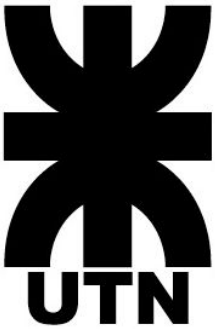


Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos
Trabajo Práctico
1° Cuatrimestre 2015
FRBA-Pago Electrónico

Nombre y Apellido	Legajo
Ariel Folino	144.182-6
Federico Sena	144.777-4
Federico Hipperdinger	141.893-2
Cristian Maldonado	144.540-6

Curso: K3013
Nro Grupo: 18

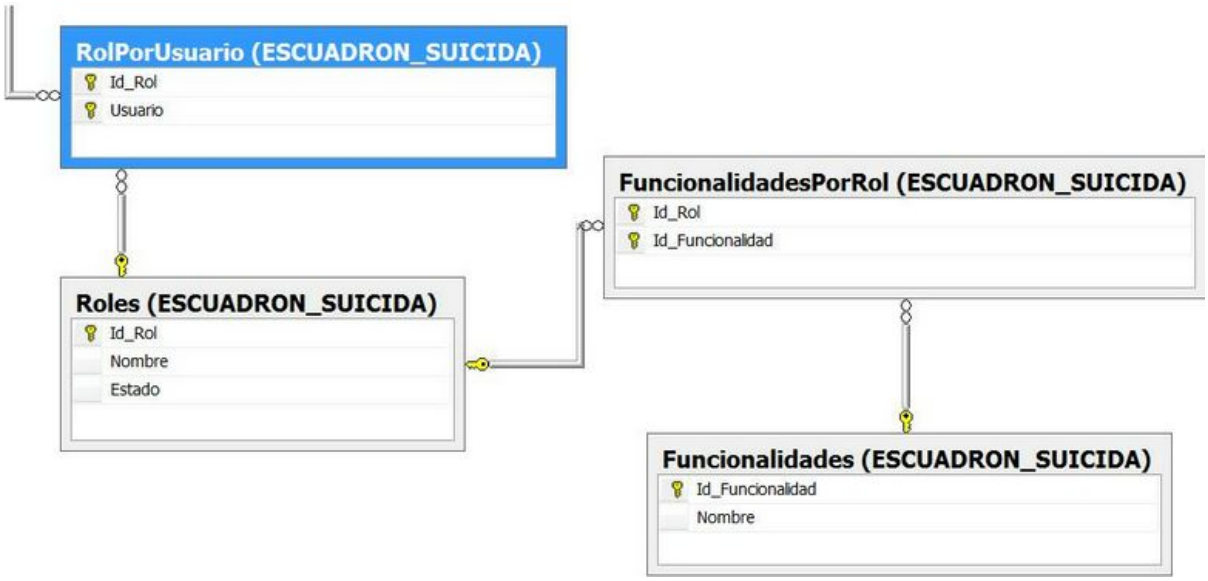
Estrategia:

El presente trabajo práctico se comenzó a partir de la tabla Maestra dada como modelo a normalizar y migrarando los datos ya existentes, actualmente desnormalizados. En primera instancia, se analizó esta tabla y fuimos identificando las diferentes tablas a normalizar y creando nuevas con el fin de obtener la información necesaria para cumplir con las funcionalidades del sistema.
Con las tablas ya creadas, definimos las distintas relaciones y la identificación de las correspondientes Primary Keys y Foreign Keys.
Luego, comenzamos con el desarrollo de cada una de las ABMs teniendo en cuenta todo lo dicho anteriormente. A lo largo de la implementación, se tuvo que ir corrigiendo el modelo planteado en una primer instancia ya sea agregando, quitando, o modificando relaciones entre tablas como también datos sobre las mismas.

1. ABM de Rol

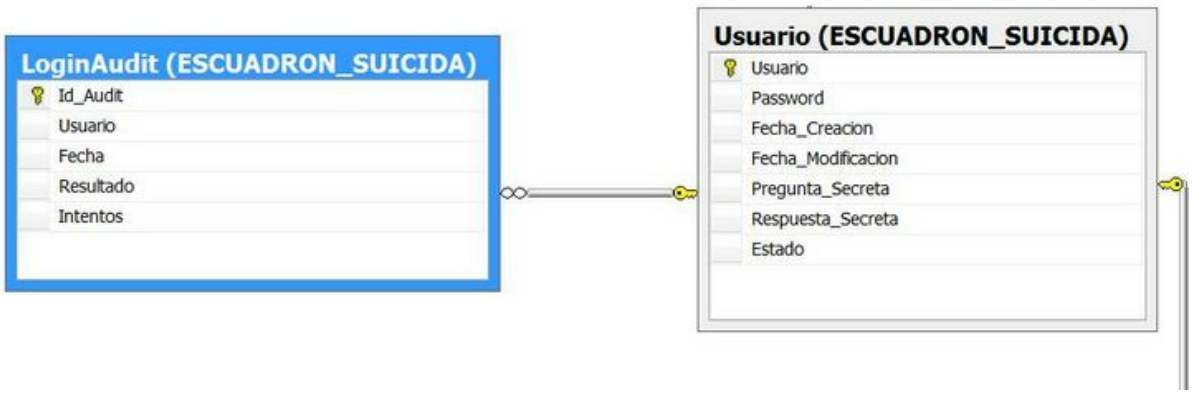
En esta funcionalidad creamos nuevas tablas, una de ellas denominada Roles, que contiene los roles que posee el sistema (en este caso los dos mencionados en el TP), que están identificados por un Id_Rol (código) y por su nombre, conteniendo un estado con formato de bit que representa la habilitación o no de ese rol. Además, esto permite realizar un alta, baja o modificación, aportando una mayor facilidad.
También creamos la tabla Funcionalidades que contiene cada una de las funcionalidades que posee el sistema, identificadas por un Id y el nombre. Al hacer esto se generó una relación muchos a muchos, por lo tanto, para solucionar este inconveniente tuvimos que crear la tabla FuncionalidadesPorRol que asocia cada rol con las funcionalidades del sistema correspondientes al mismo.
A su vez, también creamos la tabla RolPorUsuario para poder relacionar cada usuario de nuestro sistema con el rol/es que posee y resolver

nuevamente la relación muchos a muchos entre las tablas Roles y Usuario.



2. Login y Seguridad

En esta ABM, se permite dar el logueo, por medio del inicio de sesión de usuario, validando las restricciones necesarias.
En caso de ser administrador, se generan todas las funcionalidades a las cuales puede acceder, y lo mismo sucede en caso de ser cliente.
A su vez, creamos una tabla LoginAudit para poder manejar los intentos fallidos de cada usuario y su respectiva registración. Esta tabla respeta el esquema de tipo log, conteniendo al usuario que se trató loguear, la fecha y hora, si el intento fue fallido o no y, en caso de ser fallido, el número de intento.
Asumimos que el Administrador puede tener el rol de administrador y el de cliente en este caso, pero también podría darse que no. En el enunciado encontramos ambiguo este aspecto. Igualmente, cumplimos con la condición de que un Administrador pueda acceder a todas las funcionalidades del sistema, y en caso de también ser cliente, acceda sólo a las acotadas por ese rol.
Nosotros lo utilizamos para demostrar que la aplicación permite seleccionar el rol correspondiente en caso de tener más de uno.



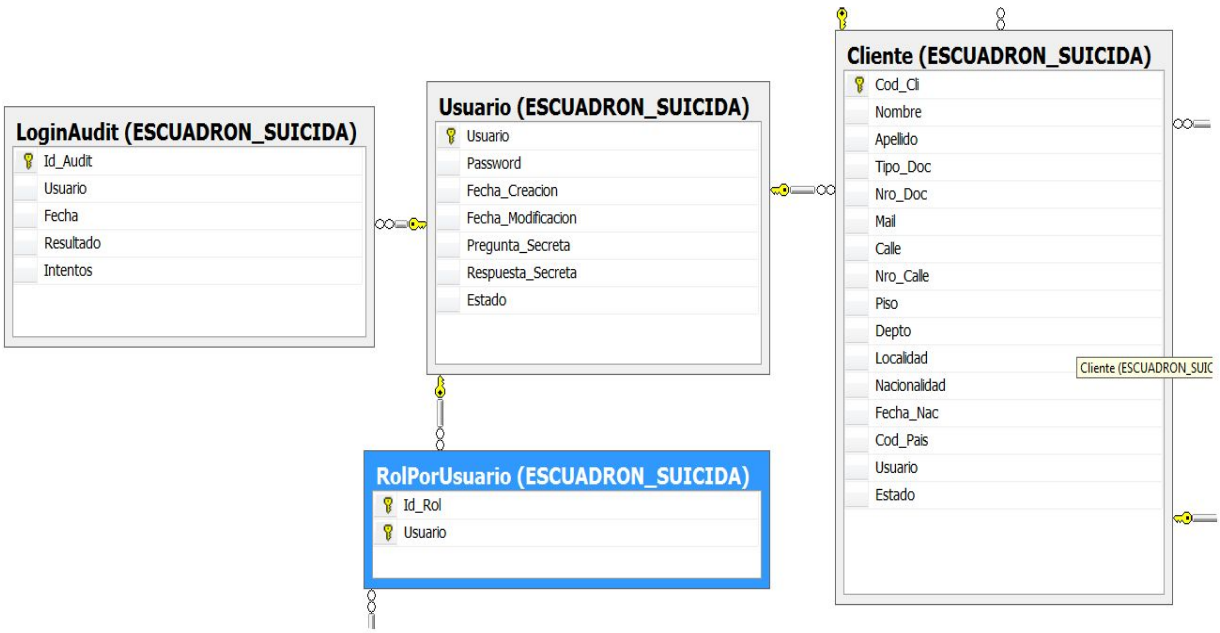
Con Intentos, acumulamos los intentos fallidos de un usuario y, a la vez, en caso de realizar correctamente el logueo, se inicializa en 0 nuevamente. No contemplamos que haya ya un usuario logueado. También guardamos en la misma tabla el Resultado, si fue exitoso o no el logueo.

Además, configuramos la validación de Login en la aplicación para que cumpla las condiciones requeridas en cuanto a visibilidad de las funcionalidades y la selección de los roles.

3. ABM de Usuario

En esta funcionalidad comenzamos con la creación de la tabla Usuario, ya que no había ninguna referencia ni mención de los usuarios en la tabla dada por el TP. Para llevar a cabo esto, nos conectamos a la base de datos para poder realizar las inserciones correspondientes a través de la ejecución de ciertas consultas.
Se tuvieron que crear todos los usuarios a los clientes ya existentes, de manera tal que, estos clientes tengan ya un único usuario asignado. Para realizar esto, vimos conveniente asignarle como username el número de documento de cada cliente, ya que es un número único que identifica a cada uno, corroborando la no duplicación del mismo.

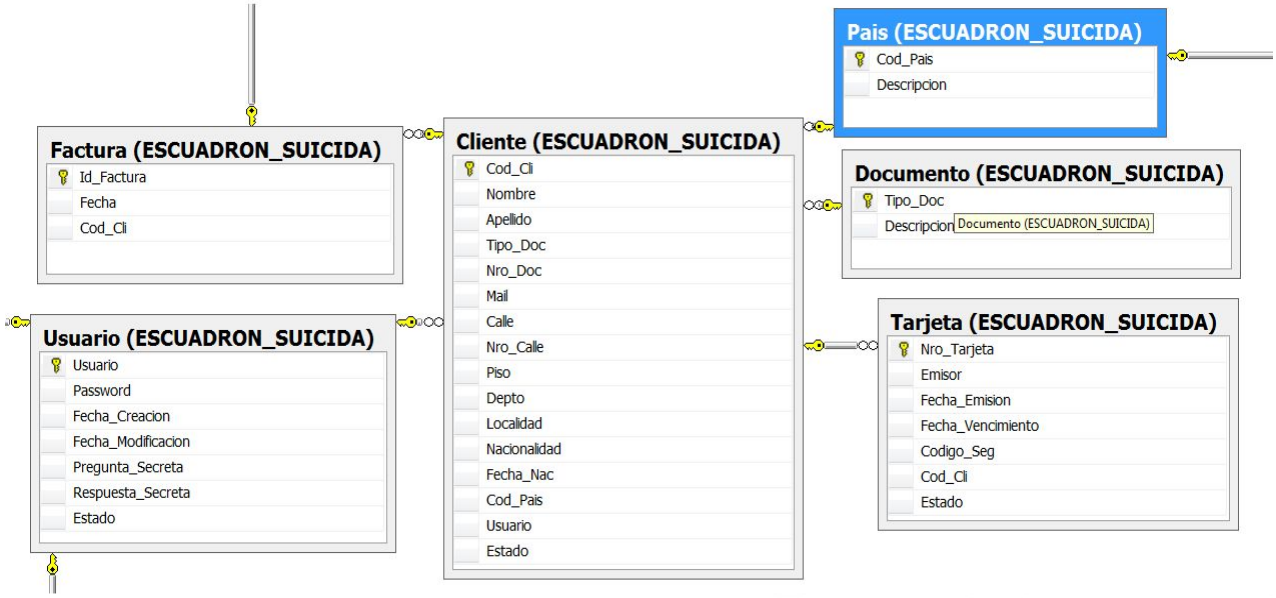
Para el alta, modificación o baja de los usuarios se tuvo en cuenta el rol asignado, y los diferentes datos obligatorios como Username, la pregunta y respuesta secreta, Password, Fecha de creación, Fecha de última modificación, se verificó que todos los campos cumplieran los requisitos pedidos.
Una vez realizado esto, buscamos la manera de poder encriptar la password pedida utilizando el algoritmo SHA256, a través de una biblioteca en C# . Con esto, almacenamos la password ya encriptada, asociada al usuario correspondiente.
También le creamos un estado al usuario para poder guardar su condición de habilitado/inhabilitado.
Además, otro punto muy importante fue que tuvimos que utilizar el Usuario como clave primaria para relacionarlo con la tabla cliente, sabiendo que existe una relación única entre ambas tablas, asegurando que cada cliente tenga un único usuario asignado.



Vale aclarar que a todos los clientes existentes, les asignamos como password 1234 ya que no se mencionaba información alguna de esto en el enunciado.

4. ABM de Clientes

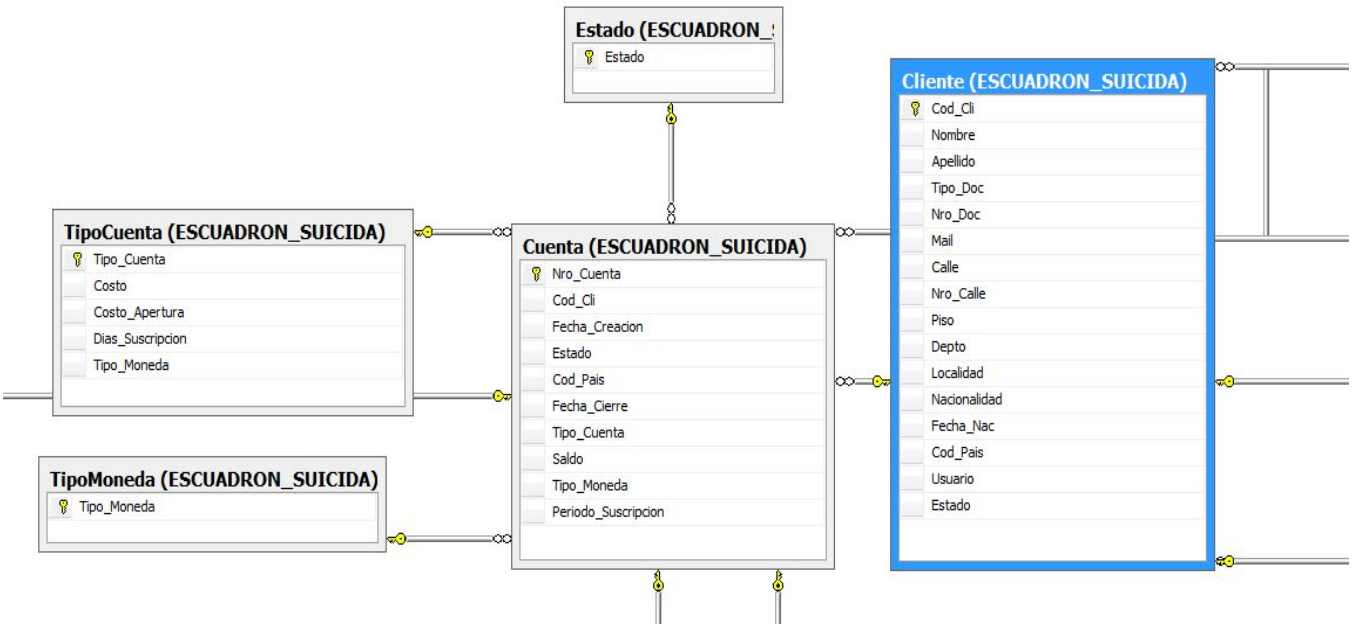
Para esta funcionalidad se creó la tabla Cliente y sus respectivas relaciones, se migraron los datos correspondientes de cada cliente, ya sean los datos personales como también se les asignó un código de cliente como identificador.
Para el alta de esta funcionalidad, se le pide el ingreso de los datos personales y una vez validado el hecho de que todos los campos estén llenos se insertan los datos en la tabla. En esta ventana en particular, en caso de ser administrador, habilitamos la opción de que él mismo genere un usuario y contraseña al cliente, controlando los datos de documentos duplicados y los mails.
También le asignamos un estado al cliente, para poder diferenciar , de acuerdo al estado, la posibilidad o no de realizar las distintas funcionalidades permitidas (transacciones, retiros de dinero o depósitos) y también para utilizarlo en caso de una baja lógica. Esto permite, poder volver habilitar al cliente desde un estado inhabilitado.
Hicimos un formulario que le permite al usuario filtrar una búsqueda por ciertos filtros establecidos en el enunciado, en donde se devuelven los resultados y se procede a la selección de alguno de ellos, ya sea para baja o modificación.
En el caso de las bajas, se actualiza el estado por medio de un update, y, en el caso de las modificaciones, se actualizan los mismos datos del cliente solicitados en el alta, pero modificados, por medio del update.
Vale destacar, que en las modificaciones del cliente, se habilitó la posibilidad de la asociación o desasociación de tarjetas por medio de botones que se configuraron para que el cliente sea derivado a esa funcionalidad y puede realizar la acción correspondiente. Esto permite también la modificación de la tarjeta de crédito.



5. ABM de Cuenta

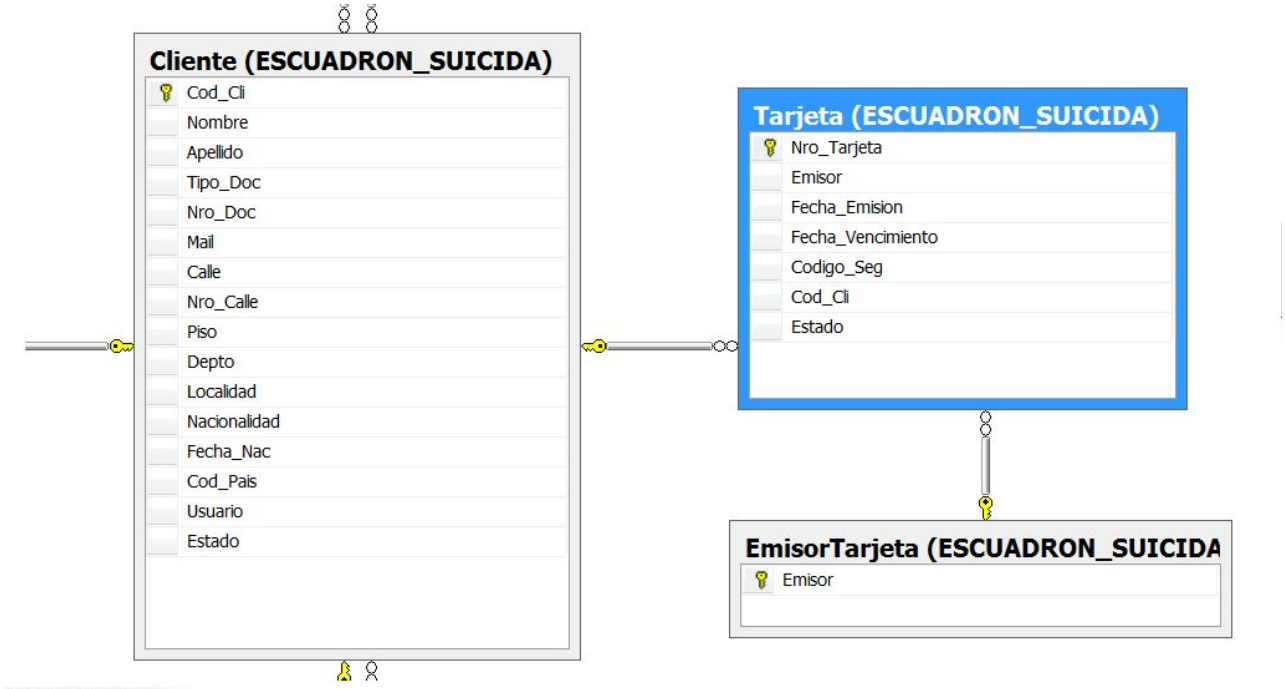
Para esta ABM, creamos la tabla Cuenta, migrando todas las cuentas existentes en el sistema.
Tuvimos que crear una tabla de países para poder guardar la información y a su vez, creamos también una tabla TipoCuenta con los diferentes tipos de cuenta con las que cuenta este sistema. A cada una de estas cuentas le asignamos la información que necesitaba para poder relacionarse con las distintas operaciones a nivel transaccional. Esto significa que ya nos decían que el tipo de moneda que se maneja era el dólar, pero que se podría agregar otro tipo, por eso creamos la tabla TipoMoneda.
También le asignamos (a cada tipo de cuenta) una duración en el tiempo en días (Dias_Suscripción) y un correspondiente costo de apertura y costo de transferencia. Estos valores optamos por asignarlos nosotros porque no nos daban información alguna sobre ellos.

Contemplamos que el cliente pueda tener más de una cuenta, por eso utilizamos una relación de uno a muchos entre las tablas Cuenta y Cliente.
Además, le agregamos y calculamos el saldo correspondiente a cada cuenta (dato que no existía, o no estaba explícitamente en la tabla Maestra).
Tomamos la decisión de que todas las cuentas migradas tenían la característica de ser de tipo Gratuitas ya interpretamos que como sus costos de transferencias eran de $0, cumplían esta condición por lo leído en el enunciado.
Vale aclarar que, además, decidimos que la apertura de una cuenta gratuita no quede pendiente de facturación ya que no tiene costo alguno por su apertura. Esto no se especificaba en el enunciado y lo decidimos de esa manera.
La tabla TipoCuenta nos permite que los distintos usuarios puedan mantener su condición de tipo de cuenta ó puedan cambiar el tipo de la misma a cuentas pagas.
A su vez, decidimos que los costos de transferencias de todas las cuentas sea 0 ya que no se especificaba nada acerca de ello y a fines prácticos los inicializamos en 0. En cambio, a los costos de apertura sí les seteamos diferentes valores de acuerdo al tipo de cuenta, con la excepción de que la cuenta gratuita tenga costo $0 de apertura.



6. Asociar/Desasociar Tarjetas de Crédito

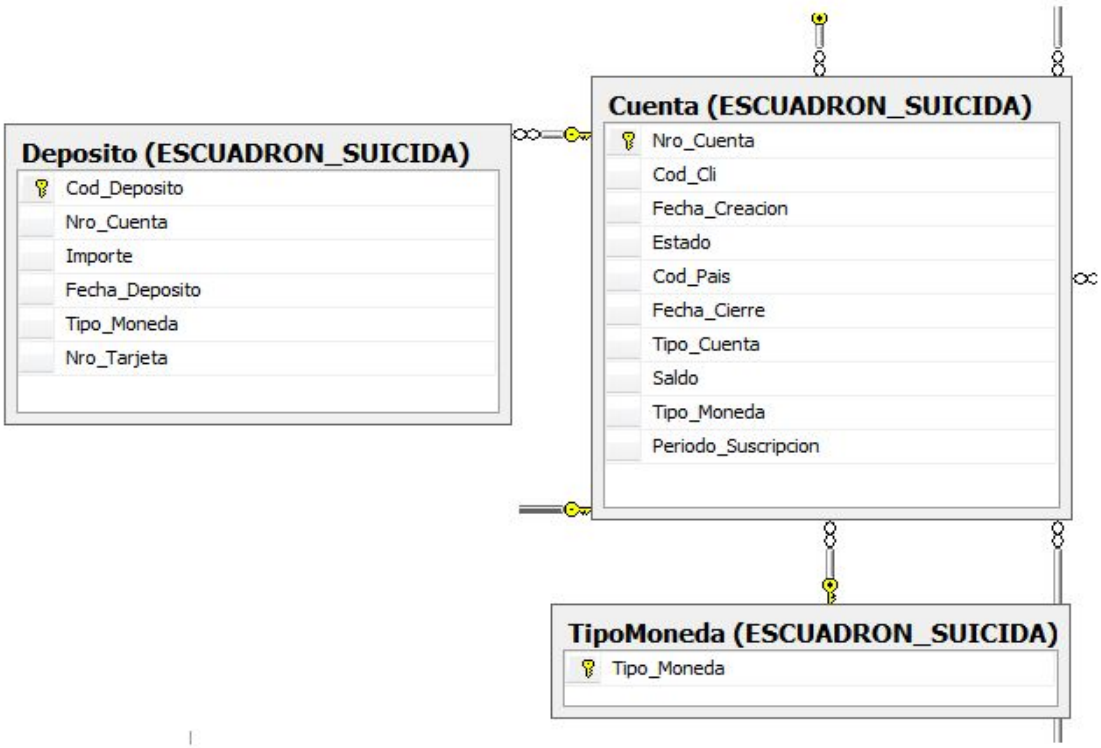
En esta sección, normalizamos las distintas tarjetas de crédito que tenía un cliente, registrando el número, emisor, fecha de emisión y vencimiento y su código de seguridad.
Se implementó un buscador para poder seleccionar las diferentes tarjetas que el cliente tiene asociadas por medio de un listado.
Esto permite la modificación o posterior desasociación de la misma.
Creamos también la entidad EmisorTarjeta que contiene los diferentes emisores de las tarjetas de crédito.



Esta funcionalidad está relacionada o, dicho de otra manera, incluida en la sección de modificación del cliente.

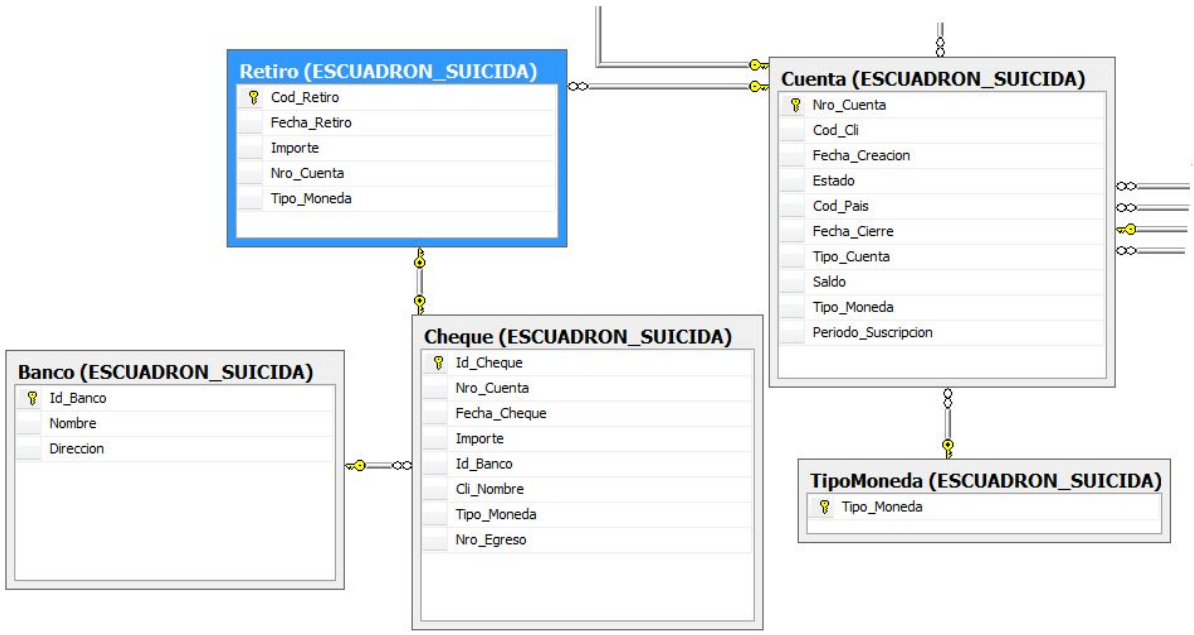
7. ABM de Depósitos

Para esta ABM, implementamos una tabla Depósitos la cual se identifica por el código de depósito como Clave Primaria y se relaciona con la tabla Cuenta por medio de Nro_Cuenta como Clave Foránea, el campo Tipo_Moneda proporciona mayor escalabilidad, ya que se contempla que en un futuro se puedan implementar otros tipos de moneda en el sistema. El ABM está diseñado de tal forma que cuando el usuario quiere realizar un depósito debe seleccionar una cuenta propia, que no se puede ingresar digito a digito si no que se deriva a una ventana de listado para poder buscar una cuenta por medio de distintos filtros. Finalmente se debe seleccionar una tarjeta de crédito de las que el cliente tenga asociadas (la cual se valida que no se encuentre vencida), se inserta el movimiento en la tabla depósitos y se actualiza el saldo en la tabla Cuenta.



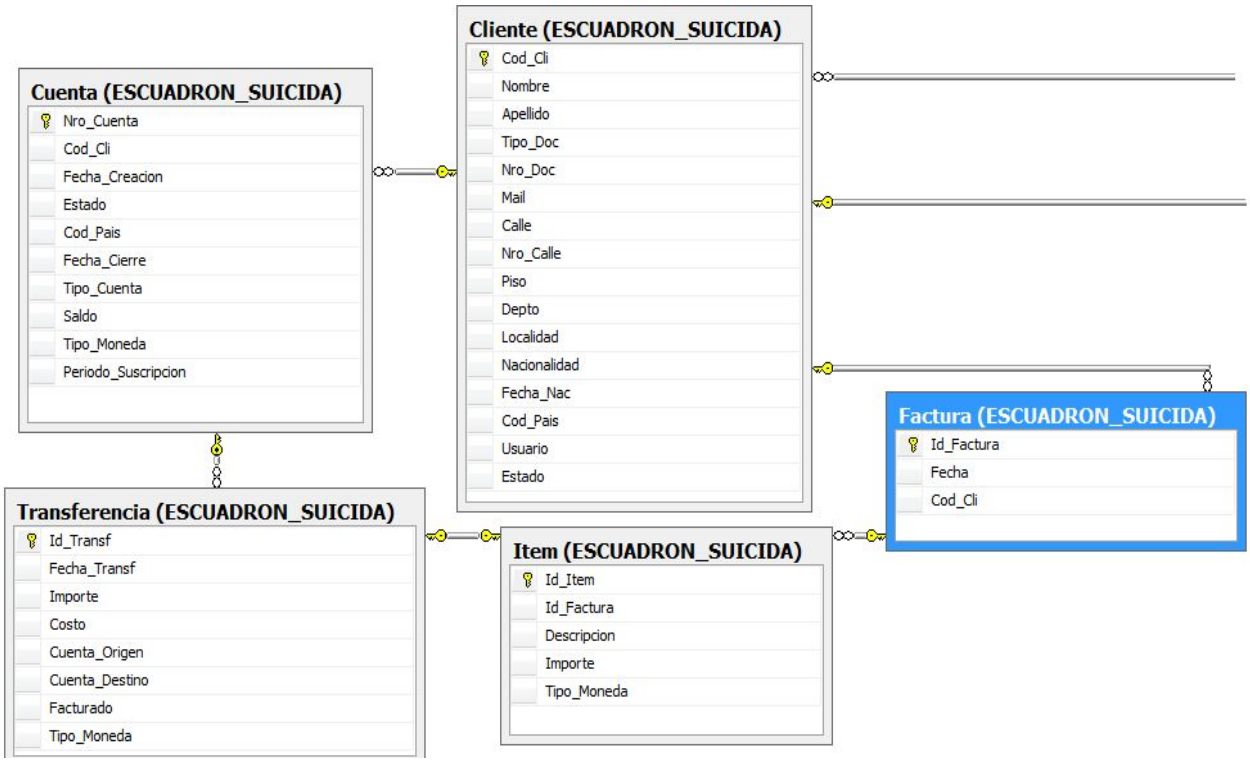
8. ABM de Retiros de Efectivo

Para dicha funcionalidad tuvimos que crear una tabla de Retiros la cual está relacionada con la tabla Cuenta, de esta forma se pueden corroborar las distintas validaciones que debe cumplir un cliente para realizar dicho retiro (que la Cuenta se encuentre habilitada, tener saldo, etc..)
Al momento de realizar el retiro no solo se debía registrar este, sino también generar un cheque con datos tanto propios del cliente (importe, cuenta) como también datos migrados desde la tabla maestra (código, fecha, e importe).
Por lo dicho anteriormente, se creó la entidad Cheque y se la relacionó con Retiro, teniendo en cuenta que un retiro genera un cheque único a ese retiro. Los relacionamos por medio de Id_Cheque / Nro_Egreso en ambas tablas.



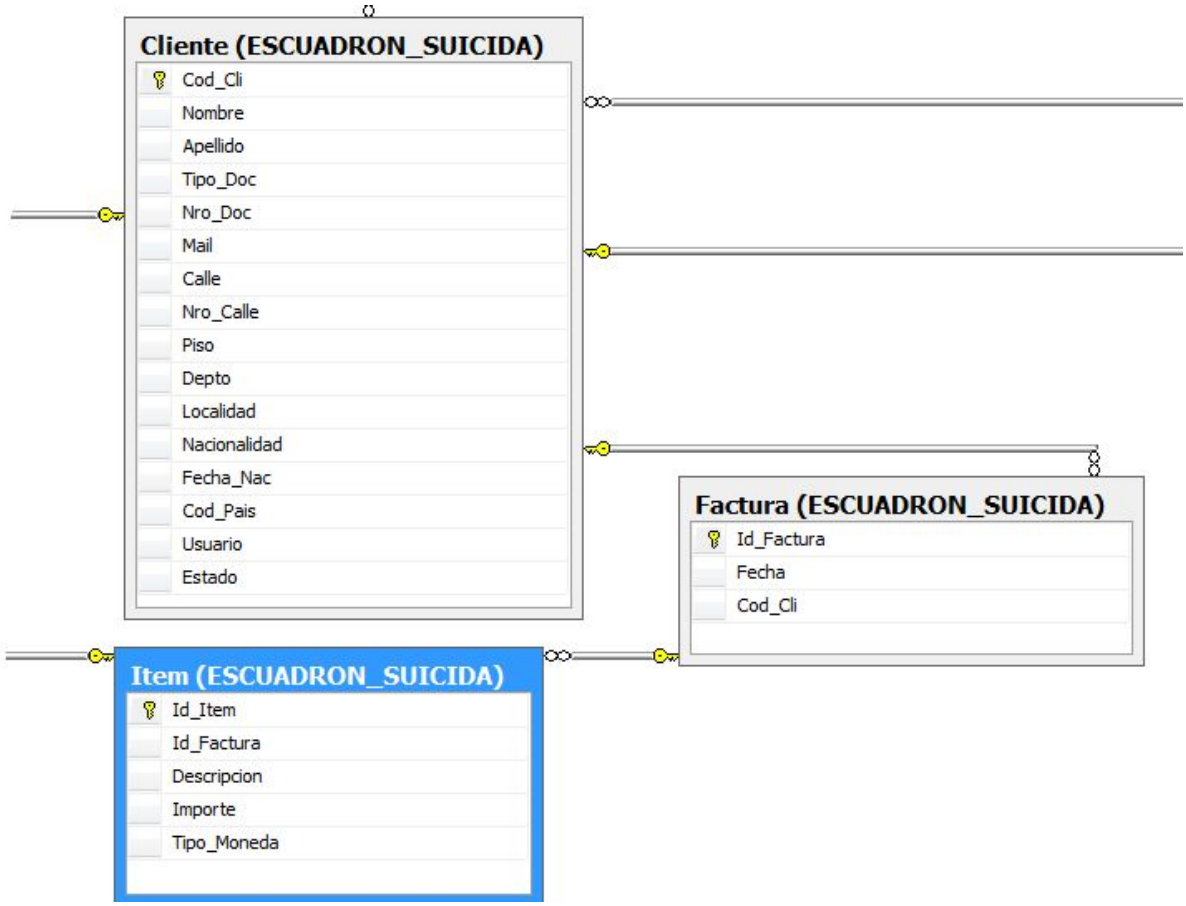
9. Transferencias entre Cuentas

Para esta ABM, creamos la tabla Transferencia relacionada con la tabla Cuenta e Item y migramos las transferencias existentes en el sistema.
Por cada transferencia registramos su número, fecha, importe, el costo de la misma, la cuenta origen y la cuenta destino.
Además agregamos un campo bit denominado Facturado para registrar si se facturó o no esa transferencia.
Esto nos permitió registrar el movimiento entre las cuentas.
A su vez, la relacionamos con Item con una relación uno a uno porque para que se correspondan la cantidad de ítems con la cantidad de transferencias, ya que éstas son movimientos que se facturan.
Por otro parte, las transferencias son las que tienen cargo por los cobros de las suscripciones de los tipos de cuenta, y también, para saber si se facturó o no la misma.
En la implementación de esta ABM, verificamos las restricciones en cuanto a los importes y los saldos de las cuentas, validando estos aspectos.
En cuanto al modelo, en la tabla Transferencia tomamos la decisión de registrar todo lo que es transacción pero referido a transferencia en sí misma, cambio de tipo de cuenta y apertura, entendiéndose que en los últimos dos casos generarían una transferencia de la misma cuenta como origen y destino y con el Importe en 0, teniendo en cuenta el costo correspondiente. Esto lo hicimos así por una cuestión de que el modelo que hicimos lo habían corregido como que estaba bien y no quisimos cambiar el mismo por un tema de que se nos ocurrió de esta manera. Tal vez se podría encarar mejor, pero decidimos esto ya que en el enunciado no se aclara nada.
Suponemos nosotros que una cuenta con estado Pendiente de activación no puede realizar transferencias a una cuenta, ya que el enunciado no aclara sobre esta situación.



10. Facturación de Costos

Para esta sección, creamos la tabla Factura y la tabla Item, estableciendo una relación entre ambas tablas por medio del Id_Factura, para poder relacionar todos los ítems ligados a una factura, que a su vez está relacionada con un Cliente.
En esta tabla, se almacenan todas las facturaciones de los clientes, de acuerdo a sus transacciones, lo que nos permite no sólo registrar de manera sencilla los movimientos de cada cliente sino también que resulte más eficiente consultar las facturaciones de cada uno.
Una vez que se facturan las facturaciones pendientes, generamos la factura correspondiente.



11. Consulta de Saldos

Esta funcionalidad la desarrollamos utilizando las tablas Retiros, Depósitos y Transferencias (ya que en ellas se registran sus respectivos movimientos), como todas están relacionadas con la tabla Cuenta, la consulta del saldo de la cuenta y las últimas transacciones, es una funcionalidad propia de la aplicación que las ejecutamos a través de las distintas consultas necesarias por medio del modelo ya creado.
La implementamos de manera que dado un usuario y una cuenta del mismo, nos informe el saldo de los últimos depósitos, transferencias o retiros, y, nos muestre un listado de los mismos con su debida información.

12. Listado Estadístico

Esta ABM se desarrolló de manera tal que sólo los administradores del sistema puedan utilizarla. La estructura final de las relaciones entre entidades (después de varias modificaciones) concluyó en un desarrollo el cual llevó más tiempo en el diseño estético de la misma que en pensar qué consultas emplear para que cumpla con su funcionalidad. Esto permite que quien utilice esta funcionalidad debe seleccionar

- El año que desee listar
- El trimestre del mismo
- Total facturado de acuerdo al tipo de cuenta

Una vez hecha la selección deseada, esta ABM permite mostrar los resultados pedidos en esta sección, filtrándose por dichas selecciones. Esta funcionalidad se basa en realizar distintas consultas a la base de datos generando las estadísticas correspondientes.

Conclusión:

El desarrollo de este TP de Pago Electrónico lo hicimos teniendo en cuenta los datos provistos por la cátedra de la tabla Maestra y los migramos al nuevo esquema de modelado, permitiendo así la extensión y mejora de este sistema ya existente, manteniendo los datos preexistentes.
A lo largo del proceso de migración, nos vimos obligados a la creación de diferentes tablas y normalización de las ya existentes, estableciendo distintas relaciones de acuerdo a este sistema.
Hubo que tener en cuenta ciertas contemplaciones y tomar algunas decisiones en cuanto a algunos aspectos que no estaban aclarados en el enunciado de este TP, ó sobre algunos aspectos que generaban

duda. Tratamos de justificar todas las decisiones tomadas en las distintas ABMs, en cada una de las secciones de esta estrategia.
Una vez migrado el sistema, nos dedicamos a las funcionalidades del mismo y su debida implementación utilizando el Visual Studio C# .NET 3.5.
Con esto, pudimos realizar las distintas ventanas para que el usuario final (ya sea un cliente o administrador) pueda interactuar con el sistema, a un nivel más bajo.
En conclusión, tratamos de realizar la ampliación de este sistema de pago electrónico de acuerdo a los requerimientos pedidos y los datos ya existentes, tomando ciertas decisiones en cuanto al desarrollo e implementación del mismo.

Aclaraciones varias:

- Decidimos que una vez inhabilitada una cuenta que posee un cliente no pueda dar de alta otra cuenta ya que tiene facturaciones pendientes, tampoco el administrador puede realizar esta acción en este caso.
- Para un usuario inhabilitado el administrador puede administrar/modificar las cuentas de ese usuario, ya que no depende de la inhabilitación de las cuentas por transacciones pendientes sino que, depende del logueo erróneo. Es por eso, que decidimos que el administrador pueda tener acceso en este caso para administrar las cuentas de ese usuario.
- Si una cuenta se encuentra cerrada, el cliente no puede realizar ninguna operación sobre la misma.
- Los costos de transferencias los seteamos en 0, 5, 10, 15 para las cuentas de tipo Gratuita, Bronce, Plata y Oro por una cuestión de que no se especificaba nada en el enunciado, pero los costos de apertura los asignamos con los valores 0, 10, 20,30 respectivamente.
- La tabla Transferencia la usamos para registrar todas las transferencias y también los distintos cambios de tipo de cuenta y

aperturas. En este punto queríamos explicar que lo dejamos así por una cuestión de que al tener bien el modelo en la corrección, no quisimos modificarlo. Es por esto, que se nos ocurrió mantener la estructura que teníamos y lo que hacemos para diferenciar una transferencia de una apertura o del cambio de tipo de cuenta, es que al realizar alguna de éstas dos últimas, definimos como cuenta origen y destino a la misma, asignándole un Importe de $0 y el costo correspondiente a su apertura o cambio de tipo de cuenta. Sabemos que seguramente se podría haber realizado de otra manera mejor, pero optamos por dejarlo así y explicar el por qué.

- En cuanto a las correcciones de las entregas anteriores, corregimos todo lo solicitado.