

Desarrollo de una página web y un sistema de
apoyo a la gestión de las Arquidiócesis, Diócesis,
Vicariatos y Congregaciones asociadas a
Inpreclero.

-Apéndices-

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR:

Edward Georges Jebian Bastouri

TUTOR EMPRESARIAL:

Dilcia Delgado

TUTOR ACADÉMICO:

Lúcia Cardoso

FECHA:

08 de junio de 2012

Apéndice 1
(M-E-R)

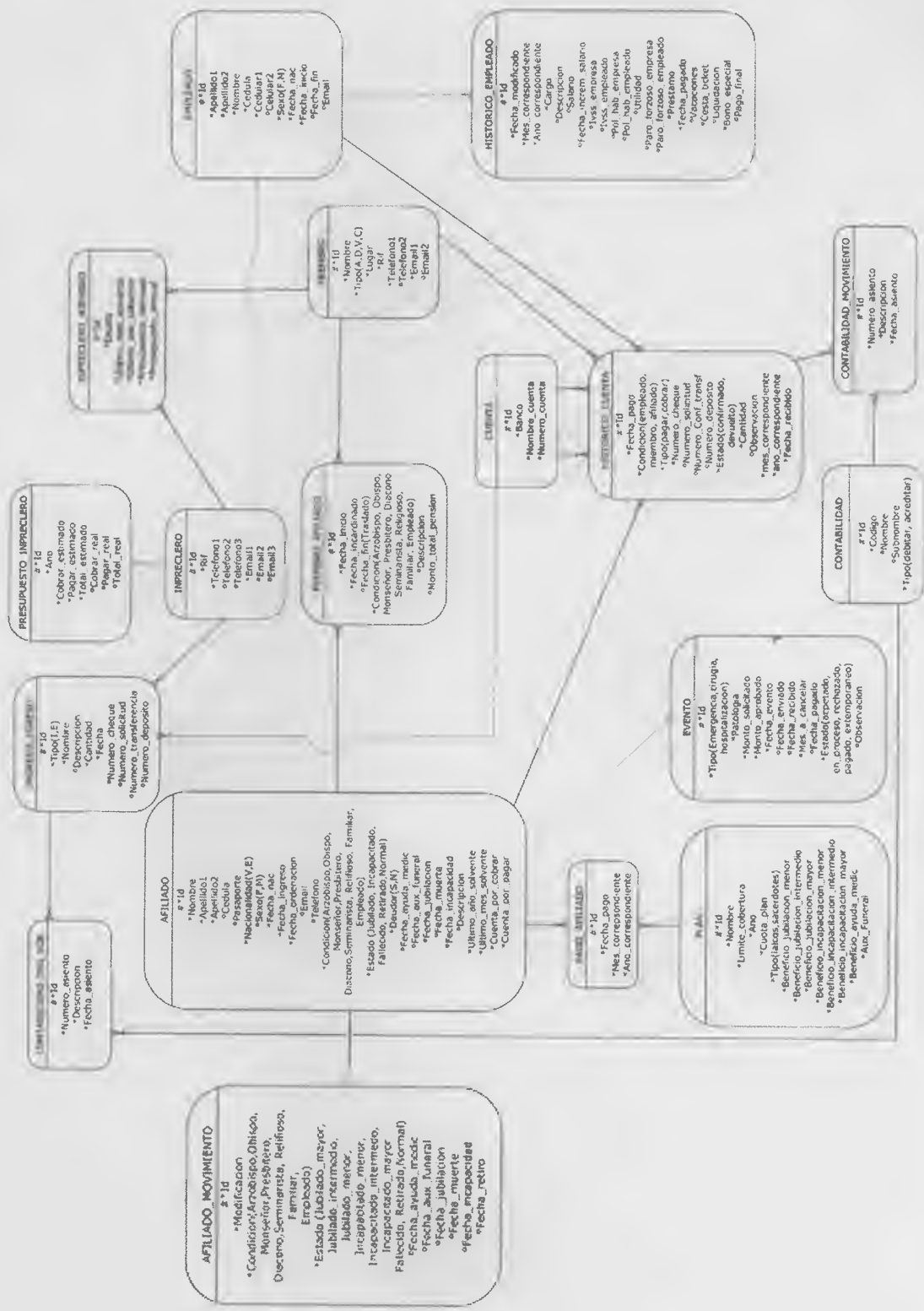
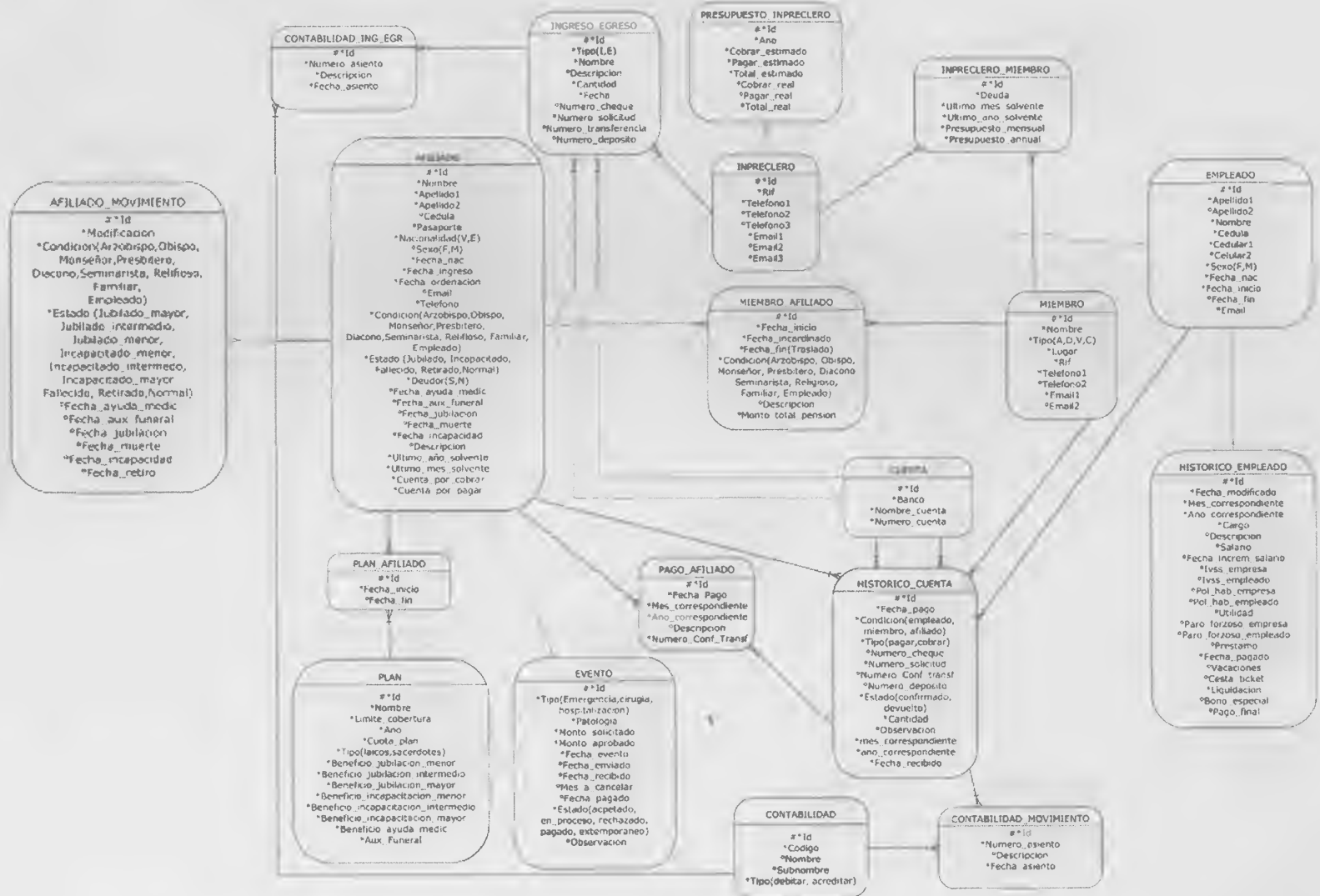


Figura 86. M-E-R (V1)

(Elaborado por Edward Jebian)

(Elaborado por Edward Jebian)

Figura 87. M-E-R (V2)



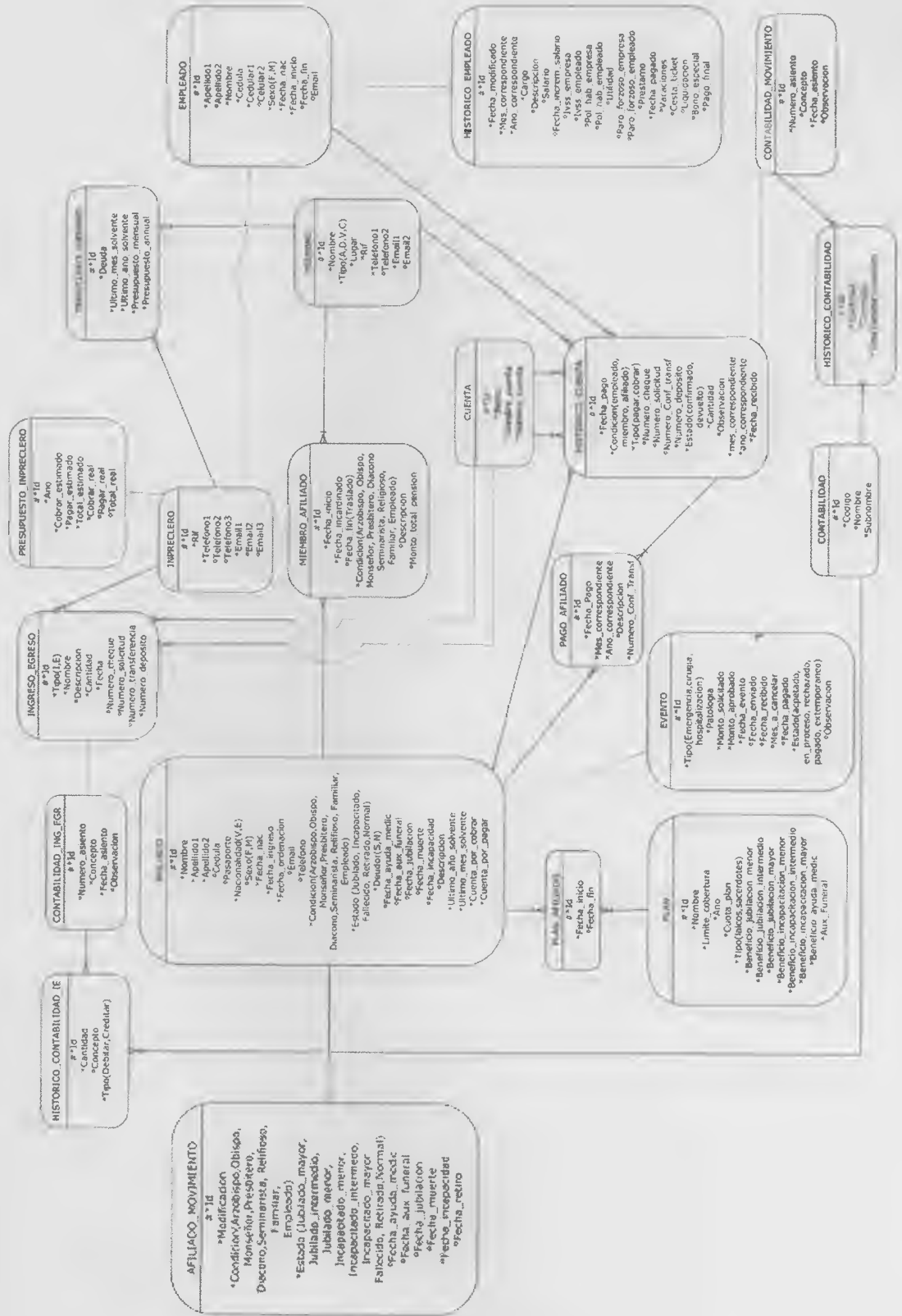


Figura 88. M-E-R (V3)

(Elaborado por Edward Jebian)

Apéndice 2
(Descripción de las tablas de la BD)

Descripción de las Tablas de la Base de Datos

Tabla: Ley

Se utiliza para manejar ciertos datos legales requeridos para las operaciones de empleados.

Id: INT, clave primaria de la tabla.

Ano_correspondiente INT, año correspondiente de la ley.

Fecha DATE, fecha en que se creó la ley.

Ivss_empresa DECIMAL(10,2), porcentaje de ivss de la empresa.

Ivss_empleado DECIMAL(10,2), porcentaje de ivss del empleado.

Politica_hab_empresa DECIMAL(10,2), porcentaje de la ley de política habitacional de la empresa.

Politica_hab_empleado DECIMAL(10,2), porcentaje de la ley de política habitacional del empleado.

Paro_forzoso_empresa DECIMAL(10,2), porcentaje de paro forzoso de la empresa.

Paro_forzoso_empleado DECIMAL(10,2), porcentaje de paro forzoso del empleado.

Unidad_tributaria DECIMAL(10,2), valor de la unidad tributaria.

Porcentaje_ut DECIMAL(10,2), porcentaje de la unidad tributaria que usa la empresa.

Dias_cesta_ticket INT, cantidad de días de cesta ticket que paga la empresa.

Fk_ley_inpreclero INT, clave foránea que hace referencia a Inpreclero.

TABLA afiliado_movimiento

Se utiliza para guardar historico de status de los afiliados a la hora de que cambió estado, condición o alguna fecha.

Id INT, clave primaria de la tabla.

Modificacion VARCHAR(50), nombre del atributo a modificar (ej.: estado, condición).

Condicion VARCHAR(30), condición del afiliado (ej.: arzobispo, obispo...).

Estado VARCHAR(20), estado del afiliado (ej.: en vida, retirado...).

Fecha_ayuda_medic DATE, fecha en que empieza la ayuda médica.

Fecha_aux_funeral DATE, fecha del auxilio funeral.

Fecha_jubilacion DATE, fecha de jubilación.

Fecha_muerte DATE, fecha de muerte.

Fecha_incapacidad DATE, fecha de incapacidad.

Fecha_retiro DATE, fecha de retiro.

Fk_afiliado_movimiento_afiliado INT, clave foránea que hace referencia a un afiliado.

TABLA control

Se utiliza para almacenar los datos de los usuarios que ingresen mediante un login a la página web.

Id clave primaria de la tabla,

Ip VARCHAR(30) NOT NULL, ip de la computadora desde donde se inicio la sesión.

Browser VARCHAR(30) , navegador desde el cual se accedió a la **PW**.

Version_browser VARCHAR(30) , versión del navegador desde el cual se accedió a la **PW**.

OS VARCHAR(30), nombre del sistema operativo de la computadora desde la cual se accedió a la **PW**.

Datecontrol DATE NOT NULL, fecha en que accedió el usuario a la **PW**.

Fk_control_afiliado, clave foránea que hace referencia a un afiliado.

Fk_control_miembro INT, clave foránea que hace referencia a un miembro.

Fk_control_admin INT, clave foránea que hace referencia a un administrador.

CREATE TABLA dinero

Se utiliza para poder llevar el control de dinero que hay en cada una e las cuentas de Inpreclero.

Id INT, clave primaria de la tabla.

Cantidad DECIMAL(10,2) NOT NULL, cantidad de dinero que se reporta existe en la cuenta pertinente.

Descripcion VARCHAR(100), descripción de este dinero.

Fecha DATE NOT NULL, fecha correspondiente al ingreso del dato dinero.

Fk_dinero_inpreclero INT, clave foránea que hace referencia a Inpreclero.

Fk_dinero_cuenta INT, clave foránea que hace referencia a una cuenta.

CREATE TABLA noticia

Se utiliza para almacenar las noticias que ingrese el administrador y que se verán en la página web.

Id INT, clave primaria de la tabla.

Fecha DATE NOT NULL, fecha en que se publica la noticia.

Nombre VARCHAR(40) NOT NULL, nombre de la noticia.

Asunto VARCHAR(100), asunto de la noticia.

Descripcion VARCHAR(500) NOT NULL, descripción de la noticia.

Imagen VARCHAR(60) NOT NULL, nombre con formato de la imagen de la noticia.

Fk_noticia_administrador INT, clave foránea que hace referencia a un administrador.

CREATE TABLA administrador

Se utiliza para almacenar los datos de los administradores.

Id INT, clave primaria de la tabla.

Nombre VARCHAR(30) NOT NULL, nombre del administrador con el que iniciara sesión.

Apellido1 VARCHAR(30) NOT NULL, primer apellido del administrador.

Apellido2 VARCHAR(30), Segundo apellido del administrador.

Clave INT NOT NULL, clave con que iniciará sesión el administrador.

Fecha_nac DATE, fecha de nacimiento del administrador.

Cedula INT NOT NULL, cédula del administrador.

CREATE TABLA presupuesto_inpreclero

Se utiliza para almacenar los presupuestos que realice Inpreclero.

Id INT, clave primaria de la tabla.

Ano INT NOT NULL, año en que se desea realizar el presupuesto.

Monto_cobrar_sacerdotes DOUBLE, monto total a cobrar a los sacerdotes

Monto_cobrar_laicos DOUBLE, monto a cobrar a los laicos.

Monto_cobrar_total DOUBLE NOT NULL, monto a cobrar total por todos.

Monto_pagar_jubilados DOUBLE, monto a pagar a los jubilados.

Monto_pagar_incapacitados DOUBLE, monto a pagar a los incapacitados.

Monto_pagar_ayuda_medica DOUBLE, monto a pagar por concepto de ayuda médica.

Monto_pagar_total DOUBLE NOT NULL, monto total a pagar.

Monto_total DOUBLE NOT NULL, monto total a ingresar después de cobrar y pagar.

Afiliados_cobrar_sacerdotes INT, número de afiliados sacerdotes a los que se les cobra cuota.

Afiliados_cobrar_laicos INT, número de afiliados laicos a los que se les cobra cuota.

Afiliados_cobrar_total INT NOT NULL, número total de afiliados que pagan cuota.

Afiliados_pagar_jubilados INT, número total de afiliados a los que se le paga beneficio de jubilación.

Afiliados_pagar_incapacitados INT, número total de afiliados a los que se le paga beneficio de incapacitación.

Afiliados_pagar_ayuda INT, número total de afiliados a los que se les paga beneficio de ayuda médica.

Afiliados_pagar_total INT NOT NULL, número total de afiliados a los que se les debe pagar.

Fecha DATE NOT NULL, fecha en que se realiza el presupuesto.

Beneficiario VARCHAR(30) NOT NULL, nombre al que se le hará el presupuesto.

Tipo VARCHAR(30), tipo de presupuesto (real recibido, real correspondiente o presupuesto).

Fk_presupuesto_inpreclero INT, clave foránea que hace referencia a Inpreclero.

CREATE TABLA historico_contabilidad_ie

Se utiliza para guardar los datos de cada concepto dentro de un asiento de ingreso/egreso.

Id INT, clave primaria de la tabla.

Cantidad INT NOT NULL, monto del concepto.

Concepto VARCHAR(80), nombre del concepto.

Tipo VARCHAR(20), tipo del concepto (debe, haber).

Fk_hisconie_conie INT, clave foránea que hace referencia a un asiento de contabilidad_ie.

Fk_hisconie_con INT, clave foránea que hace referencia a una cuenta de contabilidad.

CREATE TABLA contabilidad_ing_egr

Se utiliza para guardar los asientos de ingreso/egreso.

Id INT, clave primaria de la tabla.

Concepto VARCHAR(80), nombre del asiento.

Fecha_asiento DATE NOT NULL, fecha del asiento.

Observacion VARCHAR (150), observación del asiento.

Fk_contaie_ingegr INT, clave foránea que hace referencia a un ingegr.

CREATE TABLA ingreso_egreso

Se utiliza para guardar los ingreso/egreso.

Id INT, clave primaria de la tabla.

Tipo VARCHAR(10) NOT NULL, tipo (ingreso/egreso).

Nombre VARCHAR(30) NOT NULL, nombre del ingreso/egreso.

Descripcion VARCHAR(150), descripción del ingreso/egreso.

Descuento_abono DECIMAL(10,2), saldo del del ingreso/egreso.

Cantidad DECIMAL(10,2) NOT NULL, monto del ingreso/egreso.

Fecha DATE NOT NULL, fecha del ingreso/egreso.

Numero_cheque VARCHAR(30), número del cheque del del ingreso/egreso.

Numero_solicitud VARCHAR(30), número de solicitud del ingreso/egreso.

Numero_transferencia VARCHAR(30), número de transferencia del ingreso/egreso.

Numero_deposito VARCHAR(30), número de deposito del ingreso/egreso.

Salario_final DECIMAL(10,2), salario final con todas las adiciones y descuentos por los concepto mencionados anteriormente.

Fk_histemp_empleado INT, clave foránea que hace referencia a un empleado.

CREATE TABLA historico_contabilidad

Se utiliza para guardar los datos de los conceptos de los asientos de los histórico_cuenta.

Id INT, clave primaria de la tabla.

Cantidad INT NOT NULL, monto del concepto.

Concepto VARCHAR(80), nombre del concepto.

Tipo VARCHAR(20), tipo del concepto (debe, haber).

Fk_hiscon_conmov INT, clave foránea que hace referencia a un asiento de contabilidad_movimiento.

Fk_hiscon_con INT, clave foránea que hace referencia a una cuenta de contabilidad.

CREATE TABLA contabilidad_movimiento.

Se utiliza para guardar los datos de los asientos de los histórico_cuenta.

Id INT, clave primaria de la tabla.

Concepto VARCHAR(80), concepto del asiento.

Fecha_asiento DATE NOT NULL, fecha del asiento.

Observacion VARCHAR(100), observación del asiento.

Fk_contamov_hiscta INT, clave foránea que hace referencia a una transacción histórico_cuenta.

CREATE TABLA pago_afiliado

Se utiliza para guardar los datos de los pagos de los afiliados.

Id INT, clave primaria de la tabla.

Mes_correspondiente VARCHAR(20) NOT NULL, mes correspondiente al pago del afiliado.

Ano_correspondiente INT NOT NULL, año correspondiente al pago del afiliado.

Descripcion VARCHAR(150), descripción del pago del afiliado.

Monto INT, monto del pago del afiliado.

Numero_conf_transf VARCHAR(30) NOT NULL, número de confirmación asociado al pago del afiliado.

Fk_pagafil_hiscta INT, clave foránea que hace referencia a una transacción histórico_cuenta.

Fk_pagafil_afiliado INT, clave foránea que hace referencia a un afiliado.

CREATE TABLA historico_cuenta

Se utiliza para guardar los datos de las transacciones (histórico_cuenta)

Id INT, clave primaria de la tabla.

Fecha_pago DATE NOT NULL, fecha de la transacción.

Condicion VARCHAR(30) NOT NULL, condición de la transacción (miembro, afiliado o empleado).

Tipo VARCHAR(10) NOT NULL, tipo de la transacción (pagar, cobrar).

Numero_cheque VARCHAR(30), número de cheque de la transacción.

Numero_solicitud VARCHAR(30), número de solicitud de la transacción.

Numero_conf_transf VARCHAR(30), número de confirmación de la transacción.

Numero_deposito VARCHAR(30), número de depósito de la transacción.

Estado VARCHAR(15) NOT NULL, estado de la transacción (confirmado o devuelto).

Cantidad DECIMAL(10,2) NOT NULL, monto de la transacción.

Observacion VARCHAR(200), observación de la transacción.

Descuento_abono DECIMAL(10,2), saldo del pagador después de la transacción.

Mes_correspondiente VARCHAR(15), mes correspondiente de la transacción.

Ano_correspondiente INT NOT NULL, año correspondiente de la transacción.

Fk_hiscta_afiliado INT, clave foránea que hace referencia a un afiliado.

Fk_hiscta_empleado INT, clave foránea que hace referencia a un empleado.

Fk_hiscta_miembro INT, clave foránea que hace referencia a un miembro.

Fk_hiscta_cuentadebitar INT, clave foránea que hace referencia a una cuenta.

Fk_hiscta_cuentacreditar INT, clave foránea que hace referencia a una cuenta.

Fecha_recibido DATE, fecha recibida de la transacción.

CREATE TABLA empleado

Se utiliza para guardar los datos de los empleados.

Id INT, clave primaria de la tabla.

Apellido1 VARCHAR(50) NOT NULL, primer apellido del empleado.

Apellido2 VARCHAR(50), segundo apellido del empleado.

Nombre VARCHAR(50) NOT NULL, nombre del empleado.

Tipo VARCHAR (20) NOT NULL, tipo de empleado (honorarios profesionales o fijo).

Cedula VARCHAR(20) NOT NULL UNIQUE, número de cédula del empleado.
Celular1 VARCHAR(20) NOT NULL, celular/teléfono # 1 del empleado.
Celular2 VARCHAR(20), celular/teléfono # 2 del empleado.
Sexo VARCHAR(2) NOT NULL, sexo del empleado.
Fecha_nac DATE NOT NULL, fecha de nacimiento del empleado.
Fecha_inicio DATE NOT NULL, fecha de inicio en Inpreclero del empleado.
Fecha_fin DATE, fecha de salida de trabajo de Inpreclero.
Email VARCHAR(30), email del empleado.
Fk_emp_inpreclero INT, clave foránea que hace referencia a Inpreclero.
CREATE TABLA cuenta

Se utiliza para guardar los datos de las cuentas bancarias.

Id INT, clave primaria de la tabla.
Banco VARCHAR(30) NOT NULL, nombre del banco perteneciente a la cuenta.
Nombre_cuenta VARCHAR(30) NOT NULL, nombre del titular de la cuenta.
Numero_cuenta VARCHAR(30) NOT NULL UNIQUE, número de la cuenta.

CREATE TABLA contabilidad

Se utiliza para guardar los datos de las cuentas de contabilidad.

Id INT, clave primaria de la tabla.
Codigo VARCHAR(80) NOT NULL, código de la cuenta contable.
Nombre VARCHAR(100), nombre de la cuenta contable.

CREATE TABLA evento

Se utiliza para guardar los datos de los eventos de los afiliados.

Id INT, clave primaria de la tabla.
Tipo VARCHAR(20) NOT NULL, tipo de evento (emergencia, cirugía y hospitalización).
Patologia VARCHAR(40) NOT NULL, patología del evento.
Numero_conf_transf VARCHAR(20) NOT NULL, número de confirmación del evento.
Monto_solicitado DECIMAL(10,2) NOT NULL, monto solicitado por el afiliado.
Monto_aprobado DECIMAL(10,2) NOT NULL, monto aprobado por Inpreclero.
Fecha_evento DATE NOT NULL, fecha del evento.
Fecha_enviado DATE, fecha en que se envió el evento a Inpreclero.
Fecha_recibido DATE, fecha en que Inpreclero recibió la solicitud de reintegro del evento.
Mes_a_pagar VARCHAR(15), mes a realizar el reintegro del evento.

Fecha_pagado DATE, fecha en que se pago el reintegro del evento.
Estado VARCHAR(20) NOT NULL, estado del reintegro del evento (aceptado, en proceso, rechazado, pagado y extemporaneo).
Observacion VARCHAR(200), observación del evento.
Año_correspondiente INT NOT NULL, año correspondiente del evento.
Fk_evento_afiliado INT, clave foránea que hace referencia a un afiliado.

CREATE TABLA plan

Se utiliza para guardar los datos de los planes.

Id INT, clave primaria de la tabla.
Nombre VARCHAR(80) NOT NULL, nombre del plan.
Limite_cobertura DOUBLE NOT NULL, límite de cobertura del plan.
Año INT NOT NULL, año del plan.
Cuota_plan DOUBLE NOT NULL, cuota del plan.
Tipo VARCHAR(15) NOT NULL, tipo de plan (sacerdotes o laicos).
Beneficio_jubilacion_menor DECIMAL(10,2), monto correspondiente al beneficio de jubilación menor.
Beneficio_jubilacion_intermedio DECIMAL(10,2), monto correspondiente al beneficio de jubilación intermedio.
Beneficio_jubilacion_mayor DECIMAL(10,2), monto correspondiente al beneficio de jubilación mayor.
Beneficio_incapacitacion_menor DECIMAL(10,2), monto correspondiente al beneficio de incapacidad menor.
Beneficio_incapacitacion_intermedio DECIMAL(10,2), monto correspondiente al beneficio de incapacidad intermedio.
Beneficio_incapacitacion_mayor DECIMAL(10,2), monto correspondiente al beneficio de incapacidad mayor.
Beneficio_ayuda_medic DECIMAL(10,2), monto correspondiente al beneficio de ayuda médica.
Aux_funeral DECIMAL(10,2), monto correspondiente al beneficio de auxilio funeral.
Fk_plan_inpreclero INT, clave foránea que hace referencia a inpreclero.

CREATE TABLA miembro_afiliado

Se utiliza para guardar las relaciones entre miembro_afiliado.

Id INT, clave primaria de la tabla.
Fecha_incardinado DATE NOT NULL, fecha de ingreso del afiliado al miembro.

Fecha_fin DATE, fecha en que un afiliado deja de pertenecer a un miembro.
Condicion VARCHAR(20) NOT NULL, condición del afiliado (arzobispo, obispo, monseñor, presbítero, diacono, seminarista, religioso, familiar, empleado, otro).
Descripcion VARCHAR(150), descripción de la asociación.
Fk_miemafil_miembro INT, clave foránea que hace referencia a un miembro.
Fk_miemafil_afiliado INT, clave foránea que hace referencia a un afiliado.

CREATE TABLA afiliado

Se utiliza para guardar los datos de los afiliados.

Id INT, clave primaria de la tabla.
Nombre VARCHAR(30) NOT NULL, nombre del afiliado.
Apellido1 VARCHAR(30) NOT NULL, primer apellido del afiliado,
Apellido2 VARCHAR(30), segundo apellido del afiliad.
Nombre_miembro VARCHAR(50), nombre del miembro al que pertenece el afiliado.
Cedula VARCHAR(20), cédula del afiliado.
Pasaporte VARCHAR(30), pasaporte del afiliado.
Nacionalidad VARCHAR(2) NOT NULL, nacionalidad del afiliado.
Sexo VARCHAR(2) NOT NULL, sexo del afiliado.
Fecha_nac DATE, fecha de nacimiento del afiliado.
Fecha_ingreso DATE , fecha de ingreso a Inpreclero del afiliado.
Fecha_ordenacion DATE, fecha de ordenación del afiliado.
Email VARCHAR(30), email del afiliado.
Telefono VARCHAR(20), teléfono del afiliado.
Condicion VARCHAR(30) NOT NULL, condición del afiliado (arzobispo, obispo, monseñor, presbítero, diacono, seminarista, religioso, familiar, empleado, otro).
Estado VARCHAR(20) NOT NULL, estado del afiliado (jubilado_menor, jubilado_intermedio, jubilado_mayor, incapacitado_menor, incapacitado_intermedio, incapacitado_mayor, fallecido, retirado, en vida).
Fecha_ayuda_medic DATE, fecha de inicio de ayuda médica del afiliado.
Fecha_aux_funeral DATE, fecha de auxilio funeral del afiliado.
Fecha_jubilacion DATE, fecha de jubilación del afiliado.
Fecha_muerte DATE, fecha de fallecido del afiliado.
Fecha_incapacidad DATE, fecha de inicio de incapacidad del afiliado.
Fecha_retiro DATE, fecha de retiro del afiliado.
Descripcion VARCHAR(200), descripción del afiliado.
Ultimo_ano_solvente INT NOT NULL, último año solvente del afiliado.
Ultimo_mes_solvente VARCHAR(20) NOT NULL, último mes solvente del afiliado.

CREATE TABLA inpreclero

Se utiliza para guardar los datos de Inpreclero.

Id INT, clave primaria de la tabla.
Rif VARCHAR(15) NOT NULL UNIQUE, rif de Inpreclero.
Telefono1 VARCHAR(30) NOT NULL, teléfono # 1 de Inpreclero.
Telefono2 VARCHAR(30) , teléfono # 2 de Inpreclero.
Telefono3 VARCHAR(30), teléfono # 3 de Inpreclero.
Email1 VARCHAR(30) NOT NULL, email # 1 de Inpreclero.
Email2 VARCHAR(30), email # 2 de Inpreclero.
Email3 VARCHAR(30), email # 3 de Inpreclero.

CREATE TABLA miembro

Se utiliza para guardar los datos de los miembros.

Id INT, clave primaria de la tabla.
Nombre VARCHAR(50) NOT NULL, nombre del miembro.
Tipo VARCHAR(20) NOT NULL, tipo de miembro (arquidiócesis, diócesis, vicariato, congregación).
Lugar VARCHAR(200) NOT NULL, dirección del miembro.
Rif VARCHAR(20) NOT NULL UNIQUE, rif del miembro.
Telefono1 VARCHAR(20) NOT NULL, teléfono # 1 del miembro.
Telefono2 VARCHAR(20), teléfono # 2 del miembro.
Celular VARCHAR(20), celular del miembro.
Email1 VARCHAR(30) NOT NULL, email # 1 del miembro.
Email2 VARCHAR(30), email # 2 del miembro.
Ultimo_ano_solvente INT, último año solvente del miembro.
Ultimo_mes_solvente VARCHAR(20), último mes solvente del miembro.
Obispo_encargado VARCHAR(60), obispo encargado del miembro.
Nombre_encargado VARCHAR(30), nombre del encargado del miembro.
Celular_encargado VARCHAR(30), celular del encargado del miembro.
Email_encargado VARCHAR(30), email del encargado del miembro.

✓ **Función: agregar_mod**

```
function agregar_mov()
{
    $id=$_GET['ID'];

    $current_empleado=$this->empleado_m->get_empleado_by_id($id);

    $current_mov=$this->empleado_m->get_max_mov($id);
    $max=$current_mov[0]['max'];

    $movimiento=$this->empleado_m->get_historico_empleado_by_id($max);

    if ($movimiento)
    {
        $current_empleado[0]['prestamo_anterior']=$movimiento[0]['Prestamo'];

        if ($movimiento[0]['Prestamo']==0)
        {
            $current_empleado[0]['cuota_anterior']=0;
        }

        else
        {

        $current_empleado[0]['cuota_anterior']=$movimiento[0]['Cuota_mensual'];
        }
        $current_empleado[0]['salario_anterior']=$movimiento[0]['Salario'];
        $current_empleado[0]['cargo_anterior']=$movimiento[0]['Cargo'];
        $current_empleado[0]['tipo_anterior']=$movimiento[0]['Tipo'];
    }

    else
    {
        $current_empleado[0]['prestamo_anterior']=0;
        $current_empleado[0]['cuota_anterior']=0;
        $current_empleado[0]['salario_anterior']=0;
        $current_empleado[0]['cargo_anterior']='';
        $current_empleado[0]['tipo_anterior']='Quincena1';
    }

    $data= array('perso' => $current_empleado);

    $this->load->view("empleado/agregar_movimiento",$data);
}
```

✓ **Función: movimiento_agregar**

```
function movimiento_agregar()
{
    $Id_empleado = $_POST["Id_empleado"];
    $Mes_correspondiente = $_POST["Mes_correspondiente"];
    $Año_correspondiente = $_POST["Año_correspondiente"];
    $Cargo = $_POST["Cargo"];
    $Descripcion = $_POST["Descripcion"];
    $Salario = $_POST["Salario"];
    $Fecha_increm_sueldo = $_POST["Fecha_increm_salario"];
    $Prestamo = $_POST["Prestamo"];
    $Fecha_pagado = $_POST["Fecha_pagado"];
    $Vacaciones = $_POST["Vacaciones"];
    $Liquidacion = $_POST["Liquidacion"];
    $Bono_especial = $_POST["Bono_especial"];
    $Pago_final = $_POST["Pago_final"];

    //nuevos agregados
    $Cuota_prestamo = $_POST["Cuota_prestamo"];
    $Abono = $_POST["Abono"];
    $Cesta_ticket = $_POST["Cesta_ticket"];
    $Tipo = $_POST["Tipo"];
    $Utilidad = $_POST["Utilidad"];

    if($Mes_correspondiente=='Seleccione')
    {
        $this->form_validation->set_rules('Mes_correspondiente','Mes_correspondiente',
'callback_mescorrespondiente_check');
    }

    $this->form_validation->set_rules('Descripcion','Descripcion');

    $this->form_validation->set_rules('Salario','Salario','required');
    $this->form_validation->set_rules('Fecha_pagado','Fecha pagado','required');
    $this->form_validation->set_rules('Fecha_increm_sueldo','Fecha Increm. Sueldo');

    if ($this->form_validation->run() == FALSE)
    {
        $current_empleado=$this->empleado_m->get_empleado_by_id($Id_empleado);

        $current_mov=$this->empleado_m->get_max_mov($Id_empleado);
        $max=$current_mov[0]['max'];

        $movimiento=$this->empleado_m->get_historico_empleado_by_id($max);

        if ($movimiento)
        {
            $current_empleado[0]['prestamo_anterior']=$movimiento[0]['Prestamo'];

            if ($movimiento[0]['Prestamo']==0)
            {
                $current_empleado[0]['cuota_anterior']=0;
            }
            else
        }
    }
}
```



```

if (($Prestamo!=0)&&($Cuota_prestamo!=0))
{
    if ($Abono=='Si')
    {
        $Prestamo=$Prestamo-$Cuota_prestamo;
        $Salario_final=$Salario_final-$Cuota_prestamo;
    }
}
}

```

```
$Salario_final=$Salario_final+$Bono_especial+$Liquidacion+$Pago_final+$Vacaciones;
```

```
$datestring = "%Y-%m-%d";
```

```
$time = time();
```

```
$actualday = mdate($datestring, $time);
```

```
$new = array('ID' => "",
```

```

'Fecha_modificado' => $actualday,
'Mes_correspondiente' => $Mes_correspondiente,
'Año_correspondiente' => $Año_correspondiente,
'Cargo' => $Cargo,
'Tipo' => $Tipo,
'Descripcion' => $Descripcion,
'Salario' => $Salario,
'Fecha_increm_salario' => $Fecha_increm_sueldo,
'lvss_empresa' => $lvss_empresa,
'lvss_empleado' => $lvss_empleado,
'Pol_hab_empresa' => $pol_hab_empresa,
'Pol_hab_empleado' => $pol_hab_empleado,
'Utilidad' => $Utilidad,
'Paro_forzoso_empresa' => $paro_empresa,
'Paro_forzoso_empleado' => $paro_empleado,
'Prestamo' => $Prestamo,
'Cuota_mensual' => $Cuota_prestamo,
'Fecha_pagado' => $Fecha_pagado,
'Vacaciones' => $Vacaciones,
'Cesta_ticket' => $Cesta_ticket,
'Liquidacion' => $Liquidacion,
'Bono_especial' => $Bono_especial,
'Pago_final' => $Pago_final,
'Salario_final' => $Salario_final,
'fk_histemp_empleado' => $Id_empleado);

```

```
$this->empleado_m->insert_mov($new); //Llamando a la funcion que agrega un nuevo usuario.
```

```
$this->load->view("exito");
```

```
$this->load->view("inicio");
```

✓ Función: transaccion_agregar

```
function transaccion_agregar()
{
    $Id_miembro = $_POST["Id_miembro"];
    $Mes_correspondiente = $_POST["Mes_correspondiente"];
    $Año_correspondiente = $_POST["Año_correspondiente"];
    $Fecha_pago = $_POST["Fecha_pago"];
    $Tipo = $_POST["Tipo"];
    $Numero_cheque = $_POST["Numero_cheque"];
    $Numero_solicitud = $_POST["Numero_solicitud"];
    $Numero_conf_trans = $_POST["Numero_conf_trans"];
    $Numero_deposito = $_POST["Numero_deposito"];
    $Estado = $_POST["Estado"];

    if ($Tipo=='Cobrar')
    {
        $Meses_pagar = $_POST["Meses_pagar"];
        $Pago = $_POST["Pago"];
    }

    $Cantidad = $_POST["Cantidad"];
    $Observacion = $_POST["Observacion"];
    $Descuento_abono = $_POST["Descuento_abono"];
    $Fecha_recibido = $_POST["Fecha_recibido"];
    $Fk_hiscta_cuentadebitar = $_POST["Fk_hiscta_cuentadebitar"];
    $Fk_hiscta_cuentacreditar = $_POST["Fk_hiscta_cuentacreditar"];

    $ano=$_POST["ano"];
    $mes=$_POST["mes"];
    $abono=$_POST["abono"];

    $ano_pagar=$_POST["ano_pagar"];
    $mes_pagar=$_POST["mes_pagar"];

    //este if es para que no haya a la vez en una transaccion dos tipo de pagos

    if ($Tipo=='Cobrar')
    {
        if (!$Numero_conf_trans)
        {
            if ($Numero_deposito)
            {
                $Numero_conf_trans=$Numero_deposito;
            }
            else
            {
                if ($Numero_solicitud)
                {
                    $Numero_conf_trans=$Numero_solicitud;
                }
                else
                {
                    $this->form_validation->set_rules('Numero_cheque','Numero_cheque',
'callback_numerocheque_check');
```

```

    }
}

if ($Tipo=='Pagar')
{

    if (!$Numero_conf_trans)
    {
        if ($Numero_cheque)
        {
            $Numero_conf_trans=$Numero_cheque;
        }
        else
        {
            if ($Numero_solicitud)
            {
                $Numero_conf_trans=$Numero_solicitud;
            }
            else
            {
                $this->form_validation-
>set_rules('Numero_cheque','Numero_cheque','callback_numerocheque_check');
            }
        }
    }

    $this->form_validation->set_error_delimiters('<div class="error" align="center" style="color:#F00">','</div>');
    $this->form_validation->set_rules('Mes_correspondiente','Mes correspondiente','required');
    $this->form_validation->set_rules('Año_correspondiente','Año correspondiente','required');
    $this->form_validation->set_rules('Fecha_pago','Fecha de pago','required');
    $this->form_validation->set_rules('Cantidad','Cantidad','required|numeric');

    $this->form_validation->set_rules('Observacion','Observacion');
    $this->form_validation->set_rules('Fecha_recibido','Fecha recibido','required');
    $this->form_validation->set_rules('Numero_solicitud','Numero solicitud');
    $this->form_validation->set_rules('Numero_deposito','Numero deposito');
    $this->form_validation->set_rules('Descuento_abono','Descuento abono');

    if($Mes_correspondiente=='Seleccione')
    {
        $this->form_validation->set_rules('Mes_correspondiente','Mes correspondiente',
'callback_mescorrespondiente_check');
    }

    if($Tipo=='Seleccione')
    {
        $this->form_validation->set_rules('Tipo','Tipo','callback_tipo_check');
    }

    if($Estado=='Seleccione')
    {
        $this->form_validation->set_rules('Estado','Estado','callback_estado_check');
    }

    if (($Tipo=='Cobrar')&& ($Fk_hiscta_cuentacreditar=='Seleccione'))
    {

```

```

        $this->form_validation->set_rules('Fk_hiscta_cuentacreditar','Cuentaacreditar',
callback_cuentacreditar_check');
    }

    if (($Tipo=='Pagar')&& ($Fk_hiscta_cuentadebitar=='Seleccione'))
    {
        $this->form_validation->set_rules('Fk_hiscta_cuentadebitar',      'Cuenta      a      debitar',
'callback_cuentadeb_check');
    }

    if($Fk_hiscta_cuentadebitar=='Seleccione')
    {
        $Fk_hiscta_cuentadebitar=NULL;
    }

    if($Fk_hiscta_cuentacreditar=='Seleccione')
    {
        $Fk_hiscta_cuentacreditar=NULL;
    }

    if ($this->form_validation->run() == FALSE)
    {
        $cuentas=$this->cuenta_m->get_all_cuenta();

        $cuentas[0]['id_miembro']= $Id_miembro;

        $cuentas[0]['ano']= $ano;
        $cuentas[0]['mes']= $mes;
        $cuentas[0]['abono']= $abono;

        $cuentas[0]['ano_pagar']= $ano_pagar;
        $cuentas[0]['mes_pagar']= $mes_pagar;

        $data= array('perso' => $cuentas);
        $this->load->view("miembro/agregar_transaccion",$data);
    }

    else
    {
        $datestring = "%Y";
$time = time();
        $actualyear=$Ano_correspondiente;

        $abono_anterior=$abono;
        $abono=0;
        $sacerdotes=0;
        $laicos=0;

        if ($Tipo=='Cobrar')
        {
            $miembro= $this->miembro_m->get_miembro_by_id($Id_miembro);

            $afil= $this->afiliado_m->get_all_afiliado_by_miembro_name($miembro[0]['Nombre']);

            for ($i=0;$i<count($afil); $i++)

```

```

        if (($afil[$i]['Condicion']=='Arzobispo') ||
($afil[$i]['Condicion']=='Obispo') || ($afil[$i]['Condicion']=='Monseñor') || ($afil[$i]['Condicion']=='Presbitero'))
        {
            if ($afil[$i]['Estado']=='En vida')
            {
                $plan=$this->plan_m->get_plan_actual_sacerdotes($actualyear);

                if ($plan)
                {
                    $abono=$abono+$plan[0]['Cuota_plan'];
                    $sacerdotes=$sacerdotes+1;
                }
            }
        }
        else// si es diacono, seminarista, religioso, familiar, empleado u otro
        {
            if ($afil[$i]['Estado']=='En vida')
            {
                $plan=$this->plan_m->get_plan_actual_laicos($actualyear);

                if ($plan)
                {
                    $abono=$abono+ $plan[0]['Cuota_plan'];
                    $laicos=$laicos+1;
                }
            }
        }
    }
} //cierra el for

//cierra el tipo cobrar

if ($Tipo=='Cobrar')
{
    $abono=$abono*$Meses_pagar;
}

if ($Descuento_abono==0)
{
    if ($abono!=0)
    {
        $Descuento_abono=$Cantidad+$abono_anterior-$abono;
    }
}

$new = array('ID' => "",
            'Fecha_pago' => $Fecha_pago,
            'Condicion' => 'Miembro',
            'Tipo' => $Tipo,
            'Numero_cheque' => $Numero_cheque,
            'Numero_solicitud' => $Numero_solicitud,
            'Numero_conf_transf' => $Numero_conf_trans,
            'Numero_deposito' => $Numero_deposito,
            'Estado' => $Estado,
            'Cantidad' => $Cantidad,
            'Observacion' => $Observacion,

```

```

'Descuento_abono' => $Descuento_abono,
'Mes_correspondiente' => $Mes_correspondiente,
'Ano_correspondiente' => $Ano_correspondiente,
'Fk_hiscta_afiliado' => NULL,
'Fk_hiscta_empleado' => NULL,
'Fk_hiscta_miembro' => $Id_miembro,
'Fk_hiscta_cuentadebitar' => $Fk_hiscta_cuentadebitar,
'Fk_hiscta_cuentacreditar' => $Fk_hiscta_cuentacreditar,
'Fecha_recibido' => $Fecha_recibido);

$this->miembro_m->insert_tran($new); //Llamando a la funcion que agrega un nuevo usuario.

$agregado = $this->db->insert_id();

//esto es para actualizar ultimo mes y ano solvente del miembro///

$this->miembro_m->update_miembro_status($Id_miembro,$Mes_correspondiente,$Ano_correspondiente);

////

$miem=$this->miembro_m->get_miembro_by_id($Id_miembro);

if ($Tipo=='Cobrar')
{
    if ($Pago=='Si')///aqui se crean todos los pagos de los afiliados de ese miembro
    {
        $afiliados=$this->afiliado_m->get_afiliados_miembro_envida($miem[0]['Nombre']);

        for ($i=0;$i<count($afiliados); $i++)
        {
            $new = array('ID' => "",
                $Mes_correspondiente,
                $Ano_correspondiente,
                'Descripcion' => "",
                'Numero_conf_transf' => $Numero_conf_trans,
                'Fk_pagafil_hiscta' => $agregado,
                'Fk_pagafil_afiliado' => $afiliados[$i]['Id']);

            $this->afiliado_m->insert_pago($new); //se agregar el pago del afiliado

            $this->afiliado_m->update_afiliado_pago($afiliados[$i]['Id'],
                $Mes_correspondiente, $Ano_correspondiente); //se actualiza su ultimo mes y ano solvente
        }
    } //cierra si pago es SI
}

if ($Tipo=='Cobrar')
{
    $hiscta=$this->miembro_m->get_hiscta_by_id($agregado);

    $hiscta[0]['Miembro']=$miem[0]['Nombre'];

    $cuenta=$this->miembro_m->get_cuenta_by_id($Fk_hiscta_cuentacreditar);

```



```

$hiscta[0]['Banco']=$cuenta[0]['Banco'];

$hiscta[0]['Sacerdotes']=$sacerdotes;
$hiscta[0]['Laicos']=$laicos;
$hiscta[0]['Pago']=$Pago;

$data= array('perso' => $hiscta);

$this->load->view("miembro/hiscta_asiento",$data);

}

if ($Tipo=='Pagar')
{
    $this->load->view('exito');
    $this->load->view('inicio');
}
}

```

✓ Función: afiliados_jubilacion

```

function afiliados_jubilacion()
{
    $Ano_correspondiente = $_POST["Ano_correspondiente"];

    $afiliados=$this->afiliado_m->get_afiliados_a_jubilar();

    $datestring = "%Y";
    $datestring2 = "%m";
    $time = time();

    $actualyear = mdate($datestring, $time);
    $actualmes = mdate($datestring2, $time);

    $cantidad=0;

    $afiliado= array();

    for ($i=0;$i<count($afiliados); $i++)
    {
        if
        ((($afiliados[$i]['Condicion']=='Arzobispo') || ($afiliados[$i]['Condicion']=='Monseñor') || ($afiliados[$i]['Condicion']=='Presbitero') || ($afiliados[$i]['Condicion']=='Obispo'))&&($afiliados[$i]['Nombre_miembro']!='FLOTANTE')&&($afiliados[$i]['Fecha_nac']!='0000-00-00')&&($afiliados[$i]['Fecha_ingreso']!='0000-00-00'))

        $ano_ingreso = substr($afiliados[$i]['Fecha_ingreso'], 0, 4);

        $ano_nacimiento = substr($afiliados[$i]['Fecha_nac'], 0, 4);
    }
}

```

```
$mes_nacimiento = substr($afiliados[$i]['Fecha_nac'], 5, 2);
```

```
$edad=$Ano_correspondiente-$ano_nacimiento;
```

```
$total=$Ano_correspondiente-$ano_ingreso;
```

```
$total_mes=$actualmes-$mes_nacimiento;
```

```
if ($total<10)
```

```
{
```

```
    //no califica a jubilado
```

```
}
```

```
else
```

```
{
```

```
    if ($total<20)
```

```
    {
```

```
        //es jubilado menor
```

```
        if ($edad>69)
```

```
        {
```

```
            if ($edad==70)
```

```
            {
```

```
                ||
```

```
                X
```

```
((($total_mes>=0) || ($actualyear!=$Ano_correspondiente))
```

```
$afiliado[$cantidad]['Id']=$afiliados[$i]['Id'];
```

```
$afiliado[$cantidad]['Nombre']=$afiliados[$i]['Nombre'];
```

```
$afiliado[$cantidad]['Apellido1']=$afiliados[$i]['Apellido1'];
```

```
$afiliado[$cantidad]['Apellido2']=$afiliados[$i]['Apellido2'];
```

```
$afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];
```

```
$afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];
```

```
$afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];
```

```
$afiliado[$cantidad]['Nacionalidad']=$afiliados[$i]['Nacionalidad'];
```

```
$afiliado[$cantidad]['Sexo']=$afiliados[$i]['Sexo'];
```

```
$afiliado[$cantidad]['Fecha_nac']=$afiliados[$i]['Fecha_nac'];
```

```
$afiliado[$cantidad]['Fecha_ingreso']=$afiliados[$i]['Fecha_ingreso'];
```

```
$afiliado[$cantidad]['Fecha_ordenacion']=$afiliados[$i]['Fecha_ordenacion'];
```

```
$afiliado[$cantidad]['Email']=$afiliados[$i]['Email'];
```

```
$afiliado[$cantidad]['Telefono']=$afiliados[$i]['Telefono'];
```

```
$afiliado[$cantidad]['Condicion']=$afiliados[$i]['Condicion'];
```

```
$afiliado[$cantidad]['Estado']=$afiliados[$i]['Estado'];
```

```

    $afiliado[$cantidad]['Fecha_ayuda_medic']=$afiliados[$i]['Fecha_ayuda_medic'];
    $afiliado[$cantidad]['Fecha_aux_funeral']=$afiliados[$i]['Fecha_aux_funeral'];
    $afiliado[$cantidad]['Fecha_jubilacion']=$afiliados[$i]['Fecha_jubilacion'];
    $afiliado[$cantidad]['Fecha_muerte']=$afiliados[$i]['Fecha_muerte'];
    $afiliado[$cantidad]['Fecha_incapacidad']=$afiliados[$i]['Fecha_incapacidad'];
    $afiliado[$cantidad]['Fecha_retiro']=$afiliados[$i]['Fecha_retiro'];
    $afiliado[$cantidad]['Descripcion']=$afiliados[$i]['Descripcion'];
    $afiliado[$cantidad]['Ultimo_ano_solvente']=$afiliados[$i]['Ultimo_ano_solvente'];
    $afiliado[$cantidad]['Ultimo_mes_solvente']=$afiliados[$i]['Ultimo_mes_solvente'];
    $cantidad=$cantidad+1;
}
else
{
}
}
else //si va a
{
}

jubilado

$afiliado[$cantidad]['Id']=$afiliados[$i]['Id'];
$afiliado[$cantidad]['Nombre']=$afiliados[$i]['Nombre'];
$afiliado[$cantidad]['Apellido1']=$afiliados[$i]['Apellido1'];
$afiliado[$cantidad]['Apellido2']=$afiliados[$i]['Apellido2'];
$afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];
$afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];
$afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];
$afiliado[$cantidad]['Nacionalidad']=$afiliados[$i]['Nacionalidad'];
$afiliado[$cantidad]['Sexo']=$afiliados[$i]['Sexo'];
$afiliado[$cantidad]['Fecha_nac']=$afiliados[$i]['Fecha_nac'];
$afiliado[$cantidad]['Fecha_ingreso']=$afiliados[$i]['Fecha_ingreso'];
$afiliado[$cantidad]['Fecha_ordenacion']=$afiliados[$i]['Fecha_ordenacion'];
$afiliado[$cantidad]['Email']=$afiliados[$i]['Email'];

```

```

    $afiliado[$cantidad]['Telefono']=$afiliados[$i]['Telefono'];

    $afiliado[$cantidad]['Condicion']=$afiliados[$i]['Condicion'];

    $afiliado[$cantidad]['Estado']=$afiliados[$i]['Estado'];

    $afiliado[$cantidad]['Fecha_ayuda_medic']=$afiliados[$i]['Fecha_ayuda_medic'];

    $afiliado[$cantidad]['Fecha_aux_funeral']=$afiliados[$i]['Fecha_aux_funeral'];

    $afiliado[$cantidad]['Fecha_jubilacion']=$afiliados[$i]['Fecha_jubilacion'];

    $afiliado[$cantidad]['Fecha_muerte']=$afiliados[$i]['Fecha_muerte'];

    $afiliado[$cantidad]['Fecha_incapacidad']=$afiliados[$i]['Fecha_incapacidad'];

    $afiliado[$cantidad]['Fecha_retiro']=$afiliados[$i]['Fecha_retiro'];

    $afiliado[$cantidad]['Descripcion']=$afiliados[$i]['Descripcion'];

    $afiliado[$cantidad]['Ultimo_ano_solvente']=$afiliados[$i]['Ultimo_ano_solvente'];

    $afiliado[$cantidad]['Ultimo_mes_solvente']=$afiliados[$i]['Ultimo_mes_solvente'];

    $cantidad=$cantidad+1;

```

```

    else
    {
        if ($total<25)
        {
            //es jubilado intermedio
            if ($edad>69)
            {
                if ($edad==70)
                {

```

```

((($total_mes>=0)) | ($actualyear!=$ano_correspondiente))

```

```

    $afiliado[$cantidad]['Id']=$afiliados[$i]['Id'];

    $afiliado[$cantidad]['Nombre']=$afiliados[$i]['Nombre'];

    $afiliado[$cantidad]['Apellido1']=$afiliados[$i]['Apellido1'];

    $afiliado[$cantidad]['Apellido2']=$afiliados[$i]['Apellido2'];

    $afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];

    $afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];

    $afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];

```



```

$afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];
$afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];
$afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];
$afiliado[$cantidad]['Nacionalidad']=$afiliados[$i]['Nacionalidad'];
$afiliado[$cantidad]['Sexo']=$afiliados[$i]['Sexo'];
$afiliado[$cantidad]['Fecha_nac']=$afiliados[$i]['Fecha_nac'];
$afiliado[$cantidad]['Fecha_ingreso']=$afiliados[$i]['Fecha_ingreso'];
$afiliado[$cantidad]['Fecha_ordenacion']=$afiliados[$i]['Fecha_ordenacion'];
$afiliado[$cantidad]['Email']=$afiliados[$i]['Email'];
$afiliado[$cantidad]['Telefono']=$afiliados[$i]['Telefono'];
$afiliado[$cantidad]['Condicion']=$afiliados[$i]['Condicion'];
$afiliado[$cantidad]['Estado']=$afiliados[$i]['Estado'];
$afiliado[$cantidad]['Fecha_ayuda_medic']=$afiliados[$i]['Fecha_ayuda_medic'];
$afiliado[$cantidad]['Fecha_aux_funeral']=$afiliados[$i]['Fecha_aux_funeral'];
$afiliado[$cantidad]['Fecha_jubilacion']=$afiliados[$i]['Fecha_jubilacion'];
$afiliado[$cantidad]['Fecha_muerte']=$afiliados[$i]['Fecha_muerte'];
$afiliado[$cantidad]['Fecha_incapacidad']=$afiliados[$i]['Fecha_incapacidad'];
$afiliado[$cantidad]['Fecha_retiro']=$afiliados[$i]['Fecha_retiro'];
$afiliado[$cantidad]['Descripcion']=$afiliados[$i]['Descripcion'];
$afiliado[$cantidad]['Ultimo_ano_solvente']=$afiliados[$i]['Ultimo_ano_solvente'];
$afiliado[$cantidad]['Ultimo_mes_solvente']=$afiliados[$i]['Ultimo_mes_solvente'];
$cantidad=$cantidad+1;

```

```

}
else
{
    //es jubilado mayor
    if ($edad>69)
    {
        if ($edad==70)
        {
            if

```

```

(($total_mes>=0)||($actualyear!=$Ano_correspondiente))

```

```

    $afiliado[$cantidad]['Id']=$afiliados[$i]['Id'];

    $afiliado[$cantidad]['Nombre']=$afiliados[$i]['Nombre'];

    $afiliado[$cantidad]['Apellido1']=$afiliados[$i]['Apellido1'];

    $afiliado[$cantidad]['Apellido2']=$afiliados[$i]['Apellido2'];

    $afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];

    $afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];

    $afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];

    $afiliado[$cantidad]['Nacionalidad']=$afiliados[$i]['Nacionalidad'];

    $afiliado[$cantidad]['Sexo']=$afiliados[$i]['Sexo'];

    $afiliado[$cantidad]['Fecha_nac']=$afiliados[$i]['Fecha_nac'];

    $afiliado[$cantidad]['Fecha_ingreso']=$afiliados[$i]['Fecha_ingreso'];

    $afiliado[$cantidad]['Fecha_ordenacion']=$afiliados[$i]['Fecha_ordenacion'];

    $afiliado[$cantidad]['Email']=$afiliados[$i]['Email'];

    $afiliado[$cantidad]['Telefono']=$afiliados[$i]['Telefono'];

    $afiliado[$cantidad]['Condicion']=$afiliados[$i]['Condicion'];

    $afiliado[$cantidad]['Estado']=$afiliados[$i]['Estado'];

    $afiliado[$cantidad]['Fecha_ayuda_medic']=$afiliados[$i]['Fecha_ayuda_medic'];

    $afiliado[$cantidad]['Fecha_aux_funeral']=$afiliados[$i]['Fecha_aux_funeral'];

    $afiliado[$cantidad]['Fecha_jubilacion']=$afiliados[$i]['Fecha_jubilacion'];

    $afiliado[$cantidad]['Fecha_muerte']=$afiliados[$i]['Fecha_muerte'];

    $afiliado[$cantidad]['Fecha_incapacidad']=$afiliados[$i]['Fecha_incapacidad'];

    $afiliado[$cantidad]['Fecha_retiro']=$afiliados[$i]['Fecha_retiro'];

    $afiliado[$cantidad]['Descripcion']=$afiliados[$i]['Descripcion'];

    $afiliado[$cantidad]['Ultimo_ano_solvente']=$afiliados[$i]['Ultimo_ano_solvente'];

    $afiliado[$cantidad]['Ultimo_mes_solvente']=$afiliados[$i]['Ultimo_mes_solvente'];

    $cantidad=$cantidad+1;

```

```
//no va a jubilado le faltan meses
```

```
else
```

jubilado

else //si va a

```
$afiliado[$cantidad]['Id']=$afiliados[$i]['Id'];
$afiliado[$cantidad]['Nombre']=$afiliados[$i]['Nombre'];
$afiliado[$cantidad]['Apellido1']=$afiliados[$i]['Apellido1'];
$afiliado[$cantidad]['Apellido2']=$afiliados[$i]['Apellido2'];
$afiliado[$cantidad]['Nombre_miembro']=$afiliados[$i]['Nombre_miembro'];
$afiliado[$cantidad]['Cedula']=$afiliados[$i]['Cedula'];
$afiliado[$cantidad]['Pasaporte']=$afiliados[$i]['Pasaporte'];
$afiliado[$cantidad]['Nacionalidad']=$afiliados[$i]['Nacionalidad'];
$afiliado[$cantidad]['Sexo']=$afiliados[$i]['Sexo'];
$afiliado[$cantidad]['Fecha_nac']=$afiliados[$i]['Fecha_nac'];
$afiliado[$cantidad]['Fecha_ingreso']=$afiliados[$i]['Fecha_ingreso'];
$afiliado[$cantidad]['Fecha_ordenacion']=$afiliados[$i]['Fecha_ordenacion'];
$afiliado[$cantidad]['Email']=$afiliados[$i]['Email'];
$afiliado[$cantidad]['Telefono']=$afiliados[$i]['Telefono'];
$afiliado[$cantidad]['Condicion']=$afiliados[$i]['Condicion'];
$afiliado[$cantidad]['Estado']=$afiliados[$i]['Estado'];
$afiliado[$cantidad]['Fecha_ayuda_medic']=$afiliados[$i]['Fecha_ayuda_medic'];
$afiliado[$cantidad]['Fecha_aux_funeral']=$afiliados[$i]['Fecha_aux_funeral'];
$afiliado[$cantidad]['Fecha_jubilacion']=$afiliados[$i]['Fecha_jubilacion'];
$afiliado[$cantidad]['Fecha_muerte']=$afiliados[$i]['Fecha_muerte'];
$afiliado[$cantidad]['Fecha_incapacidad']=$afiliados[$i]['Fecha_incapacidad'];
$afiliado[$cantidad]['Fecha_retiro']=$afiliados[$i]['Fecha_retiro'];
$afiliado[$cantidad]['Descripcion']=$afiliados[$i]['Descripcion'];
$afiliado[$cantidad]['Ultimo_ano_solvente']=$afiliados[$i]['Ultimo_ano_solvente'];
$afiliado[$cantidad]['Ultimo_mes_solvente']=$afiliados[$i]['Ultimo_mes_solvente'];
```



```
$cantidad=$cantidad+1;
```

```
$plan=$this->plan_m->get_plan_actual_sacerdotes($actualyear);
```

```
$afiliado[0]['jub_mayor']='Jubilado_('. $plan[0]['Beneficio_jubilacion_mayor'].)';
```

```
$afiliado[0]['jub_intermedio']='Jubilado_('. $plan[0]['Beneficio_jubilacion_intermedio'].)';
```

```
$afiliado[0]['jub_menor']='Jubilado_('. $plan[0]['Beneficio_jubilacion_menor'].)';
```

```
$afiliado[0]['inc_mayor']='Incapacitado_('. $plan[0]['Beneficio_incapacitacion_mayor'].)';
```

```
$afiliado[0]['inc_intermedio']='Incapacitado_('. $plan[0]['Beneficio_incapacitacion_intermedio'].)';
```

```
$afiliado[0]['inc_menor']='Incapacitado_('. $plan[0]['Beneficio_incapacitacion_menor'].)';
```

```
$afiliado[0]['ayuda']=$plan[0]['Beneficio_ayuda_medic'];
```

```
$data= array('perso' => $afiliado);
```

```
$this->load->view("afiliado/datatable_afiliado", $data);
```

✓ Función: view_report_efectivo

```
function view_report_efectivo()
```

```
{
```

```
    $efectivo=$this->inpreclero_m->get_all_efectivo();
```

```
    for ($i=0;$i<count($efectivo); $i++)
```

```
        $cuentas=$this->cuenta_m->get_cuenta_by_id($efectivo[$i]['Fk_dinero_cuenta']);
```

```
        $efectivo[$i]['Banco']=$cuentas[0]['Banco'];
```

```
        $efectivo[$i]['Numero_cuenta']=$cuentas[0]['Numero_cuenta'];
```

```
    $data= array('perso' => $efectivo);
```

```
    $this->load->view("inpreclero/datatable_efectivo", $data);
```

✓ Función: asiento_agregar

```
function asiento_agregar()
```

```
    {
        $Id_hiscta = $_POST["Id_hiscta"];
        $Concepto = $_POST["Concepto"];
        $Fecha_asiento = $_POST["Fecha_asiento"];
        $Observacion2 = $_POST["Observacion2"];
        $Cantidad2 = $_POST["Cantidad2"];

        $this->form_validation->set_error_delimiters('<div class="error" align="center"
style="color:#F00">', '</div>');

        $this->form_validation->set_rules('Concepto', 'Concepto', 'required');
        $this->form_validation->set_rules('Fecha_asiento', 'Fecha del asiento', 'required');

        $this->form_validation->set_rules('Observacion2', 'Observacion');

        if (!$Cantidad2)
        {
            $Cantidad2=1;
        }

        if ($this->form_validation->run() == FALSE)
        {
            $agregado = $Id_hiscta;

            $dato = array();
            $dato[0]['agregado'] = $agregado;

            $data = array('perso' => $dato);

            $this->load->view("miembro/agregar_asiento",$data);

        }
        else
        {
            $new = array('ID' => "",
                        'Concepto' => $Concepto,
                        'Fecha_asiento' => $Fecha_asiento,
                        'Observacion' => $Observacion2,
                        'Fk_contamov_hiscta' => $Id_hiscta);

            $this->miembro_m->insert_asiento($new); //Llamando a la funcion
que agrega un nuevo usuario.

            /////Lo siguiente es para crear el asiento
            $agregado2 = $this->db->insert_id();

            //traigo todas las cuentas que existen
            $dato = $this->miembro_m->get_all_cuentas();

            $dato[0]['agregado'] = $agregado2;
            $dato[0]['cantidad'] = $Cantidad2;

            $data = array('perso' => $dato);
        }
    }
}
```

```
$this->load->view("miembro/agregar_asiento_informacion",$data);
```

✓ Función: información_agregar

```
function informacion_agregar()
{
    $Total= $_POST["Total"];
    $Fk_hiscon_conmov= $_POST["Fk_hiscon_conmov"];

    //for que hace el post de los codigos del asiento
    for ($i=0;$i<$Total; $i++)

        $Cantidad3 = $_POST["Cantidad3".$i];
        $Concepto = $_POST["Concepto".$i];
        $Tipo = $_POST["Tipo".$i];
        $Fk_hiscon_con = $_POST["Fk_hiscon_con".$i];

        $this->form_validation->set_error_delimiters('<div
class="error" align="center" style="color:#F00">', '</div>');

        $this->form_validation->set_rules('Cantidad3'.$i,
'Cantidad', 'required|numeric');

        $this->form_validation->set_rules('Concepto'.$i,
'Concepto', 'required');

        $this->form_validation->set_rules('Fk_hiscon_con'.$i,
'Contabilidad', 'required');

        if($_POST["Tipo".$i]=='Seleccione')
        {
            $this->form_validation-

        }

        if($_POST["Fk_hiscon_con".$i]=='Seleccione')
        {
            $this->form_validation-

        }

        >set_rules('Tipo'.$i, 'Tipo', 'callback_tipo_check');

        >set_rules('Fk_hiscon_con'.$i, 'Fk_hiscon_con', 'callback_Fk_hiscon_con_check');

        $adulth[$i]['Cantidad'] =$Cantidad3;
        $adulth[$i]['Concepto'] =$Concepto;
        $adulth[$i]['Tipo'] =$Tipo;
        $adulth[$i]['Fk_hiscon_con'] =$Fk_hiscon_con;
        $adulth[$i]['Fk_hiscon_conmov'] =$Fk_hiscon_conmov;

    if ($this->form_validation->run() == FALSE)
    {
        $dato= $this->miembro_m->get_all_cuentas();
    }
}
```

```

$dato[0]['agregado'] = $Fk_hiscon_conmov;
$dato[0]['cantidad'] = $Total;

$data = array('perso' => $dato);

$this->load->view("miembro/agregar_asiento_informacion", $data);
}

else
{
for ($i=0; $i<$Total; $i++)

    $new = array('ID' => "",
        'Cantidad' => $adultt[$i]['Cantidad'],
        'Concepto' => $adultt[$i]['Concepto'],
        'Tipo' => $adultt[$i]['Tipo'],
        'Fk_hiscon_conmov' => $adultt[$i]['Fk_hiscon_conmov'],
        'Fk_hiscon_con' => $adultt[$i]['Fk_hiscon_con']);

    $this->miembro_m->insert_in_hiscont($new); //Llamando
}
}
$this->load->view("exito");
$this->load->view("inicio");

```

a la funcion que agrega un nuevo dato

✓ Función: nueva_noticia

```
function nueva_noticia()
```

```

$Fecha = $_POST["Fecha"];
$Nombre = $_POST["Nombre"];
$Asunto = $_POST["Asunto"];
$Descripcion = $_POST["Descripcion"];

```

```
$image_name="";
```

```
style="color:#F00">'; '</div>');
```

```
$this->form_validation->set_error_delimiters('<div class="error" align="center"');
```

```

$this->form_validation->set_rules('Fecha', 'Fecha', 'required');
$this->form_validation->set_rules('Nombre', 'Nombre', 'required');
$this->form_validation->set_rules('Asunto', 'Asunto', 'required');
$this->form_validation->set_rules('Descripcion', 'Descripcion', 'required');

```

```

if ($this->form_validation->run() == FALSE)
{
    $this->agregar_noticia();
}

else
{
    $config['upload_path'] = './uploads/';
    $config['allowed_types'] = 'gif|jpg|png';
    $config['max_size'] = '40000';
    $config['max_width'] = '100024';
    $config['max_height'] = '40768';

    $this->load->library('upload', $config);

    if ( ! $this->upload->do_upload() )
    {
        $error = array('error' => $this->upload->display_errors());

        if ( ! $this->upload->do_upload('file_name') )
        {
            $new = array('id' => NULL,

'Fecha' => $Fecha,

'Nombre' => $Nombre,

'Asunto' => $Asunto,

'Descripcion' => $Descripcion,

'Fk_noticia_administrador' => $this->session->userdata('id_usuario'));

            $this->administrador_m-
            $this->noticia_admin();
        }
        else //si el problema es diferente a que no existe
        {
            $error = array('error' => $this-
            $this->load-

>upload->display_errors());

            $data = array('upload_data' => $this->upload-

>data());

            ////////////////////////////////////////////////// para
            $imageData = $this->upload-

>data());

```

```

$image_name = $imageData['file_name'];

////////////////////////////////////////////////////////////////

$new = array('Id' => NULL,
            'Fecha' => $Fecha,
            'Nombre' => $Nombre,
            'Asunto' => $Asunto,
            'Descripcion' => $Descripcion,
            'Imagen' => $image_name,
            'Fk_noticia_administrador' => $this->
>session->userdata('id_usuario'));

//Llamando a la funcion que agrega un nuevo usuario.

$this->administrador_m->insert_noticia($new);

$this->noticia_admin();
}

```

✓ Función: login

```

function login()
{
    $username = $_POST["username"];
    $password = $_POST["password"];
    $Tipo = $_POST["Tipo"];

    $this->form_validation->set_error_delimiters('<strong><div class="error"
align="center" style="color:#FF0000">','</div></strong>');
    $this->form_validation->set_rules('username', 'Nombre de Usuario', 'required');

    if($Tipo=='Seleccione')
    {
        $this->form_validation->set_rules('Tipo', 'Tipo', 'callback_tipo_check');
    }
    else
    {
        if($Tipo=='Afiliado')
        {
            $user = $this->registro_m-
>get_user_password_afiliado_cedula($username, $password);

            if (!$user)
            {
                $user = $this->registro_m-
>get_user_password_afiliado_pasaporte($username, $password);

                if (!$user)
                {
                    $this->form_validation-
>set_rules('password', 'Password', 'callback_user_check');
}
}
}
}
}

```

```

    }
    else
    {
        if($Tipo=='Miembro')
        {
            $user = $this->registro_m-

            if(!$user)
            {
                $this->form_validation-
            }
        }
        else // es administrador
        {
            $user = $this->registro_m-

            if(!$user)
            {
                $this->form_validation-
            }
        }
    }

    if ($this->form_validation->run() == FALSE)
    {
        $this->load->view("base3");
        $this->load->view("registro/acceso");
    }
    else
    {
        $userlogin = $user[0]['Id'];
        $usernames = $user[0]['Nombre'];
        $this->session->set_userdata('id_usuario', $userlogin);
        $this->session->set_userdata('nombre_usuario',
        $usernames);

        $this->session->set_userdata('tipo', $Tipo);

        $this->useragentt($userlogin,$Tipo);

        if($Tipo=='Afiliado')
        {
            $this->load->view("base_afiliado");
            $this->load->view("mapa2");
        }
        if($Tipo=='Miembro')
        {
            $this->load->view("base_miembro");
            $this->load->view("mapa2");
        }
        if($Tipo=='Administrador')
        {

```

```

$this->load->view("base_administrador");
$this->load->view("mapa2");
}

```

✓ Función: useragentt

```
function useragentt($userlogin, $Tipo)
```

```

{
    if ($Tipo=='Afiliado')
    {
        $userafil=$userlogin;
        $usermiem=NULL;
        $useradmin=NULL;
    }
    if ($Tipo=='Miembro')
    {
        $userafil=NULL;
        $usermiem=$userlogin;
        $useradmin=NULL;
    }
    if ($Tipo=='Administrador')
    {
        $userafil=NULL;
        $usermiem=NULL;
        $useradmin=$userlogin;
    }
}

```

```
if ($this->agent->is_browser())
```

```

{
    $agent = $this->agent->browser();
    $version = $this->agent->version();
}

```

```
else
```

```

{
    $agent = 'Unidentified Browser/Celular';
    $version = 'Unidentified Version of Browser/Celular';
}

```

```
$ip = $this->getIp();
```

```
$datestring = "%Y-%m-%d";
```

```
$time = time();
```

```
$actualday = mdate($datestring, $time);
```

```
$os = $this->agent->platform(); // Platform info (Windows, Linux, Mac, etc.)
```

```
///// APARTIR DE AQUI ES PARA INSERTAR EN LA TABL control
```

```
$new = array('Id' => "",
```

```
'Ip' => $ip,
```

```
'Browser' => $agent,
```



```
'Version_browser' => $version,  
'OS' => $os,  
'Datecontrol' => $actualday,  
'Fk_control_afiliado' => $userafil,  
'Fk_control_miembro' => $usermiem,  
'Fk_control_admin' => $useradmin);
```

nuevo acceso a control.

```
$this->registro_m->control_in($new); //Llamando a la funcion que agrega un
```

Apéndice 4
(Reportes)

✓ Alerta jubilados (Año actual)

	Afiliados a Jubilación	Afiliados a Meses de Jubilación
Menor	2	1
Intermedio	1	0
Mayor	5	5

Figura 95. Reporte Alerta Jubilados.

(Elaboración propia)

- ✓ Todos los afiliados/ Afiliados por miembro/ Afiliados activos/ Historico Afiliados/ Afiliados a jubilación/ Afiliados a meses de jubilación.

The image shows a report titled "Reporte general afiliados" with a table containing several columns. The columns are mostly illegible due to blurriness, but they appear to include fields for member status, dates, and numerical values. The table is organized into several rows, each representing a different category or group of members.

Figura 96. Reporte general afiliados. (Elaboración propia)

✓ Eventos de los afiliados

Nombre de afiliado	Apellido	Nombre afiliado	Id de afiliado	Moneda	Moneda	Moneda	Fecha de evento	Fecha de inicio	Fecha de fin	Fecha de pago	Fecha de pago	Año correspondiente
Aguilera	Los Reyes	Los Reyes	150302011	150302011	150302011	150302011	23/04/2011	23/04/2011	23/04/2011	00/00/0000	00/00/0000	2011
Alexa Domínguez	Casas	Casas	150302011	150302011	150302011	150302011	03/11/2011	03/11/2011	03/11/2011	00/00/0000	00/00/0000	2011
Alexa Domínguez	Doctores	Doctores	150302011	150302011	150302011	150302011	24/11/2011	24/11/2011	24/11/2011	00/00/0000	00/00/0000	2011
Aronco	Caera	Caera	150302011	150302011	150302011	150302011	28/05/2011	28/05/2011	28/05/2011	00/00/0000	00/00/0000	2011
Afron	Mena	Mena	150302011	150302011	150302011	150302011	14/05/2011	14/05/2011	14/05/2011	00/00/0000	00/00/0000	2011
Ara Teresa	La Guara	La Guara	150302011	150302011	150302011	150302011	04/09/2011	04/09/2011	04/09/2011	00/00/0000	00/00/0000	2011
Anselmi Celis	Sequera	Sequera	150302011	150302011	150302011	150302011	01/01/2011	01/01/2011	01/01/2011	00/00/0000	00/00/0000	2011
Arge	De Das	De Das	150302011	150302011	150302011	150302011	13/01/2011	13/01/2011	13/01/2011	00/00/0000	00/00/0000	2011
Arge	De Das	De Das	150302011	150302011	150302011	150302011	24/03/2011	24/03/2011	24/03/2011	00/00/0000	00/00/0000	2011
Arge	De Das	De Das	150302011	150302011	150302011	150302011	02/05/2011	02/05/2011	02/05/2011	00/00/0000	00/00/0000	2011

Figura 97. Reporte evento de los afiliados. (Elaboración propia)

✓ Eventos generales

Fecha	Evento	Monto	Monto Agrobio	Fecha Emis.	Fecha Pago	Mes a Pago	Fecha Pago	Año
01/01/2011	Derechos II			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Baja Cto. Acord.			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Cartera op. deud.			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Carga alianza op. deud.			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Quere demand	124.5		15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Impensum	1830.0		15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Cartera sin c.	2000.0		15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Influencia veza			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Bolsa lunar			15/01/2011	15/01/2011		15/01/2011	2011
01/01/2011	Carroba			15/01/2011	15/01/2011		15/01/2011	2011

Figura 98. Reporte eventos generales. (Elaboración propia)

✓ Transacción de los afiliados

Fecha	Descripción	Debe	Haber
1/1/2012	Saldo inicial		10000
15/1/2012	Deposito		5000
30/1/2012	Retiro	2000	
15/2/2012	Deposito		3000
30/2/2012	Retiro	1000	
15/3/2012	Deposito		4000
30/3/2012	Retiro	1500	
15/4/2012	Deposito		2000
30/4/2012	Retiro	800	
15/5/2012	Deposito		1000
30/5/2012	Retiro	500	
15/6/2012	Deposito		3000
30/6/2012	Retiro	1200	
15/7/2012	Deposito		2500
30/7/2012	Retiro	700	
15/8/2012	Deposito		1500
30/8/2012	Retiro	400	
15/9/2012	Deposito		2000
30/9/2012	Retiro	600	
15/10/2012	Deposito		1000
30/10/2012	Retiro	300	
15/11/2012	Deposito		1500
30/11/2012	Retiro	200	
15/12/2012	Deposito		1000
30/12/2012	Retiro	100	
31/12/2012	Saldo final		10000

Figura 99. Reporte transacción de los afiliados. (Elaboración propia)

✓ Todas las cuentas

País	Nombre del Titular	Numero de la Cuenta
Venezuela	Inpreclero	0102-0127-67-0006-0990-28
Mercantil	Inpreclero	0105-0021-45-1021-5232-32
Banesto	Inpreclero	0134-0332-51-3321-0349-25
Copbanca	Inpreclero	0121-0170-85-0010-7739-83
Arq. de Barquisimeto	Arq. de Barquisimeto	0134-0504-17-0043-0924-24
Arq. de Calabozo	Arq. de Calabozo	0108-0169-95-0100-0578-08
Arzobispado de Caracas	Arzobispado de Caracas	0105-0601-11-8801-0933-30
Arq. de Coad. Bolívar	Arq. de Coad. Bolívar	0105-0489-17-8689-0020-49.
Arq. de Coro	Arq. de Coro	0121-0209-70-0108-5364-93
CORP BANCA	Arq. de Cumana	0105-0166-11-1066-2639-82

Figura 104. Reporte cuentas bancarias. (Elaboración propia)

✓ Plan de cuentas (contabilizadas)

ID	Codigo	ACTIVO
1	1	ACTIVO CIRCULANTE
2	1.1	CAJA Y BANCO
3	1.1.1	CAJA
4	1.1.1.1	CAJA CHICA
5	1.1.1.1.01	BANCOS NACIONALES
6	1.1.1.1.02	BANCO REPUBLICA Nº 041-9
7	1.1.1.1.03	BANCO VENEZUELA Nº 6099028
8	1.1.1.1.04	BANCO DEL CARIBE Nº 976-8
9	1.1.1.1.05	BANCO CORP BANCA Nº 442-7

Figura 105. Reporte cuentas contabilidad. (Elaboración propia)

✓ Asientos generales

Concepto Asiento	Fecha Asiento	Observación	Numero Conf Trans.	Codigo Cuenta	Nombre Cuenta	Monto	Concepto
Pago del Mes de Marzo del Caracas	15-05-2012		556465				Individual
Pago de Mes de Junio del Caracas	23-05-2012	51111	47591		ACTIVO	5000	4000 Debe
Pago de Mes de Julio del Caracas	23-05-2012	51111	47591		ACTIVO	115	4000 Haber
Pago del Mes de Agosto a: Daquin Jose	23-05-2012		405	1:1:2:11	BANCO PLAZA	515	6004
Pago de Mes de Septiembre a: NICOLASA	09-05-2012			1:1:2:11	BANCO PLAZA	5886	8686 Haber

Figura 106. Reporte asientos generales. (Elaboración propia)

✓ Transacciones cobradas/ transacciones pagadas

Nº de transacción	Fecha	Descripción	Cuenta	Debe	Haber	Saldo
1	01/01/2014	Deposito	10010010000000000000		10000000000000000000	10000000000000000000
2	02/01/2014	Deposito	10010010000000000000		10000000000000000000	20000000000000000000
3	03/01/2014	Deposito	10010010000000000000		10000000000000000000	30000000000000000000
4	04/01/2014	Deposito	10010010000000000000		10000000000000000000	40000000000000000000
5	05/01/2014	Deposito	10010010000000000000		10000000000000000000	50000000000000000000
6	06/01/2014	Deposito	10010010000000000000		10000000000000000000	60000000000000000000
7	07/01/2014	Deposito	10010010000000000000		10000000000000000000	70000000000000000000
8	08/01/2014	Deposito	10010010000000000000		10000000000000000000	80000000000000000000
9	09/01/2014	Deposito	10010010000000000000		10000000000000000000	90000000000000000000
10	10/01/2014	Deposito	10010010000000000000		10000000000000000000	100000000000000000000

Figura 107. Reporte transaccions generales. Elaboración propia)

✓ Presupuesto

Categoría	Presupuesto	Realizado
Materiales	100	100
Mano de obra	200	200
Equipos	300	300
Transporte	400	400
Alquileres	500	500
Seguros	600	600
Impuestos	700	700
Comunicaciones	800	800
Alimentación	900	900
Alquiler de maquinaria	1000	1000
Manutención	1100	1100
Seguros de vida	1200	1200
Seguros de salud	1300	1300
Seguros de accidentes	1400	1400
Seguros de incendio	1500	1500
Seguros de robo	1600	1600
Seguros de huracán	1700	1700
Seguros de terremoto	1800	1800
Seguros de inundación	1900	1900
Seguros de explosión	2000	2000
Seguros de guerra	2100	2100
Seguros de terrorismo	2200	2200
Seguros de fraude	2300	2300
Seguros de negligencia	2400	2400
Seguros de responsabilidad	2500	2500
Seguros de contaminación	2600	2600
Seguros de privacidad	2700	2700
Seguros de propiedad	2800	2800
Seguros de vida de terceros	2900	2900
Seguros de salud de terceros	3000	3000
Seguros de accidentes de terceros	3100	3100
Seguros de incendio de terceros	3200	3200
Seguros de robo de terceros	3300	3300
Seguros de huracán de terceros	3400	3400
Seguros de terremoto de terceros	3500	3500
Seguros de inundación de terceros	3600	3600
Seguros de explosión de terceros	3700	3700
Seguros de guerra de terceros	3800	3800
Seguros de terrorismo de terceros	3900	3900
Seguros de fraude de terceros	4000	4000
Seguros de negligencia de terceros	4100	4100
Seguros de responsabilidad de terceros	4200	4200
Seguros de contaminación de terceros	4300	4300
Seguros de privacidad de terceros	4400	4400
Seguros de propiedad de terceros	4500	4500
Seguros de vida de terceros	4600	4600
Seguros de salud de terceros	4700	4700
Seguros de accidentes de terceros	4800	4800
Seguros de incendio de terceros	4900	4900
Seguros de robo de terceros	5000	5000
Seguros de huracán de terceros	5100	5100
Seguros de terremoto de terceros	5200	5200
Seguros de inundación de terceros	5300	5300
Seguros de explosión de terceros	5400	5400
Seguros de guerra de terceros	5500	5500
Seguros de terrorismo de terceros	5600	5600
Seguros de fraude de terceros	5700	5700
Seguros de negligencia de terceros	5800	5800
Seguros de responsabilidad de terceros	5900	5900
Seguros de contaminación de terceros	6000	6000
Seguros de privacidad de terceros	6100	6100
Seguros de propiedad de terceros	6200	6200
Seguros de vida de terceros	6300	6300
Seguros de salud de terceros	6400	6400
Seguros de accidentes de terceros	6500	6500
Seguros de incendio de terceros	6600	6600
Seguros de robo de terceros	6700	6700
Seguros de huracán de terceros	6800	6800
Seguros de terremoto de terceros	6900	6900
Seguros de inundación de terceros	7000	7000
Seguros de explosión de terceros	7100	7100
Seguros de guerra de terceros	7200	7200
Seguros de terrorismo de terceros	7300	7300
Seguros de fraude de terceros	7400	7400
Seguros de negligencia de terceros	7500	7500
Seguros de responsabilidad de terceros	7600	7600
Seguros de contaminación de terceros	7700	7700
Seguros de privacidad de terceros	7800	7800
Seguros de propiedad de terceros	7900	7900
Seguros de vida de terceros	8000	8000
Seguros de salud de terceros	8100	8100
Seguros de accidentes de terceros	8200	8200
Seguros de incendio de terceros	8300	8300
Seguros de robo de terceros	8400	8400
Seguros de huracán de terceros	8500	8500
Seguros de terremoto de terceros	8600	8600
Seguros de inundación de terceros	8700	8700
Seguros de explosión de terceros	8800	8800
Seguros de guerra de terceros	8900	8900
Seguros de terrorismo de terceros	9000	9000
Seguros de fraude de terceros	9100	9100
Seguros de negligencia de terceros	9200	9200
Seguros de responsabilidad de terceros	9300	9300
Seguros de contaminación de terceros	9400	9400
Seguros de privacidad de terceros	9500	9500
Seguros de propiedad de terceros	9600	9600
Seguros de vida de terceros	9700	9700
Seguros de salud de terceros	9800	9800
Seguros de accidentes de terceros	9900	9900
Seguros de incendio de terceros	10000	10000

Figura 108. Reporte presupuestos. (Elaboración propia)

✓ Ver efectivo

Fecha	Detalle	Debe	Haber	Saldo
01/01/2010	Saldo inicial			30.000,00
02/01/2010	Compra	50.000,00		20.000,00
03/01/2010	Venta		20.000,00	40.000,00
04/01/2010	Pago	10.000,00		30.000,00
05/01/2010	Recepción		10.000,00	40.000,00
06/01/2010	Saldo final			40.000,00

Figura 110. Reporte efectivo. (Elaboración propia)

✓ Ver leyes

Item	Descripción	Cantidad	Valor	Observaciones
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Figura 111. Reporte leyes. (Elaboración propia)

✓ Todos los empleados

Id	Nombre	Apellido	Celular 1	Celular 2	Correo	Sexo	Edad	Fecha de nacimiento	Fecha de ingreso	Fecha de salida	Estado	Observaciones
1	3493 2017			M	34	1983-01-15	2017-01-15	2017-01-15	Activo	
2	3493 2017			F	34	1983-01-15	2017-01-15	2017-01-15	Activo	
3	3493 2017			M	34	1983-01-15	2017-01-15	2017-01-15	Activo	
4	3493 2017			F	34	1983-01-15	2017-01-15	2017-01-15	Activo	
5	3493 2017			M	34	1983-01-15	2017-01-15	2017-01-15	Activo	
6	3493 2017			F	34	1983-01-15	2017-01-15	2017-01-15	Activo	

Figura 112. Reporte general empleados. (Elaboración propia)

✓ Movimientos de los empleados

Figura 114. Reporte movimientos de los empleados. (Elaboración propia)

✓ Asientos de ingresos/egresos

Uso	Concepto	Fecha Asiento	Observación	Debe	Haber	Debe	Haber
	Ingreso ...	2017/07/07	Esta para	6400			
	Egreso ...				6400		
						REPUELO	6400

Figura 116. Reporte asientos de ingresos/egresos. (Elaboración propia)

✓ Control de acceso

	IP	Browser	Version Browser	OS	Fecha Ingreso	Nombre
1	127.0.0.1	Chrome	19.0.1084.46	Unknown Windows OS	23/05/2012	Boisid
2	127.0.0.1	Chrome	19.0.1084.46	Unknown Windows OS	23/05/2012	Boisid
3	127.0.0.1	Chrome	19.0.1084.46	Unknown Windows OS	23/05/2012	Boisid
4	127.0.0.1	Chrome	19.0.1084.46	Unknown Windows OS	23/05/2012	Josquin José
5	127.0.0.1	Chrome	19.0.1084.46	Unknown Windows OS	23/05/2012	Caracas

Figura 117. Reporte control de acceso. (Elaboración propia)