

**ITCR – Escuela de Ing. en Computación – Base de Datos 1 – Prof. fquiros –
Agosto 2021**

Proyecto para su desarrollo en 3 tareas programadas.

Uno. Objetivo: Implementar una base de datos física, así como escribir el **código en capa lógica y física para el mantenimiento de entidades (CRUD)** y el procesamiento de datos.

Dos. Descripción general del proyecto.

Cuenta de Ahorros personal, beneficiarios, subcuentas objetivo y movimientos en colones y dólares.

La cuenta de ahorros se orienta a personas físicas, una cuenta se asocia con un tipo de cuenta que configura los siguientes valores:

- el nombre del tipo de cuenta,
- la moneda,
- el saldo mínimo,
- la multa por incumplimiento de saldo mínimo,
- el monto mensual de cargos por servicio,
- el número máximo de operaciones en cajero humano sin cobro de comisión,
- el número máximo de operaciones en cajero automático sin cobro de comisión,
- el monto de comisión por cada operación de cajero humano en exceso al número máximo
- el monto de comisión por cada operación en cajero automático en exceso al número máximo, y
- la tasa de interés anualizada.

La cuenta se asocia con único dueño, el cual define en el momento de su creación hasta 3 beneficiarios, los cuales se reparten un porcentaje del saldo en caso de muerte del dueño de la cuenta, la suma de tales porcentajes debe sumar 100, y en cualquier momento el cliente dueño puede cambiar los porcentajes o cambiar los beneficiarios. Un beneficiario puede ser dueño de una cuenta de ahorro, en ese caso los campos en común de ambas entidades (beneficiario y cliente) deben existir una sola vez.

De las personas, ya sean dueñas o beneficiarios se almacena:

- su número de documento de identidad,
- su parentesco con el dueño de la cuenta (en caso de ser beneficiarios),
- el nombre,
- la fecha de nacimiento,
- al menos dos teléfonos, y
- el email.

Los movimientos de una cuenta de ahorros pueden ser al menos, las siguientes:

- depósito en cajero humano (+),
- depósito en cajero automático (+)
- depósito por transferencia (+)
- compras (-),
- devolución de compra (+),
- retiro en cajero automático (-),
- retiro en cajero humano (-),
- retiro por transferencia (+),
- comisión por operaciones en cajero humano después del máximo permitido gratis (-),
- comisión por operaciones en cajero automático después del máximo permitido gratis (-),
- crédito por transferencia bancaria (+),
- débito por transferencia bancaria (-),
- comisión por transferencia bancaria (-),
- cargos por servicio mensual (-),
- intereses mensuales (+),
- aplicación de monto ahorrar en cuenta a subcuenta objetivo (-),
- redención de subcuenta objetivo (+),
- redención de intereses acumulados de subcuenta objetivo (+),
- multa por incumplimiento de saldo mínimo (-),
- débito por ajuste (-)
- crédito por ajuste (+).

Los movimientos pueden ser en colones o dólares, según la moneda asociada a la cuenta. Sin embargo, para el caso de las compras, los retiros o depósitos, el movimiento original puede ser en una moneda que no es la que corresponde a la cuenta, en este caso se hará una conversión según el tipo de cambio de conversión de una moneda a otra. Por ejemplo, si la cuenta es en dólares y se hace una compra (o un retiro o una transferencia) en colones, el movimiento se registra en dólares convertido desde el monto en colones, usando el tipo de cambio para la “compra de dólares” vigente al momento de la operación. Si la cuenta es en colones, y se hace una compra en dólares (o transferencia, o retiro), el movimiento se registra en colones convertido desde el monto en dólares usando el tipo de cambio para la “venta de dólares” vigente al momento de la operación.

Se usa el tipo de cambio de “compra de dólares”, si la operación es en colones, pero la cuenta es en dólares. Se usa el tipo de cambio de “venta de dólares”, si la operación es en dólares, pero la cuenta es en colones.

Se mantiene información histórica por fecha, del valor de tipo de cambio para compra o venta, utilizado por el banco.

Todos los meses, en el día del mes que corresponde al día de la fecha de apertura de la cuenta, se hace un “cierre” de operaciones del “mes” y se emite un estado de cuenta, el cual, partiendo de un saldo inicial (del inicio de “mes”, que es el saldo final del “mes” anterior), reporta los movimientos en orden cronológico, indicando para cada movimiento el nuevo saldo, y aplicando los movimientos que se aplican al final del mes (o sea al cierre, respecto del día de la fecha de la apertura). Ejemplo de los movimientos que se aplican al cierre son: crédito por intereses, multa por incumplimiento de saldo mínimo, cargos por servicio, etc.

Si la cuenta se apertura un 05 de Marzo, los cierres se harán los 05 de cualquier mes, o sea 05 de Abril, 05 de Mayo, etc. Y resume los movimientos, de los 06 hasta los 05 del siguiente mes. Hay que tomar en cuenta excepciones cuando la cuenta se abrió 31 para los meses de 30 días, o si se abrió 29, 30 o 31 para el mes de febrero.

Los intereses se calculan una vez al mes, en el momento de emisión de estado de cuenta (tiempo de cierre), se hace con respecto del saldo mínimo que mantuvo la cuenta durante el “mes calendario” anterior.

La subcuenta objetivo funciona así: el cliente entra al sitio web y establece que hará un ahorro mensual por un monto fijo en una cuenta paralela adjunta a la cuenta de ahorros, la cual se debitará cada mes en mismo día del mes (en los días que el cliente configura), por ejemplo, los días que se configuran son 15 y 30, ello implicara que en tales un débito es realizado automáticamente por el sistema, como parte de un proceso masivo (que corre una vez diaria para todas aquellas cuentas que la fecha de marras - la del proceso- corresponde al día de ahorro), en la cual debita de la cuenta de ahorro fuente y acredita en la cuenta objetivo.

Si la cuenta de ahorro no tiene saldo suficiente para realizar el débito, este no se realiza. El cliente establece el período del ahorro en la cuenta objetivo, mínimo 3 meses, máximo 24 meses, durante el mismo el monto ahorrado no podrá ser retirado y se mantendrá un saldo independiente que muestre la suma del monto ahorrado. También de manera independiente se acumulan intereses, que se sumaran al saldo al final del periodo, estos intereses se calculan y acumulan diariamente. El cliente le da un nombre al objetivo del ahorro: ejemplo: ‘Compra de carro’, ‘Viaje al mundial de Qatar’, ‘ahorro para pago de impuestos’, etc. Al finalizar el período, el saldo de lo ahorrado en la cuenta objetivo, y los intereses acumulados se acreditan en la cuenta de ahorros. El estado de cuenta informa sobre el saldo en la subcuenta objetivo y los intereses acumulados. Si al aplicar el monto a ahorrar, la cuenta no tiene suficiente saldo, se aplicará el saldo disponible de manera que el saldo quedará en cero. Ejemplo, si el monto a ahorrar es 1000, pero la cuenta de ahorro tiene en saldo 750, se aplican los 750.

La tasa de interés de las subcuentas objetivo es de un 6% anual, al cual se le suma 0.5% por cada mes del período de ahorro en exceso al mínimo que son 3 meses. Ejemplo: si el periodo es de 3 meses, la tasa será 6.0%, si el periodo es 4 meses la tasa será 6.5% anual ($6.0 + 0.5 * 1$ mes), si el periodo es 7 meses es 8.0 ($6.0 + 0.5 * 4$

meses). Para el cálculo del interés diario, la tasa de interés se divide entre la 365 o 366 días del año.

Tres. ¿Qué funcionalidades debe implementar?

Los objetivos de la etapa 1 son:

- familiarizarse con la tecnología o plataforma que se use para programar la capa lógica, y la tecnología del SABD,
- la creación de una base de datos física, la creación de procedimientos almacenados no complejos para hacer CUD,
- la escritura de scripts para procesar documentos XML y finalmente,
- conectar la BD y la invocación de SP des la capa lógica.

Para la etapa 1 de esta tarea

+Un sitio web, en el cual el cliente (ahorrante), ingresa con Usuario y Password.

En el cual puede hacer las siguientes acciones:

+ Editar beneficiarios, los campos que se editan son:

Nombre (string de 40 caracteres),
referencia a un catálogo de nombres de parentesco (string de 20 caracteres),
porcentaje de beneficio (entero, no puede ser más de 100),
fecha de nacimiento (date), documento de identificación (string numérico de 20), email y,
2 teléfonos.

El catálogo de parentesco tiene valores para representar: hijo, hija, suegro, suegra, padre, madre, primo, prima, nuera, yerno, etc.

Las acciones son:

Agregar un beneficiario, tal que un cliente no puede tener más de 3 beneficiarios.

Actualizar nombre, parentesco o porcentaje de un beneficiario, o eliminar un beneficiario.

La eliminación es lógica, implica cambiar un campo que se llama activo o "enabled" (true si el beneficiario está activo, falso de otra manera), y almacenar una fecha de desactivación.

La edición debe validar los campos.

- + Alerta acerca de beneficiarios. Si la suma de los porcentajes de los beneficiarios no suma 100, debe mostrarse una alerta (rotulito llamativo), en algún lado de la página que diga: "la suma de los porcentajes de sus beneficiarios no suma 100, favor corregir".

- + Un script para la inserción de datos en catálogos, entidades (clientes, beneficiarios, tipo documento identidad, parentesco, monedas); e inserción de clientes, cuentas.

Respecto de usuarios, hay 2 tipos, administrador y usuario cliente. El usuario cliente a su vez debe asociarse con las cuentas de ahorro que puede acceder. El usuario administrador, puede acceder los datos de cualquier cuenta de ahorro, aunque para esta primera tarea no se implementan funcionalidades para usuarios administradores.

Cuatro. Datos de prueba.

Para la carga de datos de prueba respecto de catálogos, se hará a través de la carga desde un archivo XML con la siguiente estructura.

```
<!-- catalogos-->

<Tipo_Doc>
  <TipoDocuIdentidad Id="1" Nombre="Cedula Nacional"/>
  <TipoDocuIdentidad Id="2" Nombre="Cedula Residente"/>
  <TipoDocuIdentidad Id="3" Nombre="Pasaporte"/>
  <TipoDocuIdentidad Id="4" Nombre="Cedula Juridica"/>
  <TipoDocuIdentidad Id="5" Nombre="Permiso de Trabajo"/>
  <TipoDocuIdentidad Id="6" Nombre="Cedula Extranjera"/>
</Tipo_Doc>

<Tipo_Moneda>
  <TipoMoneda Id="1" Nombre="Colones" Simbolo="¢"/>
  <TipoMoneda Id="2" Nombre="Dolares" Simbolo="$"/>
  <TipoMoneda Id="3" Nombre="Euros" Simbolo="€"/>
</Tipo_Moneda>

<Parentezcos>
  <Parentezco Id="1" Nombre="Padre"/>
  <Parentezco Id="2" Nombre="Madre"/>
  <Parentezco Id="3" Nombre="Hijo"/>
  <Parentezco Id="4" Nombre="Hija"/>
  <Parentezco Id="5" Nombre="Hermano"/>
  <Parentezco Id="6" Nombre="Hermana"/>
  <Parentezco Id="7" Nombre="amigo"/>
  <Parentezco Id="8" Nombre="amiga"/>
</Parentezcos>
```

```
<Tipo_Cuenta_Ahorros>
  <TipoCuentaAhorro
    Id="1"
    Nombre="Proletario"
    IdTipoMoneda="1"
    SaldoMinimo="25000.00"
    MultaSaldoMin="3000.00"
    CargoAnual = "5000"
    NumRetirosHumano="5"
    NumRetirosAutomatico ="8"
    comisionHumano="300"
    omisionAutomatico="300"
    interes ="10" />
  <TipoCuentaAhorro
    Id="2"
    Nombre="Profesional"
    IdTipoMoneda="1"
    SaldoMinimo="50000.00"
    MultaSaldoMin="3000.00"
    CargoAnual = "15000"
    NumRetirosHumano="5"
    NumRetirosAutomatico ="8"
    comisionHumano="500"
    comisionAutomatico="500"
    interes ="15" />
  <TipoCuentaAhorro
    Id="3"
    Nombre="Exclusivo"
    IdTipoMoneda="1"
    SaldoMinimo="100000.00"
    MultaSaldoMin="3000.00"
    CargoAnual = "30000"
    NumRetirosHumano="5"
    NumRetirosAutomatico ="8"
    comisionHumano="1000"
    comisionAutomatico="1000"
    interes ="20" />
</Tipo_Cuenta_Ahorros>
<Monedas>
<Moneda Id="1" Nombre="Colones"/>
<Moneda Id="2" Nombre="Dolares"/>
</Monedas>
```

Las llaves de las tablas catálogo, se insertan tal cual vienen en el archivo XML. O sean no son llaves identity, ni autoincrementales, todos los equipos de desarrollo, comparten la misma llave (ID) de los catálogos.

Para las tablas no-catálogo, sus llaves SI son identity, por lo tanto, cada proyecto genera llaves propias. El *mapeo* entre tablas para poder establecer las relaciones, dando el valor apropiado a los FK, será a través de llaves alternas. En Cuenta será el numero de cuenta, en Persona será el ValorDocumentoidentidad.

Los nodos XML para tablas no-catálogo:

```
<!-- entidades no-catalogos-->
<Personas>
  <Persona
    TipoDocuIdentidad="1"
    Nombre="Juan de la Barca"
    ValorDocumentoIdentidad="117370445"
    FechaNacimiento="1999-03-20"
    Email="aguerojavith@gmail.com"
    telefono1="85343403"
    telefono2="24197636"/>
  <Persona
    TipoDocuIdentidad="1"
    Nombre="Pedro Camacho Fernandez"
    ValorDocumentoIdentidad="12738545"
    FechaNacimiento="1994-10-13"
    Email="osadage@gmail.com"
    telefono1="87541766"
    telefono2="24197545"/>
</Personas>

<Cuentas>
  <Cuenta ValorDocumentoIdentidadDelCliente="117370445"
    TipoCuentaId="1"
    NumeroCuenta="11000001"
    FechaCreacion="2020-10-13"
    Saldo="1000000.00"/>
</Cuentas>

<!-- Entre 20 y 30-->

<Beneficiarios>
  <Beneficiario
```

```

    NumeroCuenta="11000001"
    ValorDocumentoIdentidadBeneficiario="117370445"
    ParentezcoId="5"
    Porcentaje="25" />
</Beneficiarios>

<Usuarios>
  <Usuario User="jaguero" Pass="LaFacil" EsAdministrador="0" />
  <Usuario User="fquiros" Pass="MyPass123*" EsAdministrador="1" />
</Usuarios>

<Usuarios_Ver>
  <UsuarioPuedeVer User="jaguero" NumeroCuenta="11000001" />
</Usuarios_Ver>

```

Para la siguiente tarea, la estructura de los XML puede variar para cumplir con nuevos requerimientos.

Cinco. ¿Qué se pide?

El código en capa lógica para la creación del sitio del cliente.

Los SP para realizar la edición de beneficiarios.

Scripts para llenado de catálogos.

Script de llenado de tablas no-catalogo: personas, cuentas y beneficiarios.

La documentación que es una bitácora.

Seis. Reglas.

Documentación: una bitácora escrita en un blog en el cual día a día, debe utilizar un Blogger, se indica la cantidad de horas trabajadas por el equipo de trabajo, se hace un relato de los avances, acerca de los problemas encontrados, como fueron resueltos; dudas, divergencias de criterio, forma en que trabajó el equipo de trabajo, problemas con la instalación del software, problemas de aprendizaje del framework, investigaciones, pruebas de concepto, experiencias, moralejas, ayuda recibida, consejos a dar, buenas prácticas descubiertas, incluir preguntas que se hagan al profe en el foro o en comunicaciones privadas, etc.; que describa el proceso de solución de la tarea programada; una descripción sincera y detallada será bien evaluada. Incluya referencias externas a recursos utilizados en internet para solventar dudas o resolver problemas, debe incluir los mensajes de error y como fueron resueltos. La última entrada del blog es un resumen que indique la cantidad de horas total de trabajo, por persona, así como un análisis de resultados de los que se está

entregado. Será muy apreciado que usen github para dar seguimiento al código de capa lógica y capa de datos, si es así debe mostrar las métricas del github en el análisis de resultados. Debe ser exhaustivo.

Todo el código referido a base de datos, debe ser un procedimiento almacenado. No puede haber SQL incrustado en capa lógica.

Grupos de 2 personas. Motor de base de datos: MS SQL cualquier versión superior a 2014. Código en capa lógica, en el lenguaje o framework de su preferencia. Se premia con 10 puntos, si la tarea se entrega en días previos.

Fecha de entrega: 20 de Setiembre.