**Manual de Usuario para el Área de Contabilidad**

**Jefry Morera**

**Clase CuentaContable**

La clase CuentaContable se utiliza para gestionar la información de las cuentas contables en el sistema de la Cooperativa de Ahorro y Crédito. Esta clase permite registrar, actualizar, eliminar y mostrar las cuentas contables.

*Ejemplo en código php*

require\_once 'db\_conexion.php';

class CuentaContable {

private $idCuentaContable;

private $nombreCuentaContable;

private $monedaCuentaContable;

private $nivelCuentaContable;

private $saldoCuentaContable;

private $tipoCuentaContable;

public function \_\_construct($id = null, $nombre = null, $moneda = null, $nivel = null, $saldo = null, $tipo = null) {

$this->idCuentaContable = $id;

$this->nombreCuentaContable = $nombre;

$this->monedaCuentaContable = $moneda;

$this->nivelCuentaContable = $nivel;

$this->saldoCuentaContable = $saldo;

$this->tipoCuentaContable = $tipo;

}

// Getters

public function getIdCuentaContable() {

return $this->idCuentaContable;

}

public function getNombreCuentaContable() {

return $this->nombreCuentaContable;

}

public function getMonedaCuentaContable() {

return $this->monedaCuentaContable;

}

public function getNivelCuentaContable() {

return $this->nivelCuentaContable;

}

public function getSaldoCuentaContable() {

return $this->saldoCuentaContable;

}

public function getTipoCuentaContable() {

return $this->tipoCuentaContable;

}

// Setters

public function setIdCuentaContable($id) {

$this->idCuentaContable = $id;

}

public function setNombreCuentaContable($nombre) {

$this->nombreCuentaContable = $nombre;

}

public function setMonedaCuentaContable($moneda) {

$this->monedaCuentaContable = $moneda;

}

public function setNivelCuentaContable($nivel) {

$this->nivelCuentaContable = $nivel;

}

public function setSaldoCuentaContable($saldo) {

$this->saldoCuentaContable = $saldo;

}

public function setTipoCuentaContable($tipo) {

$this->tipoCuentaContable = $tipo;

}

// Método para registrar una nueva cuenta contable

public function crearCuentaContable() {

$conn = conectar();

$sql = "INSERT INTO cuentas\_contables (nombre, moneda, nivel, saldo, tipo) VALUES (:nombre, :moneda, :nivel, :saldo, :tipo)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':nombre', $this->nombreCuentaContable);

$stmt->bindParam(':moneda', $this->monedaCuentaContable);

$stmt->bindParam(':nivel', $this->nivelCuentaContable);

$stmt->bindParam(':saldo', $this->saldoCuentaContable);

$stmt->bindParam(':tipo', $this->tipoCuentaContable);

$stmt->execute();

$conn = null;

}

// Método para actualizar una cuenta contable existente

public function actualizarCuentaContable() {

$conn = conectar();

$sql = "UPDATE cuentas\_contables SET nombre = :nombre, moneda = :moneda, nivel = :nivel, saldo = :saldo, tipo = :tipo WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':nombre', $this->nombreCuentaContable);

$stmt->bindParam(':moneda', $this->monedaCuentaContable);

$stmt->bindParam(':nivel', $this->nivelCuentaContable);

$stmt->bindParam(':saldo', $this->saldoCuentaContable);

$stmt->bindParam(':tipo', $this->tipoCuentaContable);

$stmt->bindParam(':id', $this->idCuentaContable);

$stmt->execute();

$conn = null;

}

// Método para eliminar una cuenta contable

public function eliminarCuentaContable() {

$conn = conectar();

$sql = "DELETE FROM cuentas\_contables WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':id', $this->idCuentaContable);

$stmt->execute();

$conn = null;

}

// Método para mostrar la información de una cuenta contable

public function mostrarCuentaContable() {

$conn = conectar();

$sql = "SELECT \* FROM cuentas\_contables WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':id', $this->idCuentaContable);

$stmt->execute();

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

$conn = null;

return $result;

}

}

**Clase AsientoContable**

La clase AsientoContable se utiliza para gestionar la información de los asientos contables en el sistema de la Cooperativa de Ahorro y Crédito. Esta clase permite registrar, actualizar, eliminar y mostrar los asientos contables.

*Ejemplo del código en php:*

require\_once 'db\_conexion.php';

class AsientoContable {

private $idAsientoContable;

private $fechaAsientoContable;

private $descripcionAsientoContable;

private $totalDebeAsiento;

private $totalHaberAsiento;

private $estadoAsientoContable;

private $tipoAsiento;

public function \_\_construct($id = null, $fecha = null, $descripcion = null, $debe = null, $haber = null, $estado = null, $tipo = null) {

$this->idAsientoContable = $id;

$this->fechaAsientoContable = $fecha;

$this->descripcionAsientoContable = $descripcion;

$this->totalDebeAsiento = $debe;

$this->totalHaberAsiento = $haber;

$this->estadoAsientoContable = $estado;

$this->tipoAsiento = $tipo;

}

// Getters

public function getIdAsientoContable() {

return $this->idAsientoContable;

}

public function getFechaAsientoContable() {

return $this->fechaAsientoContable;

}

public function getDescripcionAsientoContable() {

return $this->descripcionAsientoContable;

}

public function getTotalDebeAsiento() {

return $this->totalDebeAsiento;

}

public function getTotalHaberAsiento() {

return $this->totalHaberAsiento;

}

public function getEstadoAsientoContable() {

return $this->estadoAsientoContable;

}

public function getTipoAsiento() {

return $this->tipoAsiento;

}

// Setters

public function setIdAsientoContable($id) {

$this->idAsientoContable = $id;

}

public function setFechaAsientoContable($fecha) {

$this->fechaAsientoContable = $fecha;

}

public function setDescripcionAsientoContable($descripcion) {

$this->descripcionAsientoContable = $descripcion;

}

public function setTotalDebeAsiento($debe) {

$this->totalDebeAsiento = $debe;

}

public function setTotalHaberAsiento($haber) {

$this->totalHaberAsiento = $haber;

}

public function setEstadoAsientoContable($estado) {

$this->estadoAsientoContable = $estado;

}

public function setTipoAsiento($tipo) {

$this->tipoAsiento = $tipo;

}

// Método para registrar un nuevo asiento contable

public function crearAsientoContable() {

$conn = conectar();

$sql = "INSERT INTO asientos\_contables (fecha, descripcion, total\_debe, total\_haber, estado, tipo) VALUES (:fecha, :descripcion, :debe, :haber, :estado, :tipo)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':fecha', $this->fechaAsientoContable);

$stmt->bindParam(':descripcion', $this->descripcionAsientoContable);

$stmt->bindParam(':debe', $this->totalDebeAsiento);

$stmt->bindParam(':haber', $this->totalHaberAsiento);

$stmt->bindParam(':estado', $this->estadoAsientoContable);

$stmt->bindParam(':tipo', $this->tipoAsiento);

$stmt->execute();

$conn = null;

}

// Método para actualizar un asiento contable existente

public function actualizarAsientoContable() {

$conn = conectar();

$sql = "UPDATE asientos\_contables SET fecha = :fecha, descripcion = :descripcion, total\_debe = :debe, total\_haber = :haber, estado = :estado, tipo = :tipo WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':fecha', $this->fechaAsientoContable);

$stmt->bindParam(':descripcion', $this->descripcionAsientoContable);

$stmt->bindParam(':debe', $this->totalDebeAsiento);

$stmt->bindParam(':haber', $this->totalHaberAsiento);

$stmt->bindParam(':estado', $this->estadoAsientoContable);

$stmt->bindParam(':tipo', $this->tipoAsiento);

$stmt->bindParam(':id', $this->idAsientoContable);

$stmt->execute();

$conn = null;

}

// Método para eliminar un asiento contable

public function eliminarAsientoContable() {

$conn = conectar();

$sql = "DELETE FROM asientos\_contables WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':id', $this->idAsientoContable);

$stmt->execute();

$conn = null;

}

// Método para mostrar la información de un asiento contable

public function mostrarAsientoContable() {

$conn = conectar();

$sql = "SELECT \* FROM asientos\_contables WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':id', $this->idAsientoContable);

$stmt->execute();

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

$conn = null;

return $result;

}

}

**Clase DetalleAsiento**

La clase DetalleAsiento se utiliza para gestionar los detalles de los asientos contables en el sistema de la Cooperativa de Ahorro y Crédito. Esta clase permite añadir, eliminar y aplicar detalles de asientos contables.

*Ejemplo del código en php:*

require\_once 'db\_conexion.php';

class DetalleAsiento {

private $idDetalleAsiento;

private $debeAsiento;

private $haberAsiento;

private $idAsientoContable;

private $idCuentaContable;

public function \_\_construct($idDetalle = null, $debe = null, $haber = null, $idAsiento = null, $idCuenta = null) {

$this->idDetalleAsiento = $idDetalle;

$this->debeAsiento = $debe;

$this->haberAsiento = $haber;

$this->idAsientoContable = $idAsiento;

$this->idCuentaContable = $idCuenta;

}

// Getters

public function getIdDetalleAsiento() {

return $this->idDetalleAsiento;

}

public function getDebeAsiento() {

return $this->debeAsiento;

}

public function getHaberAsiento() {

return $this->haberAsiento;

}

public function getIdAsientoContable() {

return $this->idAsientoContable;

}

public function getIdCuentaContable() {

return $this->idCuentaContable;

}

// Setters

public function setIdDetalleAsiento($idDetalle) {

$this->idDetalleAsiento = $idDetalle;

}

public function setDebeAsiento($debe) {

$this->debeAsiento = $debe;

}

public function setHaberAsiento($haber) {

$this->haberAsiento = $haber;

}

public function setIdAsientoContable($idAsiento) {

$this->idAsientoContable = $idAsiento;

}

public function setIdCuentaContable($idCuenta) {

$this->idCuentaContable = $idCuenta;

}

// Método para añadir un nuevo detalle de asiento

public function añadirDetalleAsiento() {

$conn = conectar();

$sql = "INSERT INTO detalle\_asientos (debe, haber, id\_asiento, id\_cuenta) VALUES (:debe, :haber, :idAsiento, :idCuenta)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':debe', $this->debeAsiento);

$stmt->bindParam(':haber', $this->haberAsiento);

$stmt->bindParam(':idAsiento', $this->idAsientoContable);

$stmt->bindParam(':idCuenta', $this->idCuentaContable);

$stmt->execute();

$conn = null;

}

// Método para eliminar un detalle de asiento

public function eliminarDetalleAsiento() {

$conn = conectar();

$sql = "DELETE FROM detalle\_asientos WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':id', $this->idDetalleAsiento);

$stmt->execute();

$conn = null;

}

// Método para aplicar un detalle de asiento

public function aplicarDetalleAsiento() {

$conn = conectar();

$sql = "UPDATE detalle\_asientos SET debe = :debe, haber = :haber, id\_asiento = :idAsiento, id\_cuenta = :idCuenta WHERE id = :id";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':debe', $this->debeAsiento);

$stmt->bindParam(':haber', $this->haberAsiento);

$stmt->bindParam(':idAsiento', $this->idAsientoContable);

$stmt->bindParam(':idCuenta', $this->idCuentaContable);

$stmt->bindParam(':id', $this->idDetalleAsiento);

$stmt->execute();

$conn = null;

}

}

**Clase TipoAsiento**

La clase TipoAsiento se utiliza para gestionar los tipos de asientos contables en el sistema de la Cooperativa de Ahorro y Crédito. Esta clase permite registrar tipos de asientos.

*Ejemplo del código en php:*  
require\_once 'db\_conexion.php';

class TipoAsiento {

private $idTipoAsiento;

private $nombreTipoAsiento;

private $descripcionTipoAsiento;

public function \_\_construct($id = null, $nombre = null, $descripcion = null) {

$this->idTipoAsiento = $id;

$this->nombreTipoAsiento = $nombre;

$this->descripcionTipoAsiento = $descripcion;

}

// Getters

public function getIdTipoAsiento() {

return $this->idTipoAsiento;

}

public function getNombreTipoAsiento() {

return $this->nombreTipoAsiento;

}

public function getDescripcionTipoAsiento() {

return $this->descripcionTipoAsiento;

}

// Setters

public function setIdTipoAsiento($id) {

$this->idTipoAsiento = $id;

}

public function setNombreTipoAsiento($nombre) {

$this->nombreTipoAsiento = $nombre;

}

public function setDescripcionTipoAsiento($descripcion) {

$this->descripcionTipoAsiento = $descripcion;

}

// Método para registrar un nuevo tipo de asiento

public function registrarTipoAsiento() {

$conn = conectar();

$sql = "INSERT INTO tipo\_asiento (nombre, descripcion) VALUES (:nombre, :descripcion)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':nombre', $this->nombreTipoAsiento);

$stmt->bindParam(':descripcion', $this->descripcionTipoAsiento);

$stmt->execute();

$conn = null;

}

}

**Clase TipoCuentaContable**

La clase TipoCuentaContable se utiliza para gestionar los tipos de cuentas contables en el sistema de la Cooperativa de Ahorro y Crédito. Esta clase permite registrar tipos de cuentas contables.

*Ejemplo del código en php:*

require\_once 'db\_conexion.php';

class TipoCuentaContable {

private $idTipoCuentaContable;

private $nombreTipoCuenta;

private $descripcionTipoCuenta;

public function \_\_construct($id = null, $nombre = null, $descripcion = null) {

$this->idTipoCuentaContable = $id;

$this->nombreTipoCuenta = $nombre;

$this->descripcionTipoCuenta = $descripcion;

}

// Getters

public function getIdTipoCuentaContable() {

return $this->idTipoCuentaContable;

}

public function getNombreTipoCuenta() {

return $this->nombreTipoCuenta;

}

public function getDescripcionTipoCuenta() {

return $this->descripcionTipoCuenta;

}

// Setters

public function setIdTipoCuentaContable($id) {

$this->idTipoCuentaContable = $id;

}

public function setNombreTipoCuenta($nombre) {

$this->nombreTipoCuenta = $nombre;

}

public function setDescripcionTipoCuenta($descripcion) {

$this->descripcionTipoCuenta = $descripcion;

}

// Método para registrar un nuevo tipo de cuenta contable

public function registrarTipoCuentaContable() {

$conn = conectar();

$sql = "INSERT INTO tipo\_cuenta (nombre, descripcion) VALUES (:nombre, :descripcion)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':nombre', $this->nombreTipoCuenta);

$stmt->bindParam(':descripcion', $this->descripcionTipoCuenta);

$stmt->execute();

$conn = null;

}

}

**Relaciones entre tablas:**

Una Cuenta Contable puede estar asociada a muchos Detalle Asiento.

* Un Detalle Asiento pertenece a un solo Asiento Contable y una Cuenta Contable.
* Un Asiento Contable puede tener muchos Detalle Asiento y está asociado a un solo Tipo Asiento.
* Un Tipo Asiento puede estar asociado a muchos Asiento Contable.
* Un Tipo Cuenta Contable puede estar asociado a muchos Cuenta Contable.