

INSTITUTO TECNOLÓGICO SUPERIOR

JOSÉ OCHOA LEÓN.

ANÁLISIS DE SISTEMAS



4^{to} “A”

MANUAL TÉCNICO.

**Software para control de pagos (multas y
aportaciones) de socios de la Cooperativa
de Taxis 15 de Octubre**

AUTORES

DUSTIN JAVIER BLACIO ERAS

MELLISA NICOLE CONTRERAS DIOSES

RICARDO ANDRÉS DÍAZ PÉREZ

ANGGIE JAMILETT FERNANDEZ CRUZ

DIEGO ANDRÉS JIMENEZ PONCE

LISSETH CAROLINA OVIEDO ARMIJOS

ZOILA ANABEL PÉREZ BERMEO

Pasaje /10/2018

INDICE

1.	INTRODUCCIÓN.....	3
2.	OBJETIVO.....	3
3.	DESARROLLO.....	4
3.1	MODELADO DEL NEGOCIO	4
3.2	REQUISITOS.....	5
3.3	ANALISIS Y DISEÑO.....	11
3.3	IMPLEMENTACION.....	34
3.5	PRUEBAS.....	44
4	CONCLUSIONES.....	44
5	RECOMENDACIONES.....	44
6	ANEXOS.....	44

1. INTRODUCCIÓN.

Este documento ayuda al manejo del software y pueda tener conocimiento de la interacción del sistema se desarrollara en la Cooperativa de Taxis 15 de Octubre perteneciente al Cantón Santa Rosa, el manual es una herramienta que tiene como funcionamiento con el modelado que son las fases del sistema mediante los procesos realizados.

Inicio.

En esta fase empezamos con el análisis luego de haber reconocido el problema en la cooperativa, realizamos los diagramas de caso de uso, luego tenemos fases de identificación de los problemas, se desarrolló los escenarios de caso de uso para una mejor perspectiva, derivar diagramas de actividad de los casos de uso, desarrollar los diagramas de secuencia crear diagramas de clases y diagramas de estado.

Elaboración

En el sistema elaboramos los diseños que son utilizados en la aplicación del escritorio para establecer las funciones que tendrá cada interfaz, por el cual tenemos los requisitos.

Construcción.

Es la parte de programación de los botones de cada interfaz que permite dar un funcionamiento en el sistema y los procesos a realizarse.

Transición

Es la parte de las pruebas que se realiza del programa de escritorio por el cual se visualiza si cumple la aplicación con lo requerido de cada función realizada.

En el programa constara de 5 módulos los cuales son:

Gestión de Socios

Gestión de Multas

Gestión Aportación

Cobro de Multas

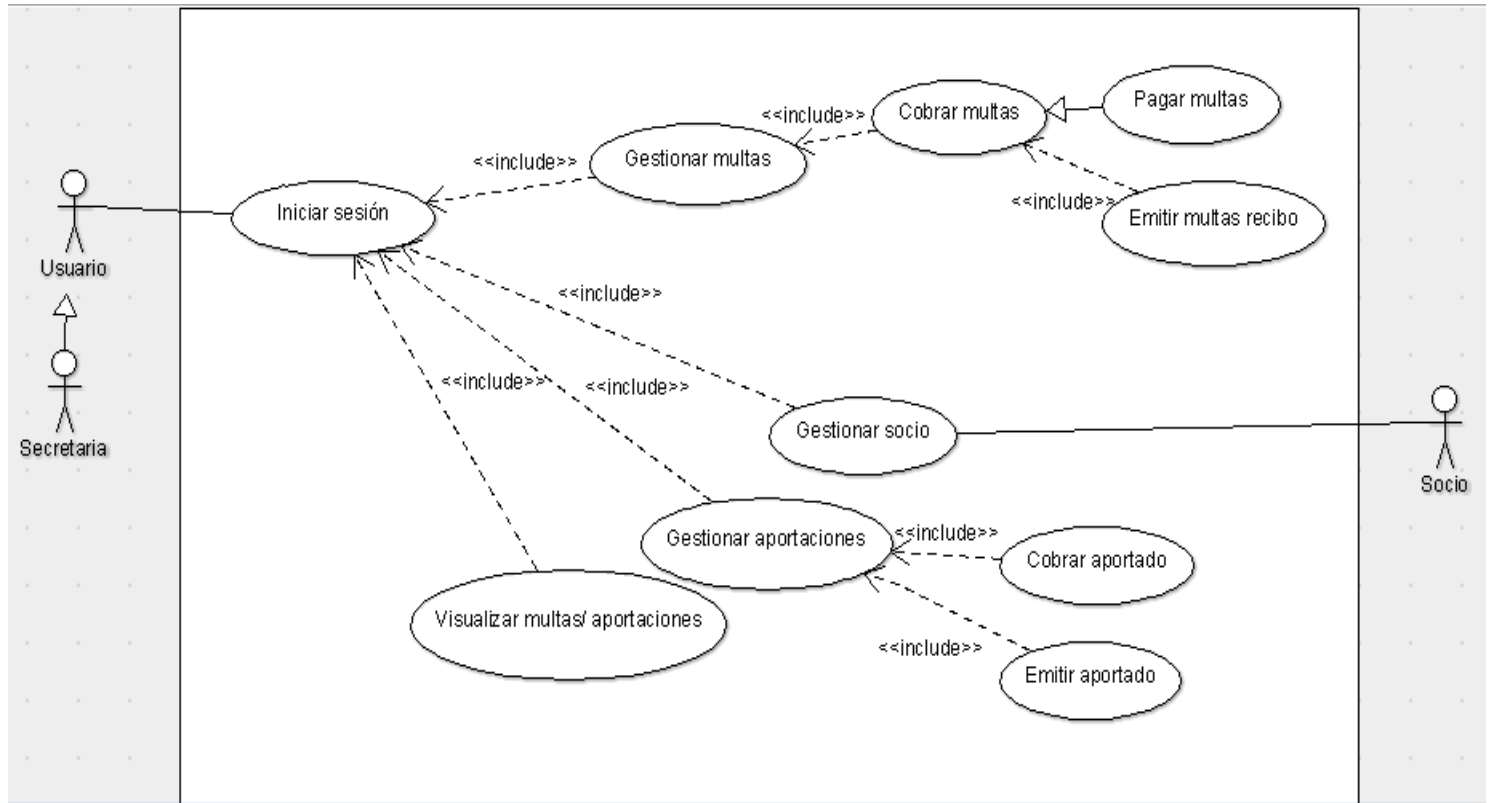
Cobro de Aportaciones

2. OBJETIVO.

Demostrar el proceso que se ha realizado en la construcción del software en la Cooperativa de Taxis 15 de Octubre, con la aplicación de la Metodología Orientada Objetos con sus respectivas fases, para servir como herramienta de ayuda para el personal técnico de la empresa.

3. DESARROLLO

3.1 MODELADO DEL NEGOCIO




3.2 REQUISITOS.

Iniciar sesión.

Login

INICIO DE SESION





Secretaria

Ingreso

USUARIO

CLAVE


CONTINUAR


REGRESAR

Identificación del requerimiento:	RF 01
Nombre del Requerimiento:	Iniciar sesión
Características:	Seguridad de datos
Descripción del requerimiento:	<ul style="list-style-type: none"> • Permite inicio de sesión secretaria • Permitir ingresar clave usuario • Mostrar el botón de continuar • Mostrar el botón regresar
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta

Gestionar Socio

Codigo	Cedula	Movil	Nombre	Apellido	Direccion
2	0703349852	2	FREDDY	MALDONADO SAL	SANTA ROSA
3	0700961600	3	GILBERTO	GUANGA LEON	SANTA ROSA
4	0703689190	4	CARLOS MAGNO	ERIQUE PINEDA	SANTA ROSA
5	0700761539	5	MANUEL DAVID	CRESPO PALADIN	SANTA ROSA

Identificación del requerimiento:	RF 02
Nombre del Requerimiento:	Registro de socio
Características:	Permitir registrar un socio
Descripción del requerimiento:	<ul style="list-style-type: none"> • Mostrar las cajas bloqueadas dar clic en el botón agregar. • Permitir el registro de socio por el cual se debe de registrar sus datos personales. • Mostrar el botón agregar • Mostrar el botón guardar • Mostrar el botón modificar por si sea necesario • Mostrar el botón cancelar • Mostrar el botón cerrar • Permitir una Grid donde se muestra los datos del socio.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta

Gestionar Multas.

Codigo	Cedula	Nombre	Apellido	Multa
22	0703349852	FREDDY	MALDONADO SALAZAR	TARDE DEPORTIVA
17	0703349852	FREDDY	MALDONADO SALAZAR	SALTARSE EL TURNI
1	070361600	GILBERTO	GUANGA LEON	SALTARSE EL TURNI
18	0703689190	CARLOS MAGNO	ERIQUE PINEDA	TARDE DEPORTIVA

Identificación del requerimiento:	RF 03
Nombre del Requerimiento:	Registro de multas
Características:	Ingreso de datos
Descripción del requerimiento:	<ul style="list-style-type: none"> • Se presentan cajas bloqueadas hacer clic en agregar para acceder • Permitir la búsqueda del socio donde se despliega otra ventana. • Permitir agregar la multa mediante un combo box y un botón agregar multas donde tendrá que llenar los campos vacíos. • Permitir mostrar la fecha • Permitir dar clic en debe y cancelado y todos según el estado de cada socio. • Permitir mostrar la información en un data Grid • Permitir el botón para guardar los registros. • Permitir el botón modificar • Mostrar el botón cancelar • Mostrar el botón reporte • Permitir el botón cerrar.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento.	Alto

Cobro de multas.



Identificación del requerimiento:	RF 04
Nombre del Requerimiento:	Cobranzas de multas.
Características:	Permitir el cobro de la multa.
Descripción del requerimiento:	<ul style="list-style-type: none"> • Mostrar las cajas bloqueadas dar clic en el botón cobrar. • Permitir buscar el socio hacemos clic y se despliega una ventana donde tenemos que llenar todos los campos, donde pasaran a las cajas de texto. • Permitir dar clic en el botón cargar multas. • Permitir visualizar los datos en el check dar clic señalando los datos de la multas. • Permitir la fecha. • Permitir agregar la observación • Permitir agregar el valor y saldo pendiente. • Permitir visualizar el Grid con todos los datos correspondientes. • Mostrar el botón agregar • Mostrar el botón guardar • Mostrar el botón modificar por si sea necesario • Mostrar el botón cancelar • Mostrar el botón cerrar. • Permitir el botón imprimir
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta

Gestionar Aportaciones.

Aportaciones

SOCIO

NOMBRE

APELLIDO

TIPO APORTACION

FECHA

☐ DEBE ☐ CANCELADO ☐ TODOS

Codigo	Cedula	Nombre	Apellido	Aportacion
13	0700961600	GILBERTO	GUANGA LEON	MORTORIA
9	0703689190	CARLOS MAGNO	ERIQUE PINEDA	MORTORIA
5	0700761539	MANUEL DAVID	CRESPO PALADIN	LINEA
2	0700761539	MANUEL DAVID	CRESPO PALADIN	MORTORIA

AGREGAR GUARDAR MODIFICAR CANCELAR REPORTE CERRAR

Identificación del requerimiento:	RF 05
Nombre del Requerimiento:	Aportaciones
Características:	Ingreso de datos
Descripción del requerimiento:	<ul style="list-style-type: none"> • Mostrar caja bloqueada dar clic para activar la acción. • Se debe Permitir la búsqueda del socio, hacer clic • Permitir ingresar el tipo de aportación mediante un combo box • Permitir dar clic en agregar para agregar el tipo de aportación. • Permitir mostrar la fecha • Permitir dar clic debe, cancelado y todos. • Permitir mostrar la información en un check • Mostrar el botón agregar • Mostrar el botón guardar • Mostrar el botón modificar • Mostrar el botón cancelar • Mostrar el botón reporte. • Mostrar el botón cerrar
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Cobro de Aportaciones.

Cobranzas de Aportaciones

CEDULA

NOMBRE

APELLIDO

APORTACIONES

FECHA	VALOR	OBSERVACION
07/10/2018		

FECHA




OBSERVACION

VALOR

Identificación del requerimiento:	RF 06
Nombre del Requerimiento:	Cobranza de Aportaciones.
Características:	Ingreso de datos
Descripción del requerimiento:	<ul style="list-style-type: none"> • Mostrar caja bloqueada dar clic para activar la acción en cobrar. • Se debe Permitir la búsqueda del socio hacer clic • Permitir cargar aportaciones. • Permitir mostrar la fecha • Permitir mostrar la información en un check • Mostrar el botón cobro • Mostrar el botón guardar • Mostrar el botón cancelar • Mostrar el botón imprimir • Mostrar el botón cerrar
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF04 • RNF05
Prioridad del requerimiento:	Alta

3.3 ANALISIS Y DISEÑO.

Iniciar Sesión.

Nombre de caso de uso	Iniciar Sesión	ID Único:01
Área		
Actor(es)	Usuarios	
Interesados	Todos los usuarios	
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir que el usuario pueda iniciar sesión	
Evento desencadenador	El usuario acceda al sistema al sistema de cobro de multas y aportaciones de la cooperativa de Taxis 15 de Octubre, mediante su icono.	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)	Información para los pasos	
1. Acceder a sistema de Coop. 15 de Octubre mediante su icono.	Icono de Coop. 15 de Octubre.	
2. Escribir usuario y contraseña.	Usuario, contraseña	
3. Verificar usuario y contraseña	Usuario y contraseña , registro de usuario	
4. Acceder a la interfaz de usuario	Interfaz de sesión	
Precondiciones:	El usuario está registrado en sistema Coop 15 de Octubre.	
Postcondiciones:	El usuario ha iniciado sesión correctamente.	
Suposiciones:	El usuario cuenta con un usuario y contraseña válidos.	
Garantías de Éxito:	Se inicia sesión y muestra la interfaz correcta.	
Garantía Mínima:	El usuario accede al sistema Coop. 15 de Octubre.	
Requerimientos cumplidos:	Permitir que el usuario inicie sesión	
Cuestiones Pendientes:	¿Qué pasaría si la contraseña es incorrecta?	
Prioridad	Media	
Riesgo:	Media	




Gestión de socio.

Nombre de caso de uso	Gestionar Socio		ID Único:02
Área			
Actor(es)	Secretaria		
Interesados	Secretaria ,Gerente		
Nivel:	<input type="radio"/> Blanco <input type="radio"/> Cometa <input checked="" type="radio"/> Azul		
Descripción	Permitir la gestión de socio		
Evento desencadenador	Iniciar Sesión.		
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>	
Pasos realizados (ruta principal)	Información para los pasos		
1. Iniciar Sesión.	Usuario, contraseña validos		
2. Accede al menú principal	Opción Gestión de socio		
3. Buscar socio.	búsqueda del socio		
4. Registrar al socio.	Datos del socio		
5. Validar datos del socio	Datos validados		
6. Guardar datos del socio.	Datos Guardados del socio.		
Precondiciones:	Iniciar sesión.		
Postcondiciones:	Se ha realizado correctamente la gestión de socios.		
Suposiciones:	La secretaria pudo realizar el registro del socio correctamente.		
Garantías de Éxito:	Datos guardados exitosamente del socio		
Garantía Mínima:	Haber podido realizar inicio de sesión.		
Requerimientos cumplidos:	Permitir que se pueden la secretaria pueda gestionar socio.		
Cuestiones Pendientes:	¿Qué pasaría si no se guarda los datos del socio?		
Prioridad	Media		
Riesgo:	Media		






Gestión de multas.

Nombre de caso de uso	Gestionar Multas.		ID Único:03
Área			
Actor(es)	Secretaria		
Interesados	Gerente		
Nivel:	<input type="radio"/> Blanco <input type="radio"/> Cometa <input checked="" type="radio"/> Azul		
Descripción	Permitir que la Secretaria realice la gestión de Multas.		
Evento desencadenador	La Secretaria realiza inicio de sesión		
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>	
Pasos realizados (ruta principal)	Información para los pasos		
1. Iniciar sesión.	Usuario y contraseña validas		
2. Acceder al menú principal	Opción gestionar multas		
3. Buscar socio por cedula	Datos del socio		
4. Procesa información búsqueda	Datos validados del socio.		
5. Registrar al socio	Ingresar, modificar y guardar multa.		
6. Validar datos	Datos validados del socio		
7. Guardar multa	Guardar los datos de la multa		
Precondiciones:	El usuario registra al socio en el sistema		
Postcondiciones:	La Secretaria registrará las multas		
Suposiciones:	Se ingresa la multa		
Garantías de Éxito:	Se ha Iniciado sesión y se ha registrado la multa		
Garantía Mínima:	El usuario ingreso al sistema		
Requerimientos cumplidos:	El usuario debe iniciar sesión		
Cuestiones Pendientes:	¿Qué pasaría si la secretaria registró mal al socio en el sistema?		
Prioridad	Media		
Riesgo:	Media		

Cobro de Multas.

Nombre de caso de uso	Cobrar multas	ID Único:04
Área		
Actor(es)	Secretaria	
Interesados		
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir al sistema que realice el cobro de multas correctamente	
Evento desencadenador	Gestionar Multas	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)		Información para los pasos
1. Inicio de sesión		Iniciar sesión
2. Acceder a la interfaz del sistema		Multas y aportaciones
3. Buscar socio		Icono buscar
4. Verificar socio		Datos de socio
5. Realizar cobro de multa		Datos de multa
6. Realizar tipo de pago		Tipos de pago
7. Guardar datos de cobranzas		Datos del cobro de multa
Precondiciones:	Gestionar Multas	
Postcondiciones:	Sistema de cobro de multas ha realizado correctamente el cobro	
Suposiciones:	Datos de Multas	
Garantías de Éxito:	Permitir que el sistema realice exitosamente el cobro de multas	
Garantía Mínima:	Que se registre la multa	
Requerimientos cumplidos:	Permitir el cobro de multas	
Cuestiones Pendientes:	¿Qué pasaría si la no se registran los datos del socio?	
Prioridad	Media	
Riesgo:	Media	




Pagar Multas.

Nombre de caso de uso	Pagar multas	ID Único:05
Área	Cooperativa de taxis	
Actor(es)	Secretaria	
Interesados	Gerente, secretaria, socio.	
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir que el usuario pueda pagar	
Evento desencadenador	El cobro de multas.	
Tipo de desencadenador:	Externo 	Temporal 
Pasos realizados (ruta principal)		Información para los pasos
1. Iniciar sesión		Usuario y contraseña
2. Acceder al menú cobro de multas		interfaz
3. Accede a la opción pago de multas		Datos de multa
4. buscar valor de multa		Valor de multa
5. Verificar fecha de pago		Fecha de pago
6. Recibir el pago de multa		Pago de multa
7. Guardar pago		Guarda el pago
Precondiciones:	Cobro de multas	
Postcondiciones:	Se ha recibido el pago de la multa	
Suposiciones:	Se asume que el pago de la multa es por haber realizado una infracción.	
Garantías de Éxito:	Se ha recibido el pago de la multa correctamente.	
Garantía Mínima:	Al menos se ha cobrado la multa	
Requerimientos cumplidos:	Permitir que el usuario pueda pagar la multa.	
Cuestiones Pendientes:	¿Qué pasaría si no paga la multa?	
Prioridad	Media	
Riesgo:	Media	




Emitir Multas Recibo

Nombre de caso de uso	Emitir Recibo de Multas	ID Único:06
Área		
Actor(es)	Secretaria	
Interesados	Gerente	
Nivel:	<input type="radio"/> Blanco <input type="radio"/> Cometa <input checked="" type="radio"/> Azul	
Descripción	Permitir que la Secretaria emita recibo de multas.	
Evento desencadenador	La Secretaria realiza inicio de sesión	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)	Información para los pasos	
1. Iniciar sesión	Usuario y clave	
2. Validar usuario y clave	Validar usuario y clave	
3. Buscar socio por cedula	Cedula correcta	
4. Validar socio por cedula	cedula	
5. Cobro de multas	Multas	
6. Emitir recibo de la multa	Pago de la multa	
Precondiciones:	Inicia sesión	
Postcondiciones:	Emite recibo de multa	
Suposiciones:	La secretaria emite recibo correctamente	
Garantías de Éxito:	El recibo se emite exitosamente	
Garantía Mínima:	Ingrese sesión	
Requerimientos cumplidos:	Permitir que la Secretaria logre emitir en recibo.	
Cuestiones Pendientes:	¿Qué pasaría si no se pude emitir el recibo de multas?	
Prioridad	Media	
Riesgo:	Media	

Gestiones de aportaciones

Nombre de caso de uso	Gestionar Aportaciones	ID Único:07
Área		
Actor(es)	Secretaria	
Interesados	Gerente	
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir que la Secretaria realice la gestión de aportaciones.	
Evento desencadenador	La Secretaria realiza inicio de sesión	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)	Información para los pasos	
7. Accede al menú principal	Interfaz del menú principal	
8. Accede a la opción gestión de aportaciones	Opción de gestión de aportaciones	
9. Realiza la acción respectiva: Registrar, visualizar y modificar aportaciones.	Acción: registrar, visualizar y modificar	
10. Validar los datos de aportaciones	Datos de aportaciones	
11. Confirma la validación	Mensaje de confirmación de aportaciones.	
Precondiciones:	Realiza el Acceso al menú principal	
Postcondiciones:	La Secretaria accede a la opción de aportaciones.	
Suposiciones:	La Secretaria gestiona aportaciones de socios correctamente.	
Garantías de Éxito:	La Secretaria pudo realizar la gestión de aportaciones	
Garantía Mínima:	La Secretaria ingresa al menú principal	
Requerimientos cumplidos:	Permitir que la Secretaria realice la gestión de aportaciones.	
Cuestiones Pendientes:	¿Qué pasaría si no se pudo realizar el registro de las aportaciones?	
Prioridad	Media	
Riesgo:	Media	




Cobro de Aportado.

Nombre de caso de uso	cobrar Aportado	ID Único:08
Área		
Actor(es)	Secretaria	
Interesados	Gerente	
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir que la Secretaria realice el cobro Aportado.	
Evento desencadenador	La Secretaria realiza inicio de sesión	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)	Información para los pasos	
1. Accede al menú principal.	Interfaz del menú principal.	
2. Accede a la opción cobrar aportado.	Opción de cobrar aportado.	
3. Realiza la acción respectiva: Registrar, visualizar y modificar aportaciones.	Acción: registrar, visualizar y modificar	
4. Validar los datos de cobrar aportado.	Datos de aportado	
5. Confirma la validación.	Mensaje de confirmación de aportado.	
Precondiciones:	Realiza el Acceso al menú principal	
Postcondiciones:	La Secretaria accede a la opción de aportaciones.	
Suposiciones:	La Secretaria gestiona aportaciones de socios correctamente.	
Garantías de Éxito:	La Secretaria pudo realizar el cobro aportado.	
Garantía Mínima:	La Secretaria ingresa al menú principal.	
Requerimientos cumplidos:	Permitir que la Secretaria realice el cobro aportado.	
Cuestiones Pendientes:	¿Qué pasaría si no se pudo realizar el cobro de aportado?	
Prioridad	Media	
Riesgo:	Media	

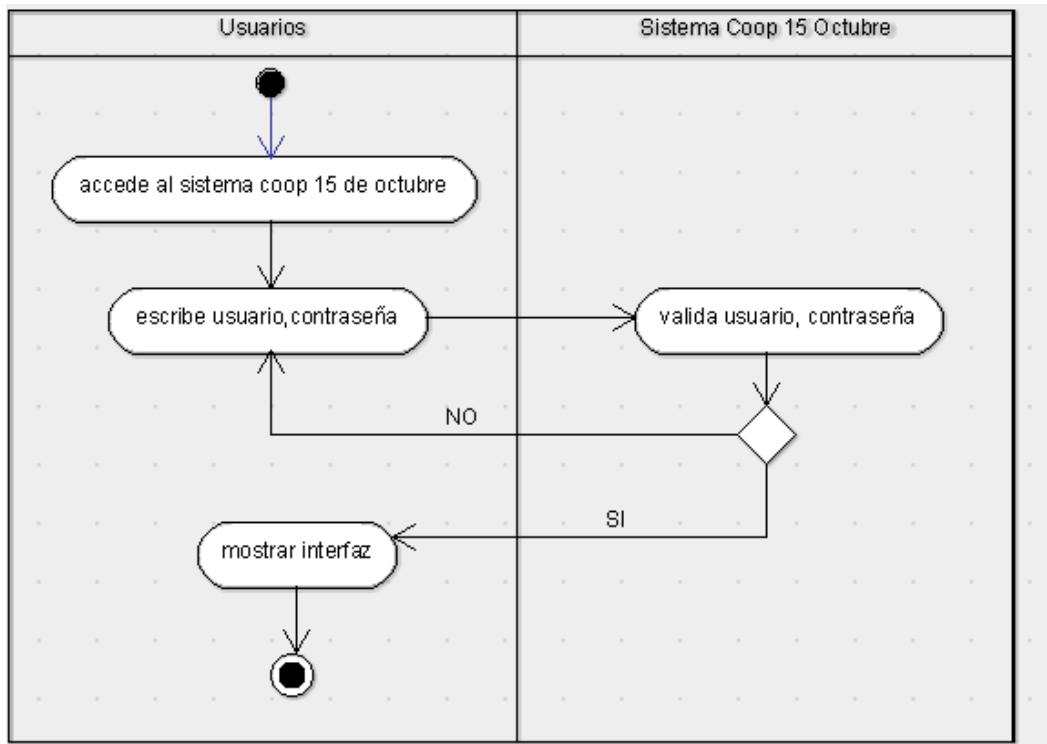
Emitir Aportado.

Nombre de caso de uso	Emitir Recibo de aportaciones	ID Único:09
Área		
Actor(es)	Secretaria	
Interesados	Gerente	
Nivel:	<input type="radio"/> Blanco <input type="radio"/> Cometa <input checked="" type="radio"/> Azul	
Descripción	Permitir que la Secretaria emita recibo de aportaciones.	
Evento desencadenador	La Secretaria realiza inicio de sesión	
Tipo de desencadenador:	Externo <input type="checkbox"/>	Temporal <input checked="" type="checkbox"/>
Pasos realizados (ruta principal)	Información para los pasos	
1. Iniciar sesión	Usuario y clave	
2. Validar usuario y clave	Validar usuario y clave	
3. Buscar socio por cedula	Cedula correcta	
4. Validar socio por cedula	cedula	
5. Cobro de aportaciones	aportaciones	
6. Emitir recibo de la aportación	Pago de la aportación	
Precondiciones:	Inicia sesión	
Postcondiciones:	Emite recibo de aportación	
Suposiciones:	La secretaria emite recibo correctamente	
Garantías de Éxito:	El recibo se emite exitosamente	
Garantía Mínima:	Ingrese sesión	
Requerimientos cumplidos:	Permitir que la Secretaria logre emitir en recibo.	
Cuestiones Pendientes:	¿Qué pasaría si no se pude emitir el recibo de las aportaciones?	
Prioridad	Media	
Riesgo:	Media	

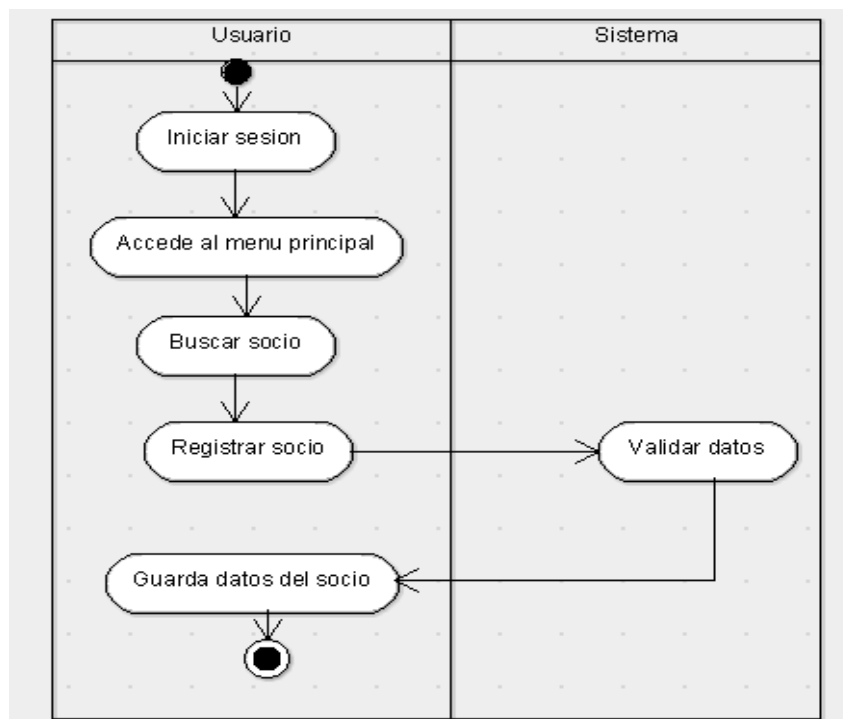
Visualizar Multas y Aportaciones

Nombre de caso de uso	Visualizar multas y aportaciones	ID Único:10
Área		
Actor(es)	usuario	
Interesados	gerente	
Nivel:	 Blanco  Cometa  Azul	
Descripción	Permitir que el usuario pueda visualizar las multas y aportaciones	
Evento desencadenador	El usuario inicie sesión	
Tipo de desencadenador:	Externo <input type="text"/>	Temporal <input type="text" value="*"/>
Pasos realizados (ruta principal)		Información para los pasos
1. Inicio de sesión		Ingresar usuario y contraseña
2. Validar usuario y contraseña		Usuario y contraseña registrados
3. Buscar socio por cedula		Mostrar los datos
4. Validar cedula socio		Datos visualizados
5. Visualizar multas y aportaciones		Multas y aportaciones
Precondiciones:	Inicio de sesión correctamente	
Postcondiciones:	Se ha Visualizado las multas y aportaciones	
Suposiciones:	Se asume que se debe registrar	
Garantías de Éxito:	Se visualiza las multas y aportaciones correctamente	
Garantía Mínima:	Poder inicias sesión	
Requerimientos cumplidos:	Permitir que el gerente pueda visualizar las multas y aportaciones	
Cuestiones Pendientes:	¿Qué pasaría si el usuario no puede iniciar sesión?	
Prioridad	Media	
Riesgo:	Media	

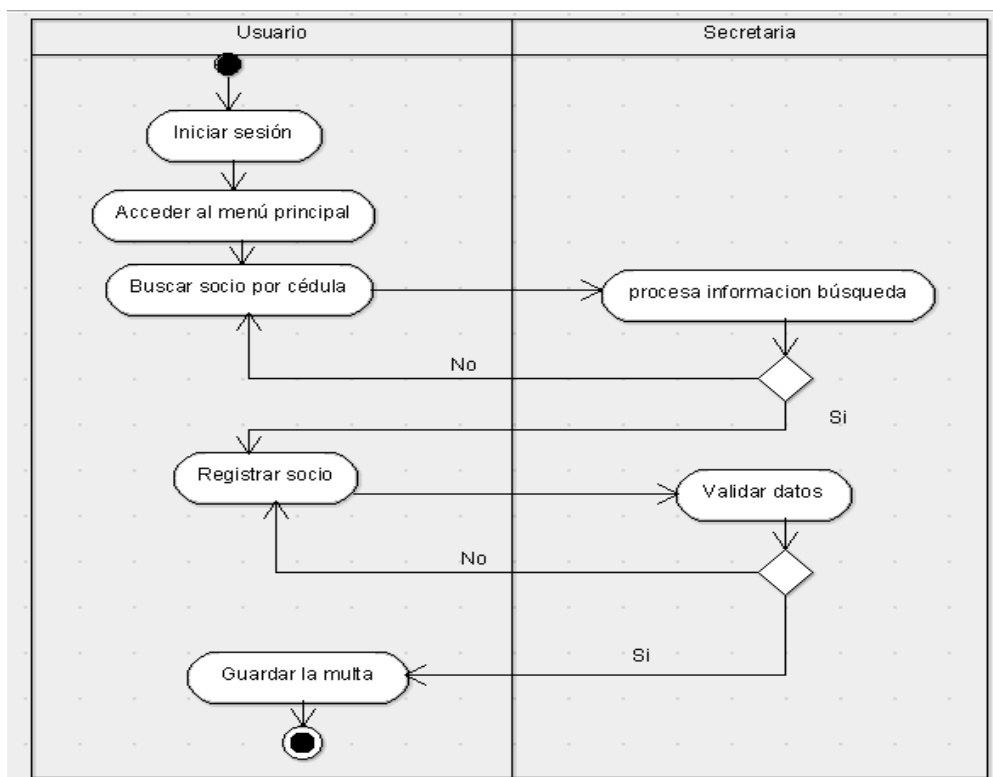
INICIAR SESIÓN



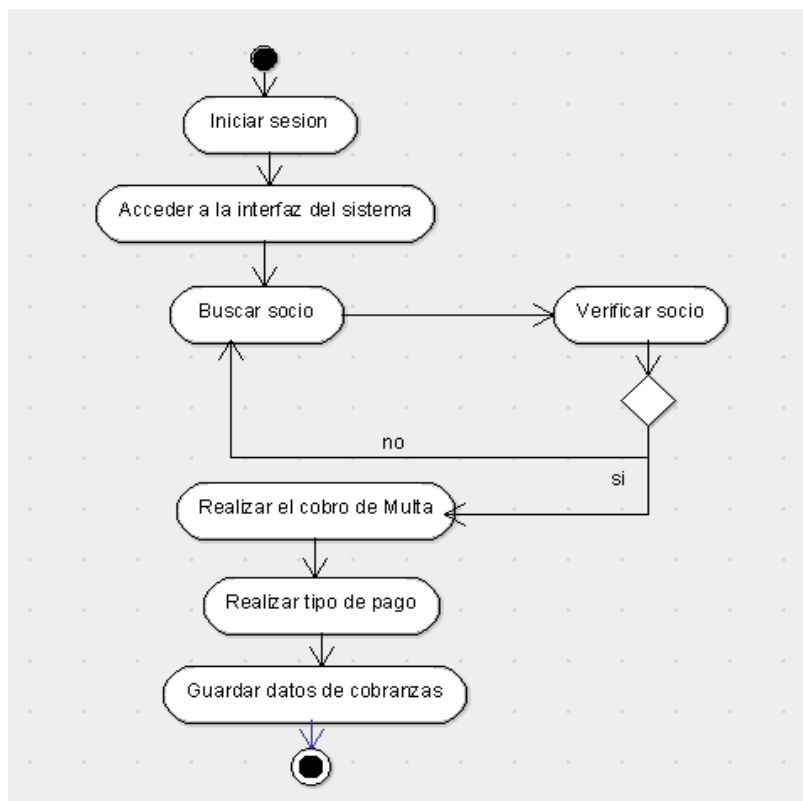
GESTIÓN DE SOCIO



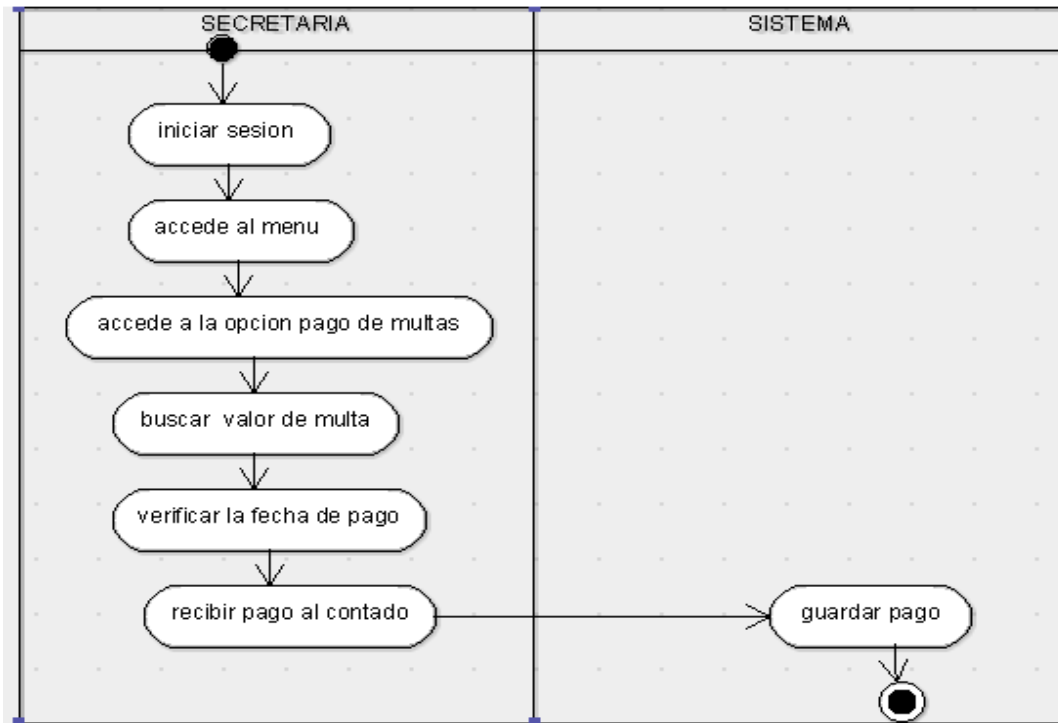
GESTIÓN DE MULTAS



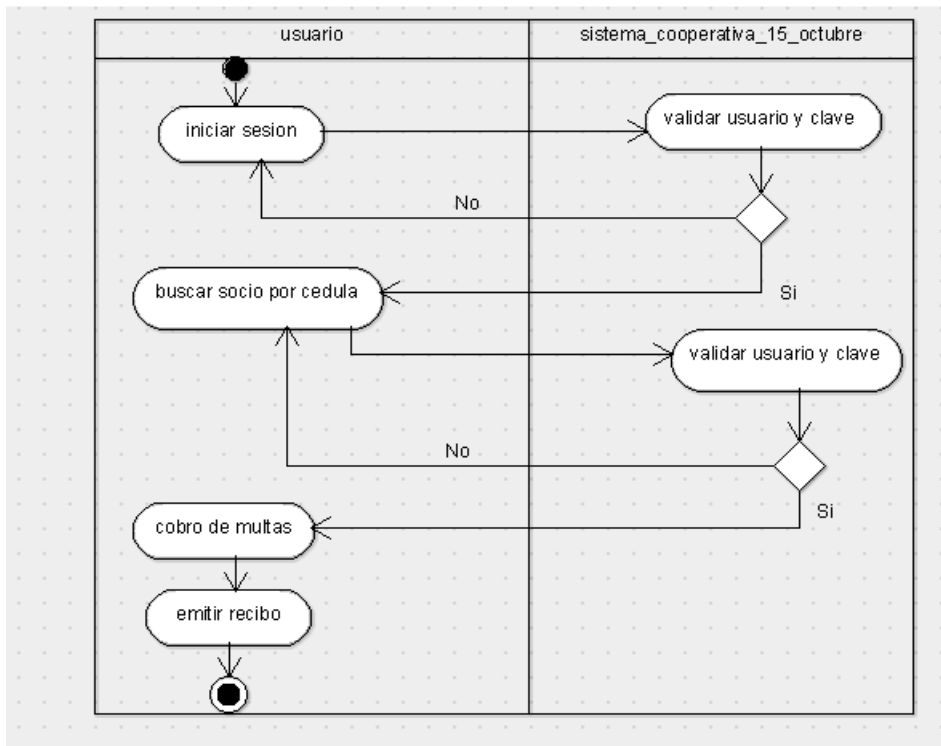
COBRAR MULTAS.



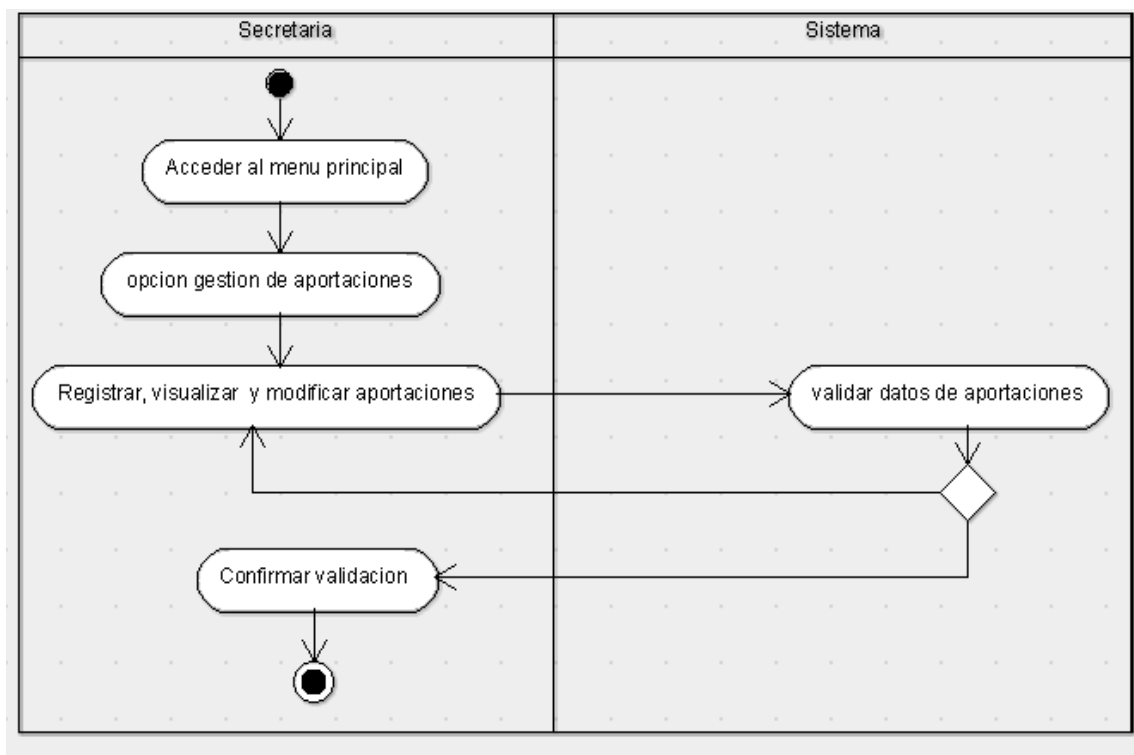
PAGAR MULTAS.



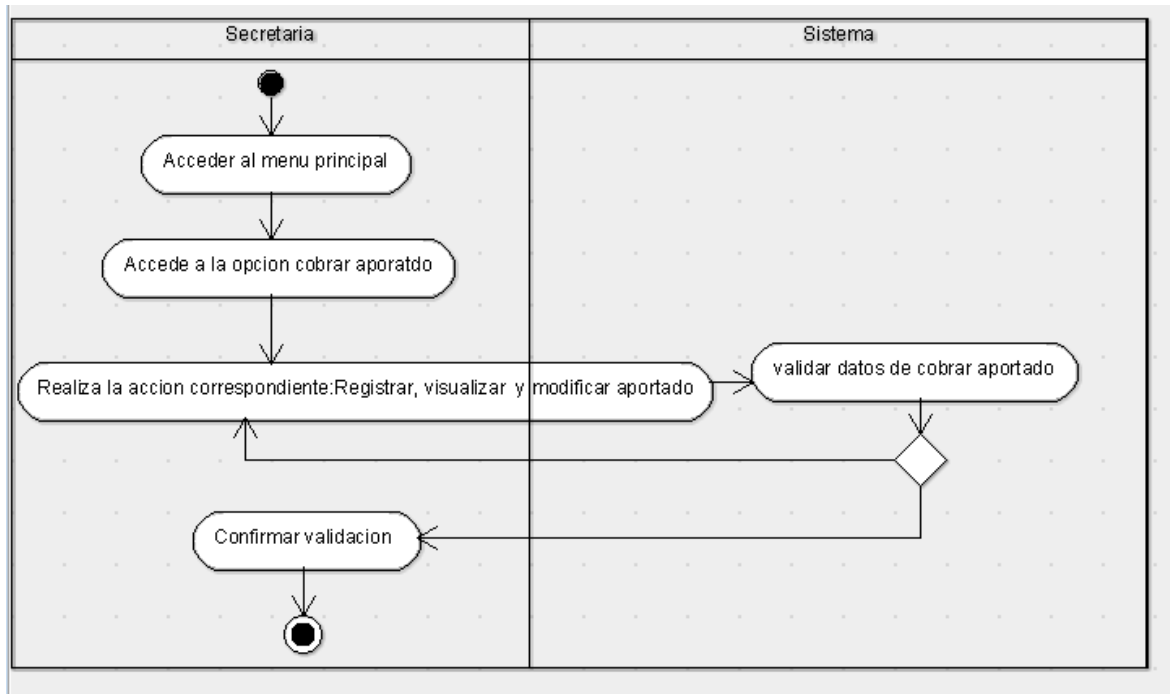
EMITIR MULTAS RECIBO



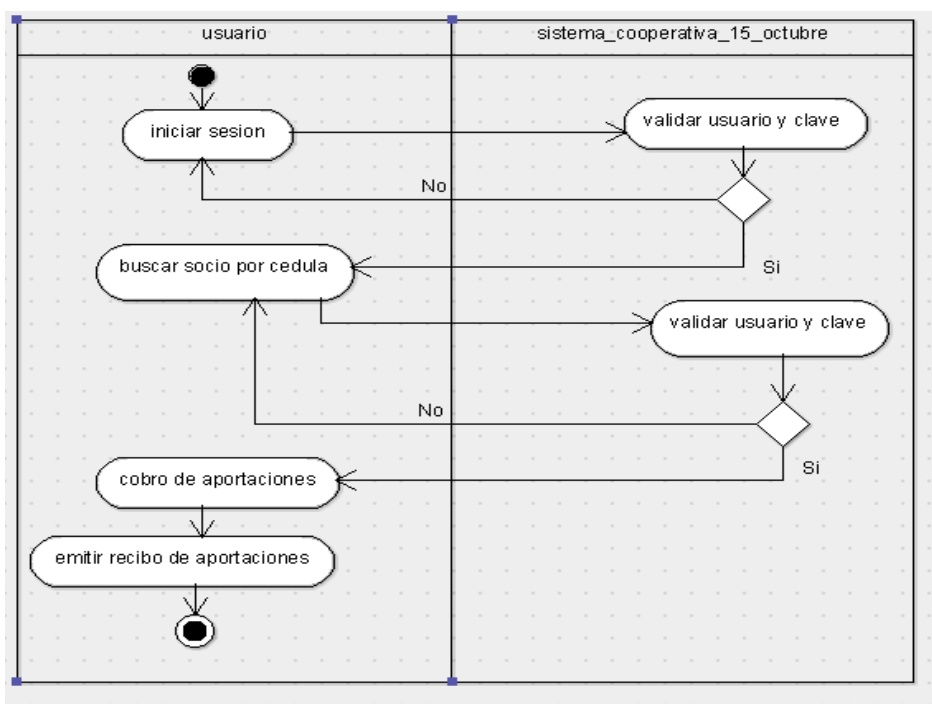
GESTIONAR APORTACIONES



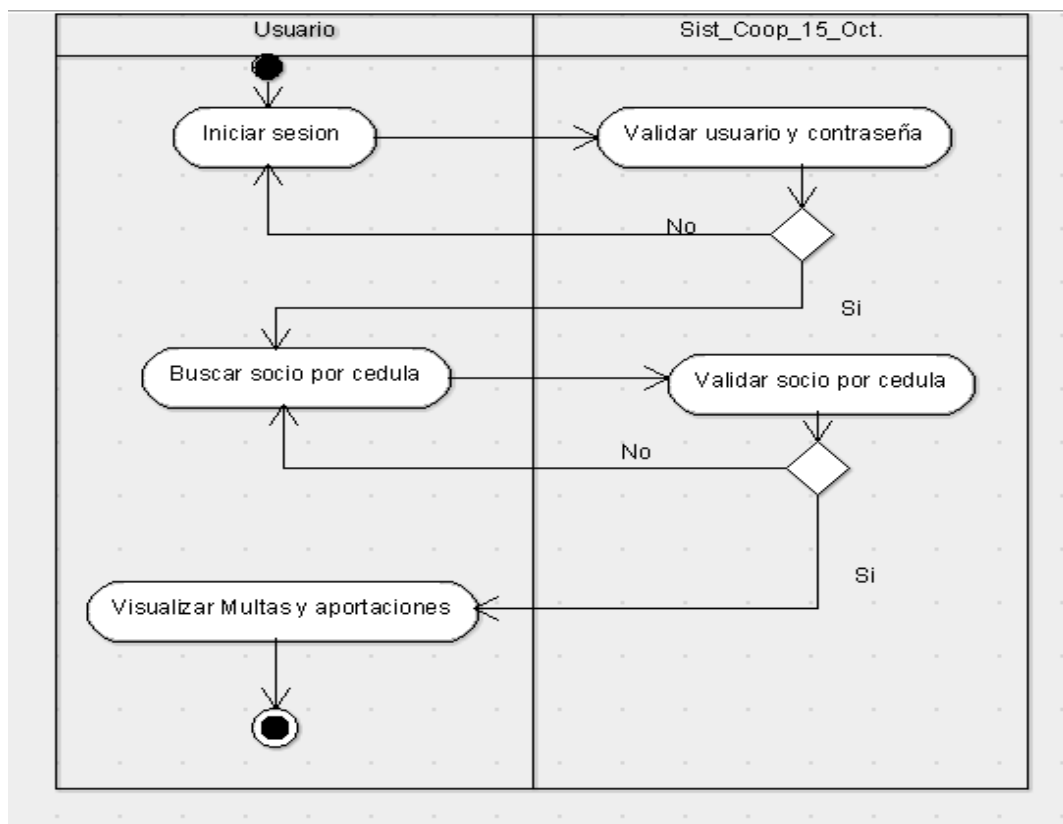
COBRAR APORTADO



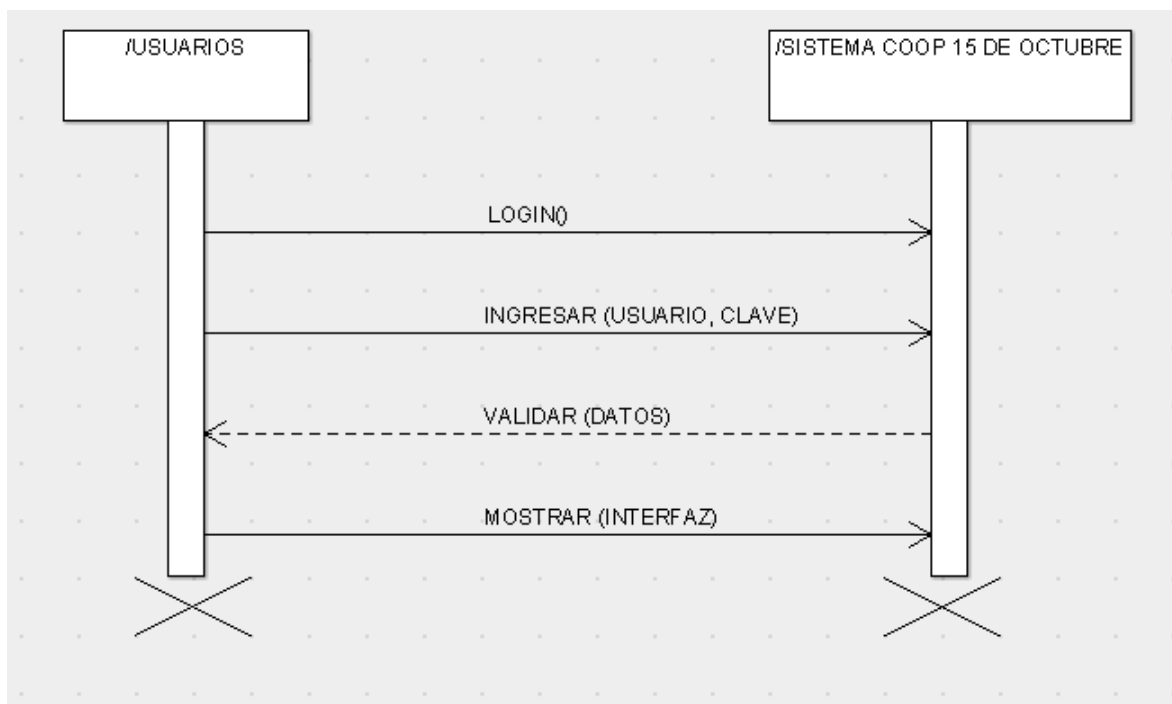
EMITIR APORTADO



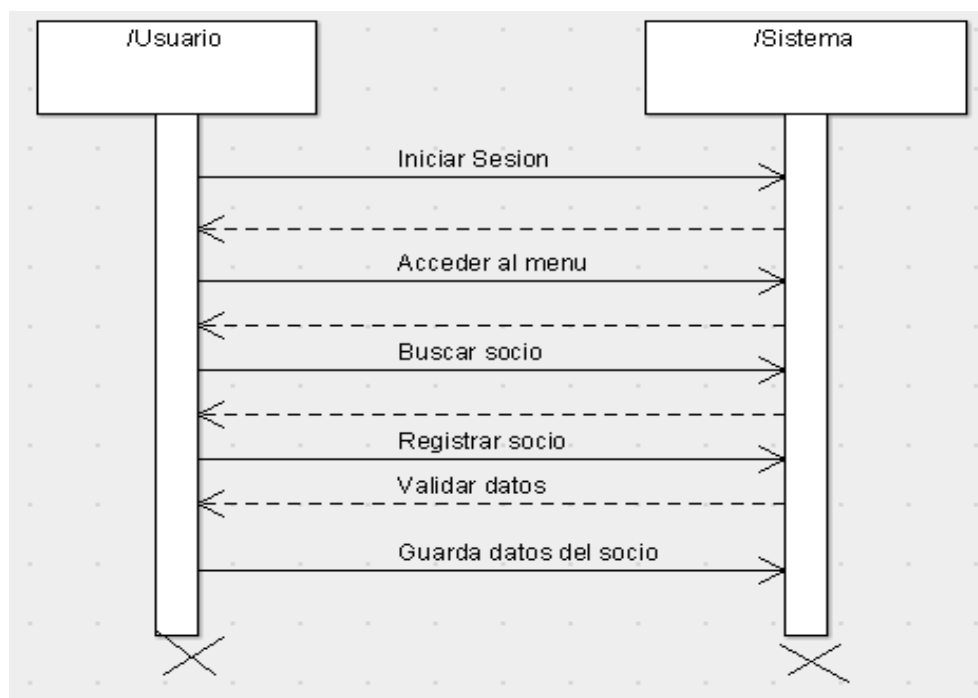
VISUALIZAR MULTAS Y APORTACIONES



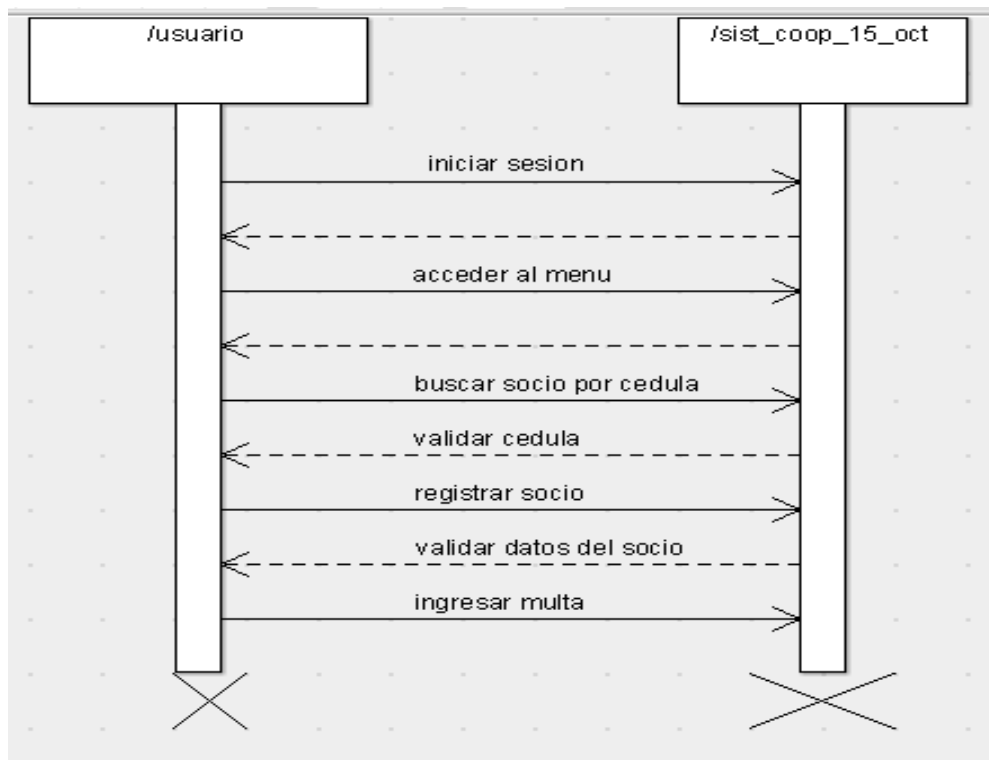
INICIO DE SESIÓN



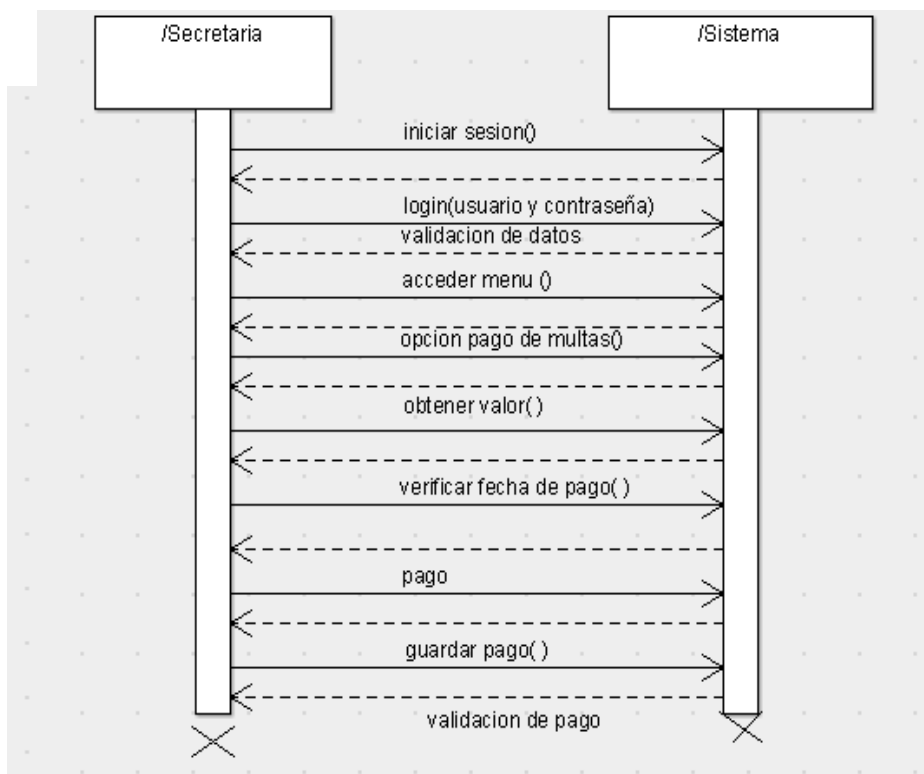
GESTIÓN DE SOCIO



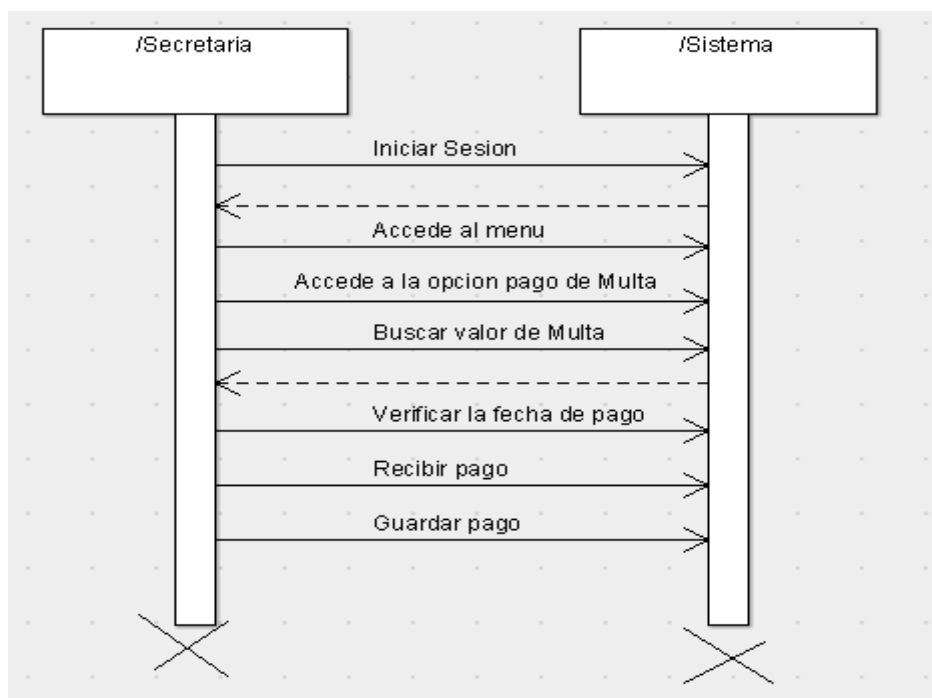
GESTIONAR DE MULTAS



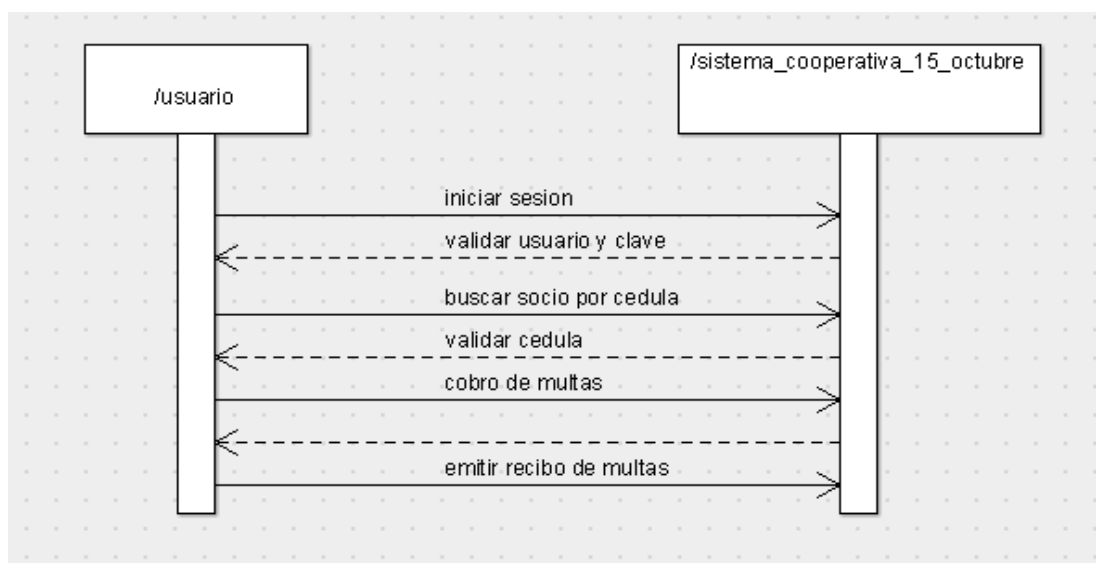
COBRAR MULTAS



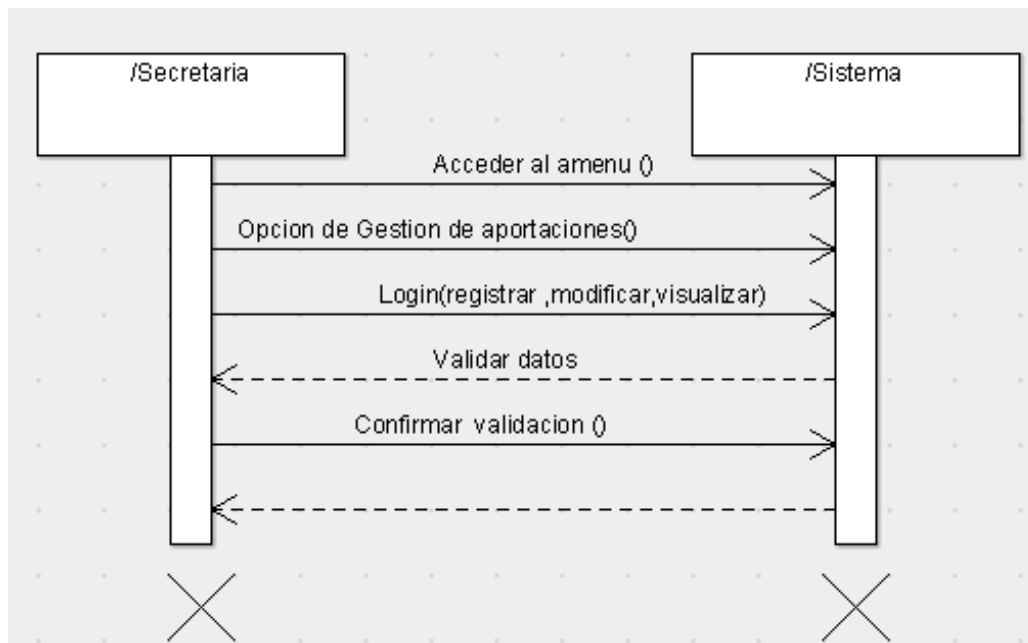
PAGAR MULTAS.



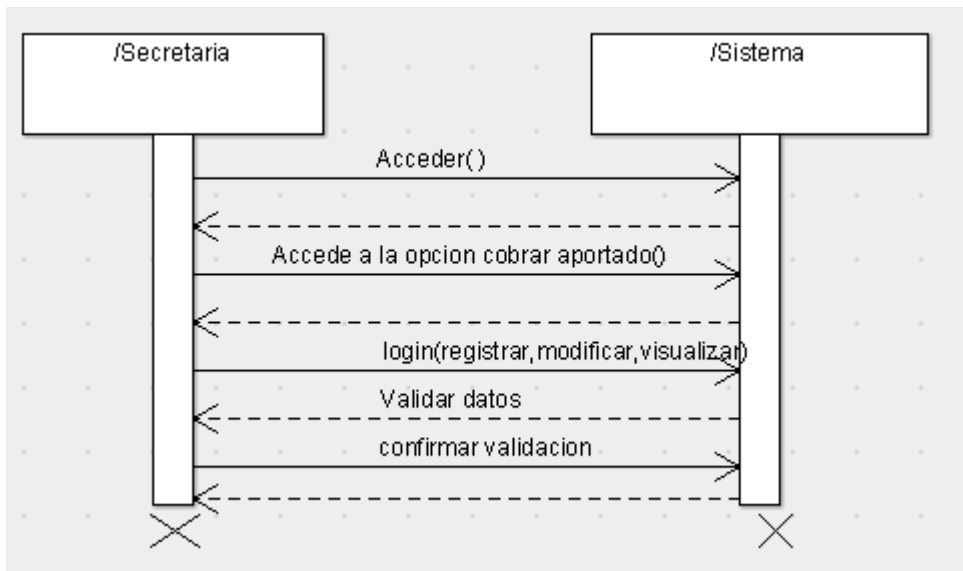
EMITIR MULTAS RECIBO.



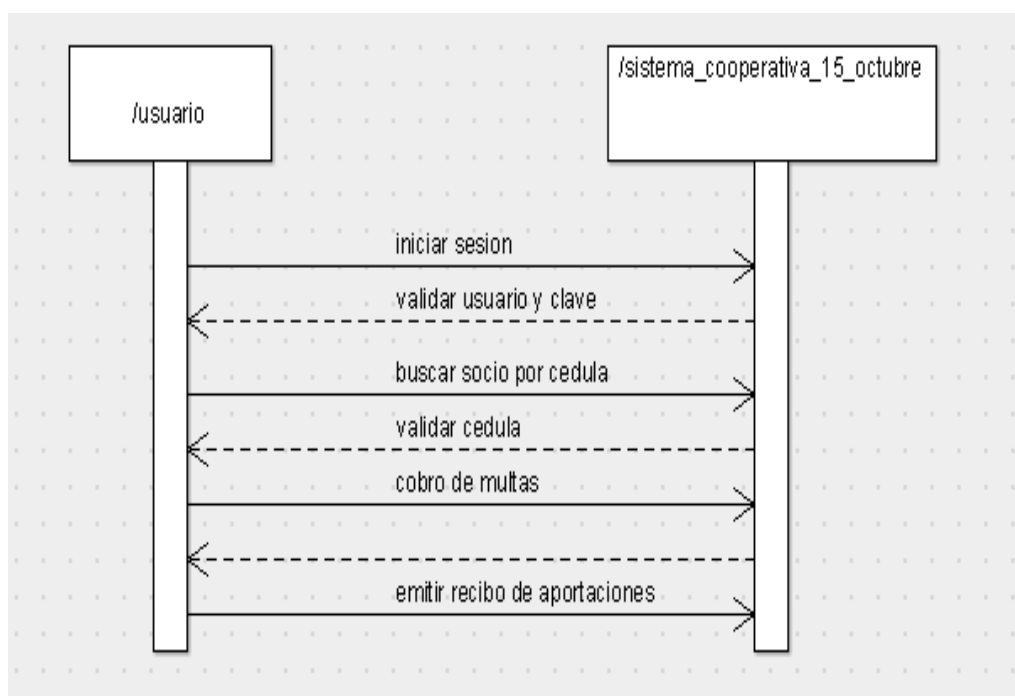
GESTIÓN DE APORTACIONES



COBRAR APORTADO



EMITIR APORTADO



VISUALIZAR MULTAS Y APORTACIONES.

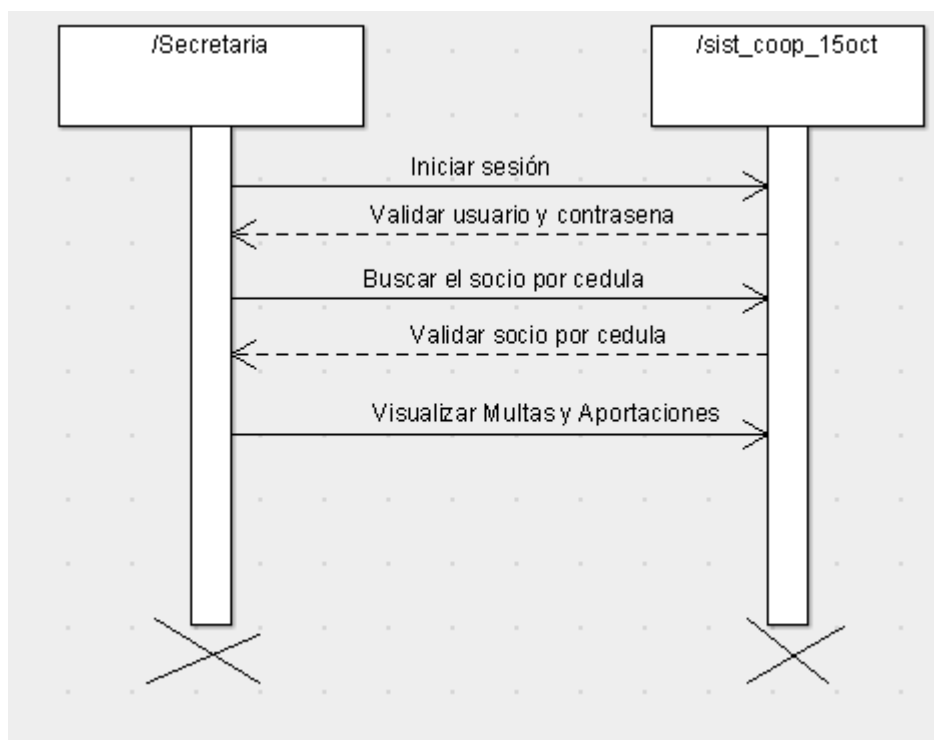
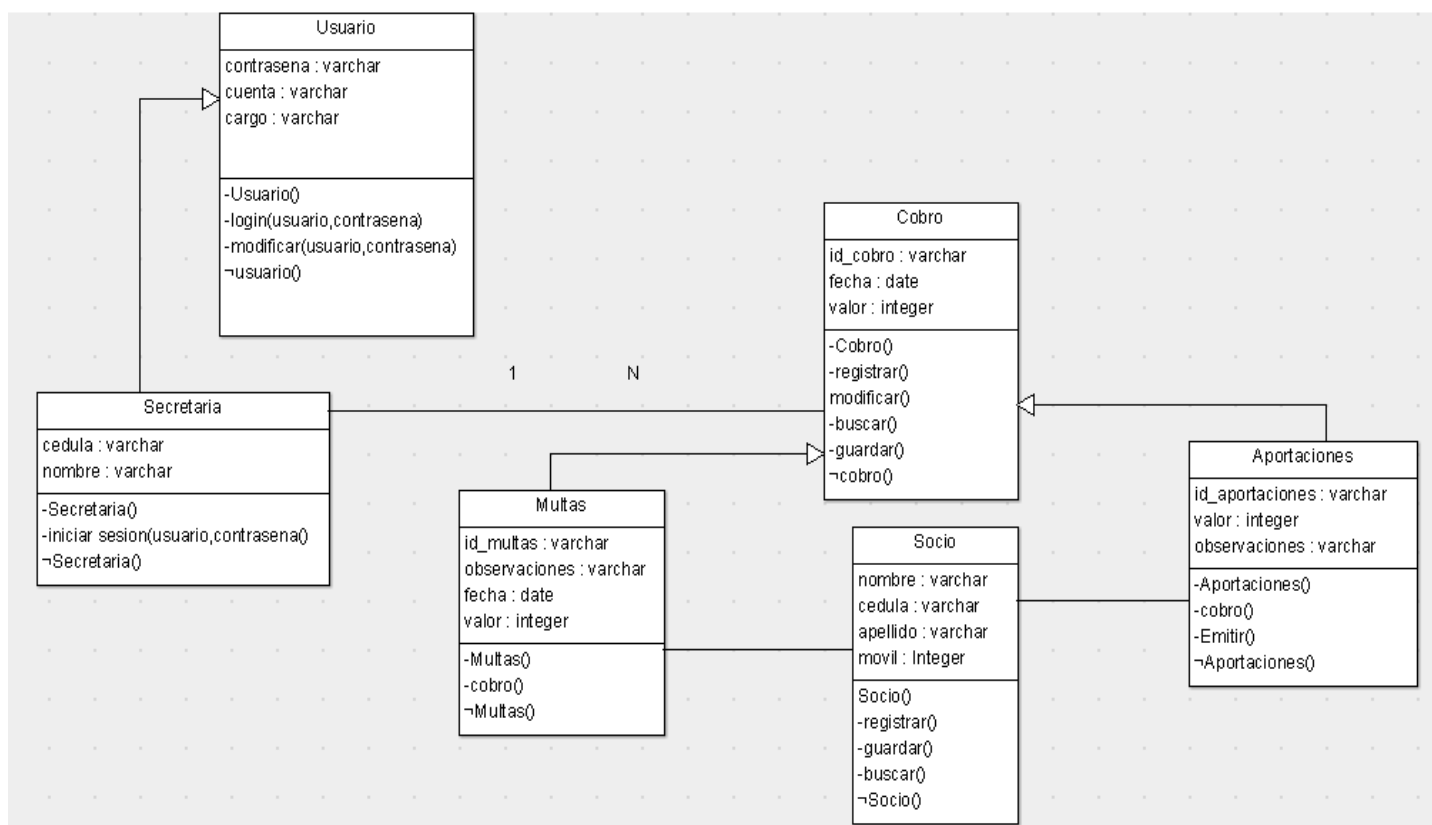
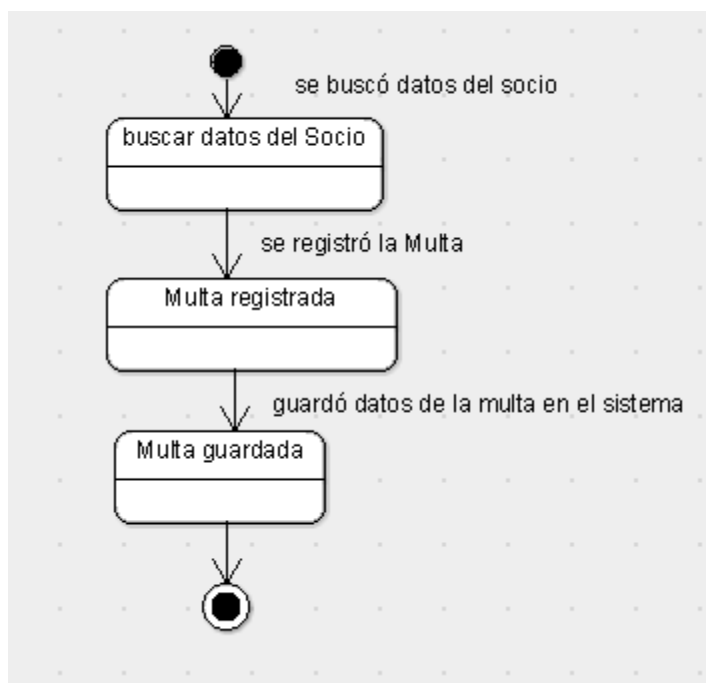


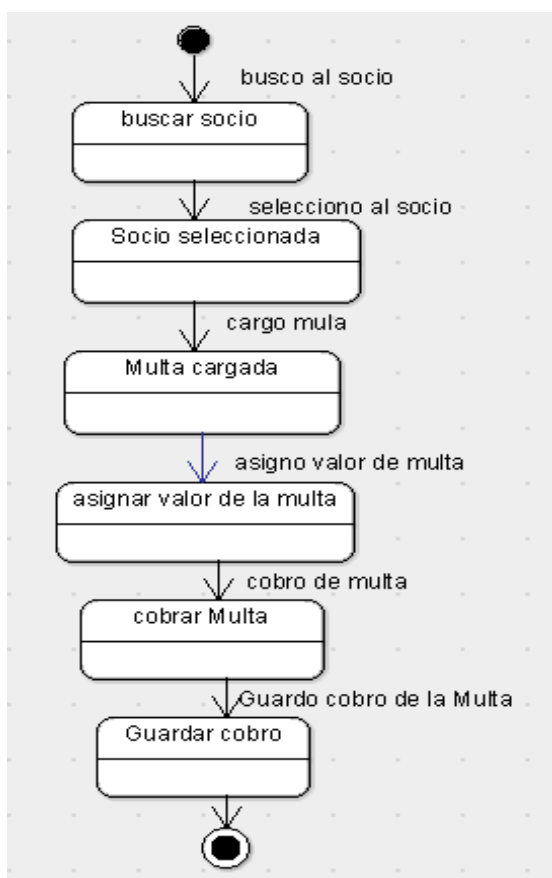
DIAGRAMA DE CLASES



Registro de Multa.



Cobrar Multa



Registro de Aportaciones

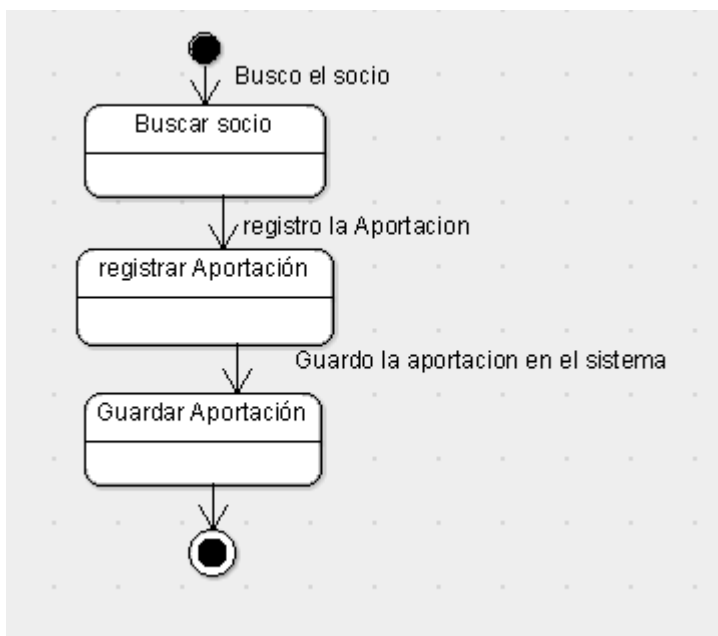
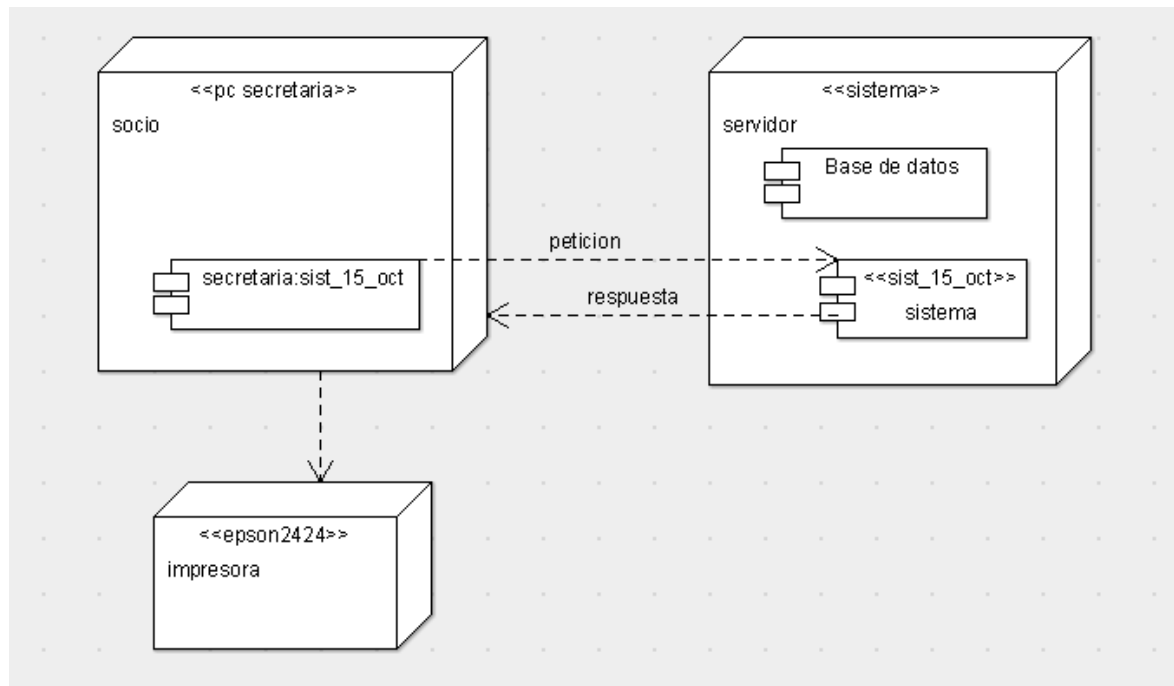


Diagrama de despliegue



3.3 IMPLEMENTACIÓN.

Iniciar sesión

```
Dim tbl_inicio As New ADODB.Recordset
Set tbl_inicio = CONEXION.Execute(" select * from tipo_usuario")
If Me.txt_usuario.Text = tbl_inicio!usuario And
Me.txt_contraseña.Text = tbl_inicio!contrasenia Then
    Frm_menu_asistente.Show
    Unload Me
End If
Set tbl_inicio = CONEXION.Execute(" select * from
tipo_usuario")
If Me.txt_usuario.Text = tbl_inicio!usuario And
Me.txt_contraseña.Text = tbl_inicio!contrasenia Then
    frm_jefe_talento_humano.Show
    Unload Me
End If
```

Guardar socios.

```
Private Sub cmdGuardar_Click()
If Len(Trim(txtcedula.Text)) > 9 And Len(Trim(txtMovil.Text)) > 0 And
Len(Trim(txtnombre.Text)) > 0 And Len(Trim(txtapellido.Text)) > 0 And
Len(Trim(txtDireccion.Text)) > 0 And Len(Trim(txtDireccion)) > 0 Then

    Dim respuestaID As Integer
    Set TBLGuardarSocio = Nothing
    TBLGuardarSocio.Open "select * from socio where cedula = '" &
txtcedula.Text & "'", CONEXION, adOpenDynamic, adLockOptimistic
    If Not (TBLGuardarSocio.EOF) Then
        MsgBox "Socio ya existe"
    Else
        Set TBLGuardarSocio = Nothing
        TBLGuardarSocio.Open "select * from socio ", CONEXION, adOpenDynamic,
adLockOptimistic
        TBLGuardarSocio.AddNew
        TBLGuardarSocio!cedula = txtcedula.Text
        TBLGuardarSocio!movil = txtMovil.Text
        TBLGuardarSocio!nombre = Trim(UCASE(txtnombre.Text))
        TBLGuardarSocio!apellido = Trim(UCASE(txtapellido.Text))
        TBLGuardarSocio!Direccion = Trim(UCASE(txtDireccion.Text))
        TBLGuardarSocio!telefono = Trim(UCASE(txtTelefono.Text))
        g
        TBLGuardarSocio.Update

        Call desactivarcajas
        Call cargargrid

    End If
Else
    MsgBox "Imposible guardar!.No se permiten campos vacios, ingresar
valores en los campos", vbExclamation
End If
```

End Sub

Modificar socios.

```
If Len(Trim(txtcedula.Text)) > 9 And Len(Trim(txtMovil.Text)) > 0 And
Len(Trim(txtnombre.Text)) > 0 And Len(Trim(txtapellido.Text)) > 0 And
Len(Trim(txtDireccion.Text)) > 0 And Len(Trim(txtDireccion)) > 0 Then

    Dim respuestaID As Integer

    Set TBLGuardarSocio = Nothing
    TBLGuardarSocio.Open "select * from socio where idsocio=" & IDSOCIO,
CONEXION, adOpenDynamic, adLockOptimistic
    TBLGuardarSocio!cedula = txtcedula.Text
    TBLGuardarSocio!movil = txtMovil.Text
    TBLGuardarSocio!nombre = Trim(UCase(txtnombre.Text))
    TBLGuardarSocio!apellido = Trim(UCase(txtapellido.Text))
    TBLGuardarSocio!Direccion = Trim(UCase(txtDireccion.Text))
    TBLGuardarSocio!telefono = Trim(UCase(txtTelefono.Text))
    TBLGuardarSocio.Update
    Call desactivarcajas
    Call cargargrid
Else
    MsgBox "Imposible guardar!.No se permiten campos vacios, ingresar
valores en los campos", vbExclamation
End If
```

Gestión de Multas

Búsqueda de socios

```
Dim z As Integer
z = GridBusqueda.Row
If z > 0 Then
    cisocio = GridBusqueda.TextMatrix(GridBusqueda.Row, 1)
    IDSOCIO = ModuloFunciones.buscarID("socio", "cedula", cisocio)
    Set TBLsocio = Nothing
    TBLsocio.Open "select * from socio where cedula='" & cisocio &
"'", CONEXION, adOpenDynamic, adLockOptimistic
    If Not (TBLsocio.EOF) Then
        TBLsocio.MoveFirst
        frmGestionMultas.txtSocio.Text = TBLsocio.Fields("cedula").Value
        frmGestionMultas.txtnombre.Text = TBLsocio.Fields("nombre").Value
        frmGestionMultas.txtapellido.Text =
TBLsocio.Fields("apellido").Value
    Else
        End If
        Unload Me
    End If
```

Agregar Multa

```
If Len(TxtMultas.Text) > 1 And Len(txtObservacion.Text) > 1 And
Len(txtValor.Text) > 1 Then
    Dim respuestaID As Integer
```

```

        Set TBLGuardarMulta = Nothing
        TBLGuardarMulta.Open "select * from multas where
nombre_aportacion= '" & UCase(TxtMultas.Text) & "'", CONEXION,
adOpenDynamic, adLockOptimistic
        If Not (TBLGuardarMulta.EOF) Then
            MsgBox "La multa ya existe"

        Else

            Set TBLGuardarMulta = Nothing
            TBLGuardarMulta.Open "select * from multas", CONEXION,
adOpenDynamic, adLockOptimistic
            TBLGuardarMulta.AddNew
            TBLGuardarMulta!nombre_multa =
Trim(UCase(TxtMultas.Text))
            TBLGuardarMulta!observacion =
Trim(UCase(Me.txtObservacion.Text))
            TBLGuardarMulta!valor = txtValor.Text

            TBLGuardarMulta.Update

            Call CargarMulta
        End If
    Else
        MsgBox "Ingrese registro"
    End If
    TxtMultas.Text = ""
    txtObservacion.Text = ""
    txtValor.Text = ""

```

Modificar Multas

```

If Len(TxtMultas.Text) > 1 And Len(txtObservacion.Text) > 1 And
Len(txtValor.Text) > 1 Then
    Dim respuestaID As Integer
    Set TBLGuardarMulta = Nothing
        Set TBLGuardarMulta = Nothing
        TBLGuardarMulta.Open "select * from multas where
idmulta=" & cimulta, CONEXION, adOpenDynamic, adLockOptimistic
        TBLGuardarMulta!nombre_multa =
Trim(UCase(TxtMultas.Text))
        TBLGuardarMulta!observacion =
Trim(UCase(Me.txtObservacion.Text))
        TBLGuardarMulta!valor = txtValor.Text
        TBLGuardarMulta.Update
        Call CargarMulta
    End If
    TxtMultas.Text = ""
    txtObservacion.Text = ""
    txtValor.Text = ""

```

Guardar Multa al Socio

```

Dim tblguardarsociomulta1 As New ADODB.Recordset
    socio_ingresado = buscarID("socio", "cedula", IDSOCIO)

```

```

        multa_seleccionada = buscarID("multas", "nombre_multa",
Me.cmbMultas.Text)

        Set tblgurdarsociomulta1 = Nothing
        tblguardarsociomulta1.Open "select * from socio_multas",
CONEXION, adOpenDynamic, adLockOptimistic
        tblguardarsociomulta1.AddNew
        tblguardarsociomulta1!IDSOCIO = socio_ingresado
        tblguardarsociomulta1!idmulta = multa_seleccionada

        tblguardarsociomulta1!fecha = Me.DTPFecha.Value
        tblguardarsociomulta1.Update
        Call desactivarcajas
        Call cargargrid

```

Modificar Multa al socio

```

If Len(Trim(cmbMultas.Text)) > 0 Then

    Dim tblguardarsociomulta1 As New ADODB.Recordset
    Dim z As Integer
    z = MsgBox("Desea Guardar el Socio?", vbYesNo, "Gestion Multas")
    If z = vbYes Then
        multa_seleccionada = buscarID("multas", "nombre_multa",
cmbMultas.Text)

        Set tblgurdarsociomulta1 = Nothing
        tblguardarsociomulta1.Open "select * from socio_multas",
CONEXION, adOpenDynamic, adLockOptimistic
        tblguardarsociomulta1!idmulta = multa_seleccionada
        tblguardarsociomulta1!fecha = Me.DTPFecha.Value
        tblguardarsociomulta1.Update
        Call cargargrid
    End If
Else
    MsgBox "No se puede Guardar Datos vacios!,Ingresar valores en los
campos", vbInformation, ""

```

Reporte de la Multa

```

Dim nombreseccion As String
Dim rsReporte As New ADODB.Recordset
Dim sql As String

Set rsReporte = Nothing
If optTodos.Value = True Then
    sql = " select
s.cedula,s.nombre,s.apellido,m.nombre_multa,sm.fecha,m.observacion,m.valo
r from socio s join socio_multas sm on s.idsocio=sm.idsocio join multas
m on m.idmulta=sm.idmulta"
End If
If optcancelado.Value = True Then
    sql = "select
s.cedula,s.nombre,s.apellido,m.nombre_multa,sm.fecha,sm.estado_pago from

```

```

socio s join socio_multas sm on s.idsocio=sm.idsocio join multas m on
m.idmulta=sm.idmulta where sm.estado_pago='CANCELADO'"
End If

If optdebe.Value = True Then
    sql = "select
s.cedula,s.nombre,s.apellido,m.nombre_multa,sm.fecha,sm.estado_pago from
socio s join socio_multas sm on s.idsocio=sm.idsocio join multas m on
m.idmulta=sm.idmulta where sm.estado_pago='DEBE'"
End If
Set rsReporte = CONEXION.Execute(sql)
If Not rsReporte Is Nothing Then

    nombreseccion = "sesion1"

    With ReporteMultas

        .LeftMargin = 0
        .RightMargin = 0
        .TopMargin = 0
        .BottomMargin = 0

        .Sections(nombreseccion).Controls.Item("txtcedula").DataField =
"cedula"
        .Sections(nombreseccion).Controls.Item("txtnombre").DataField =
"nombre"
        .Sections(nombreseccion).Controls.Item("txtapellido").DataField =
"apellido"
        .Sections(nombreseccion).Controls.Item("txtmulta").DataField =
"nombre_multa"
        .Sections(nombreseccion).Controls.Item("txtfecha").DataField =
"fecha"
        Set .DataSource = rsReporte
        .Show vbModal

        Set rsReporte = Nothing

    End With
End If

```

Gestion Aportaciones

Busqueda de socios

```

Dim z As Integer
z = GridBusqueda.Row
If z > 0 Then
    cisocio = GridBusqueda.TextMatrix(GridBusqueda.Row, 1)
    IDSOCIO = ModuloFunciones.buscarID("socio", "cedula", cisocio)

    Set TBLSocio = Nothing
    TBLSocio.Open "select * from socio where cedula='" & cisocio &
"'", CONEXION, adOpenDynamic, adLockOptimistic
    If Not (TBLSocio.EOF) Then
        TBLSocio.MoveFirst
    End If
End If

```

```

        frmAportaciones.txtSocio.Text = TBLSocio.Fields("cedula").Value
        frmAportaciones.txtnombre.Text = TBLSocio.Fields("nombre").Value
        frmAportaciones.txtapellido.Text =
TBLSocio.Fields("apellido").Value
    Else
    End If
    Unload Me
End If

```

Agregar Aportaciones

```

If Len(TxtAportaciones.Text) > 1 And Len(TxtDescripcion.Text) > 1 And
Len(TxtValor.Text) > 1 Then
    Dim respuestaID As Integer
    Set TBLGuardarAportacion = Nothing
    TBLGuardarAportacion.Open "select * from aportaciones where
aportacion= '" & UCase(TxtAportaciones.Text) & "'", CONEXION,
adOpenDynamic, adLockOptimistic
    If Not (TBLGuardarAportacion.EOF) Then
        MsgBox "La aportacion ya existe"
    Else

        Set TBLGuardarAportacion = Nothing
        TBLGuardarAportacion.Open "select * from aportaciones",
CONEXION, adOpenDynamic, adLockOptimistic
        TBLGuardarAportacion.AddNew
        TBLGuardarAportacion!valor = Me.TxtValor.Text
        TBLGuardarAportacion!aportacion =
Trim(UCase(Me.TxtAportaciones.Text))
        TBLGuardarAportacion!descripcion =
Trim(UCase(TxtDescripcion.Text))

        TBLGuardarAportacion.Update
        Call CargarAportacion
    End If
Else
    MsgBox "Ingrese registro"
End If
TxtAportaciones.Text = ""
TxtDescripcion.Text = ""
TxtValor.Text = ""

```

Guardar Aportaciones.

```

If Len(Me.cmbAportaciones.Text) > 0 And Me.DTPFecha.Value = False Then
    Dim TBLguardarsocioaportacion As New ADODB.Recordset
    socio_ingresado = buscarID("socio", "cedula", Me.txtSocio.Text)
    aportacion_seleccionada = buscarID("aportaciones", "aportacion",
Me.cmbAportaciones.Text)
    Set TBLguardarsocioaportacion = Nothing
    TBLguardarsocioaportacion.Open "select * from
socio_aportaciones", CONEXION, adOpenDynamic, adLockOptimistic
    TBLguardarsocioaportacion.AddNew
    TBLguardarsocioaportacion!IDSOCIO = socio_ingresado

```

```

        TBLguardarsocioaportacion!idaportaciones =
aportacion_seleccionada
        TBLguardarsocioaportacion!fecha = Me.DTPFecha.Value
        TBLguardarsocioaportacion.Update
        Call desactivarcajas
        Call cargargrid
    Else
        MsgBox "Imposible guardar!.No se permiten campos vacios, ingresar
valores en los campos", vbExclamation
    End If

```

Modificar Aportaciones

If Len(TxtAportaciones.Text) > 1 And **Len**(TxtDescripcion.Text) > 1 And **Len**(TxtValor.Text) > 1 **Then**

```

        Dim respuestaID As Integer
        Set TBLGuardarAportacion = Nothing
        Set TBLGuardarAportacion = Nothing
        TBLGuardarAportacion.Open "select * from aportaciones
where idaportaciones=" & ciaportacion, CONEXION, adOpenDynamic,
adLockOptimistic

        TBLGuardarAportacion!valor = Me.TxtValor.Text
        TBLGuardarAportacion!aportacion =
Trim(UCase(Me.TxtAportaciones.Text))
        TBLGuardarAportacion!descripcion =
Trim(UCase(TxtDescripcion.Text))

        TBLGuardarAportacion.Update
        Call CargarAportacion
    End If
    TxtAportaciones.Text = ""
    TxtDescripcion.Text = ""
    TxtValor.Text = ""

```

Reporte de Aportaciones

```

Dim nombreseccion As String
Dim rsReporte As New ADODB.Recordset
Dim sql As String

Set rsReporte = Nothing
If optTodos.Value = True Then
    sql = " select
s.cedula,s.nombre,s.apellido,a.aportacion,sa.fecha,a.descripcion,a.valor
from socio s join socio_aportaciones sa on s.idsocio=sa.idsocio join
aportaciones a on a.idaportaciones=sa.idaportaciones"
End If
If optcancelado.Value = True Then
    sql = "select
s.cedula,s.nombre,s.apellido,a.aportacion,sa.fecha,sa.estado_pago from
socio s join socio_aportaciones sa on s.idsocio=sa.idsocio join
aportaciones a on a.idaportaciones=sa.idaportaciones where
sa.estado_pago='CANCELADO'"
End If

```



```

    If optdebe.Value = True Then
        sql = "select
s.cedula,s.nombre,s.apellido,a.aporte,sa.fecha,sa.estado_pago from
socio s join socio_aportaciones sa on s.idsocio=sa.idsocio join
aportaciones a on a.idaportaciones=sa.idaportaciones where
sa.estado_pago='DEBE'"
    End If
    Set rsReporte = CONEXION.Execute(sql)
    If Not rsReporte Is Nothing Then

        nombreseccion = "sesion1"

        With ReporteAportaciones

            .LeftMargin = 0
            .RightMargin = 0
            .TopMargin = 0
            .BottomMargin = 0

            .Sections(nombreseccion).Controls.Item("txtcedula").DataField =
"cedula"
            .Sections(nombreseccion).Controls.Item("txtnombre").DataField =
"nombre"
            .Sections(nombreseccion).Controls.Item("txtapellido").DataField =
"apellido"
            .Sections(nombreseccion).Controls.Item("txtaporte").DataField =
"aporte"
            .Sections(nombreseccion).Controls.Item("txtfecha").DataField =
"fecha"
            Set .DataSource = rsReporte
            .Show vbModal

            Set rsReporte = Nothing

        End With
    End If

```

Cobranzas de multas.

```

Dim z As Integer
z = GridBusquedaCobranzas.Row
If z > 0 Then
    cisocio =
GridBusquedaCobranzas.TextMatrix(GridBusquedaCobranzas.Row, 0)
    idsociomulta = ModuloFunciones.buscarID("socio", "cedula",
cisocio)
    IDSOCIO1 = idsociomulta
    Set TBLCobrar = Nothing
    TBLCobrar.Open "select * from socio where cedula= '" & cisocio &
"'", CONEXION, adOpenDynamic, adLockOptimistic

    TBLCobrar.MoveFirst
    FrmCobranzasMultas.txtcedula.Text =
TBLCobrar.Fields("cedula").Value
    FrmCobranzasMultas.txtnombre.Text =
TBLCobrar.Fields("nombre").Value

```

```

        FrmCobranzasMultas.txtapellido.Text =
TBLCobrar.Fields("apellido").Value

```

```

Else
End If
Unload Me

```

Enviar a Excel Multas

```

Dim fso, f, i, objeto1
Dim cadena As String
Dim NUM As Integer
Dim Objeto As Object

Set Objeto = Nothing
Set objeto1 = CreateObject("Excel.Application")
objeto1.Visible = True
objeto1.workbooks.Open FileName:=App.Path & "\Recibos\RECIBO.xlsx"
cadena = App.Path & "\Recibos\RECIBO.xlsx"

objeto1.cells(4, 4).Value = Trim(Str(Date))

With objeto1 '.worksheets("RecibosMultas")

.cells(4, 4).Value = Trim(Str(Date))
.cells(8, 2).Value = txtcedula.Text
.cells(9, 2).Value = Me.txtnombre.Text & " " & Me.txtapellido.Text
.cells(19, 4).Value = Me.lblValor.Caption

NUM = 13
Dim cod As Integer
cod = 1
For i = 1 To UBMULTAS.Rows
    If UBMULTAS.TextMatrix(i, 1) = 1 Then
        .cells(NUM, 1).Value = cod
        .cells(NUM, 2).Value = UBMULTAS.TextMatrix(i, 3)
        .cells(NUM, 3).Value = UBMULTAS.TextMatrix(i, 5)
        .cells(NUM, 4).Value = UBMULTAS.TextMatrix(i, 4)
        cod = cod + 1
        NUM = NUM + 1
    End If
Next

End With

```

Cobranzas de Aportaciones.

```

Dim z As Integer
z = GridBusquedaCobranzas.Row
If z > 0 Then
    cisocio =
GridBusquedaCobranzas.TextMatrix(GridBusquedaCobranzas.Row, 0)

```

```

        idsociomulta = ModuloFunciones.buscarID("socio", "cedula",
cisocio)
        IDSOCIO1 = idsociomulta
        Set TBLCobrar = Nothing
        TBLCobrar.Open "select * from socio where cedula= '" & cisocio &
        "'", CONEXION, adOpenDynamic, adLockOptimistic

        TBLCobrar.MoveFirst
        FrmCobranzasAportaciones.txtcedula.Text =
TBLCobrar.Fields("cedula").Value
        FrmCobranzasAportaciones.txtnombre.Text =
TBLCobrar.Fields("nombre").Value
        FrmCobranzasAportaciones.txtapellido.Text =
TBLCobrar.Fields("apellido").Value
        Else
        End If
        Unload Me

```

Enviar a Excel Aportaciones

```

Dim fso, f, i, objetol
Dim cadena As String
Dim NUM As Integer
Dim Objeto As Object

Set Objeto = Nothing
Set objetol = CreateObject("Excel.Application")
objetol.Visible = True
objetol.workbooks.Open FileName:=App.Path & "\Recibos\RECIBO.xlsx"
cadena = App.Path & "\Recibos\RECIBO.xlsx"

        objetol.cells(4, 4).Value = Trim(Str(Date))

With objetol
        .cells(4, 4).Value = Trim(Str(Date))
        .cells(8, 2).Value = txtcedula.Text
        .cells(9, 2).Value = Me.txtnombre.Text
        .cells(19, 4).Value = Me.lblValor.Caption

NUM = 13

For i = 1 To UBAPORTACIONES.Rows - 1
        .cells(NUM, 1).Value = i
        .cells(NUM, 2).Value = UBAPORTACIONES.TextMatrix(i, 3)
        .cells(NUM, 3).Value = UBAPORTACIONES.TextMatrix(i, 5)
        .cells(NUM, 4).Value = UBAPORTACIONES.TextMatrix(i, 4)

        'insertar fila en excel'
        'objeto.Range(LTrim(Str(NUM)) & ":" & LTrim(Str(NUM + 1))).Select
        'objeto.Selection.Insert

        NUM = NUM + 1

Next

```

3.5 PRUEBAS.

N°	REQUISITOS	CUMPLE	
		SI	NO
Requisito 1	Iniciar sesión	✓	
Requisito 2	Gestionar Socio	✓	
Requisito 3	Gestionar Multas	✓	
Requisito 4	Cobrar Multas	✓	
Requisito 5	Emitir multas recibo	✓	
Requisito 6	Gestionar Aportaciones	✓	
Requisito 7	Gestionar cobro de Aportaciones.	✓	
Requisito 8	Emitir Aportado	✓	

4 CONCLUSIONES

De la aplicación de los procesos utilizados para el desarrollo del manual y con la aplicación metodología sirvió de ayuda en la ayuda del diseño y construcción del software con las especificaciones necesarias para su manejo.

5 RECOMENDACIONES.

Recomendamos un buen uso en el manual permitiendo indicar todos los procesos en el proyecto se desea que haya una mejora continua del mismo, por lo tanto se recomienda a futuros estudiantes que tenga interés en el proyecto, la implementación del sistema con más distribuciones para la demanda y el tiempo de espera, en el proceso de optimización.

6 ANEXOS

Semana	Actividades	JUNIO				JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Visitar a la cooperativa y realizar la entrevista al Gerente.																				
2	Emplear el análisis FODA.																				
3	Obtener las entidades en el sistema, identificación del problema en la empresa.																				
4	Elaborar el diagrama entidad/relación, diseño del diagrama relación de la base de datos.																				
5	Realizar el diagrama de caso de uso y redactar los escenarios de los casos de uso.																				
6	Elaborar los diagramas de actividades y crear los diagramas de secuencia.																				
7	Diseñar el diagrama de clase, elaborar los diagramas de estados.																				
8	Construir el diagrama de despliegue, diseño de las interfaces en VISUAL BASIC 6.0																				
9	Crear la base de datos en postgresql, realizar y establecer la conexión de postgresql en VB 6.0.																				
10	Programa de formularios en visual BASIC																				
11	Elaborar las consultas en programa postgresql																				
12	Elaborar el método determinístico																				
13	Obtener la probabilidad del proyecto																				
14	Realizar pruebas del sistema																				
15	Diseñar Manual de Usuario																				
16	Crear el Manual Técnico																				
17	Elaborar el artículo académico.																				
18	Presentación del proyecto																				

