

ANÁLISIS ORGÁNICO

BANCO

Descripción.

Programa principal para manejar los ficheros e insertar los datos en la bases de datos.

Área de almacenamiento.

Se recibirá un fichero el cual habrá que tratar, el cual vendrá separado en registros de setenta y siete caracteres.

1. Área para identificar el tipo de registro leído. Vendrá dado por el código numérico de dos dígitos al principio de cada registro. Será un Switch.

2. Un área de variables para cada tipo de registro.

2.1. Registro tipo cliente. Contiene el código de dos dígitos, nombre de cincuenta y siete caracteres, fecha de nacimiento de ocho dígitos (año, mes, día) y NIF de diez caracteres.

2.2. Registro tipo domicilio. Contiene código de dos dígitos, calle de treinta y cinco caracteres, número de tres caracteres que puede no constar, 'S/N', código postal de cinco dígitos, población y provincia de dieciséis caracteres respectivamente.

2.3. Registro tipo cuenta y tarjeta. Contiene el código de dos dígitos, número de cuenta de veinte dígitos, saldo; de ocho dígitos, signo, y dos decimales. Tipo de relación que tiene el cliente respecto a la cuenta(T=TITULAR, C=COTITULAR, A=AUTORIZADO) que tendrá un carácter, número de tarjeta de dieciséis dígitos, crédito; de ocho dígitos, signo, y dos decimales. Y, finalmente, fecha de expiración de tarjeta de cuatro dígitos representando como mes y año.

2.4. Registro tipo movimiento. Contiene código de dos dígitos, concepto de veintitrés caracteres, número de cuenta o tarjeta con la que se ha realizado, que tendrá máximo veinte dígitos, medio de pago que puede ser tarjeta o cuenta(T o C), de un carácter, importe; de ocho dígitos, signo, y dos decimales. Y, finalmente, fecha de realización de veinte dígitos que corresponde a un timestamp.

ANÁLISIS ORGÁNICO

3. Área para guardar el cliente completo. Dado que al leer un registro se perderán los datos del anterior, se creará un área para guardar todos los registros relacionados a un cliente. Posteriormente se imprimirá e insertará en la BBDD cuando se haya dectado que se ha transicionado al siguiente cliente. Esta estructura de guardado tendrá en cuenta la limitación de 5 cuentas, tarjetas o movimientos por cliente.

4. Variables auxiliares para formatear números de cuentas, tarjetas y fechas. Para una mejor lectura cuando se quiera mostrar estos datos, se dividirán las fechas por barras y los números de cuenta y tarjeta por espacio o guiones. En el caso de cuenta, agrupando los números en el siguiente orden: entidad (cuatro dígitos), oficina (cuatro dígitos), dígito de control (dos dígitos), producto o folio (cuatro dígitos) y cuenta (seis dígitos). Las fechas en formato DD/MM/AAAA, o MM/AA, dependiendo de nuestros datos de entrada.

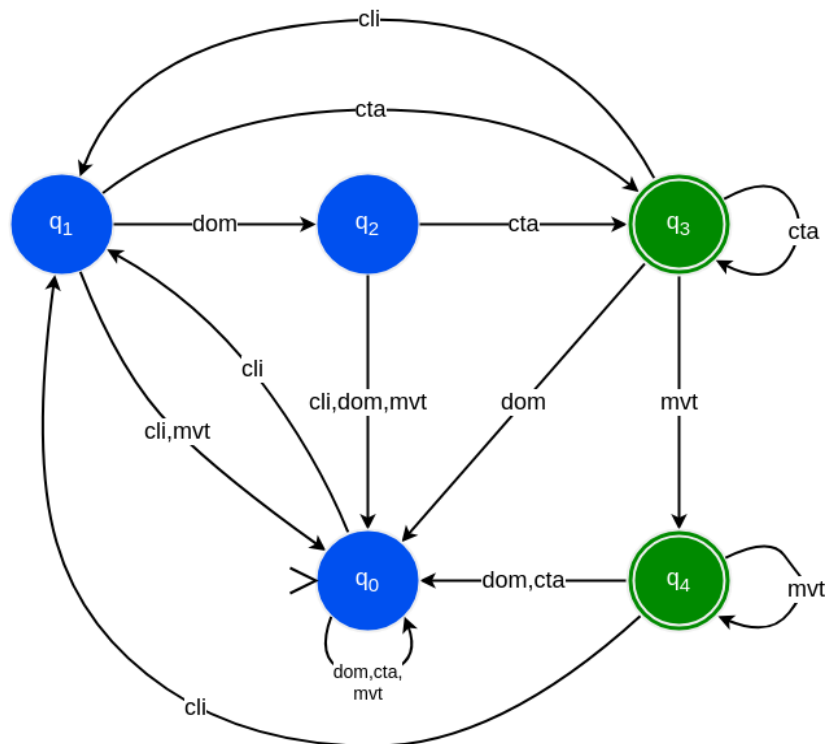
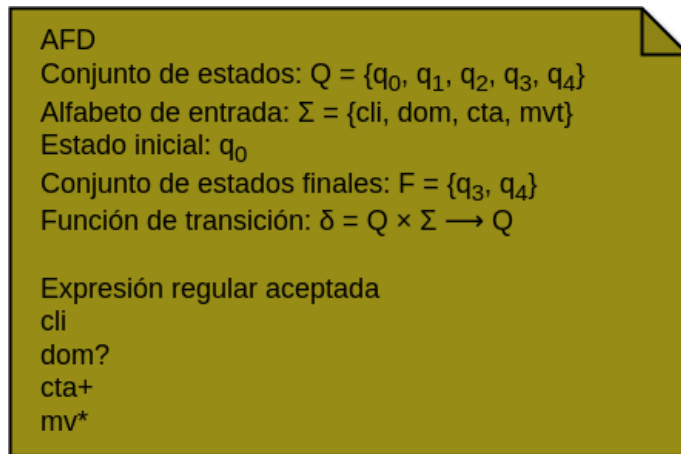
5. Un área switch que indique el estado de la operación con los posibles errores, incluyendo si todo se ha desarrollado correctamente.

6. Un área para todos los mensajes de errores que enlazarán con sus respectivos códigos de error usando los valores de estos como índice en un array.

7. Se hará uso de un automata finito determinista para transitar correctamente por los registros, identificar cuando se ha pasado de un cliente al siguiente, y solo procesar los registros que lleguen en el orden que requieren las especificaciones.

ANÁLISIS ORGÁNICO

Para ello se creará un switch con los distintos estados del AFD, especificando también si se ha alcanzado un estado final o no.



ANÁLISIS ORGÁNICO

Proceso.

El programa principal cargará todos los registros del fichero de entrada en la base de datos.

Para ello se estructurará de la siguiente forma:

1. Inicialización de variables.

2. Abrir fichero.

3. Leer fichero.

4. Procesar los distintos tipos de registro hasta fin de fichero o la ocurrencia de un error.

4.1. Se querrá leer los registros en orden secuencial clasificados por sus códigos: 01, 02, 03 y 04. El AFD determinará los distintos estados en los que se encuentra la lectura. Estos estados cambiarán dependiendo del tipo de registro que leamos seguidamente.

Entre la transición de estados se llevarán a cabo, o no, una serie de acciones:

4.1.1. Validar datos de entrada. Para todos los registros se deberá asegurar que los datos vienen informados y son válidos. En caso contrario, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

4.1.2. Cargar los datos del registro leído a su estructura correspondiente para guardarlos. En el caso de cuentas, tarjetas o movimientos se asegurará que no sobrepasen la cantidad permitida antes de añadirse. En el caso de las cuentas y tarjetas, además, tampoco podrán venir repetidas para un mismo cliente.

4.1.3. Imprimir los datos de cliente y cargarlos en la BBDD. Formatear las de cuentas, tarjetas y fechas para para la impresión. Formatear también la relación que tiene el cliente con la cuenta para que en vez de una 'T', aparezca como 'Titular' y así con las demás relaciones. Si seguidamente se tuviera que mostrar un cliente nuevo, se imprimirá también una línea divisoria entre ambos.

ANÁLISIS ORGÁNICO

CONEXMDB

Descripción.

Rutina para la conexión a base de datos.

Entrada.

1. La conexión podrá estar abierta o cerrada.
2. Los datos del data source, que será un alfanumérico de cincuenta caracteres.

Salida.

1. Estado de la operación.
 - 1.1. Resultado OK.
 - 1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.
2. Sqlcode.

Proceso.

1. Validar estado de la conexión.
2. Si la conexión es correcta validar data source en caso de querer conectar, en caso contrario, desconectar.
3. Para la conexión o desconexión se usarán las siguientes consultas sql:

```
CONNECT TO :HV-DATASOURCE
```

```
DISCONNECT
```

BUSQCLIE

Descripción.

Rutina para la búsqueda de clientes.

ANÁLISIS ORGÁNICO

Entrada.

1. Tipo de criterio de búsqueda por identificación o por nif. Será un switch.
2. Valores del filtro. El identificador será un valor numérico de diez dígitos y el nif será un valor alfabético de diez caracteres.

Salida.

1. Estado de la operación.
 - 1.1. Resultado OK, encontrado, o no encontrado.
 - 1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.
2. Sqlcode.
3. Datos del cliente. Se devolverán todas las columnas de la tabla.
 - 3.1. El id del cliente. Número de diez caracteres.
 - 3.2. Nombre del cliente. Será un valor alfabético de cuarenta y cinco caracteres.
 - 3.3. La fecha de nacimiento será un valor numérico de ocho dígitos que representarán una fecha en formato año, mes y día.
 - 3.4. El NIF. Será un valor alfabético de diez caracteres.
 - 3.5. el ID de domicilio. Valor opcional numérico diez dígitos

Proceso.

1. Validar los datos de entrada: el criterio y los valores del filtro. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Si la búsqueda es por ID, se realizará la siguiente consulta sql:

```
SELECT
    id_cliente,
    nif_cliente,
    nom_cliente,
    fec_nac_cliente,
    id_dom
INTO
    :CLIENTE-ID      :CLIENTE-ID-NULL,
    :CLIENTE-NIF     :CLIENTE-NIF-NULL,
    :CLIENTE-NOM     :CLIENTE-NOM-NULL,
    :CLIENTE-FEC-NAC :CLIENTE-FEC-NAC-NULL,
    :CLIENTE-ID-DOM  :CLIENTE-ID-DOM-NULL
FROM banco.cliente
WHERE id_cliente = :CLIENTE-ID
```

ANÁLISIS ORGÁNICO

3. Si la búsqueda es por NIF se realizará la siguiente consulta sql:

```
SELECT
    id_cliente,
    nif_cliente,
    nom_cliente,
    fec_nac_cliente,
    id_dom
INTO
    :CLIENTE-ID      :CLIENTE-ID-NULL,
    :CLIENTE-NIF     :CLIENTE-NIF-NULL,
    :CLIENTE-NOM     :CLIENTE-NOM-NULL,
    :CLIENTE-FEC-NAC :CLIENTE-FEC-NAC-NULL,
    :CLIENTE-ID-DOM  :CLIENTE-ID-DOM-NULL
FROM banco.cliente
WHERE nif_cliente = :CLIENTE-NIF
```

BUSQDOM

Descripción.

Rutina para la búsqueda de domicilios.

Entrada.

1. Tipo de criterio de búsqueda por identificación o por combinación de calle, número y código postal. Será un switch.

2. Valores del filtro. El identificador será un valor numérico de diez dígitos, la calle un valor alfabético de máximo treinta y cinco caracteres, el número alfanumérico de 3 dígitos o S/N, y el código postal de cinco dígitos.

Salida.

1. Estado de la operación.

1.1. Resultado OK, encontrado, o no encontrado.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos del domicilio. Se devolverán todas las columnas de la tabla.

3.1. El id del domicilio. Número de diez caracteres.

ANÁLISIS ORGÁNICO

3.2. Calle. Será un valor alfabético de treinta y cinco caracteres.

3.3. El número del domicilio que será un número de tres dígitos o S/N.

3.4. El código postal. Será un valor numérico de cinco dígitos.

3.5. La provincia, un valor alfabético de dieciséis caracteres.

3.6. La población, un valor alfabético de dieciséis caracteres.

Proceso.

1. Validar los datos de entrada: el criterio y los valores del filtro. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Si la búsqueda es por ID, se realizará la siguiente consulta sql:

```
SELECT
id_dom,
calle_dom,
num_dom,
lpad(cod_post_dom,5,'0'),
prov_dom,
pobl_dom
INTO
:DOMICILIO-ID      :DOMICILIO-ID-NULL,
:DOMICILIO-CALLE   :DOMICILIO-CALLE-NULL,
:DOMICILIO-NUM     :DOMICILIO-NUM-NULL,
:DOMICILIO-COD-POS :DOMICILIO-COD-POS-NULL,
:DOMICILIO-PROV    :DOMICILIO-PROV-NULL,
:DOMICILIO-POBL    :DOMICILIO-POBL-NULL
```


ANÁLISIS ORGÁNICO

```
FROM banco.domicilio
WHERE id_dom =      :DOMICILIO-ID
```

3. Si la búsqueda es por la combinación de calle, número y código postal se realizará la siguiente consulta sql:

```
SELECT
id_dom,
calle_dom,
num_dom,
lpad(cod_post_dom,5,'0'),
prov_dom,
pobl_dom
INTO
:DOMICILIO-ID      :DOMICILIO-ID-NULL,
:DOMICILIO-CALLE   :DOMICILIO-CALLE-NULL,
:DOMICILIO-NUM      :DOMICILIO-NUM-NULL,
:DOMICILIO-COD-POS  :DOMICILIO-COD-POS-NULL,
:DOMICILIO-PROV     :DOMICILIO-PROV-NULL,
:DOMICILIO-POBL     :DOMICILIO-POBL-NULL
FROM banco.domicilio
WHERE calle_dom =   :DOMICILIO-CALLE
AND IFNULL(num_dom,'@') = :DOMICILIO-NUM
AND cod_post_dom = :DOMICILIO-COD-POS
```

BUSQCTA

Descripción.

Rutina para la búsqueda de cuentas.

Entrada.

1. Tipo de criterio de búsqueda por identificación o por número de cuenta.

2. Valores del filtro. El identificador será un valor numérico de diez dígitos, el número de cuenta un dígito con un tamaño de veinte.

Salida.

ANÁLISIS ORGÁNICO

1. Estado de la operación.

1.1. Resultado OK, encontrado, o no encontrado.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos de la cuenta. Se devolverán todas las columnas de la tabla.

3.1. El id de la cuenta. Número de diez caracteres.

3.2. El número de cuenta. Número de veinte caracteres.

3.3. El saldo de la cuenta. Número con signo, ocho enteros y dos decimales.

3.4. El NIF del cliente al que pertenece la cuenta.

3.5. La relación entre el cliente y la cuenta.

Proceso.

1. Validar los datos de entrada: el criterio y los valores del filtro. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Si la búsqueda es por ID, se realizará la siguiente consulta sql:

```
SELECT
id_medio,
num_cuenta,
saldo_cuenta
INTO
:CUENTA-ID :CUENTA-ID-NULL,
:CUENTA-NUM :CUENTA-NUM-NULL,
:CUENTA-SALDO :CUENTA-SALDO-NULL,
FROM banco.cuenta
WHERE id_medio = :CUENTA-ID
```

3. Si la búsqueda es por número de cuenta, se realizará la siguiente consulta sql:

```
SELECT
id_medio,
num_cuenta,
saldo_cuenta,
INTO
:CUENTA-ID :CUENTA-ID-NULL,
```

ANÁLISIS ORGÁNICO

```
:CUENTA-NUM      :CUENTA-NUM-NULL,  
:CUENTA-SALDO    :CUENTA-SALDO-NULL,  
FROM banco.cuenta  
WHERE num_cuenta = :CUENTA-NUM
```

BUSQTAR

Descripción.

Rutina para la búsqueda de tarjetas.

Entrada.

1. Tipo de criterio de búsqueda por identificación o por número de tarjeta.
2. Valores del filtro. El identificador será un valor numérico de diez dígitos, el número de tarjeta un dígito con un tamaño de dieciséis.

Salida.

1. Estado de la operación.
 - 1.1. Resultado OK, encontrado, o no encontrado.
 - 1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.
2. Sqlcode.
3. Datos de la tarjeta. Se devolverán todas las columnas de la tabla.
 - 3.1. El id de la tarjeta. Número de diez caracteres.
 - 3.2. El número de tarjeta. Número de dieciséis caracteres.
 - 3.3. El crédito de la tarjeta. Número con signo, ocho enteros y dos decimales.
 - 3.4. El NIF del cliente al que pertenece la tarjeta.
 - 3.5. La cuenta a la que pertenece la tarjeta.

Proceso.

1. Validar los datos de entrada: el criterio y los valores del filtro. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

ANÁLISIS ORGÁNICO

2. Si la búsqueda es por ID, se realizará la siguiente consulta sql:

```
SELECT
id_medio,
id_medio_cta,
id_cliente,
num_tarjeta,
cred_tarjeta,
date_format(fec_tarjeta, '%d%m%y'),
ccv_tarjeta
INTO
:TARJETA-ID-TAR :TARJETA-ID-TAR-NULL,
:TARJETA-ID-CTA :TARJETA-ID-CTA-NULL,
:TARJETA-ID-CLI :TARJETA-ID-CLI-NULL,
:TARJETA-NUM :TARJETA-NUM-NULL,
:TARJETA-CRED :TARJETA-CRED-NULL,
:TARJETA-FEC :TARJETA-FEC-NULL,
:TARJETA-CCV :TARJETA-CCV-NULL
FROM banco.tarjeta
WHERE id_medio = :TARJETA-ID-TAR
```

3. Si la búsqueda es por número de tarjeta, se realizará la siguiente consulta sql:

```
SELECT
id_medio,
id_medio_cta,
id_cliente,
num_tarjeta,
cred_tarjeta,
date_format(fec_tarjeta, '%d%m%y'),
ccv_tarjeta
INTO
:TARJETA-ID-TAR :TARJETA-ID-TAR-NULL,
:TARJETA-ID-CTA :TARJETA-ID-CTA-NULL,
:TARJETA-ID-CLI :TARJETA-ID-CLI-NULL,
:TARJETA-NUM :TARJETA-NUM-NULL,
:TARJETA-CRED :TARJETA-CRED-NULL,
:TARJETA-FEC :TARJETA-FEC-NULL,
:TARJETA-CCV :TARJETA-CCV-NULL
FROM banco.tarjeta
WHERE num_tarjeta = :TARJETA-NUM
```

ANÁLISIS ORGÁNICO

BUSQMOV

Descripción.

Rutina para la búsqueda de movimientos.

Entrada.

1. Tipo de criterio de búsqueda por identificación o por número de tarjeta.
2. Valores del filtro. El identificador será un valor numérico de diez dígitos, la fecha será un valor numérico de ocho caracteres.

Salida.

1. Estado de la operación.
 - 1.1. Resultado OK, encontrado, o no encontrado.
 - 1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.
2. Sqlcode.
3. Datos del movimiento. Se devolverán todas las columnas de la tabla.
 - 3.1. El id del movimiento. Número de diez caracteres.
 - 3.2. El concepto del movimiento. Un alfabético de cuarenta y nueve caracteres.
 - 3.3. El importe del movimiento. Número con signo, ocho enteros y dos decimales.
 - 3.4. La fecha a la que se realizó el movimiento.

Proceso.

1. Validar los datos de entrada: el criterio y los valores del filtro. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. La búsqueda será por ID y por fecha. Se realizará la siguiente consulta sql:

```
SELECT
id_medio,
date_format(fec_mov,'%Y%m%d%H%i%s%f'),
cnpt_mov,
importe_mov
```

ANÁLISIS ORGÁNICO

```
INTO
:MOVIMIENTO-ID      :MOVIMIENTO-ID-NULL,
:MOVIMIENTO-FEC     :MOVIMIENTO-FEC-NULL,
:MOVIMIENTO-CPT     :MOVIMIENTO-CPT-NULL,
:MOVIMIENTO-IMPT    :MOVIMIENTO-IMPT-NULL
FROM banco.movimiento
WHERE id_medio = :MOVIMIENTO-ID
AND fec_mov =
str_to_date(:MOVIMIENTO-FEC,'%Y%m%d%H%i%s%f')
```

INSCLI

Descripción.

Rutina para la inserción de clientes.

Entrada.

1. El NIF, un alfanumérico de diez caracteres. El nombre, un alfabético de cincuenta y siete caracteres máximo. La fecha de nacimiento, un numérico de ocho caracteres y el id de domicilio de diez caracteres, en caso de tener un domicilio asociado.

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos del movimiento. Se devolverá el ID creado para el cliente que se ha insertado.

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

ANÁLISIS ORGÁNICO

2. Generar el ID con la siguiente consulta sql:

```
SELECT NEXT VALUE FOR banco.cliente_id_cliente_seq  
INTO :CLIENTE-ID :CLIENTE-ID-NULL
```

3. Insertar el cliente en la BBDD realizando la siguiente consulta sql:

```
INSERT INTO banco.cliente(  
id_cliente,  
nif_cliente,  
nom_cliente,  
fec_nac_cliente,  
id_dom  
)  
VALUES(  
:CLIENTE-ID          :CLIENTE-ID-NULL,  
:CLIENTE-NIF         :CLIENTE-NIF-NULL,  
:CLIENTE-NOM         :CLIENTE-NOM-NULL,  
:CLIENTE-FEC-NAC     :CLIENTE-FEC-NAC-NULL,  
:CLIENTE-ID-DOM      :CLIENTE-ID-DOM-NULL  
)
```

INSDOM

Entrada.

1. Se insertarán la calle, un alfabético de treinta y cinco caracteres máximo. El número, un alfanumérico de tres caracteres. El código postal, un numérico de cinco caracteres. La provincia y población de dieciséis caracteres respectivamente.

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos del movimiento. Se devolverá el ID creado para el domicilio que se ha insertado.

ANÁLISIS ORGÁNICO

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Generar el ID con la siguiente consulta sql:

```
SELECT NEXT VALUE FOR banco.domicilio_id_dom_seq  
INTO :DOMICILIO-ID :DOMICILIO-ID-NULL
```

3. Insertar el domicilio en la BBDD realizando la siguiente consulta sql:

```
INSERT INTO banco.cliente(  
id_dom,  
calle_dom,  
num_dom,  
cod_post_dom,  
prov_dom,  
pobl_dom  
)  
VALUES(  
:DOMICILIO-ID          :DOMICILIO-ID-NULL,  
:DOMICILIO-CALLE       :DOMICILIO-CALLE-NULL,  
:DOMICILIO-NUM         :DOMICILIO-NUM-NULL,  
:DOMICILIO-COD-POS     :DOMICILIO-COD-POS-NULL,  
:DOMICILIO-PROV        :DOMICILIO-PROV-NULL,  
:DOMICILIO-POBL        :DOMICILIO-POBL-NULL  
)
```

INSCTA

Descripción.

Rutina para la inserción de cuentas.

Entrada.

1. Se insertarán el número, un numérico de veinte caracteres. El saldo, un numérico con signo, ocho enteros y dos decimales.

ANÁLISIS ORGÁNICO

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos del movimiento. Se devolverá el ID creado para la cuenta que se ha insertado.

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Generar el ID con la siguiente consulta sql:

```
SELECT NEXT VALUE FOR banco.medio_id_medio_seq  
INTO :CUENTA-ID :CUENTA-ID-NULL
```

3. Insertar la cuenta en la BBDD realizando las siguientes consultas sql:

```
INSERT INTO banco.medio(  
id_medio,  
tip_med  
)  
VALUES(  
:CUENTA-ID :CUENTA-ID-NULL,  
'CTA'  
)
```

```
INSERT INTO banco.cuenta(  
id_medio,  
num_cuenta,  
saldo_cuenta  
)  
VALUES(  
:CUENTA-ID :CUENTA-ID-NULL,  
:CUENTA-NUM :CUENTA-NUM-NULL,  
:CUENTA-SALDO :CUENTA-SALDO-NULL
```

ANÁLISIS ORGÁNICO

)

Se necesita insertar primeramente en medio, que puede ser de dos tipos: cuenta o tarjeta. Seguidamente ya podemos insertar en la tabla cuenta.

INSTAR

Descripción.

Rutina para la inserción de tarjetas.

Entrada.

1. Se insertará el id de cliente un numérico de dieciséis caracteres. El id de la cuenta a la que se vincula la misma, un numérico de dieciséis caracteres. El número de la tarjeta, un numérico de dieciséis caracteres. El crédito, un numérico con signo, ocho enteros y dos decimales. La fecha de expiración que será un numérico de 4 carácter. Y el CCV que será un alfabético de tres caracteres.

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

3. Datos del movimiento. Se devolverá el ID creado para la tarjeta que se ha insertado.

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Generar el ID con la siguiente consulta sql:

```
SELECT NEXT VALUE FOR banco.medio_id_medio_seq
```

ANÁLISIS ORGÁNICO

INTO :TARJETA-ID-TAR :TARJETA-ID-TAR-NULL

3. Insertar la tarjeta en la BBDD realizando las siguientes consultas sql:

```
INSERT INTO banco.medio(  
id_medio,  
tip_med  
)  
VALUES(  
:TARJETA-ID-TAR :TARJETA-ID-TAR-NULL,  
'TRJ'  
)
```

```
INSERT INTO banco.tarjeta(  
id_medio,  
id_cliente,  
id_medio_cta,  
num_tarjeta,  
cred_tarjeta,  
fec_tarjeta,  
ccv_tarjeta  
)  
VALUES(  
:TARJETA-ID-TAR :TARJETA-ID-TAR-NULL,  
:TARJETA-ID-CLI :TARJETA-ID-CLI-NULL,  
:TARJETA-ID-CTA :TARJETA-ID-CTA-NULL,  
:TARJETA-NUM :TARJETA-NUM-NULL,  
:TARJETA-CRED :TARJETA-CRED-NULL,  
str_to_date(:TARJETA-FEC-X :TARJETA-FEC-NULL, '%d%m%y'),  
:TARJETA-CCV :TARJETA-CCV-NULL  
)
```

Se necesita insertar primeramente en medio, que puede ser de dos tipos: cuenta o tarjeta. Seguidamente ya podemos insertar en la tabla tarjeta.

INSMOV

Descripción.

Rutina para la inserción de movimientos.

ANÁLISIS ORGÁNICO

Entrada.

1. Se insertará el id de movimiento, un numérico de diez caracteres. La fecha que será un alfabético de veinte caracteres. El concepto que será un alfanumérico de cuarenta y nueve caracteres. El importe que tendrá signo, un entero de ocho caracteres y dos decimales.

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Generar el ID con la siguiente consulta sql:

```
SELECT NEXT VALUE FOR banco.medio_id_medio_seq  
INTO :TARJETA-ID-TAR :TARJETA-ID-TAR-NULL
```

3. Insertar el movimiento en la BBDD realizando la siguientes consulta sql:

```
INSERT INTO banco.movimiento(  
id_medio,  
fec_mov,  
cnpt_mov,  
importe_mov  
)  
VALUES(  
:MOVIMIENTO-ID :MOVIMIENTO-ID-NULL,  
str_to_date(:MOVIMIENTO-FEC, '%Y%m%d%H%i%S%f')  
:MOVIMIENTO-FEC-NULL,  
:MOVIMIENTO-CPT :MOVIMIENTO-CPT-NULL,  
:MOVIMIENTO-IMPT :MOVIMIENTO-IMPT-NULL
```

ANÁLISIS ORGÁNICO

)

INSRLN

Descripción.

Rutina para la inserción de relaciones entre cuentas y clientes.

Entrada.

1. Se insertará el id de cliente, un numérico de diez caracteres. El id de cuenta, un numérico de diez caracteres. El tipo de relación que podrá ser: Titular, cotitular o autorizado.

Salida.

1. Estado de la operación.

1.1. Resultado OK.

1.2. Código de error: que indique si ha habido un error de validación de datos de entrada, o si ha habido un error de sql.

2. Sqlcode.

Proceso.

1. Validar los datos de entrada: Los datos para realizar la inserción. En caso de que algún dato sea erróneo o no informado, se abortará la ejecución y se retornará en el área de salida el código de error correspondiente.

2. Insertar la relación en la BBDD realizando la siguiente consulta sql:

```
INSERT INTO banco.cliente_rln_cuenta(  
id_cliente,  
id_medio,  
tip_rln  
)  
VALUES(  
:RELACION-CTA-ID :RELACION-CLI-ID-NULL,  
:RELACION-CLI-ID :RELACION-CTA-ID-NULL,  
:RELACION-RLN :RELACION-RLN-NULL  
)
```

ANÁLISIS ORGÁNICO

PRBBANCO

Descripción.

Programa para probar rutinas de búsqueda e inserción.

Entrada.

1. Se hará uso de todas las rutinas de búsqueda e inserción de datos.
2. Datos de datasource que se pasará a la rutina de conexión a la que también se llamará en el programa.
3. Se introducirán datos para la comprobación de todos los errores posibles que puedan encontrarse en las rutinas.

Proceso.

1. Conectar o desconectar de BBDD llamando a la rutina **conexmdb**.
2. Probar para cada rutina de búsqueda e inserción todos los casos de error que puedan darse.

ALTACLIENTE

Descripción.

Programa para insertar todos los datos tratados por el programa **banco** en la BBDD haciendo uso de las rutinas previamente descritas.

Entrada.

1. Se hará uso de las rutinas de búsqueda a inserción usando datos trasladados del programa **banco** al area de altacliente.
2. Datos de datasource que se pasará a la rutina de conexión a la que también se llamará en el programa.

Salida.

1. Todos los stados posible de errores y estado ok.

ANÁLISIS ORGÁNICO

2. SQLCODE.

Proceso.

1. Conectar a BBDD llamando a la rutina **conexmdb**.

2. Si el domicilio viene informado, se busca. Si existe se guarda su id, y si no, se inserta y se guarda su id.

3. Se busca el cliente y se existiera, se propaga el error correspondiente, si no, se inserta. Se guarda el id.

4. Se buscan todas las cuentas y si existen, se guardan sus id. De lo contrario, se insertan y se guardan sus id.

5. Se busca la relación entre cliente y cuenta. Si existe se propaga el error correspondiente. De lo contrario, se inserta.

6. Se buscan las tarjeta. Si existen se propaga el error correspondiente. De lo contrario, se insertan. Guardamos los id.

7. Se buscan los movimientos. Si existen se propaga el error correspondiente. De lo contrario, se insertan. Si el medio por el que se ha hecho el movimiento es una cuenta, se actualiza el saldo de la misma. Si el movimiento se ha realizado a través de una tarjeta, se actualiza el crédito.

8. Si los datos para un cliente son todos correctos, se hace COMMIT de las operaciones, en caso de encontrarse un error, se aborta el programa y se hace ROLLBACK.

PRBALTACLIENTE

Descripción.

Programa para probar la rutina que recibe los datos del fichero tratados por el programa **banco** e invoca a las rutinas de búsqueda e inserción.

Entrada.

1. Se hará uso de la rutina **altacliente**.

2. Datos de datasource que se pasará a la rutina de conexión a la que también se llamará en el programa.

ANÁLISIS ORGÁNICO

3. Se introducirán datos para la comprobación de todos los errores posibles que puedan encontrarse en las rutinas invocadas por **altacliente**. Se comprobará que los códigos de error se propagan correctamente a los errores del area de **altacliente**.

Transacción.

- 1.** Conectar o desconectar de BBDD llamando a la rutina **conexmdb**.
- 2.** Probar con datos cada caso de error de **altacliente**.