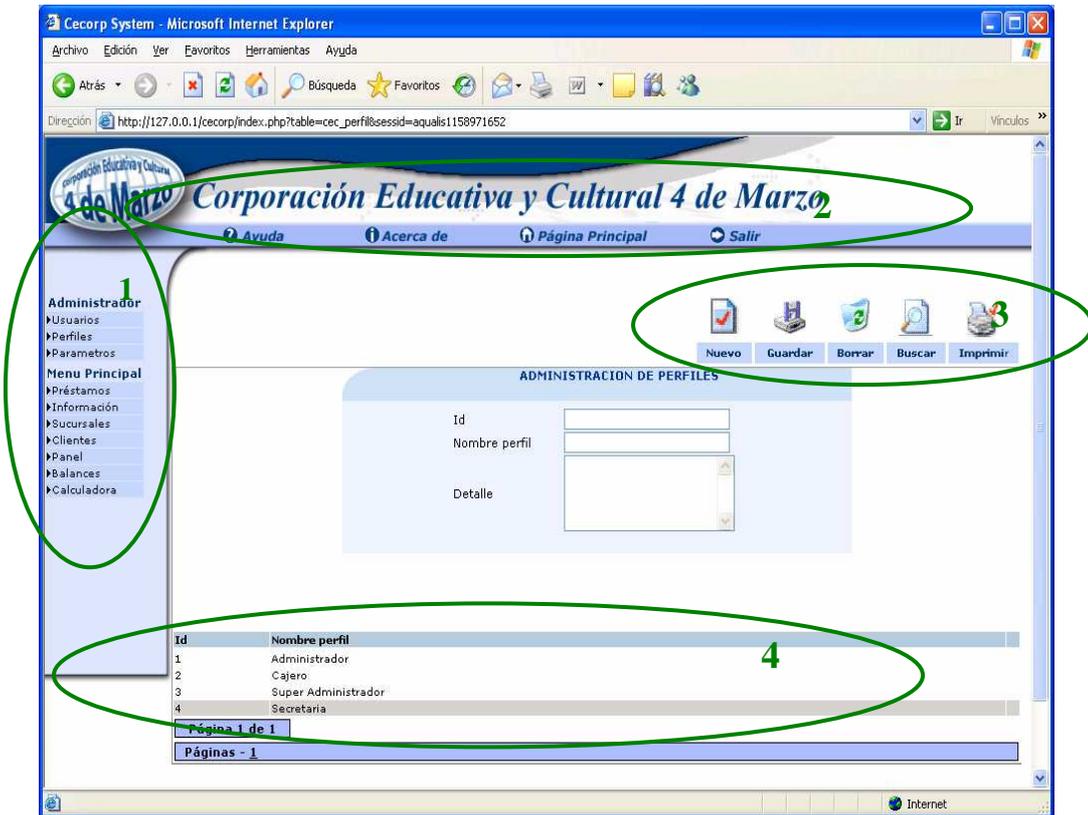


Figura 2.1 Barras

BOTONES (3)._ La aplicación utiliza los siguientes botones:



Figura 2.2 Botones

NUEVO._ Permite ingresar nueva información de los campos que usted requiera.

GUARDAR._ Permite refrescar la información ya sea nueva o modificada.

BORRAR._ Elimina el campo seleccionado.

BUSCAR._ Esta opción permite realizar búsquedas definidas de un campo o una búsqueda más compleja.

IMPRIMIR._ Permite imprimir pantallas o reportes ya establecidos.

LISTAR INFORMACIÓN (4)._ Permite desplegar una lista de datos.

3. ACCESO A LA APLICACIÓN

Para ingresar a la aplicación debemos digitar el nombre del usuario (admin) y la clave establecida (admin) para ingresar al módulo CMS (Control Manager Systems).

A continuación se desplegará la siguiente interfaz.

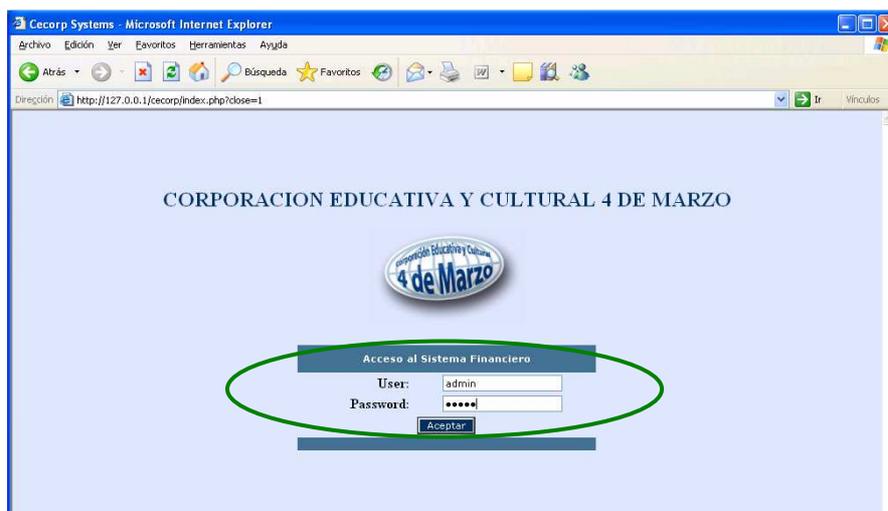


Figura 3.1 Acceso a la aplicación

Posteriormente debemos dar clic en el botón Aceptar y accedemos a la siguiente interfaz que permanecerá por unos instantes.

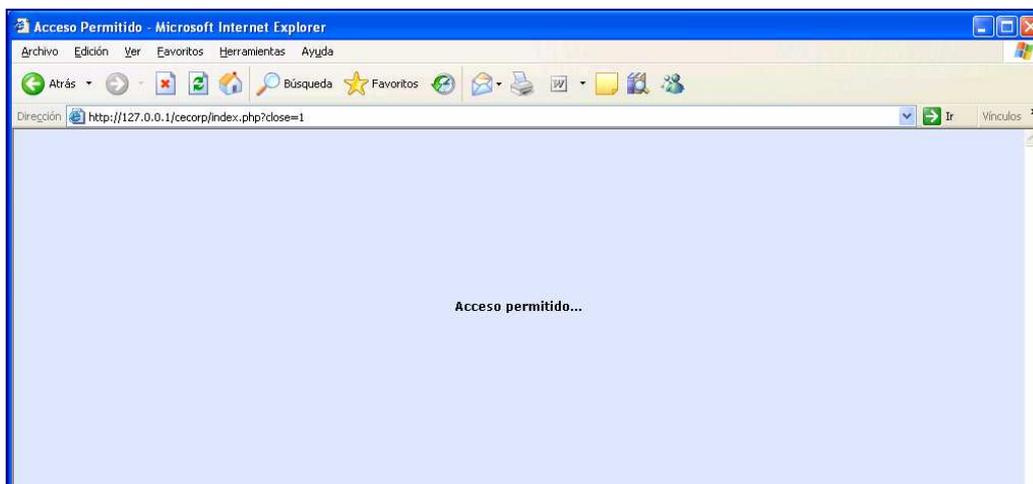


Figura 3.2 Acceso permitido

A continuación se despliega la pantalla principal en la cual encontramos información general de la Corporación Educación y Cultural 4 de Marzo, en la cual observamos el panel de menú anteriormente mencionada.

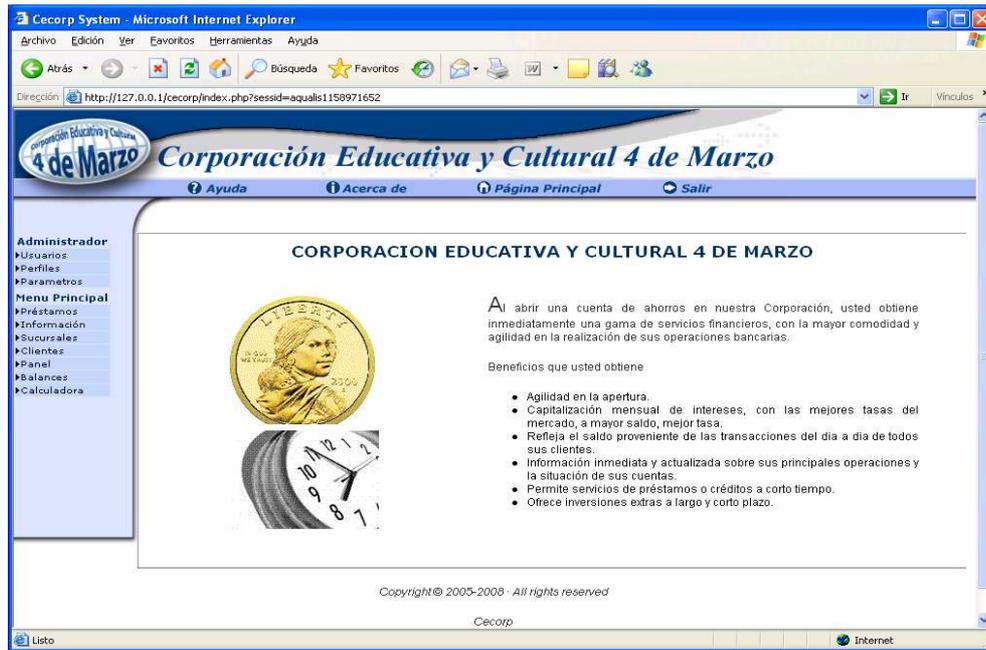


Figura 3.3 Información de la Corporación

4. ADMINISTRACIÓN DE PERFILES

En este módulo puedo ingresar los perfiles de usuario, que requiera el administrador en nuestra aplicación nuestro primer perfil de usuario es Administrador con el id 1 y su respectivo detalle, y puede realizar funciones como agregar (nuevo), guardar, borrar, buscar, e imprimir.

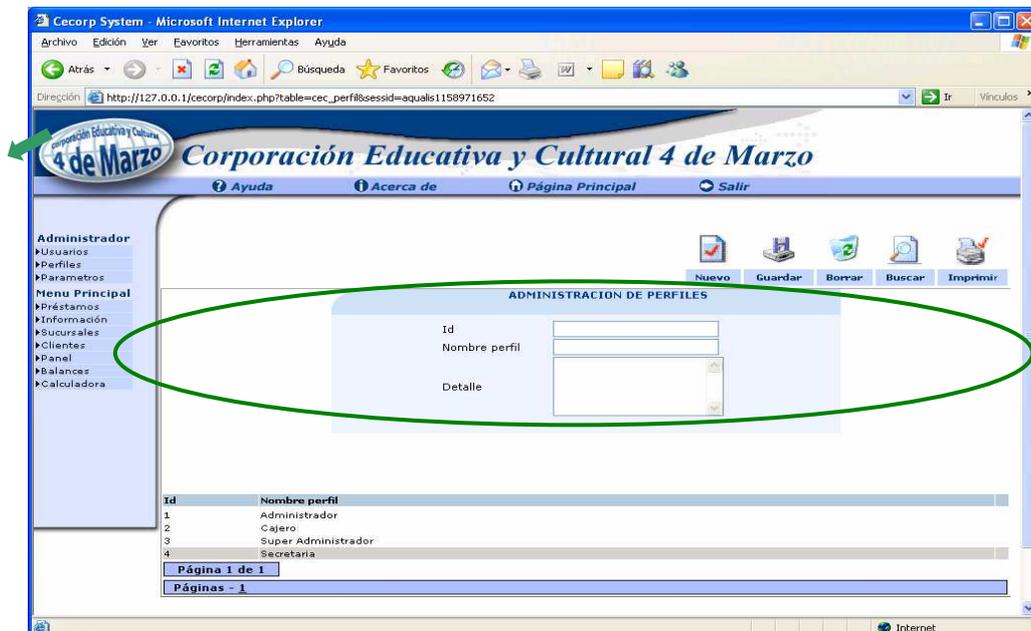


Figura 4.1. Módulo perfiles de usuario

Para ver la información detallada de los perfiles debemos dar clic en la información requerida.

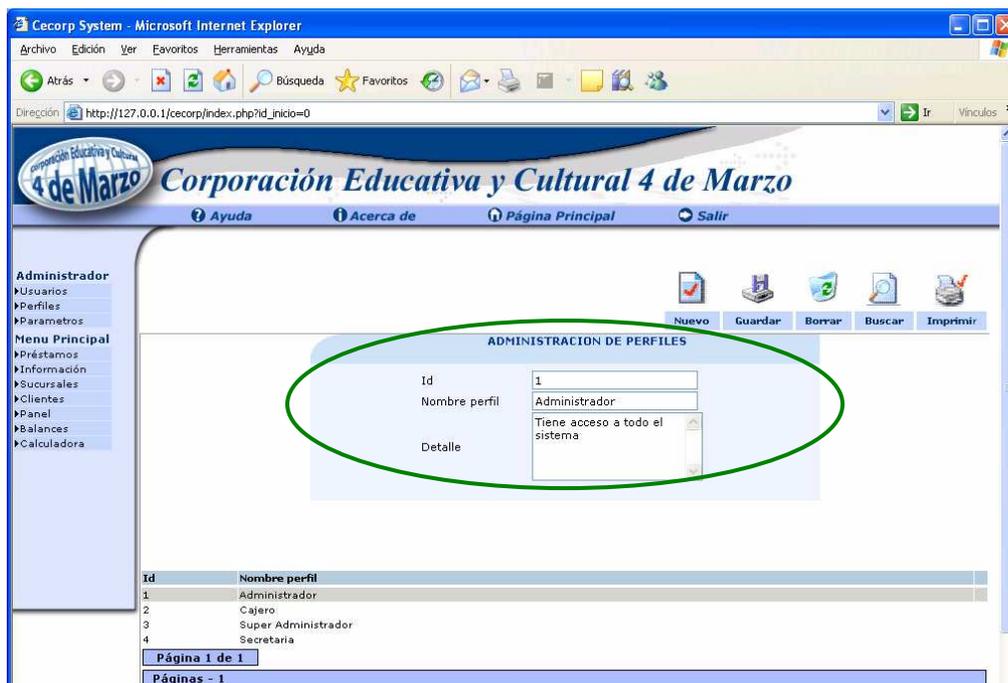


Figura 4.2 Ingreso de perfiles de usuario

De igual manera debo ingresar cada uno de los perfiles de usuarios, una vez ingresados puedo realizar una búsqueda poniendo el cursor sobre dicho botón y dando un clic en Buscar.



Al realizar una búsqueda nos presente la siguiente interfaz en la cual observamos los campos que podemos acceder a una búsqueda esto depende del módulo a ingresar.

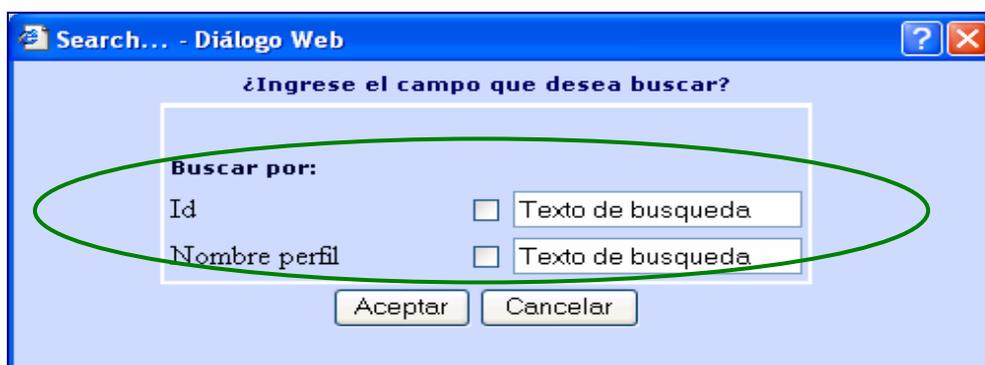


Figura 4.3 Búsqueda de perfil

Debemos dar un check en el campo de búsqueda e ingresamos el registro.

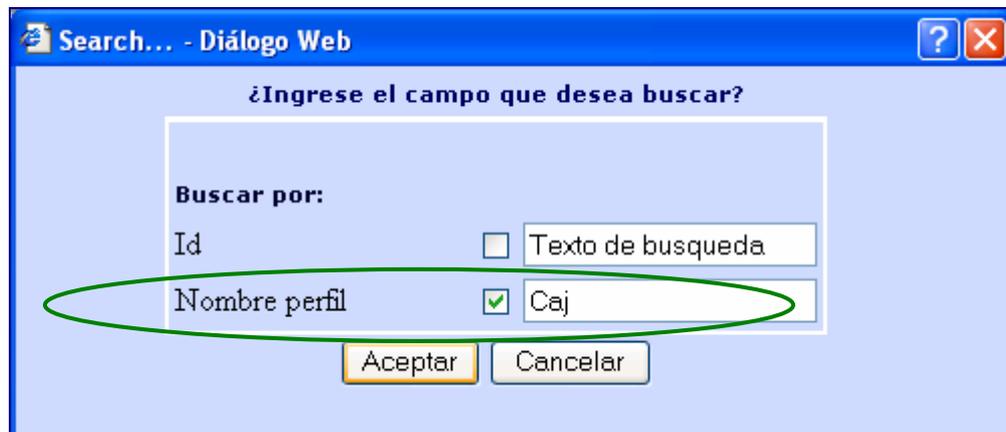


Figura 4.4 Búsqueda de cajeros

Clic en Aceptar y a continuación nos despliega el informe a imprimir.

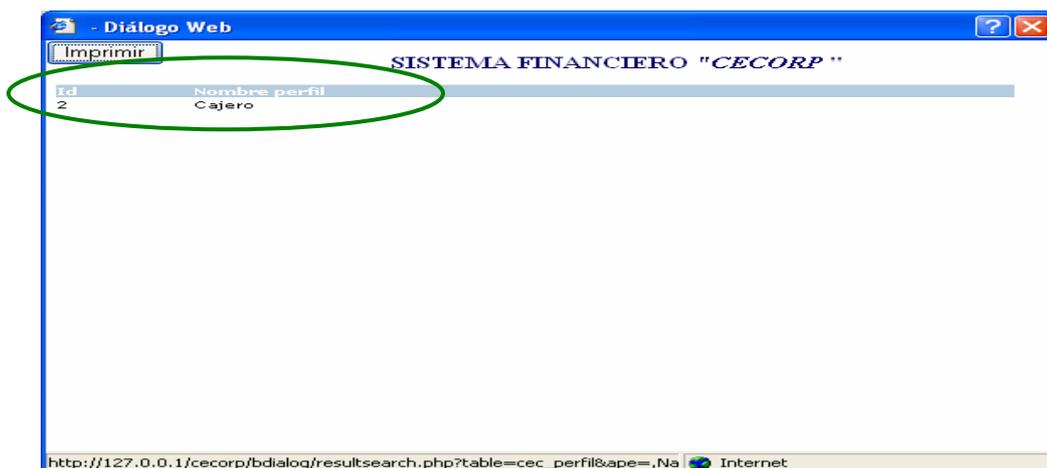


Figura 4.5 Resultado de búsqueda

Una vez realizada la búsqueda puedo imprimir el documento, para ellos debo definir la impresora

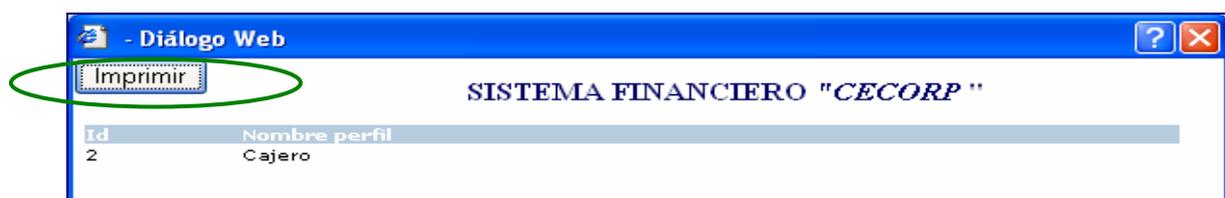


Figura 4.6 Imprimir

En caso de no encontrar la impresora predeterminada de clic en Buscar impresora local o red y presione la opción Aceptar.

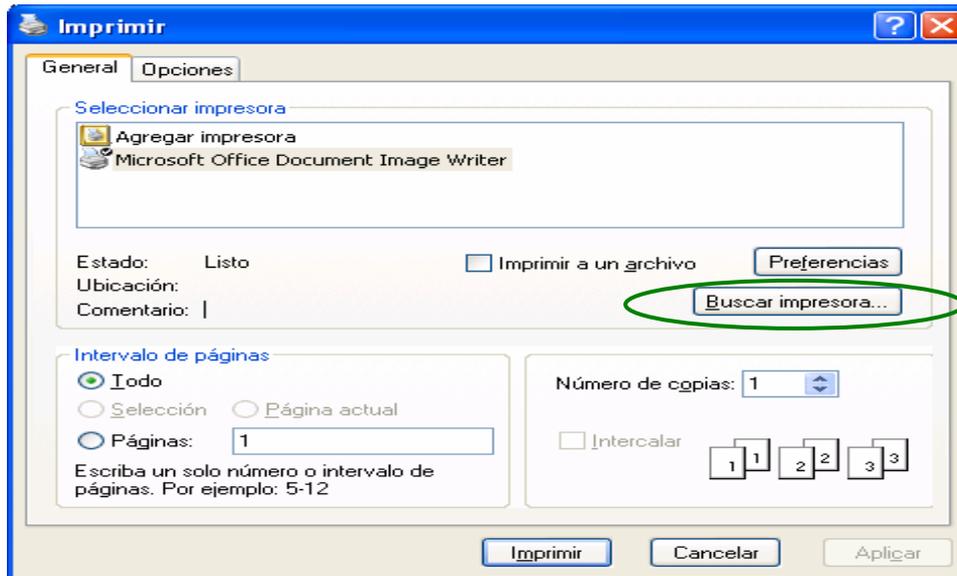


Figura 4.7 Búsqueda de la impresora

Debemos digitar el nombre de la impresora y la opción Aceptar.

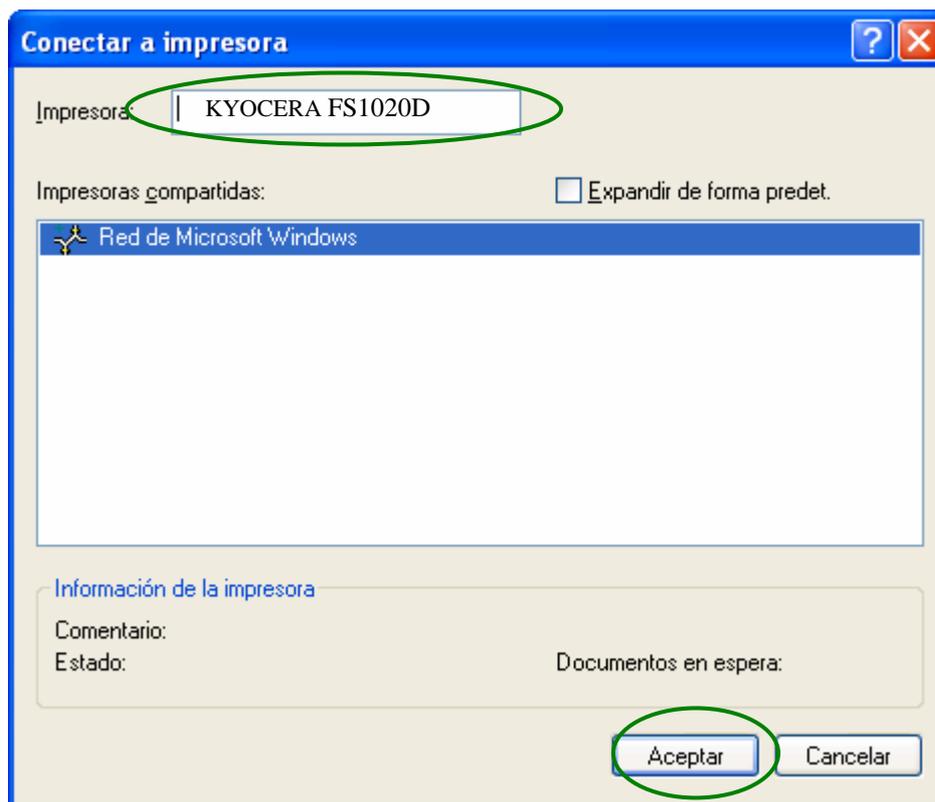


Figura 4.8 Conexión de la impresora seleccionada

5. ADMINISTRACIÓN DE USUARIOS

Para ingresar a este modulo debo colocarme en paneles y acceder a la opción Usuarios

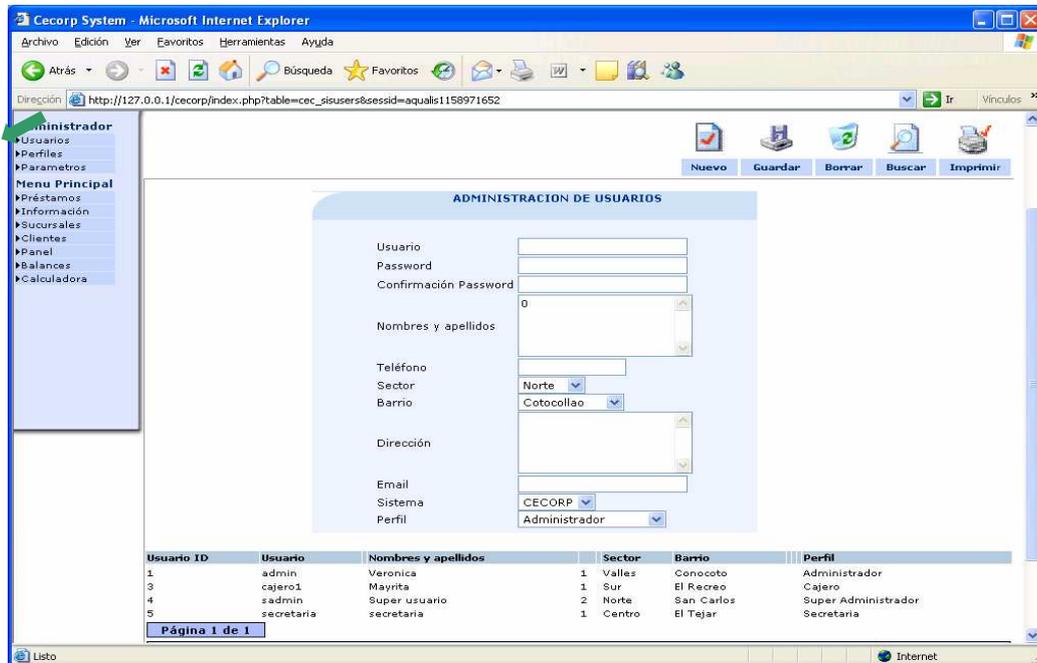


Figura 5.1 Módulo de usuarios

Para crear un usuario nuevo debo ingresar el nombre de usuario, password, etc es importante escoger que clase de perfil de usuario, ya que esto restringirá las opciones de la aplicación, por ejemplo el usuario `admin`. con la password `admin`. tiene el perfil de usuario Administrador en cual tiene acceso a todo la aplicación y puede realizar cambios de información.

Para ver información de los clientes debo dar clic en lista desplegada en la parte inferior.

Usuario ID	Usuario	Nombres y apellidos	Sector	Barrio	Perfil
1	admin	Veronica	1	Valles	Administrador
3	super	Super usuario	2	Norte	Super Administrador
4	sadmin	secretaria	1	Centro	Secretaria
5	secretaria	secretaria			

Figura 5.2 Ingreso de usuarios

Para realizar una búsqueda definida debo ingresar el campo a buscar por ejemplo el % busca todos los campos.

¿Ingrese el campo que desea buscar?

Buscar por:

Usuario ID Texto de búsqueda
 Usuario %
 Nombres y apellidos Texto de búsqueda
 Sector Texto de búsqueda
 Barrio Texto de búsqueda
 Perfil Texto de búsqueda

Aceptar Cancelar

Figura 5.3 Búsqueda de todos los usuarios

A continuación se despliega la búsqueda el cual nos permite imprimir dicho reporte.

Podemos realizar una búsqueda de los campos de parámetros para ver información.

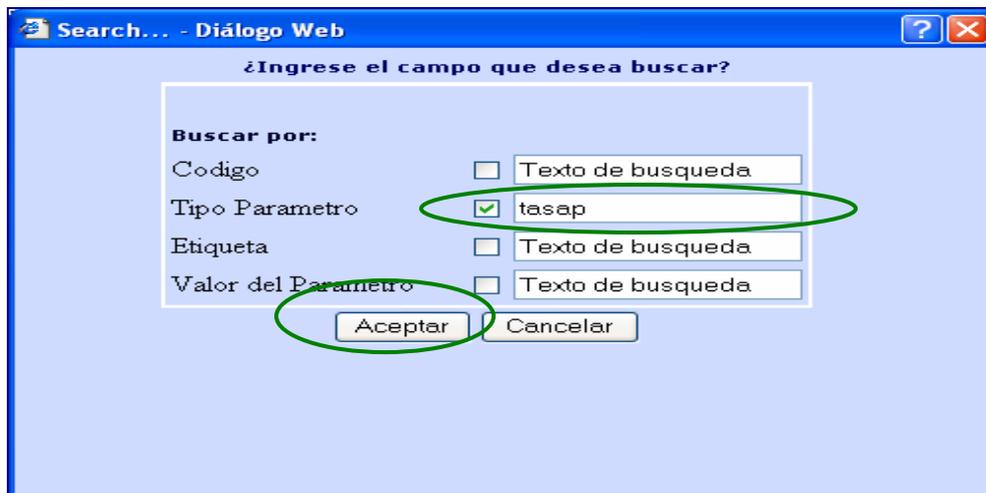


Figura 6.2 Búsqueda de parámetros

A continuación se desplegar los datos a buscar

Al encontrar el registro en una lista seleccionamos el campo solicitado y regresamos a la pantalla anterior en la que se despliega la información a la cual realizare algún cambio.

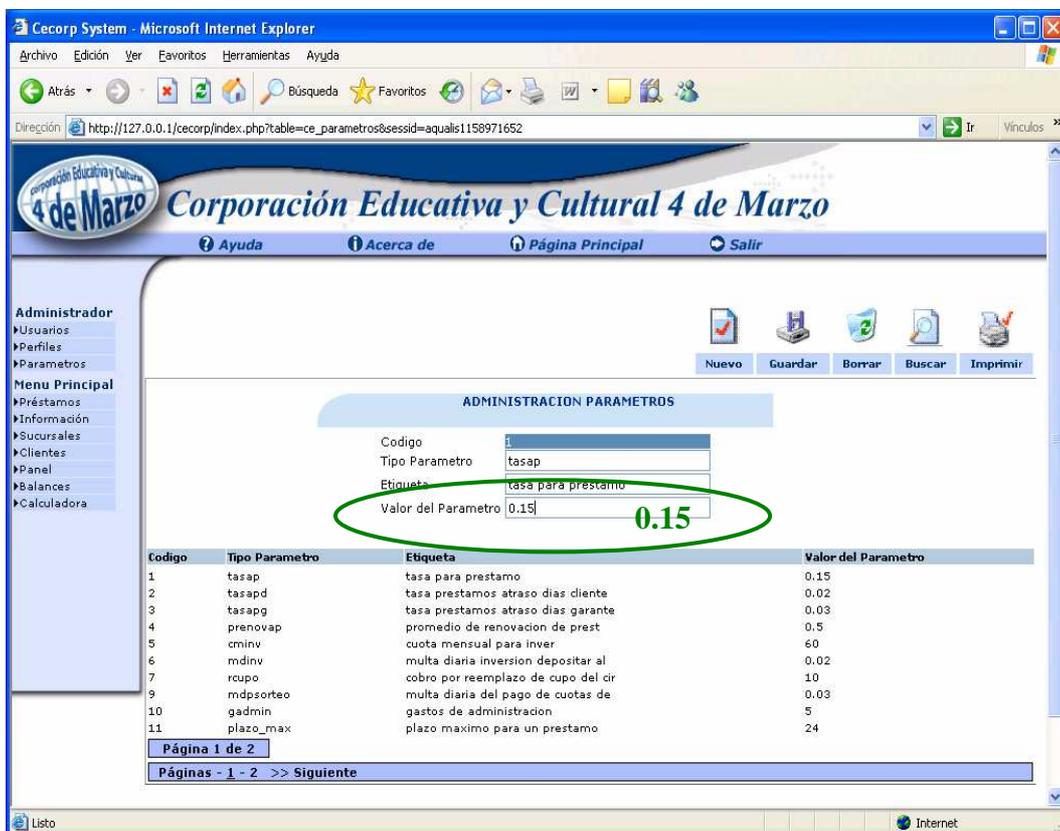


Figura 6.3 Despliegue de búsqueda

Para realizar cambio en el valor de parámetros u otros cambios, debo sobrescribir la información y dar clic en el Botón Guardar, para que el campo se actualice utilizaremos el mismo botón guardar tanto para un registro nuevo como actualizado.

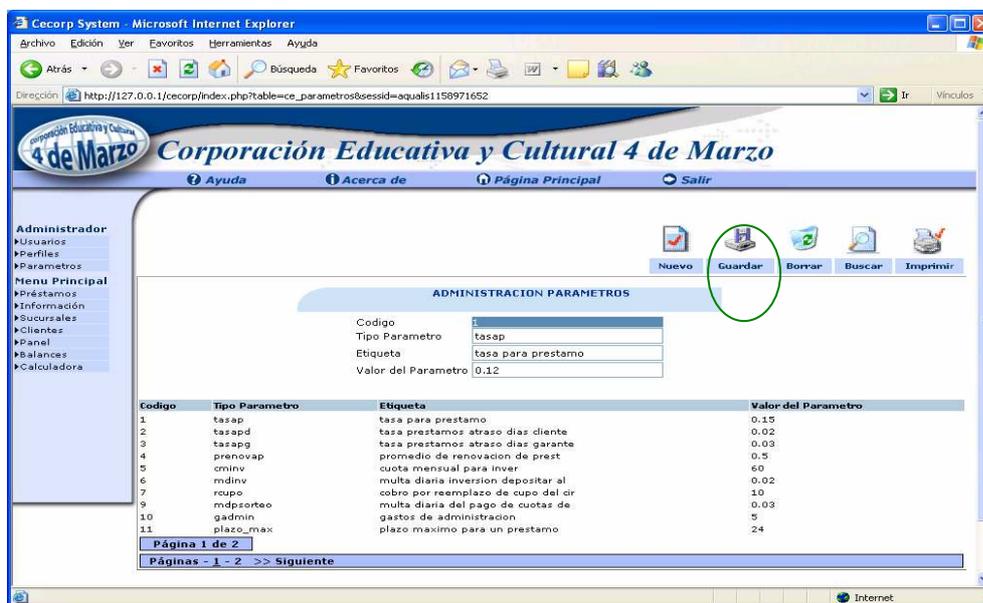


Figura 6.4 Guardar actualización

7. ADMINISTRAR SUCURSALES

El módulo Sucursales almacena la información de las Agencias que posee la Corporación e información de la misma.

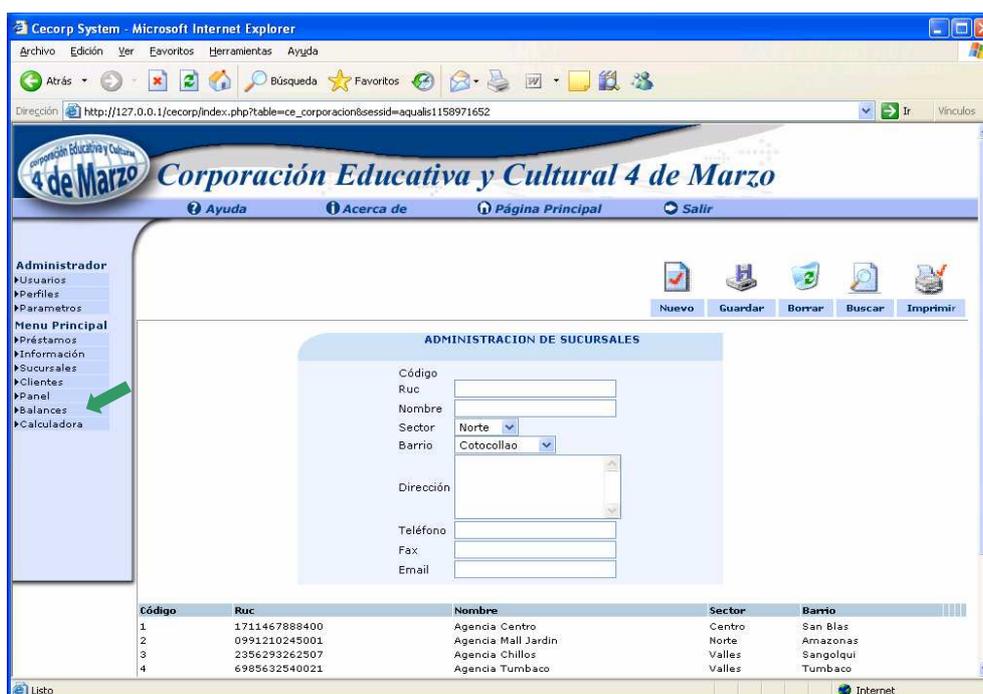


Figura 7.1 Módulo sucursales

Uno de los campos que se debe registrar es el campo Sector y el campo Barrio ya que por medio de estos cambios se debe identificar algunos clientes que realizan transacciones en Agencias diferentes.

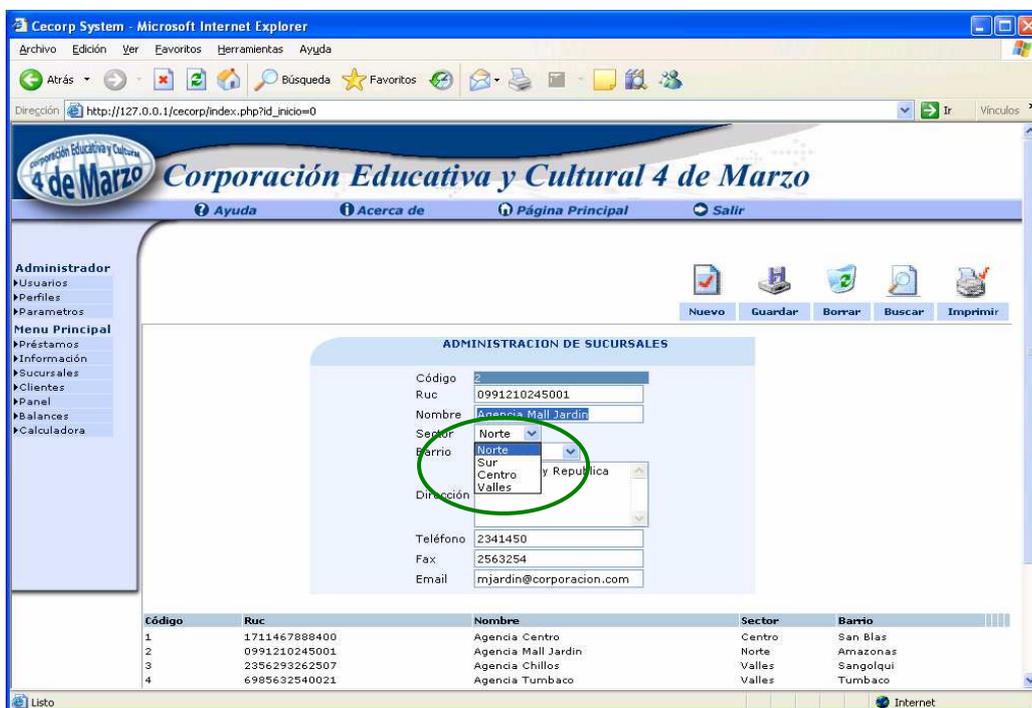


Figura 7.2 Ingreso de sucursales

De igual manera podemos realizar búsquedas sin necesidad de ingresar todo el dato del campo a buscar siempre y cuando lo active con un check.

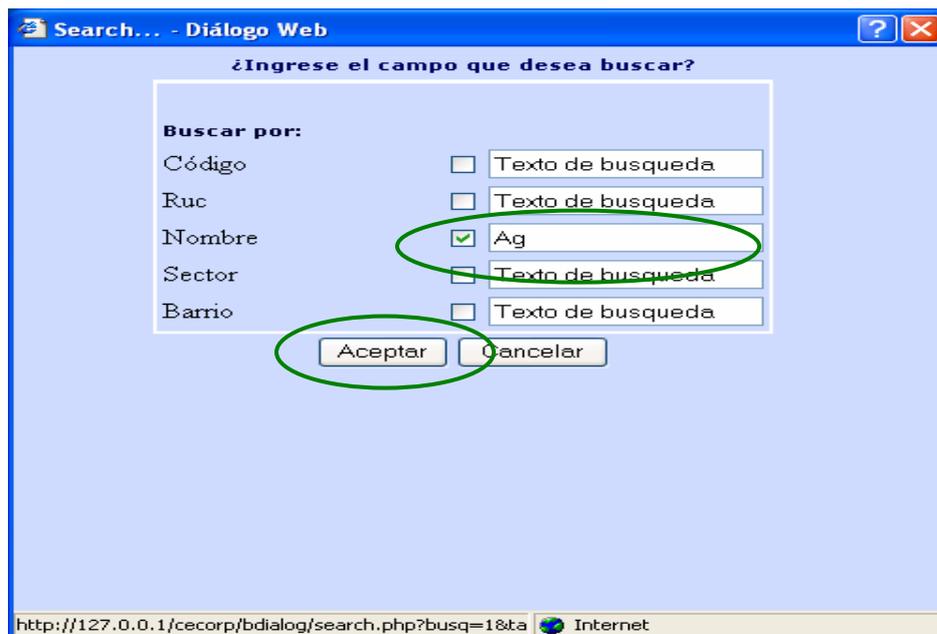
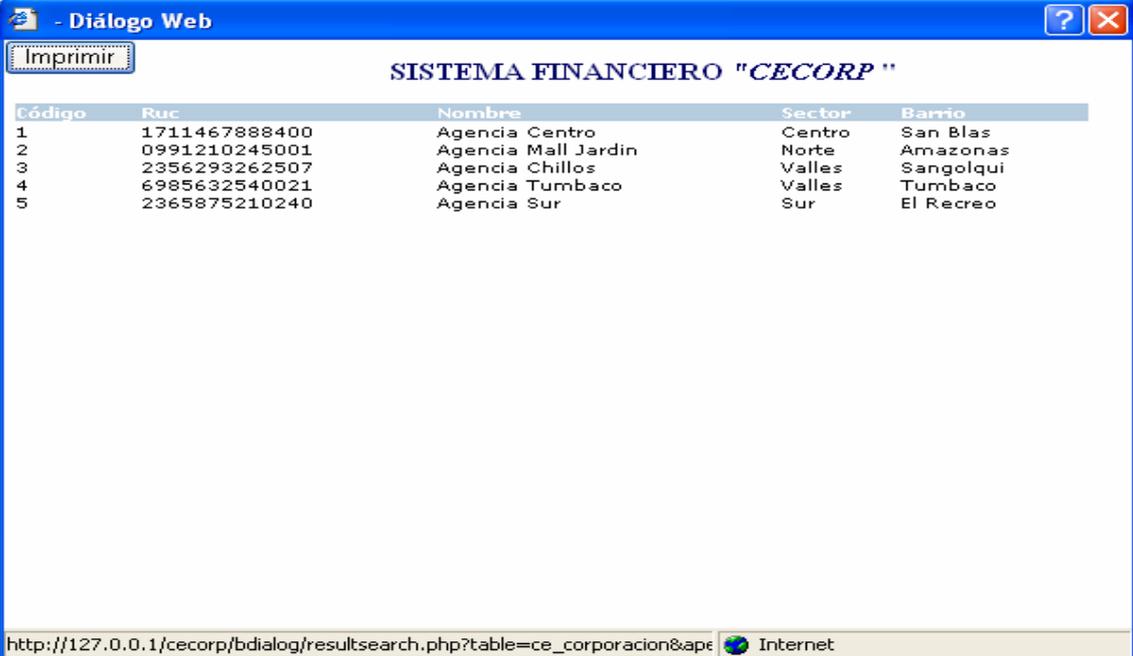


Figura 7.3 Búsqueda de sucursales

De igual manera despliega toda la información a buscar el cual me permite imprimir la información requerida.

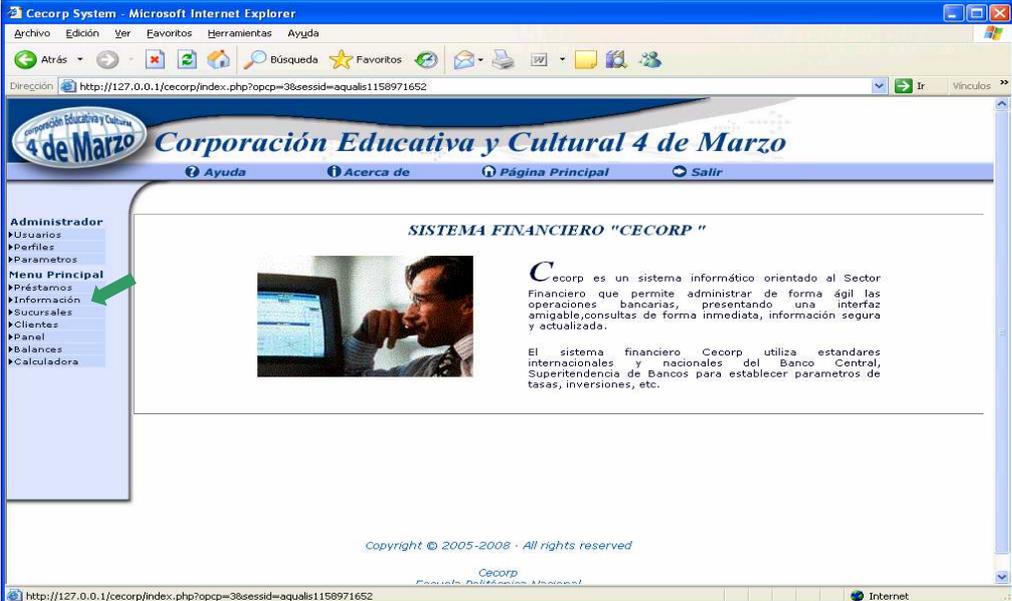


Código	Ruc	Nombre	Sector	Barrio
1	1711467888400	Agencia Centro	Centro	San Blas
2	0991210245001	Agencia Mall Jardin	Norte	Amazonas
3	2356293262507	Agencia Chillos	Valles	Sangolqui
4	6985632540021	Agencia Tumbaco	Valles	Tumbaco
5	2365875210240	Agencia Sur	Sur	El Recreo

Figura 7.4 Resultado de búsqueda

8. INFORMACIÓN GENERAL

Este módulo permite observar información de Cecorp de la aplicación Web dando clic en el menú de la parte derecha de la internas.



Administrador

- Usuarios
- Perfiles
- Parametros

Menu Principal

- Préstamos
- Información
- Seguros
- Cuentas
- Panel
- Balances
- Calculadora

SISTEMA FINANCIERO "CECORP"

Cecorp es un sistema informático orientado al Sector Financiero que permite administrar de forma ágil las operaciones bancarias, presentando una interfaz amigable, consultas de forma inmediata, información segura y actualizada.

El sistema financiero Cecorp utiliza estándares internacionales y nacionales del Banco Central, Superintendencia de Bancos para establecer parametros de tasas, inversiones, etc.

Copyright © 2005-2008 · All rights reserved

Figura 8.1 Información

9. ADMINISTRAR CLIENTE

En este módulo debemos ingresar los datos personales de nuestros clientes para ellos debemos dar clic en el menú clientes.

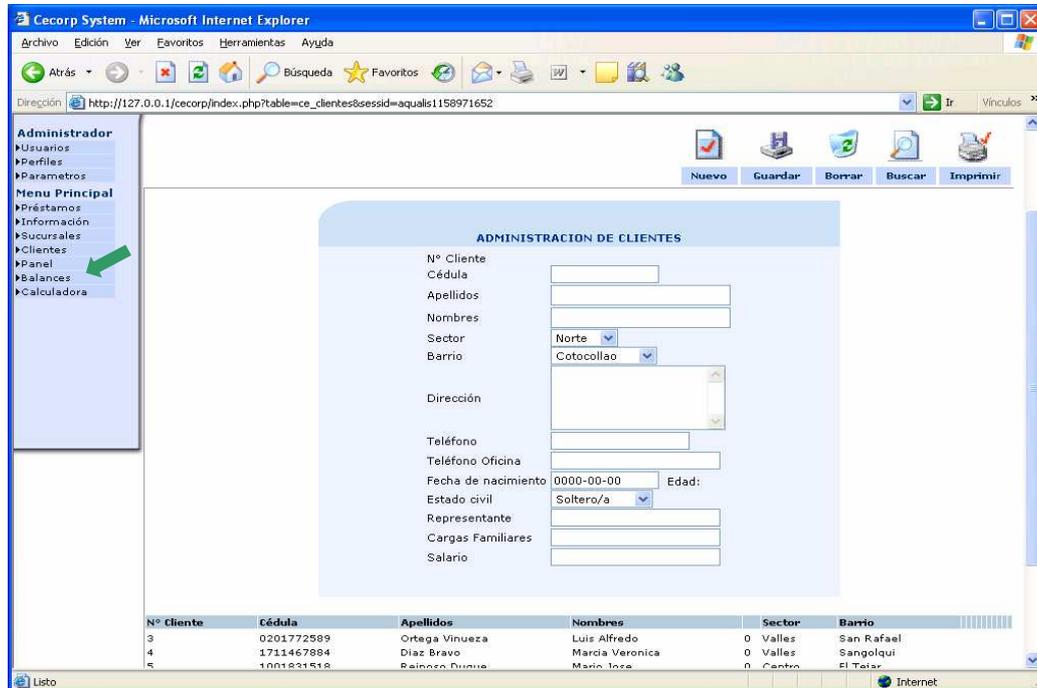


Figura 9.1 Módulo clientes

Es importante manejar una búsqueda definida ya que podrían existir muchos registros y sería imposible buscar de uno en uno así que el campo mas de búsqueda es el N° de cedula u otra forma es ingresar los 3 primeros dígitos.

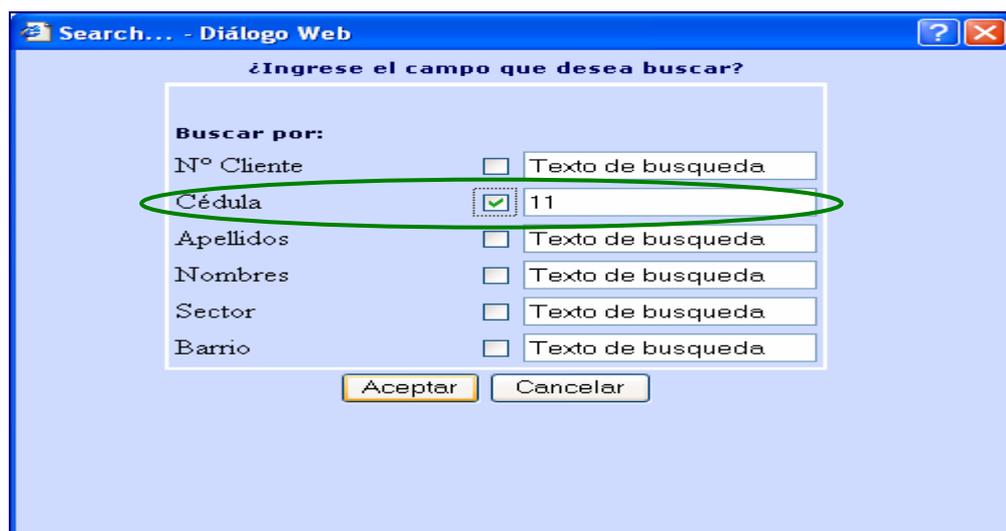


Figura 9.2 Búsqueda de clientes por el N° de cédula

Se presenta una interfaz con los campos de búsqueda si deseo imprimir debo dar clic en Imprimir o en caso de actualizar algún campo debo dar doble clic al registro mostrado para regresar a la interfaz anterior.



Figura 9.3 Resultado de la búsqueda

Al crear un cliente debo guardarlo y a continuación nos despliega la siguiente interfaz en la que observamos dos link que son CUENTAS y PRESTAMOS ya que nuestro cliente puede acceder a estas opciones siempre y cuando cumplan con las condiciones establecidas por la Corporación.

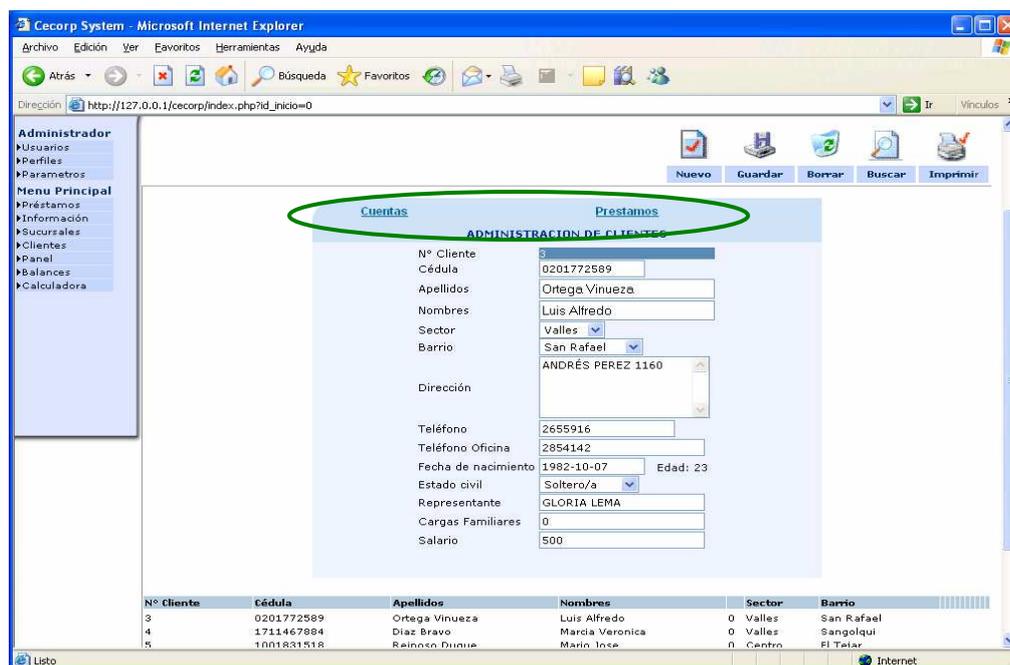


Figura 9.4 Cliente guardado

10. ADMINISTRAR CUENTAS

El módulo de cuentas se activa cuando ya he señalado a un cliente y debo dar clic en Cuentas para generar un número de cuenta junto con su N° de cédula van hacer sus identificaciones personales.

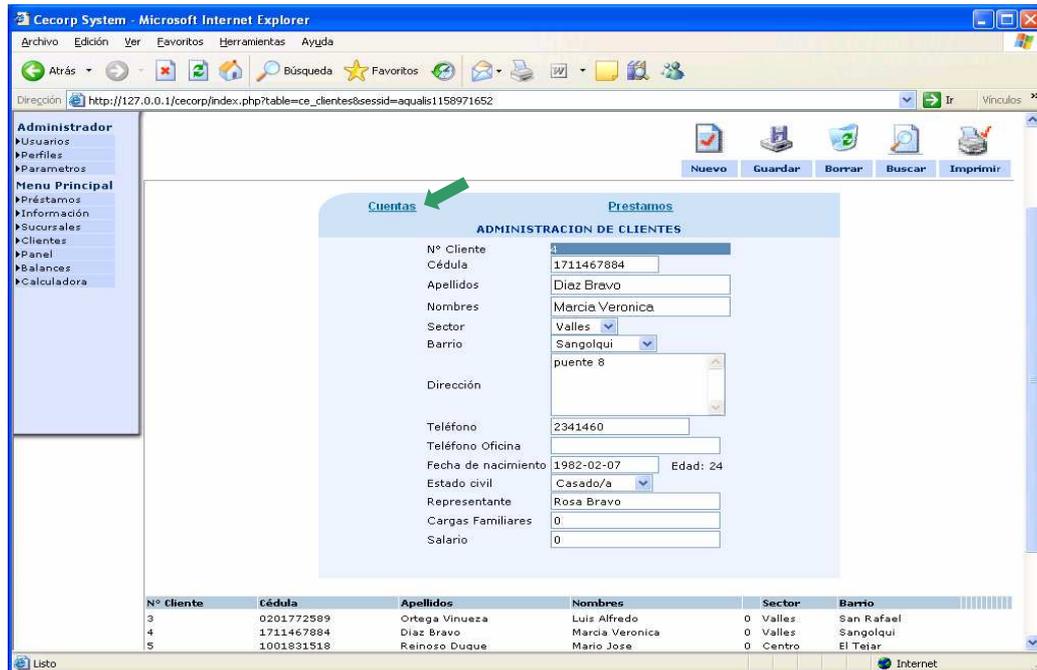


Figura 10.1 Acceso a cuentas

Asigno un número de cuenta e ingreso los demás campos y guardo dicha información.

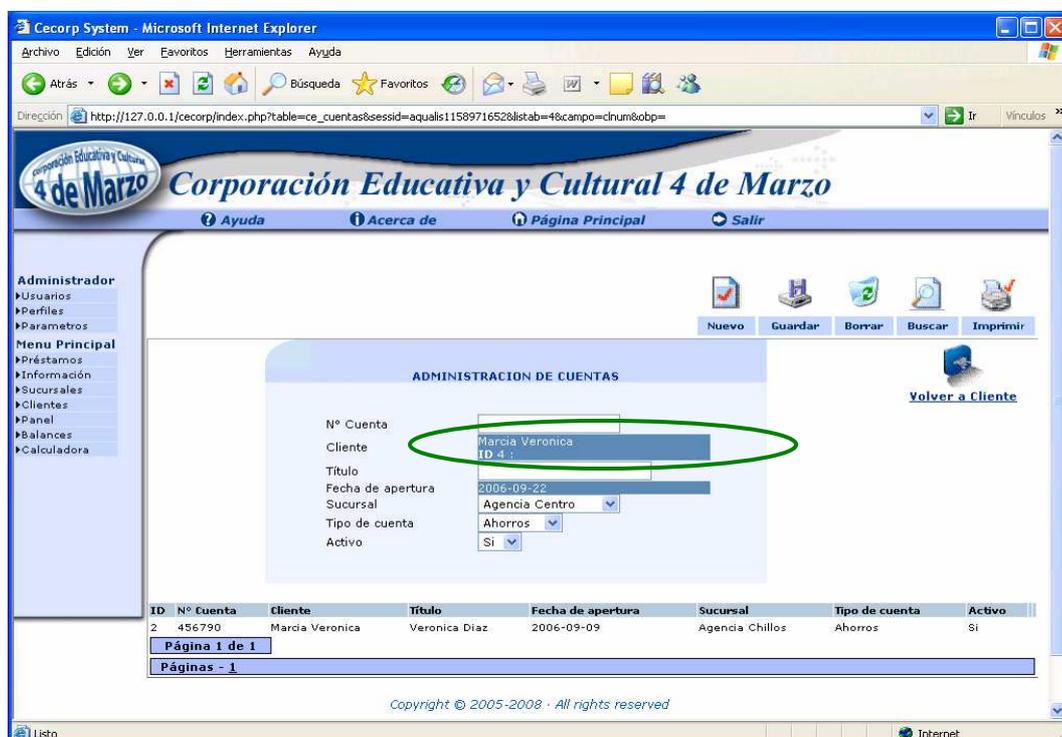


Figura 10.2 Asignación de cuenta

En caso de no guardar la información no aparecerá en la lista de la parte inferior en la cual debo señalarla para que dicha información se despliegue en cada campo de presentación, y se activará el link Detalle de Cuenta

Para registrar cada una de las transacciones realizadas, debo utilizar este módulo ya que es muy importante tener cuidado al ingresar, eliminar o modificar un campo por que se maneja el estado económico de los clientes.

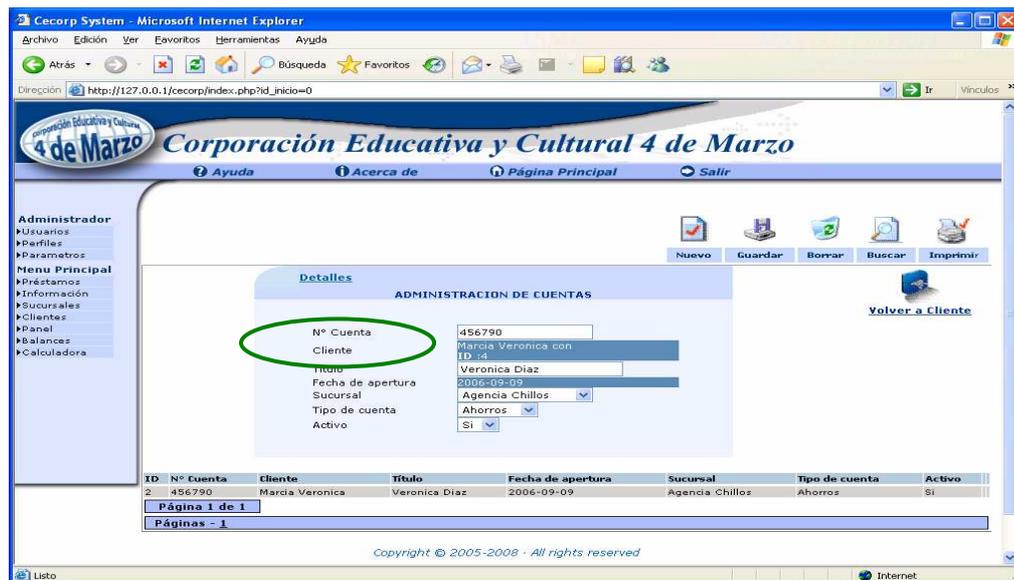


Figura 10.3 Visualización de cuenta

Al acceder al módulo Detalle de Cuenta nos despliega los datos como saldos, intereses que automáticamente se generan con las formulas establecidas en la programación.

Para registrar los movimientos de transacciones debo señalar si es un depósito, retiro siempre y cuando el valor de la movimiento no sea mayor al saldo y este a su vez si en cheque o en efectivo, calcula el total del movimiento.

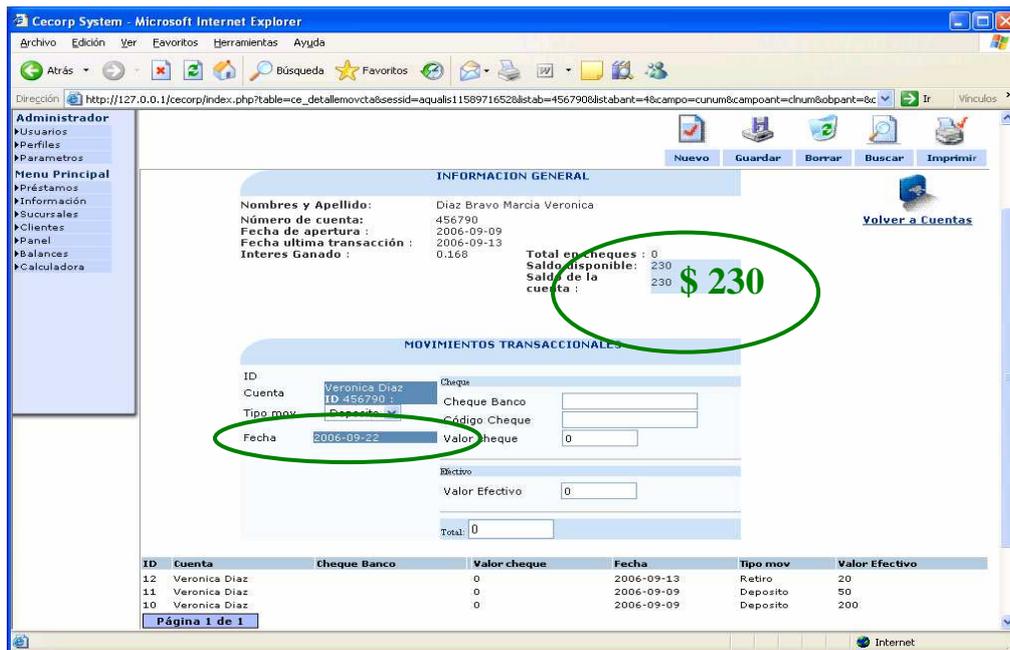


Figura 10.4 Depósito

Al momento que este seguro de los datos a ingresar debo guardar y actualizar la aplicación para que aparezca el nuevo saldo ya actualizado.

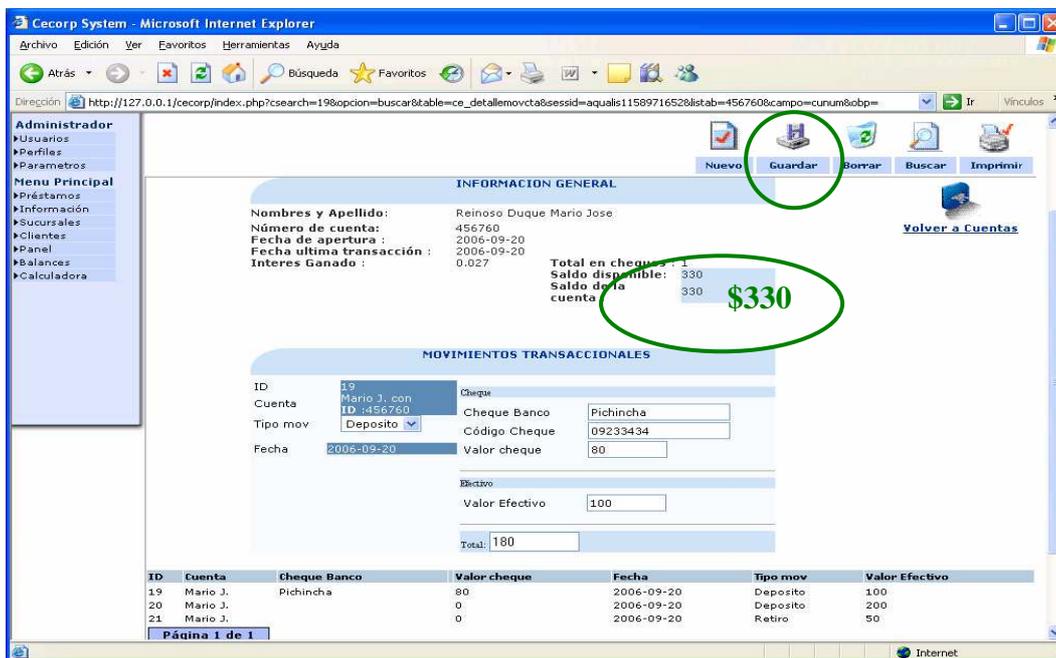


Figura 10.5 Guardar depósito

Para buscar un campo ya sea por id del cheque o nombre debo ingresar por lo menos 1 digito de búsqueda.

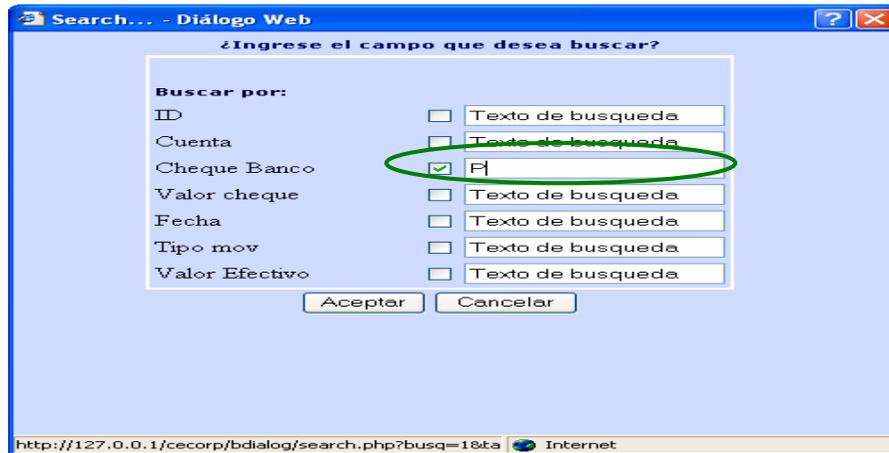


Figura 10.6 Búsqueda de cheque de banco

Por ejemplo: realizamos una búsqueda con el dígito P y nos despliega la siguiente interfaz con los campos detallados.

ID	Cuenta	Cheque Banco	Valor cheque	Fecha	Tipo mov	Valor Efectivo
8	456789	Mutualista Pichincha	300	2006-09-09	Deposito	200
13	456789	Pacifico	200	2006-09-10	Deposito	100
19	456760	Pichincha	80	2006-09-20	Deposito	100
25	456789	Pacifico	50	2006-09-21	Deposito	70

Figura 10.7 Resultado de búsqueda

11. ADMINISTRAR PRÉSTAMOS

Para acceder al módulo de préstamos damos clic en el link de Préstamos debemos seleccionar el cliente a consultar.

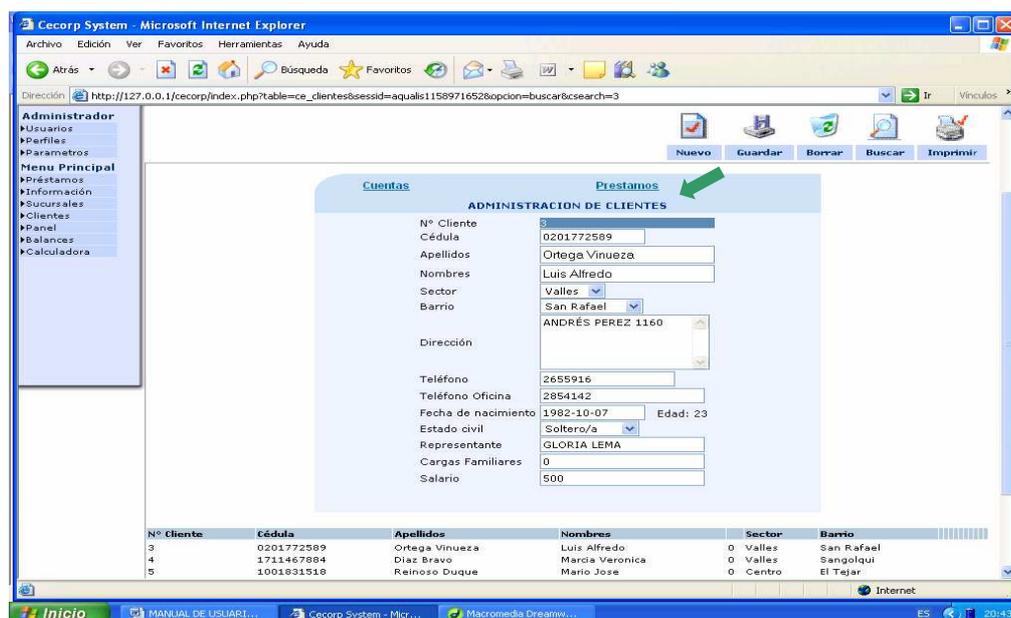


Figura 11.1 Módulo de préstamos

Para registrar el préstamo debemos ingresar el monto, fecha de solicitud, y el estado del préstamo que puede ser pendiente, aprobado, o negado, debo ingresar el número de meses a pagar (dividendos).

ADMINISTRACION DE PRESTAMOS

Nº Préstamo:

Monto:

Fecha Solicitud:

Fecha Aprobada:

Estado préstamo:

Dividendos:

Sucursal:

Nº Cliente:

Estado del Préstamo: Pendientes: 0 Mora: 0 Pagado: 0

Calcular Monto de Préstamo

Monto: \$

Plazo: 0.0 años

Interés Anual: 14 %

Periodos: mensual

Nº Préstamo	Monto	Fecha Solicitud	Fecha Aprobada	Estado préstamo	Dividendos	Sucursal	Nº Cliente
Página 1 de 0							

Copyright © 2005-2008 - All rights reserved

ADMINISTRACION DE PRESTAMOS

Nº Préstamo:

Monto:

Fecha Solicitud:

Fecha Aprobada:

Estado préstamo:

Dividendos:

Sucursal:

Nº Cliente:

Estado del Préstamo: Pendientes: 0 Mora: 0 Pagado: 0

Figura 11.2 Registrar préstamo

Todos los clientes pueden tener préstamos pero si no acceden a uno se encuentra en estado pendiente.

ADMINISTRACION DE PRESTAMOS

Nº Préstamo:

Monto:

Fecha Solicitud:

Fecha Aprobada:

Estado préstamo:

Dividendos:

Sucursal:

Nº Cliente:

Estado del Préstamo: Pendientes: 0 Mora: 0 Pagado: 0

Calcular Monto de Préstamo

Monto: \$

Plazo: 0.0 años

Interés Anual: 14 %

Periodos: mensual

Nº Préstamo	Monto	Fecha Solicitud	Fecha Aprobada	Estado préstamo	Dividendos	Sucursal	Nº Cliente
Página 1 de 0							

Copyright © 2005-2008 - All rights reserved

Figura 11.3 Estado del préstamo

Para aprobar un crédito debo digitar la fecha aprobada y guardar

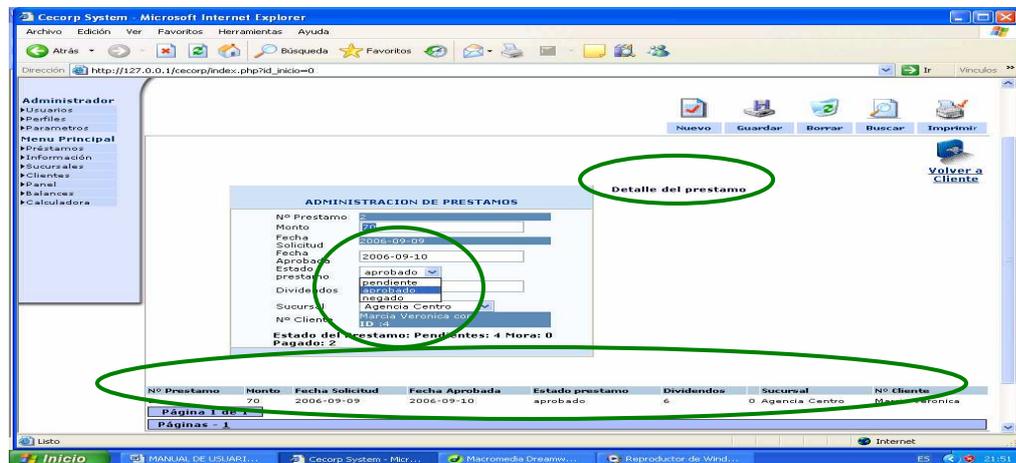


Figura 11.4 Fecha del crédito

Para calcular los detalles debo ingresar los datos requeridos clic en el botón Calcular para generar una Tabla de Amortizaciones.

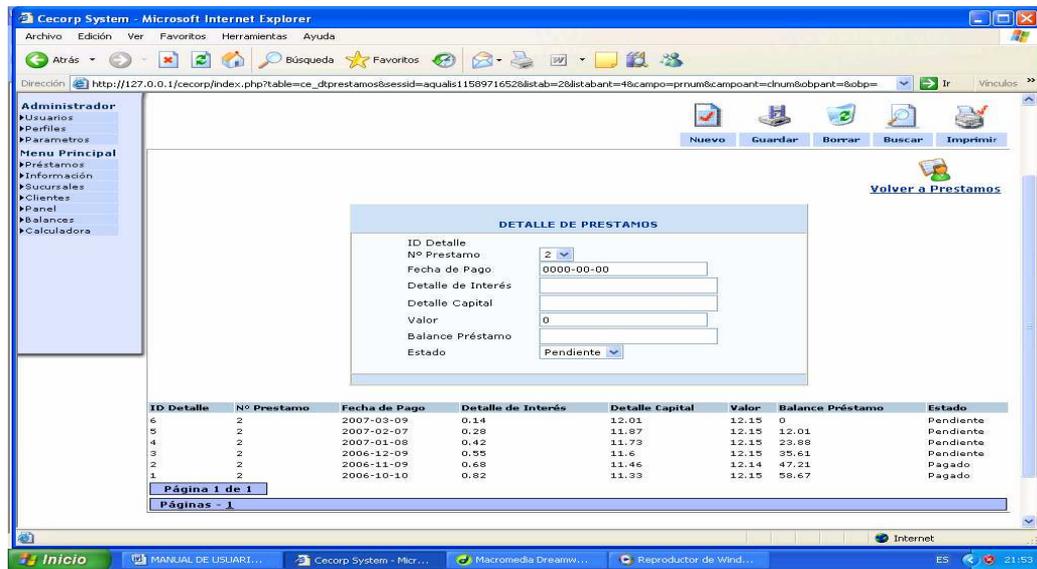


Figura 11.5 Generar amortizaciones

En la parte inferior encontraremos los dividendos describiendo la cuota a pagar mensualmente, los intereses por pagar, y el estado del detalle que puede ser pagado, pendiente o por mora.

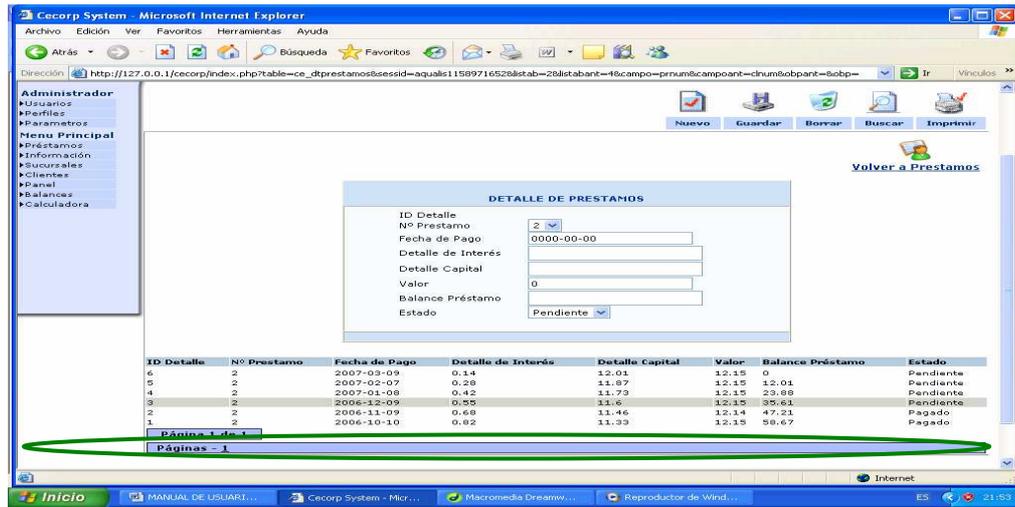


Figura 11.6 Datos del préstamo

Para cambiar el estado del detalle debo seleccionar la cuota a pagar y cambiar de estado ha pagado

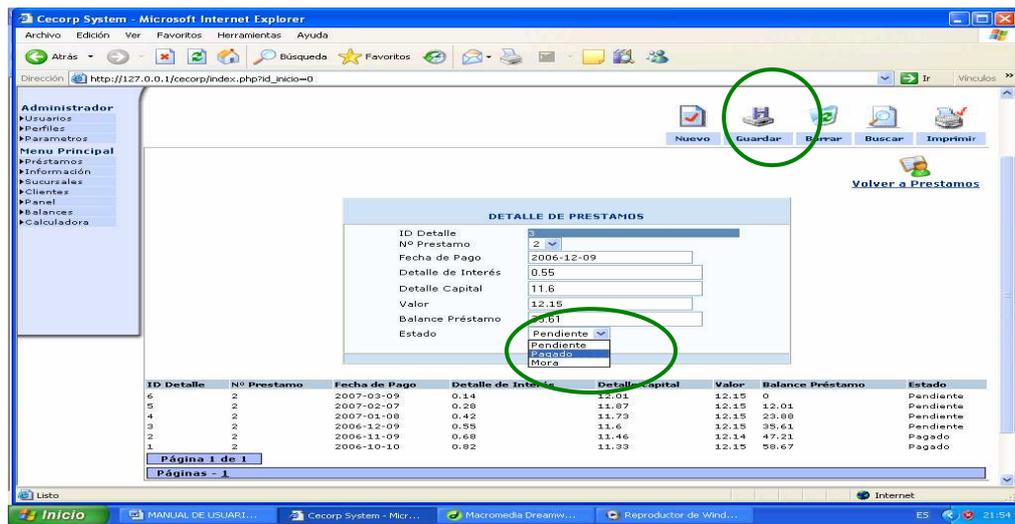


Figura 11.7 Cambio estado detalle

Al momento de cambiar el estado del detalle de préstamo se actualizará las cuotas pendientes, interés por pagar, y en caso de interés por mora cuanto se acumulado al total de la cuota pendiente.

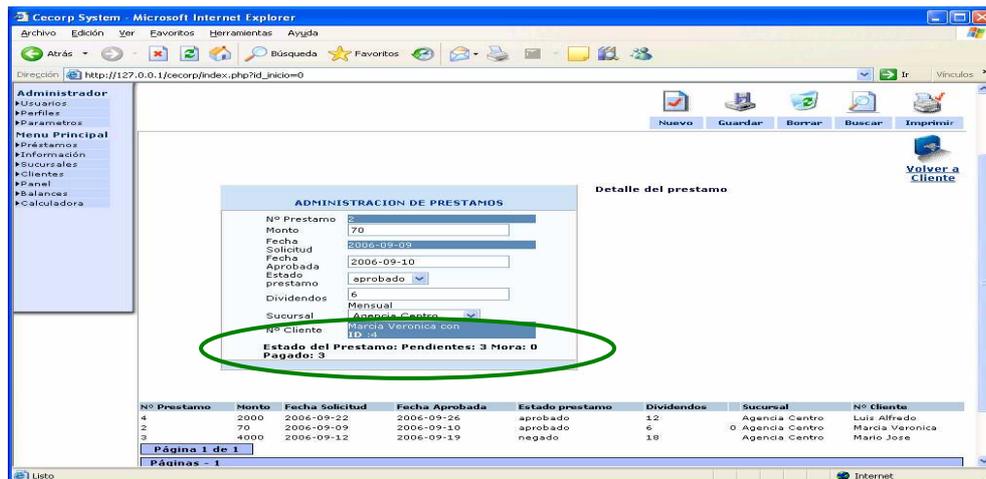


Figura 11.8 Actualización cuotas pendientes

12. ACCESO A PRÉSTAMOS

Esta interfaz nos ayuda a realizar una búsqueda rápida ingresando el código de cliente o a su vez el código del préstamo que esta información de ser impresa en la libreta del cliente.

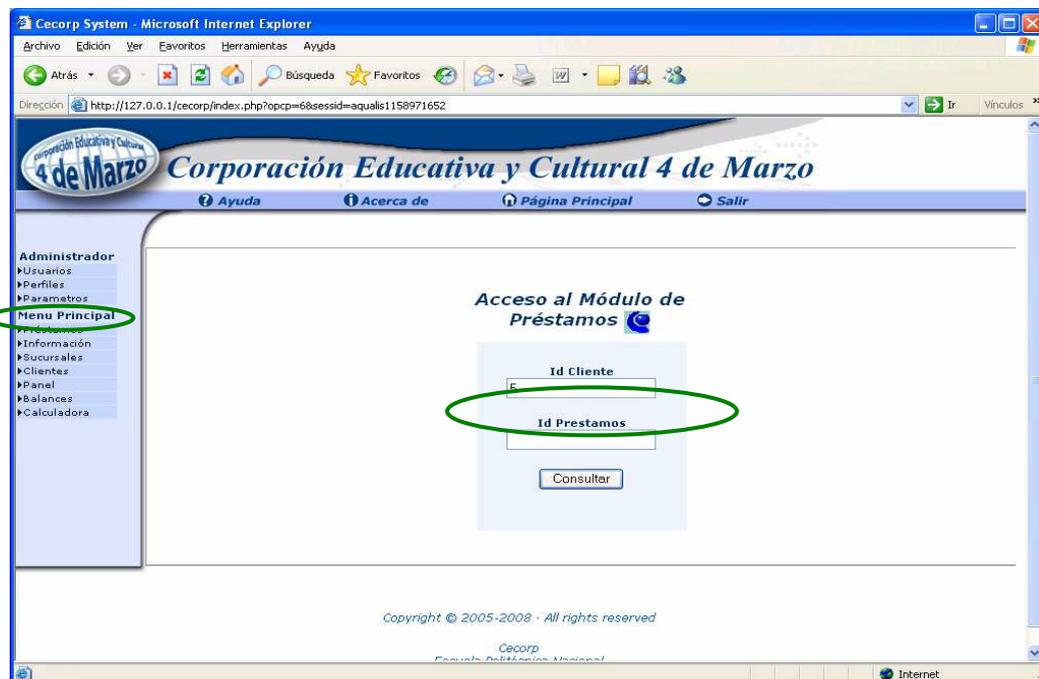


Figura 12.1 Búsqueda de préstamos

En caso de que el crédito sea negado debe constar fecha de solicitud y la fecha aprobada que en este caso será la fecha negada, y debo guardar dicha información.

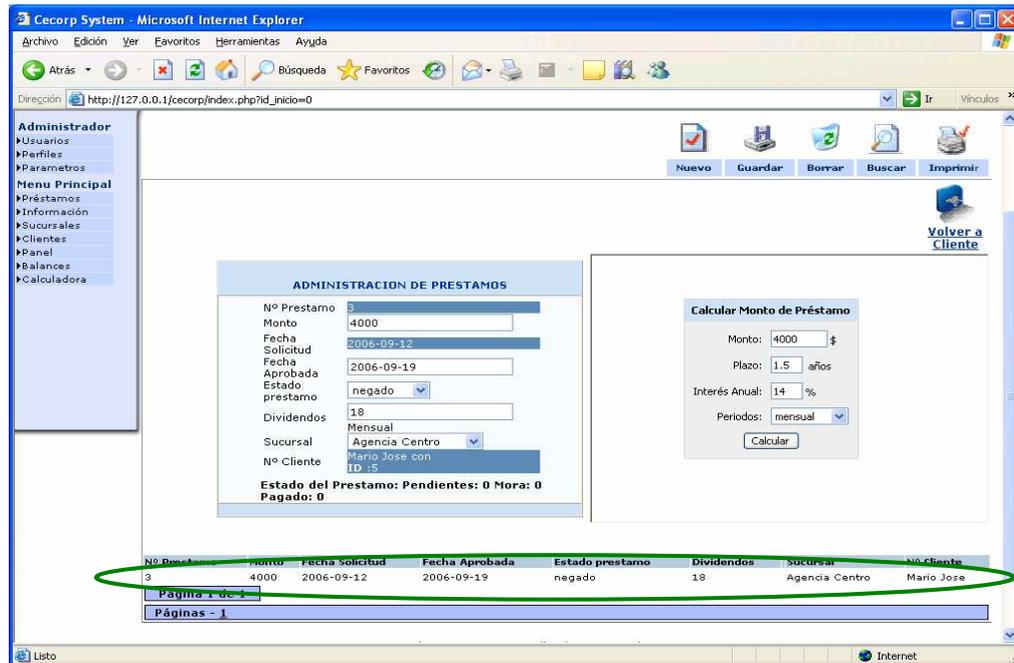


Figura 12.2 Crédito negado

13. CALCULAR PRESTAMO

Este módulo permite calcular intereses a pagar con cuotas fijas, sin necesidad de guardar dicha consulta.

Primeramente se ingresa el monto de crédito, el plazo en años, el interés anual que cobra la Corporación este % debe ser establecido por el Banco Central y Superintendencia de Bancos.

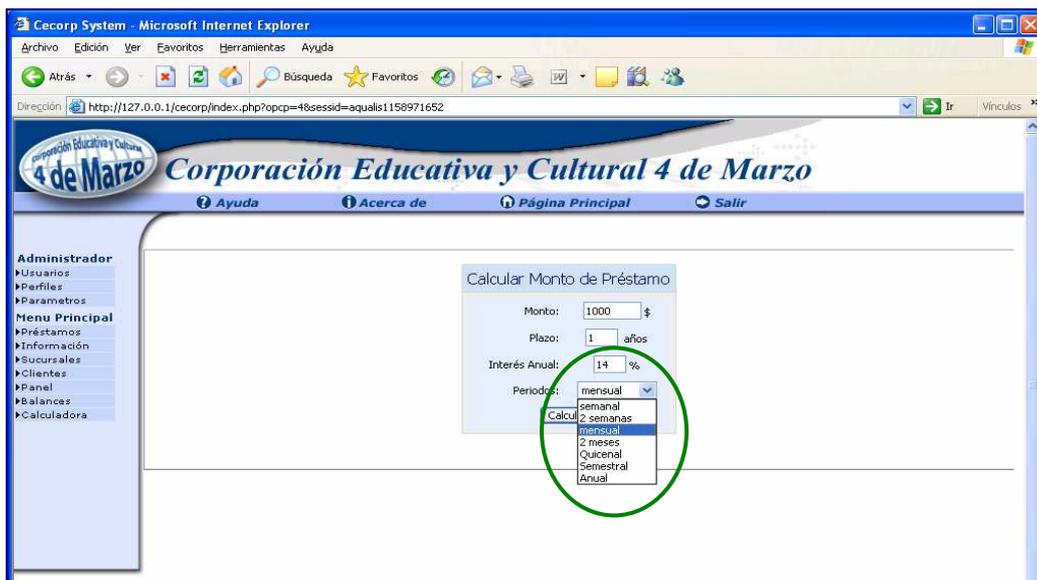


Figura 13.1 Tipo de interés

A continuación nos presenta la siguiente interfaz de Tabla de Amortización.

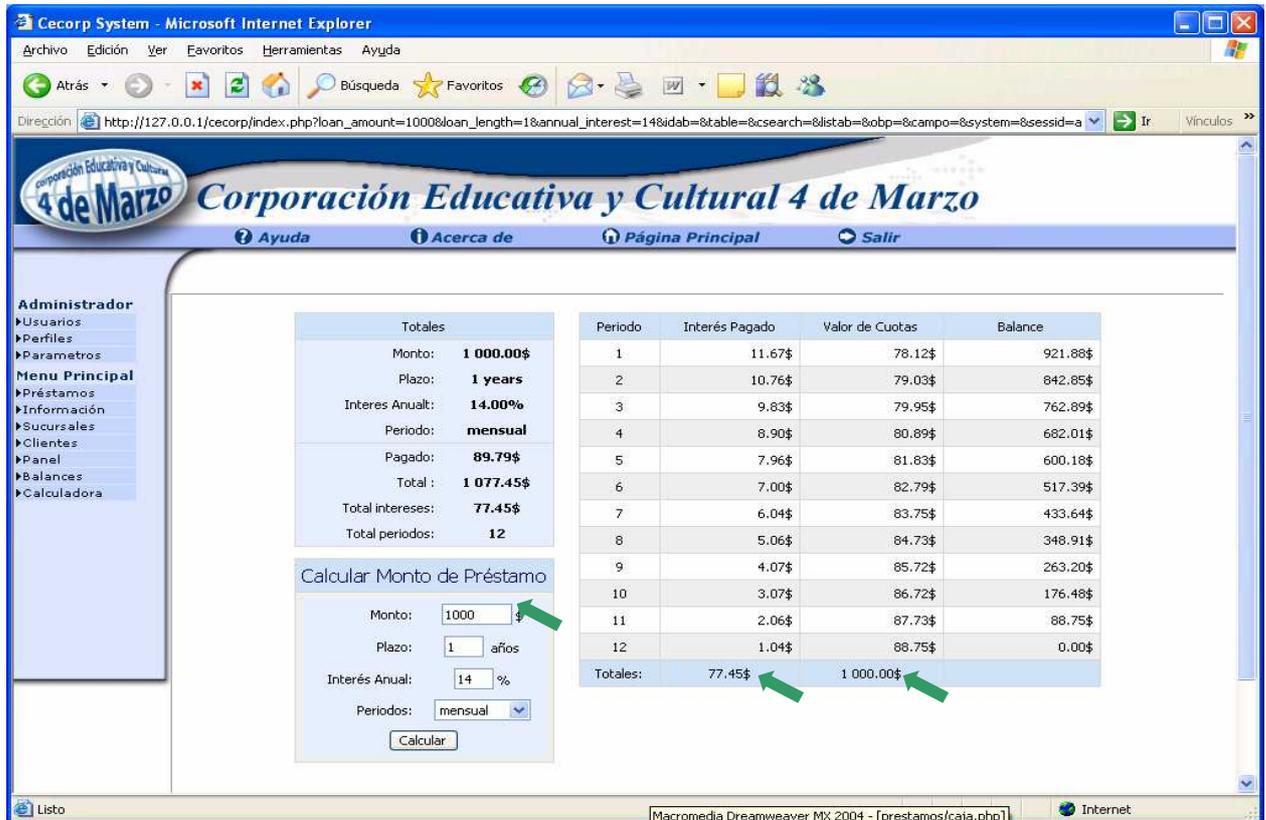


Figura 13.2 Tabla de amortizaciones

14. ADMINISTRAR PANEL

Esta interfaz facilita al administrador el ingreso al módulo Cuentas con el N° de cuenta, posteriormente damos clic Ejecutar.

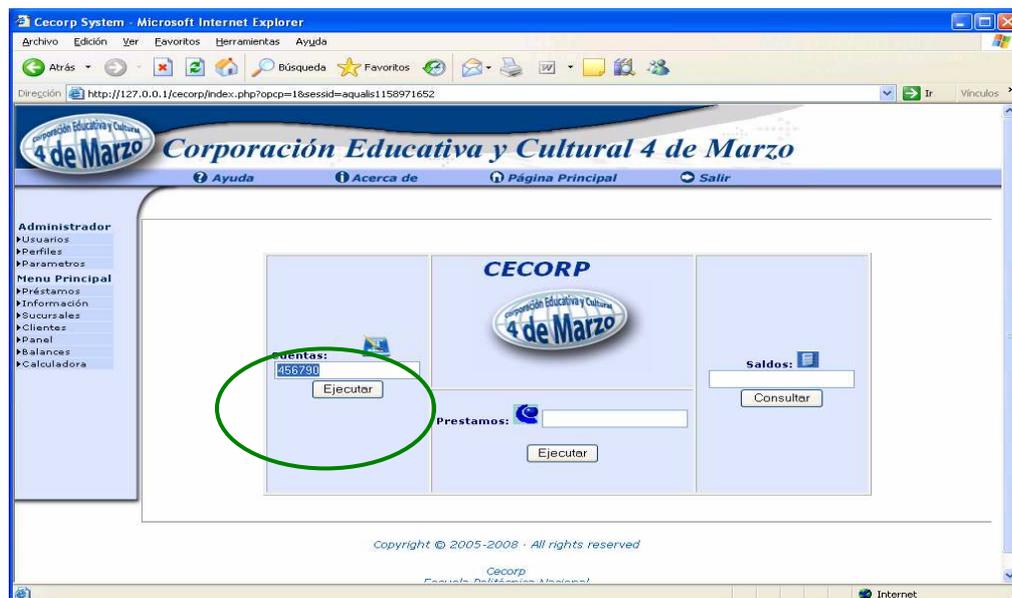


Figura 14.1 Ingreso a cuentas

Como ya ingresamos el número de cuenta directamente ingresamos al módulo de detalle movimiento e ingresamos los campos.

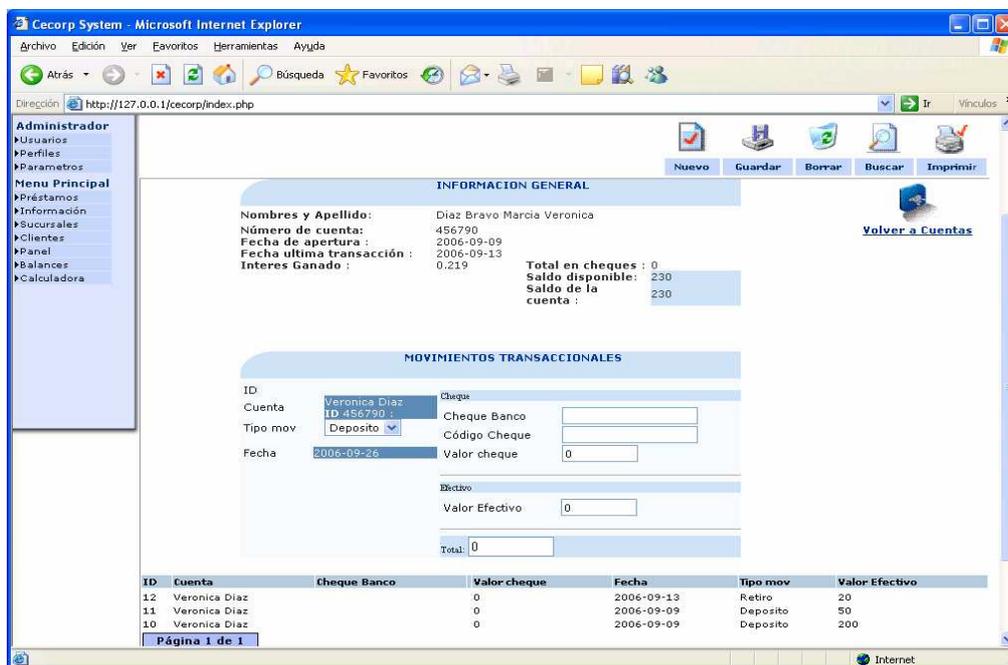


Figura 14.2 Módulo detalle de movimiento

De igual manera si deseamos ingresar al módulo préstamos lo podemos hacer de forma directa, ingresando el código de préstamos.



Figura 14.3 Ingreso al módulo de préstamos

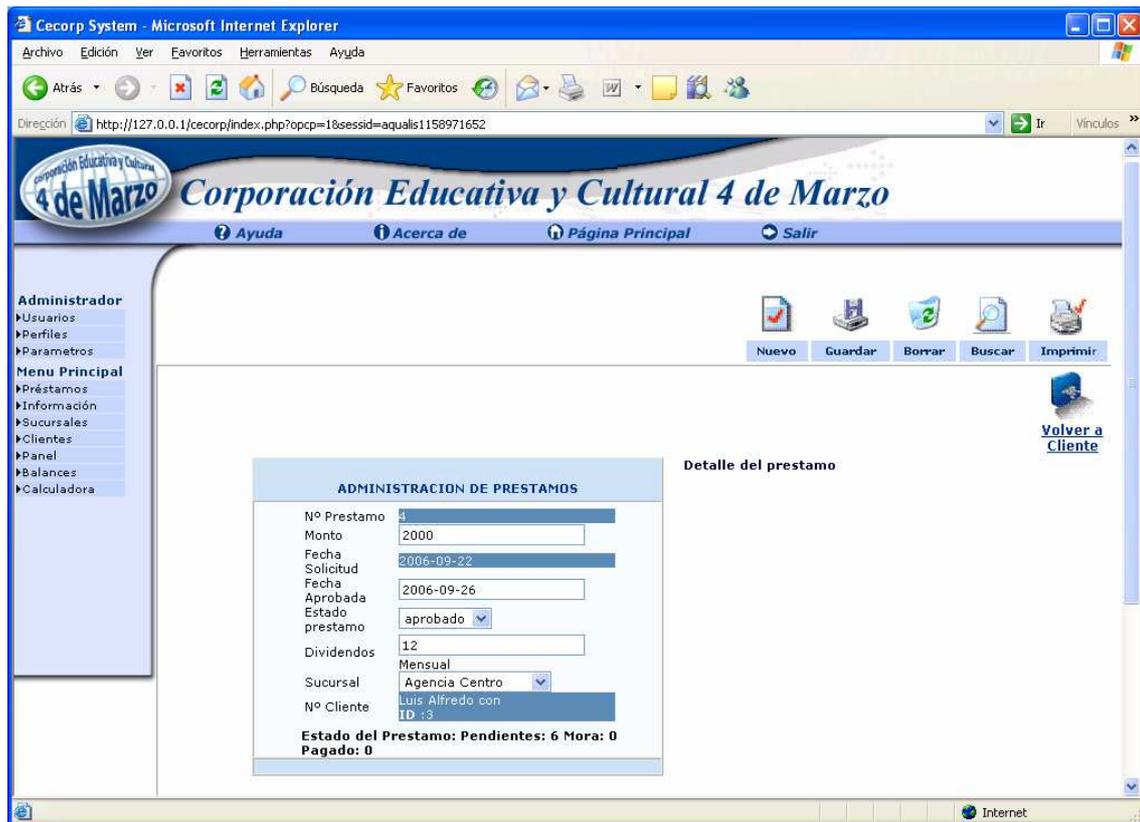


Figura 14.4 Módulo de préstamos



15. MODULO SALDOS

Este módulo nos permite consultar los saldos del cliente u otra información adicional.

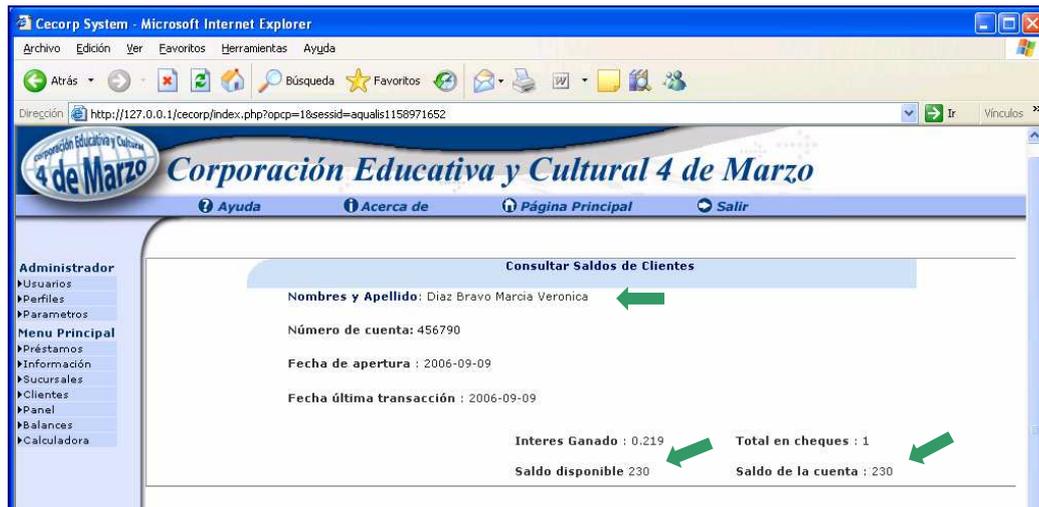


Figura 15.1 Módulo saldos

16. MODULO BALANCE

Para acceder a este módulo debo ingresar la fecha inicial, la fecha final en la que se desplegará la fecha del detalle el valor de los depósitos, retiros y sobre todo el monto total este reporte nos servirá para hacer cierre de caja.

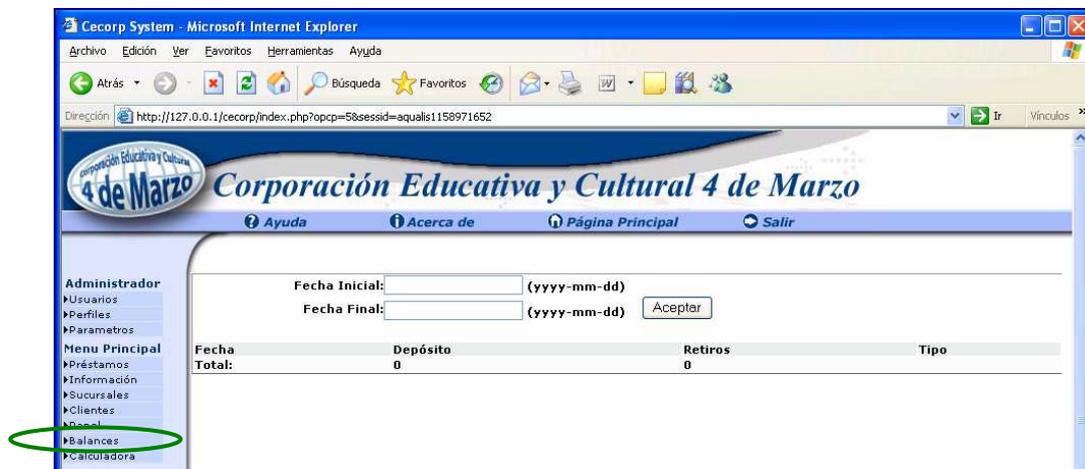


Figura 16.1 Módulo balance

Al momento de ingresar la fecha debo tomar en cuenta el formato (yyy-mm-dd) año, mes, día.

Administrador

- Usuarios
- Perfiles
- Parámetros

Menu Principal

- Préstamos
- Información
- Sucursales
- Clientes
- Panel
- Balances
- Calculadora

Fecha Inicial: 2006-09-09 (yyyy-mm-dd)

Fecha Final: 2006-09-21 (yyyy-mm-dd)

Fecha	Depósito	Retiros	Tipo
Total:	0	0	

Figura 16.2 Ingreso de fechas

Dar clic en el botón aceptar para generar el reporte.

Fecha Inicial: 2006-09-09 (yyyy-mm-dd)

Fecha Final: 2006-09-21 (yyyy-mm-dd)

Fecha	Depósito	Retiros	Tipo
2006-09-21	70		Efectivo
2006-09-21	50		Cheque
2006-09-20	35		Cheque
2006-09-20	230		Efectivo
2006-09-20	200		Efectivo
2006-09-20		50	Efectivo
2006-09-20	200		Efectivo
2006-09-20	100		Efectivo
2006-09-20	80		Cheque
2006-09-13		20	Efectivo
2006-09-12	40		Efectivo
2006-09-12		80	Efectivo
2006-09-12	400		Efectivo
2006-09-10		40	Efectivo
2006-09-10	100		Efectivo
2006-09-10	200		Cheque
2006-09-09	200		Efectivo
2006-09-09	300		Cheque
2006-09-09	10		Efectivo
2006-09-09	50		Cheque
2006-09-09	200		Efectivo
2006-09-09	50		Cheque
2006-09-09	50		Efectivo
Total:	2515	190	

Figura 16.3 Visualización de reportes

17. SALIR

Esta opción me permite salir de la aplicación para ingresar como otro usuarios.



Figura 17.1 Salir de la aplicación

Esta interfaz se presenta cuando se sale de la aplicación y dura unos minutos.

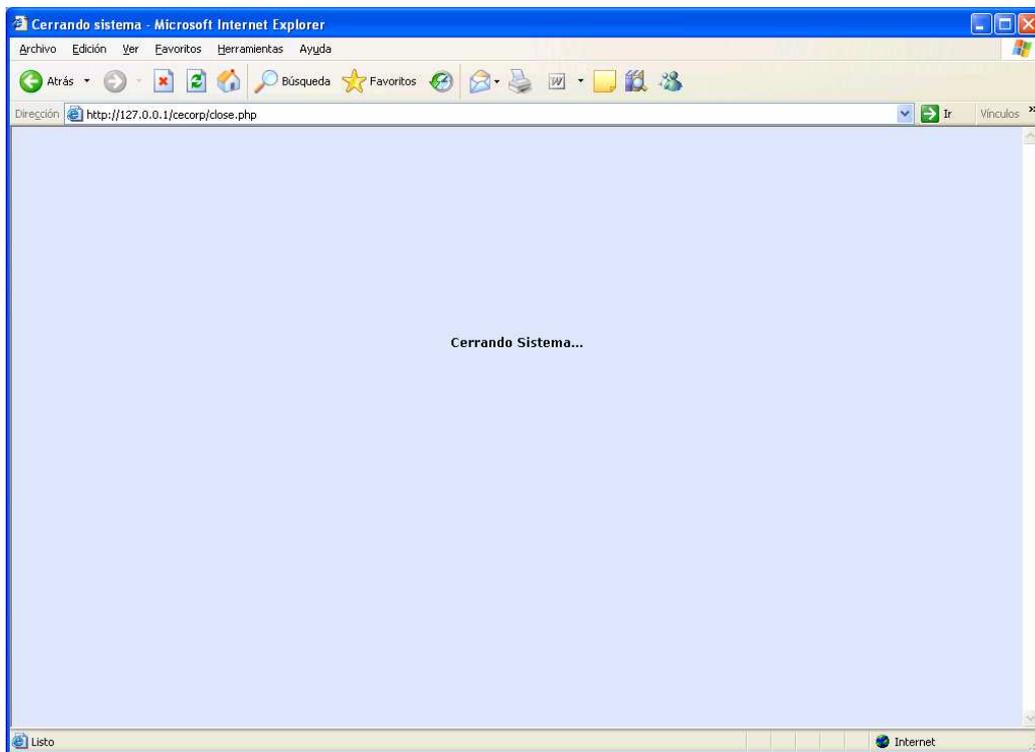


Figura 17.2 Cerrando el sistema

GUIA DE INSTALACIÓN

CONTENIDO

- 1. INSTALACIÓN DE XAMPP**
- 2. INSTALACIÓN DE MACROMEDIA DREAMWEAVER**
- 3. INSTALACIÓN DE MACROMEDIA FLASH**
- 4. INSTALACIÓN DE MACROMEDIA FIREWORKS**

1. INSTALACIÓN DE XAMPP

Como primer paso ejecutamos el setup de xampp, y escogemos el idioma en el cual vamos a instalar.



Figura 1.1 Lenguaje de instalación

Tenemos la pantalla de bienvenida y presionamos en siguiente para seguir con la instalación.

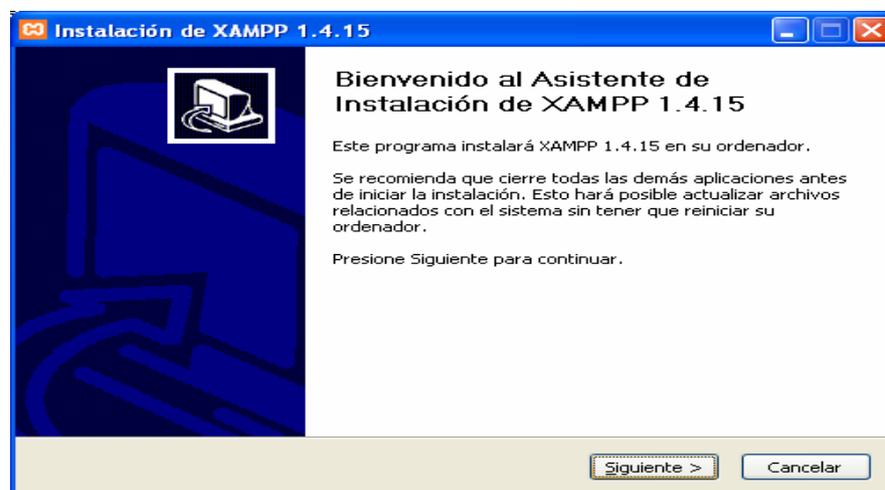


Figura 1.2 Pantalla de bienvenida

Especificamos la dirección en el cual se va a copiar los archivos de xampp.

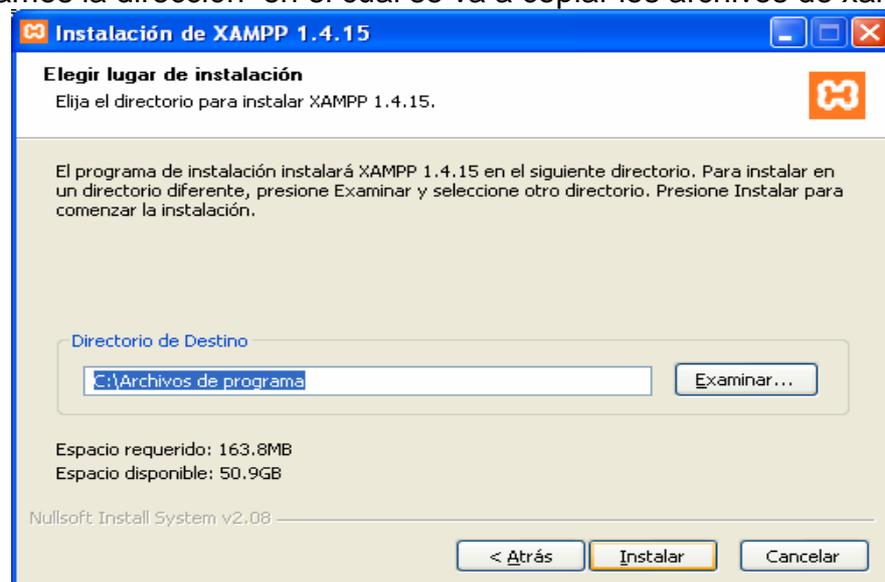


Figura 1.3 Lugar de instalación

Se están copiando todos los archivos de xampp, en la dirección que escogimos anteriormente.

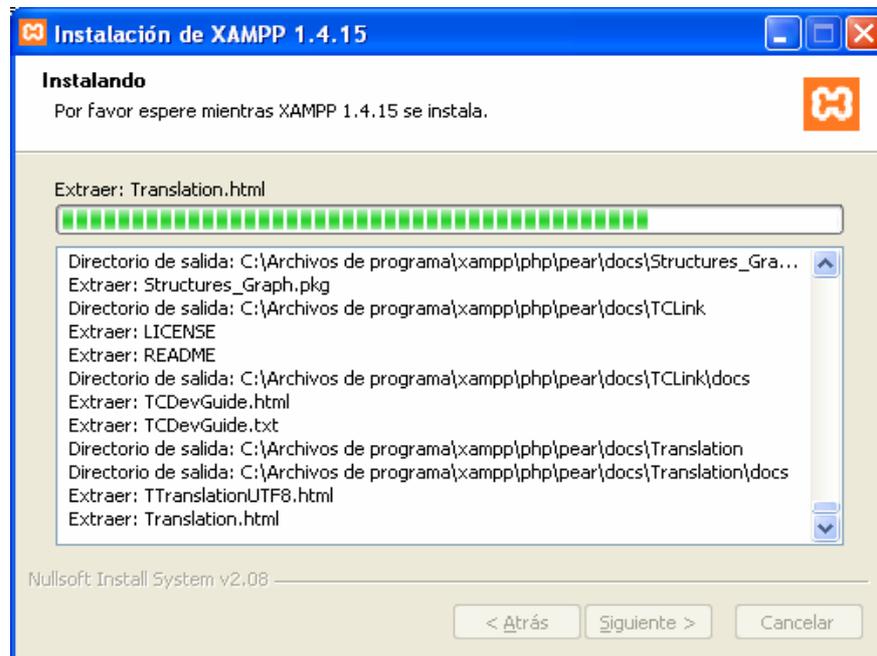


Figura 1.4 Instalación

En la siguiente pantalla se está configurando el servidor.

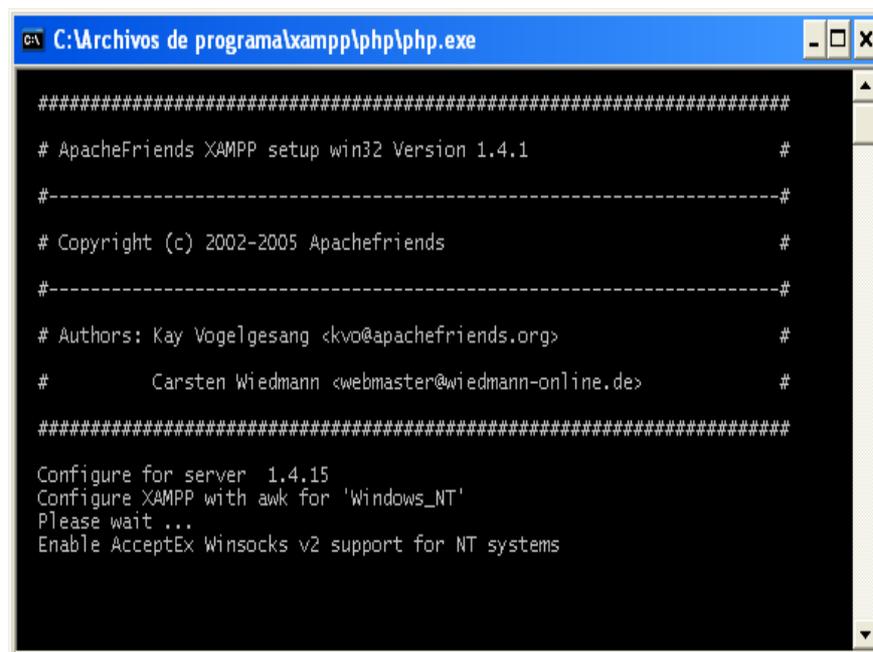


Figura 1.5 Configuración

Se completó la instalación y hacemos clic en terminar.



Figura 1.6 Terminar la instalación

Continuamos con la instalación del servidor de MySQL que viene incorporado en Xampp.

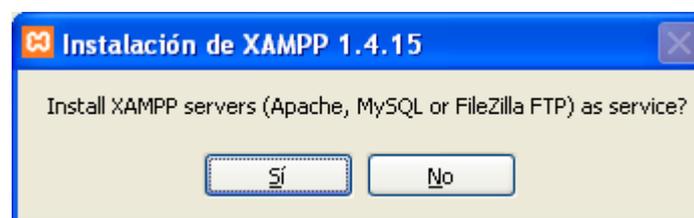


Figura 1.7 Instalación de los servicios

Chequeando los puertos necesarios para la instalación.

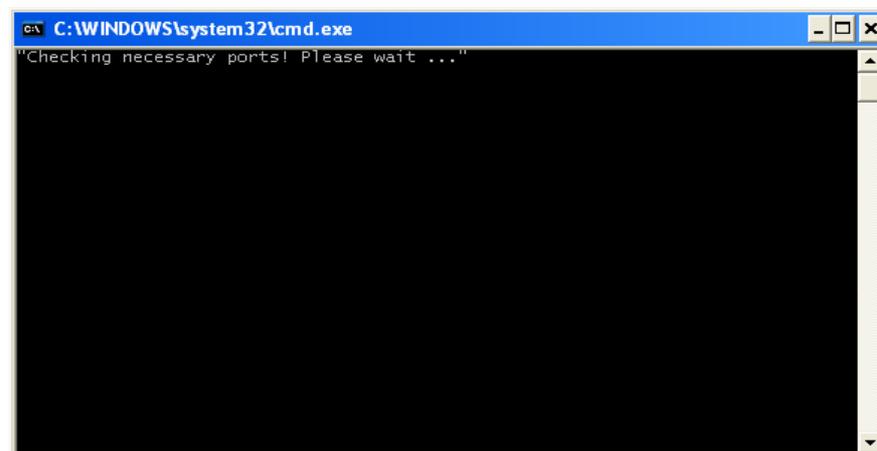


Figura 1.8 chequeo de puertos

Instalamos es servicio del apache.



Figura 1.9 Instalación del apache

Está corriendo el servicio del apache.

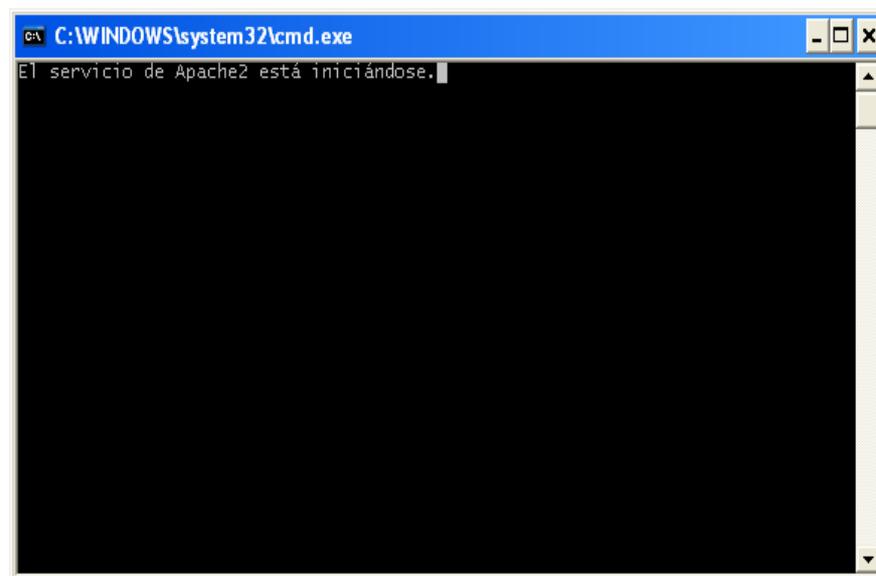


Figura 1.10 Servicio del apache

La instalación ha terminado satisfactoriamente en el puerto 80.



Figura 1.11 Instalación satisfactoria

Instalación del servicio de MySQL .



Figura 10.12 Instalación de MySql

Ha terminado satisfactoriamente.



Figura 10.13 Instalación concluida

También tenemos que instalar el Ftp. Como se muestra en la siguiente pantalla.



Figura 10.14 Instalación FTP

Hemos terminado la instalación con todos los servicios de Xampp, necesarios para el buen funcionamiento del servidor.

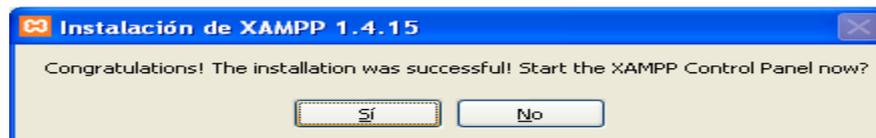


Figura 10.15 Configuración de Xampp

La siguiente pantalla nos muestra como se están ejecutando los servicios del Apache de MySql.

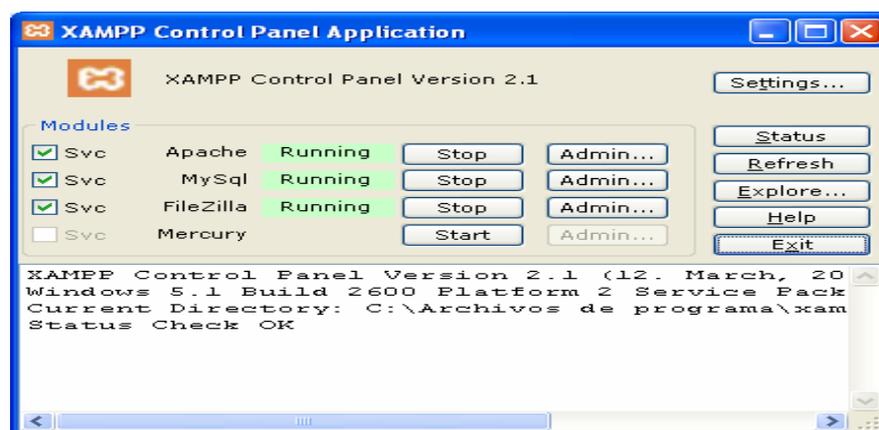


Figura 10.16 Ejecución de servicios

Lo normal es activar los dos primeros, lo que permitirá que desde cualquier navegador tecleemos la eterna IP local (<http://127.0.0.1>, o bien <http://localhost>) y accedamos a la verdadera interfaz de la aplicación, desde la cual tendremos acceso a nuestras bases de datos MySQL.

2. INSTALACIÓN DE MACROMEDIA DREAMWEAVER

Debemos ejecutarlo para poder instalar en nuestro computador, al hacerle doble clic, veremos la siguiente ventana.

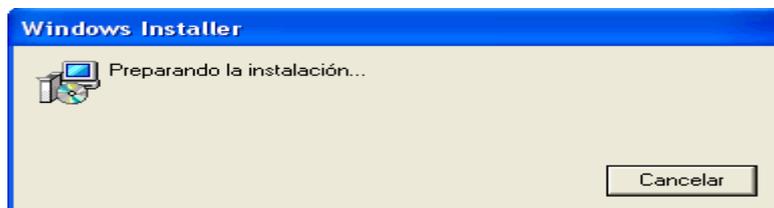
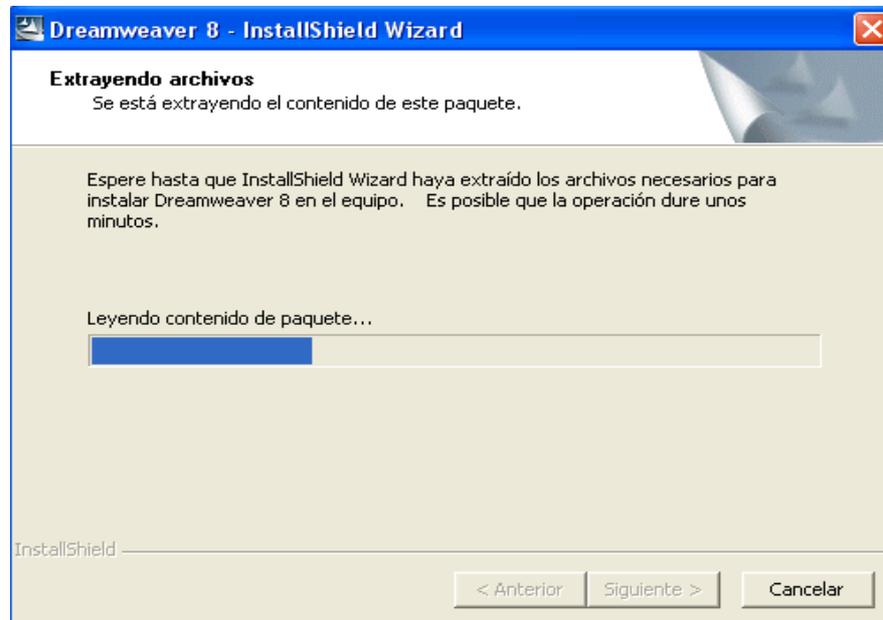


Figura 2.1 Extrayendo archivos para la instalación

Tenemos la pantalla de bienvenida para la instalación.

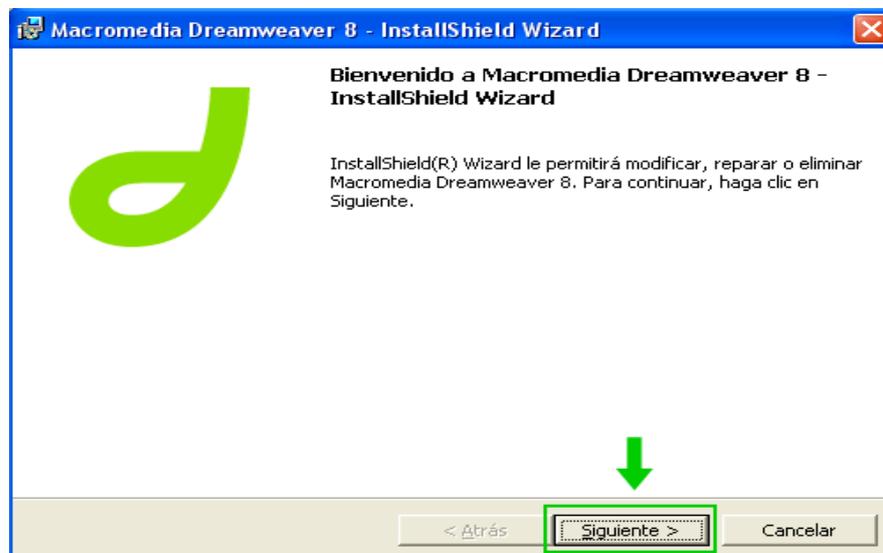


Figura 2.2 Pantalla de bienvenida

Aceptamos los términos establecidos en la licencia ara continuar.

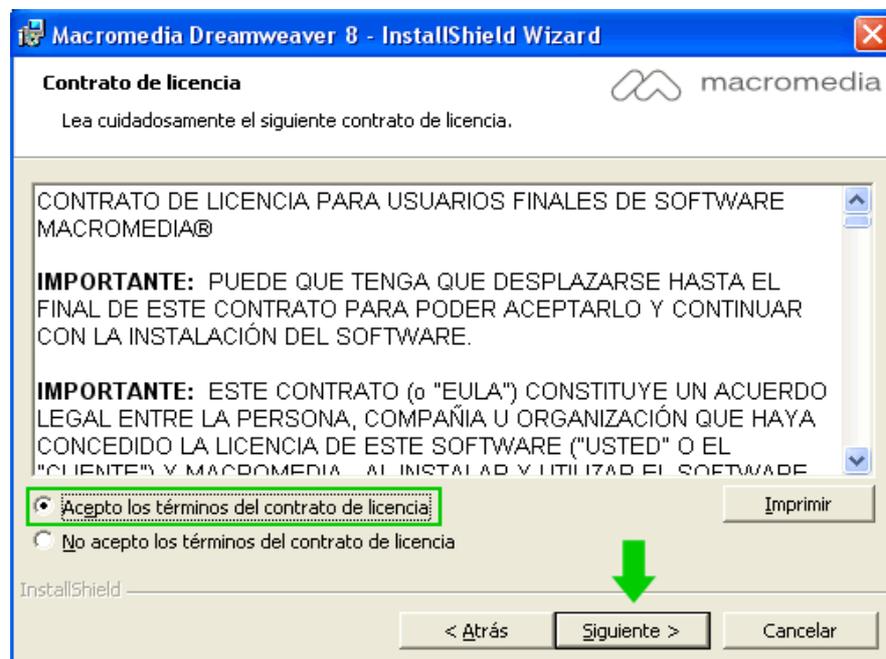


Figura 2.3 Términos de la licencia

Especificamos la ubicación para copiar los archivos de instalación.

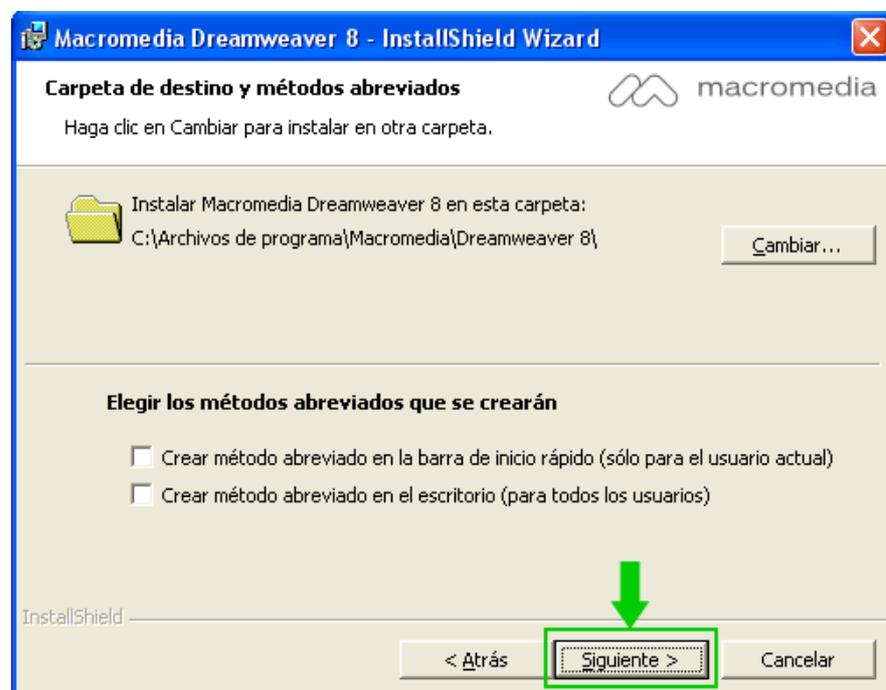


Figura 2.4 Destino de la instalación

Activamos todos los componentes a instalar.

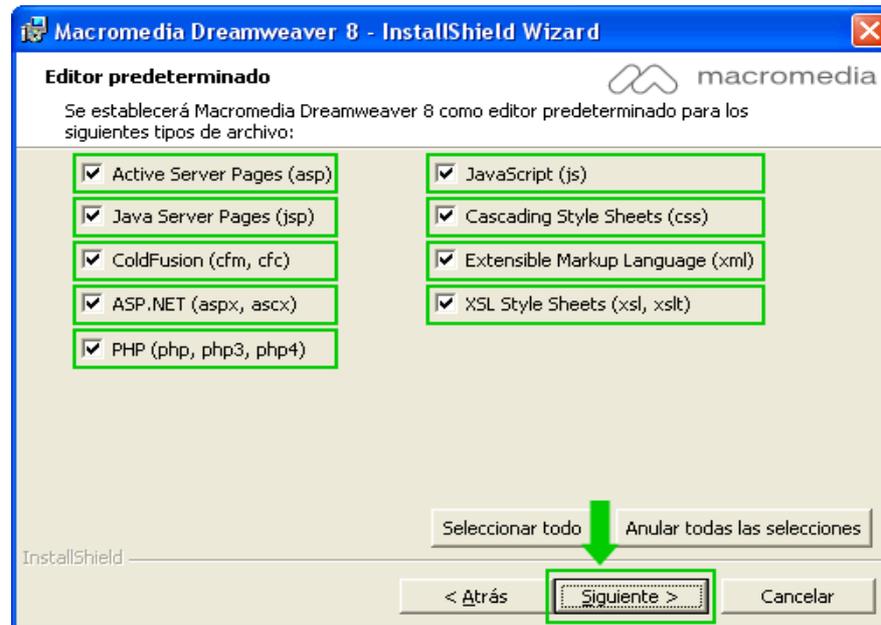


Figura 2.5 Editor predeterminado

La siguiente pantalla muestra la preparación para instalar.

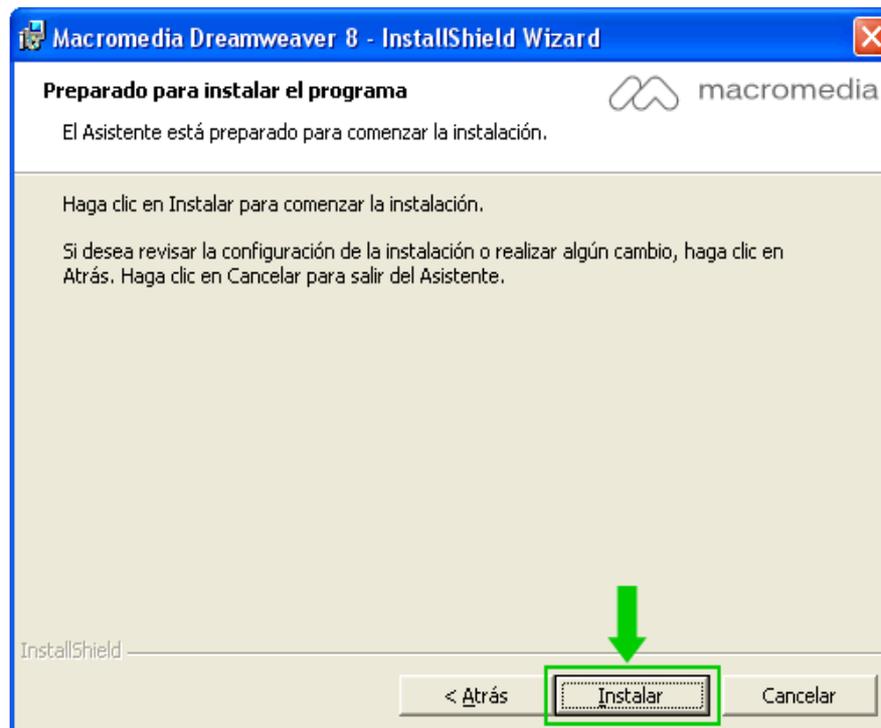


Figura 2.6 Preparación de la instalación

Proceso de instalación.

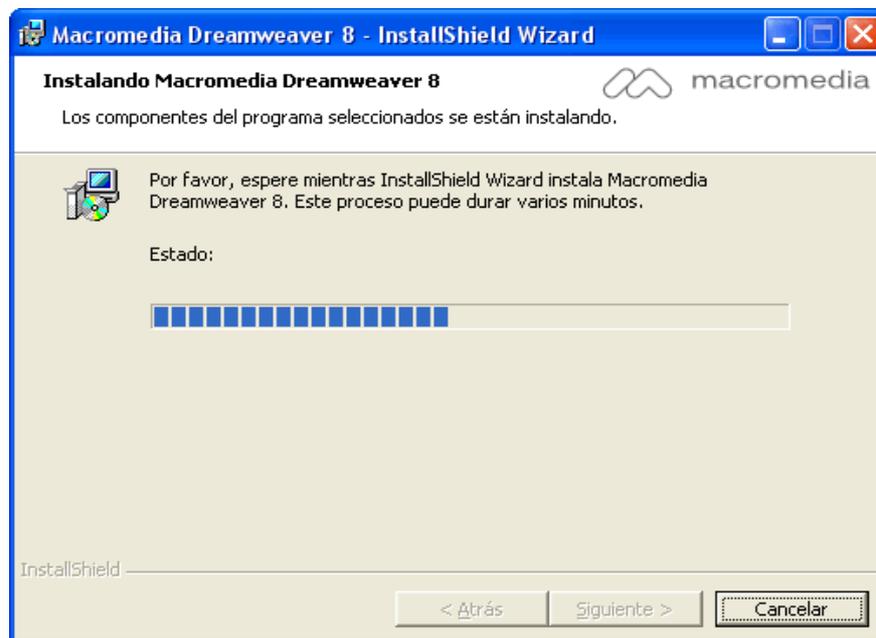


Figura 2.7 Instalando

Finalizamos la instalación que se lo ha realizado con éxito.

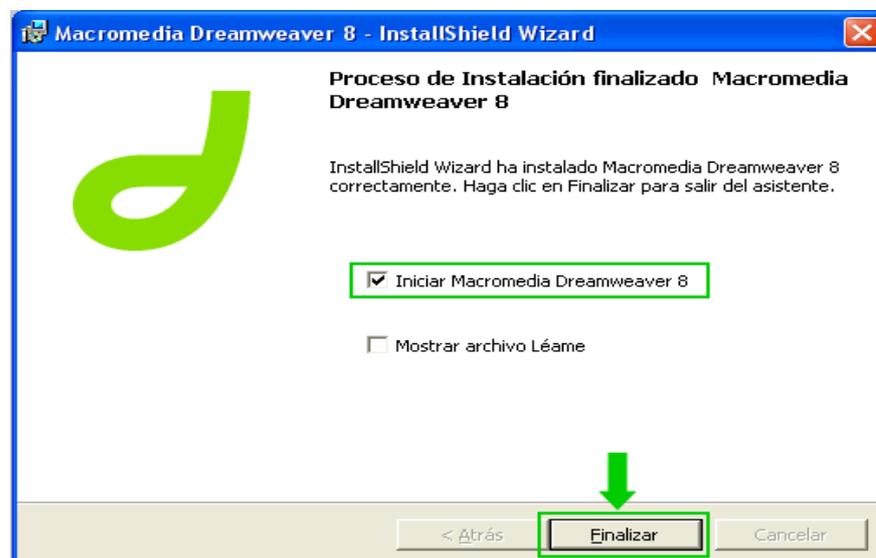


Figura 2.8 Instalación finalizada

Cuando le demos finalizar, nos pregunta si tenemos un serial para nuestro equipo, debemos seleccionar **EVALUAR**. Luego nos muestra una pequeña ventana que pregunta acerca del modo de programación, debemos seleccionar **MODO DISEÑADOR** y listo. Ya tenemos instalado nuestro **MACROMEDIA DREAMWEAVER**.

3. INSTALACIÓN DE MACROMEDIA FLASH

Ejecutar el archivo Install Flash MX 2004.exe para iniciar.



Figura 3.1 Extrayendo archivos

En la pantalla de bienvenida hacemos clic en siguiente.



Figura 3.2 Pantalla de bienvenida

Aceptamos los términos establecidos en la licencia para continuar con la instalación.

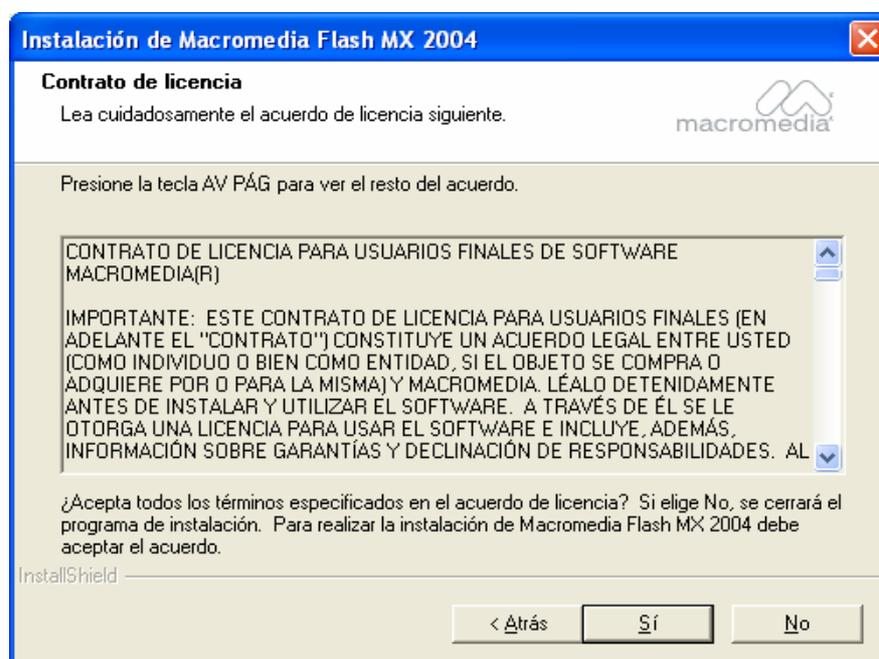


Figura 3.3 Contrato de licencia

En la siguiente pantalla nos hace caer en cuneta si queremos continuar o abandonar con la instalación.



Figura 3.4 Instalar macromedia

Especificamos la ubicación para copiar los archivos de instalación.



Figura 3.5 Copia de archivos

Proceso de instalación.

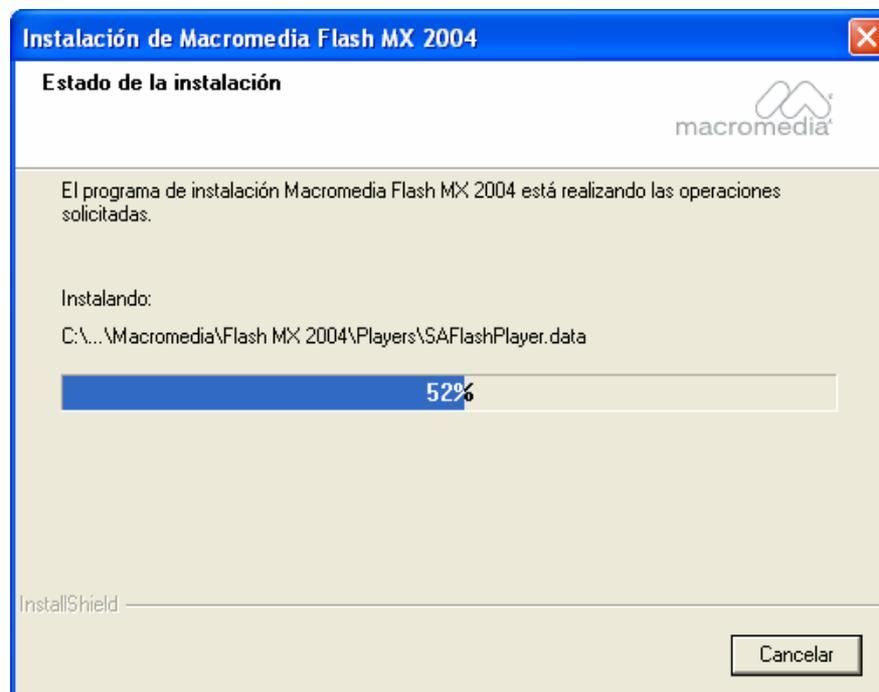


Figura 3.6 Estado de instalación

Ha concluido con la instalación y nos pide que reiniciemos el computador, para la actualización de los archivos instalados.



Figura 3.7 Instalación completa

4. INSTALACIÓN DE MACROMEDIA FIREWORKS

Extrayendo los archivos que comprimidos.

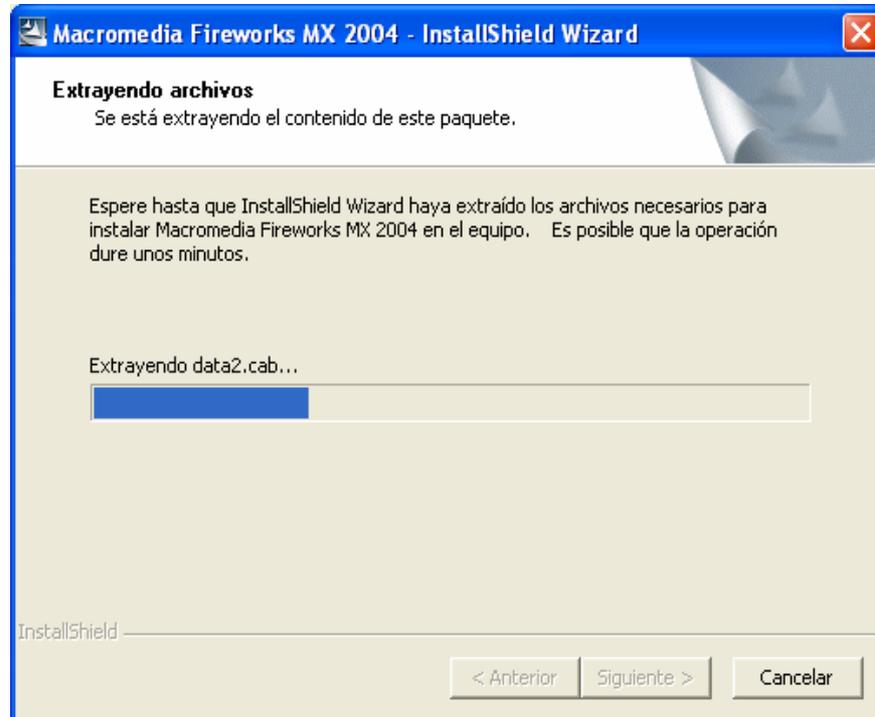


Figura 4.1 Extrayendo archivos para la instalación

Tenemos la pantalla de bienvenida a la instalación.



Figura 4.2 Pantalla de Bienvenida

Aceptamos los términos establecidos en la licencia para continuar con la instalación.

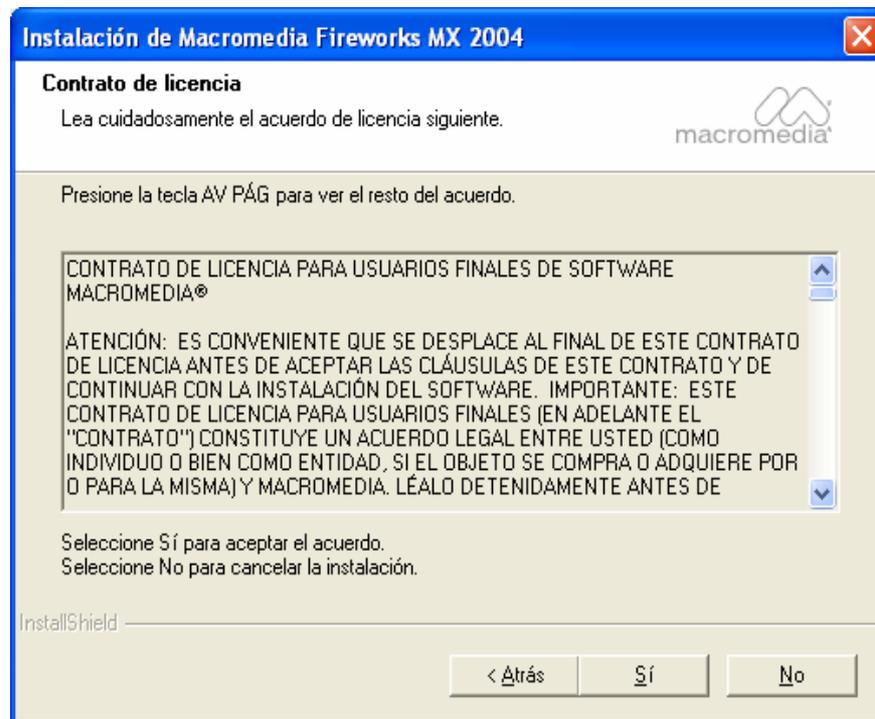


Figura 4.3 Contrato de licencia

Especificamos la dirección en donde se van a copiar los archivos a instalarse.



Figura 4.4 Ubicación de la instalación

Revisamos la configuración de los archivos a copiar si estamos de acuerdo hacemos clic en siguiente.

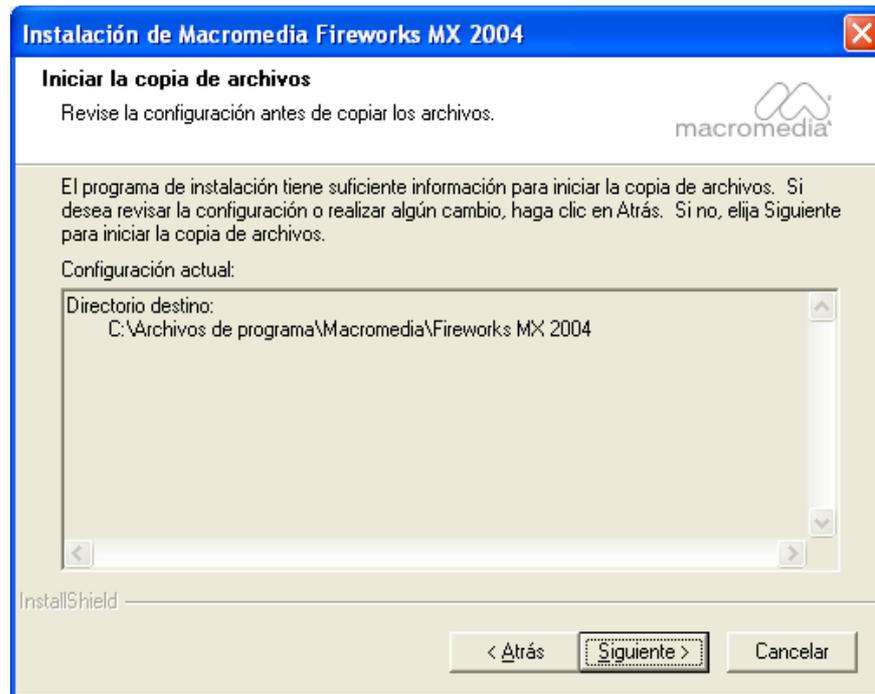


Figura 4.5 Copia de archivos

La siguiente ventana nos muestra el avance de la instalación.

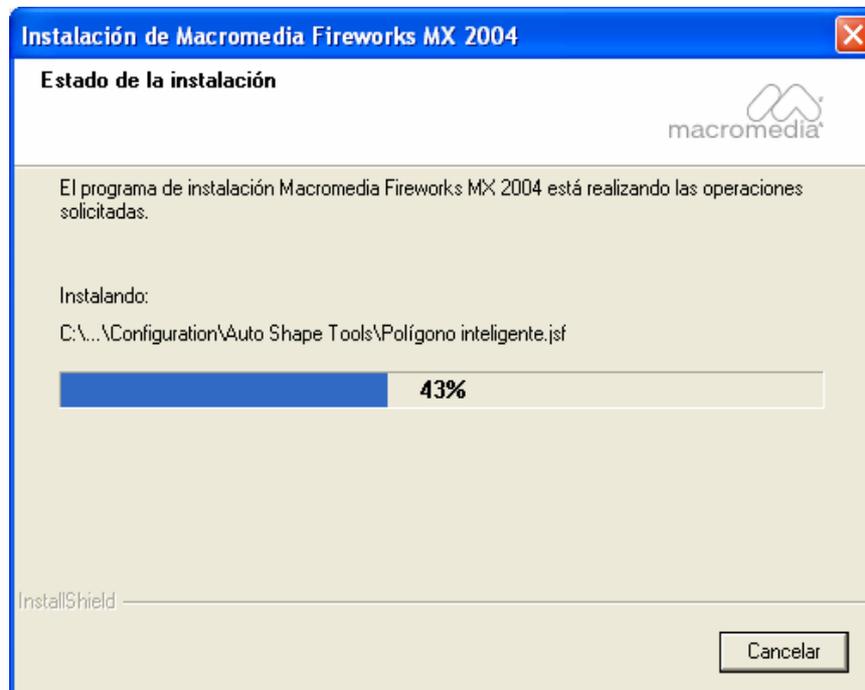


Figura 4.6 Estado de la instalación

La instalación ha terminado con éxito y por último finalizamos.

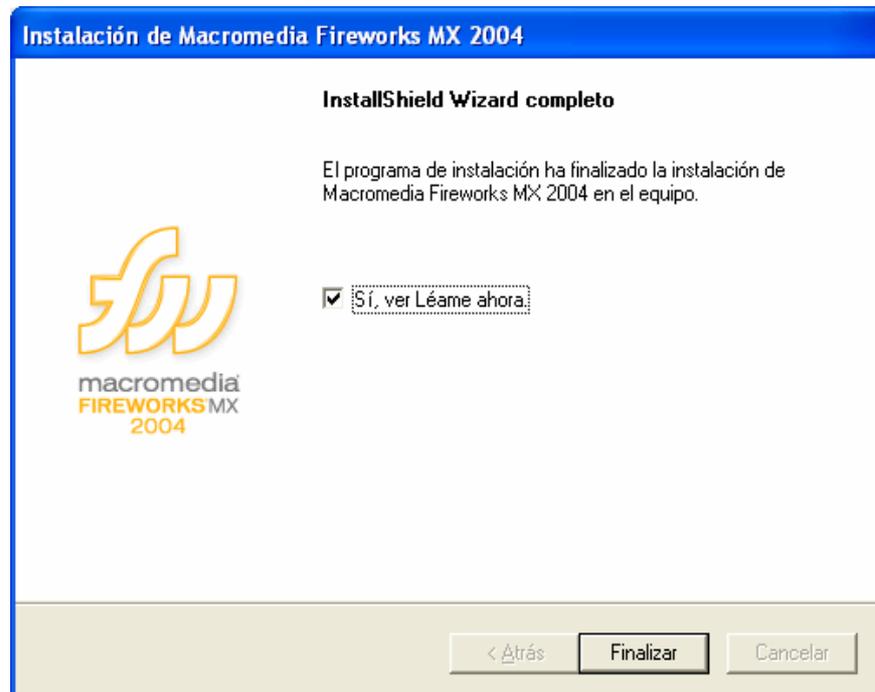


Figura 4.7 Instalación finalizada