

Trabajo Práctico de Gestión de Datos

Cuponete utn

2C 2012

Grupo de Trabajo Práctico: MR_ANDERSON

Integrantes:

- Aldana Laura Quintana Munilla
- Rodrigo Julián Candurra
- Matías David Choren
- Ezequiel Darío Gambaccini

Link a la documentación: (Por si hubiera algún problema de visualización)

https://docs.google.com/document/d/187W6i3QlXk35dMwA2ihYlbg1n_UFnanfrmC3nVGHEN0/edit

Aplicacion C#

La aplicación desktop está separada en 2 capas, una capa que es la UI, y otra capa de dominio en la que se encuentran los Adapter y la parte de comunicación con la DB. Los adapter se encargan de hacer de intermediario entre los datos obtenidos de la base de datos y la UI. Además, hacen el proceso inverso, el cual es persistir en la base de datos los datos ingresados en la UI.

La comunicación de los Adapter con la base de datos se realiza a través de otro objeto llamado DataAccess.

Esta separación en capas se hizo para que haya un mayor encapsulamiento entre las diferentes partes de la aplicación, de tal manera que si hubiese que hacer modificaciones a futuro, estas sean más sencillas de realizar, ya que solamente habría que reemplazar componentes.

Aclaración: Como se solicitó que el Administrador pueda realizar tareas por un usuario dado, se creó una funcionalidad nueva la cual se llamó "Simular usuario".

Especificaciones respecto de las funcionalidades

• **ABM de Rol**

Es la funcionalidad que permite dar de alta a un rol, modificarlo o deshabilitarlo.

En el alta se debe elegir un nombre, un tipo de usuario y una funcionalidad de forma obligatoria.

Los roles tienen un tipo de usuario que puede ser cliente o proveedor (el tipo de usuario administrador no está disponible en la ventana de alta). Esto es así porque si bien pueden existir distintos tipos de roles para estos tipos de usuarios, hay cosas que distinguen dos tipos de usuarios bien diferenciados. Por ejemplo, desde los datos, los proveedores no tienen saldo, por lo que no podrían comprar cupones o una giftcard. Además de esta manera en el alta de usuario se pueden traer solo los roles que correspondan para el usuario que se está dando de alta.

Además, el alta de todos los roles incluye la funcionalidad cambiar password y darse de baja (son funcionalidades de características especiales que no se pueden eliminar de ningún rol).

La baja es una baja lógica, que deja deshabilitado al rol y a todos los usuarios que lo poseen. Esta misma acción se puede realizar desde la ventana de Modificación, así como volver a habilitar el rol. Tanto las ventanas de baja y de modificación poseen un listado que permite ver los roles del sistema por su nombre.

Las funcionalidades de los roles se han modelado con cadenas que interpreta el menú, y es esta ventana, la de menú, que sabe qué acciones realizar (a qué ventana llamar) cuando se escoge una acción, por lo que se pueden agregar y quitar funcionalidades de los roles fácilmente.

• **Login y seguridad**

La ventana de login permite, a un usuario ya registrado, ingresar al sistema, y a uno que aún no lo ha hecho, registrarse. La aplicación informa al usuario en caso de que haya ingresado mal la contraseña o en caso de que el usuario ingresado no exista (nombre de usuario mal ingresado).

• **Registro de usuario**

Implica la carga de ciertos datos según el tipo de usuario que se elija, y permite escoger un rol entre los disponibles también según el tipo de usuario.

No se puede asignar un rol no habilitado a un nuevo usuario. Los roles no habilitados no aparecen en la ventana de registro de usuario o de alta de usuario.

Una vez elegido un tipo de usuario en el registro o en su alta, no es posible modificarlo. El sistema identifica e impide la creación de los usuarios gemelos.

• **ABM de usuarios**

Permite crear, modificar e inhabilitar a un usuario. Si bien están identificadas como dos cosas diferentes, el abm de cliente y el abm de proveedor, ambas siguen la misma lógica por lo q se presentan en el menú principalmente como una misma funcionalidad (ABM usuario). Luego otra ventana permite elegir si se quiere ingresar al abm de cliente o al de proveedor. El alta sigue la misma lógica que el registro, la baja implica una baja lógica, una deshabilitación, y la modificación, la modificación de todos sus datos incluídas las ciudades de interés y el estado de habilitación. Tanto la ventana de alta como la de modificación permiten filtrar la lista de usuarios por distintos filtros.

Interpretación de enunciado: en el alta y modificación de proveedores no incluimos el campo Ciudad porque nos dijeron que era lo mismo que el campo Localidad.

Interpretación de enunciado: como en el registro de usuarios no se puede dejar ningún campo en blanco (salvo el depto y el piso), decidimos que en la modificación de usuarios tampoco es posible esto.

Importante interpretación de enunciado: Los roles de los usuarios se pueden modificar pero siempre dentro de su tipo, no es posible asignar un rol que no sea compatible con el tipo de usuario. Por esta misma decisión que tomamos, no se deben ingresar nuevamente los datos de un usuario si se le ha cambiado el rol (porque no cambia el tipo de usuario).

• **Carga de crédito**

Para cargar crédito se decidió hacer un modo de carga con tarjetas (crédito/débito) y se agregó un tipo de pago vía paypal el cual sería similar a un pago en efectivo. El sistema para el perfecto funcionamiento debería chequear la información cargada con la tarjeta de crédito / paypal, pero esto no fue requerimiento al trabajo.

• **Comprar Giftcard**

La compra de giftcard hace que un usuario le pueda regalar crédito a otro, descontándose del crédito propio. Sin la necesidad de volver a ingresar datos de tarjetas, solo se debita de la cuenta del usuario que desea regalar la giftcard y se le acredita automáticamente al usuario beneficiario. No hicimos que se cargue el cliente origen en la ventana porque es el que opera.

• **Comprar Cupón**

Para la compra de los cupones se muestra una grilla con todos los cupones en venta, y de estos aparece el precio de compra, la descripción y el código de la promoción. El usuario debe marcar el descuento a comprar, elegir la cantidad de cupones que quiere y presionar comprar. En caso que el Stock no alcance o la cantidad comprada supere al máximo establecido para esa promo, le aparecerá un mensaje de error al usuario. El filtrado de los descuentos se hace directamente en la DB. A la aplicación le llegan solo los cupones que se deben mostrar por día y por descuentos no agotados.

• **Pedir devolución**

Para pedir la devolución de un cupón, el usuario debe tener en su poder el número de cupón que adquirió. Además, deberá ingresar un motivo por el cual se devuelve y el código de la promoción.

(De esta forma nos aseguramos que haya un matcheo entre el id propio del cupón y el código de la promoción).

• **Historial de compra de cupones**

Esta funcionalidad permite a un cliente ver los cupones que compró con su código y su estado (comprado, devuelto o consumido). Para esto solo debe ingresar un intervalo de fechas en el que ninguna puede quedar sin seleccionar y la segunda fecha debe ser mayor a la primera. De otro modo el sistema no realizará la búsqueda y mostrará un mensaje de error.

Agregamos el estado consumido porque nos pareció necesario especificarlo.

• **Armar Cupón**

Los proveedores podrán armar sus cupones ingresando el precio real, el precio ficticio, una descripción, la cantidad máxima por usuario y la cantidad disponible. Deberán ingresar para que ciudades está disponible y finalmente tres fechas: la fecha en la que se desea publicar, la fecha de finalización de la promoción y la fecha de vencimiento del cupón.

• **Registro de consumo cupón**

Esta tarea es manejada por el proveedor, el cual ingresará el id_compra del cupón (este es único) y el DNI del cliente. Luego de esto aparecerá un mensaje en pantalla donde se confirmará el consumo del cupón o en caso de error de mostrará que fue lo ocurrido. Por ejemplo, si el cupón está vencido. O si ya fue utilizado.

• **Publicar cupón**

El administrador decidirá que cupones se van a publicar y cuales quedarán pendientes. Los posibles cupones deben ser de la fecha actual. Es decir, solo se podrán publicar cupones para la fecha que se establece en el archivo de configuración. Además para la búsqueda de cupones a publicar, el administrador podrá buscarlo por el CUIT del proveedor, o realizar una búsqueda entre todos los proveedores que tienen cupones a publicar en el día actual.

• **Facturación proveedor**

Para realizar la factura a un proveedor de cupones, se debe ingresar el rango de fechas entre el cual se desea facturar. Además del cuit. Luego aparecerá el resultado en el grid de todos los cupones a facturar, y en los textbox de abajo, se mostrará el numero de factura y el monto final facturado.

• **Listado estadístico**

Permite elegir una forma de calcular el top 5 de los elementos que cumplan determinada condición (actualmente los proveedores con porcentaje mas alto de devolución de cupones y los clientes con más giftcards acreditadas) en determinado año y semestre. El sistema no informa en el caso de que no haya elementos que cumplan dichas condiciones, simplemente no muestra ninguno. Y está diseñado de manera que se muestren distintos campos según el tipo de búsqueda. Con respecto a los listados estadísticos implementados: el de devolución de cupones muestra las columnas NombreProveedor (username), la cantidad de cupones vendidos, la cantidad de cupones devueltos, y su porcentaje calculado, y el de acreditación de giftcards muestra el nombre del usuario y la cantidad de giftcards acreditadas.

Además cada listado proporciona una descripción al lado de la información que va a proporcionar.

• **Simulacion usuario**

Funcionalidad que permite a un administrador simular ser otro usuario y hacer cosas como si fuese

el.

No es posible la simulación de un usuario inhabilitado.

- **Cambiar password**

Funcionalidad de características especiales que permite a cualquier tipo de usuario cambiar su contraseña. No es posible remover esta funcionalidad y todos los roles la tienen.

- **Darse de baja**

Funcionalidad de características especiales que permite a cualquier tipo de usuario darse de baja (inhabilitar su usuario). No es posible remover esta funcionalidad y todos los roles la tienen.

Base de Datos

El modelo original de datos consistía en una sola tabla que contenía todos los datos, por lo que hubo que normalizarlo mediante la creación de entidades y relaciones en base a los datos suministrados.

Todas las bajas del modelo son lógicas, por lo que nunca se pierden datos.

Actualmente el modelo está diseñado para 2 tipos de usuarios, Clientes y Proveedores, así como un usuario administrador, con la diferencia que el usuario administrador no registra ningún dato personal más allá de un usuario y contraseña.

Los usuarios pueden tener diferentes roles, y cada rol tiene permitido realizar ciertas funcionalidades.

La relación entre roles, tipos de usuario y funcionalidades se puede ver en las tablas Tipo_Usuario, Rol_Tipo, Roles, y Funcionalidades_Roles.

Para reducir la redundancia de datos, se decidió crear una tabla que relacione todos los datos de una dirección con un usuario, ya que este era el único dato en común entre clientes y proveedores. En la dirección, los campos calle y localidad fueron completados con las direcciones y ciudades de los clientes/proveedores, mientras que el resto de los campos fue puesto en NULL, ya sea el piso, el código postal, etc, ya que al momento de realizar la migración no había forma de determinarlos.

Los clientes, compradores de productos dentro del sistema, tienen asociados a ellos diferentes entidades que se detallan a continuación:

- **Datos_Clientes**

En esta entidad se guardan los datos personales del usuario, y cuál es su saldo actual. La dirección se guarda aparte.

- **Datos_Tarjeta**

En esta entidad se guardan los datos de las tarjetas que use el cliente para pagar sus compras, ya sean de crédito o de débito.

- **Ciudades**

En esta entidad se guardan las ciudades de las cuales el cliente desea ver las promociones que aplican a las mismas. El cliente solo ve las promociones que apliquen a estas ciudades.

- **Cargas**

En esta entidad se registran todas las cargas de crédito que el cliente realiza, el monto, y el tipo de pago (tarjeta/efectivo).

- **Cliente_Origen / Cliente_Destino / Giftcard**

Estas tablas sirven como índices para relacionar los dni de los clientes que envían giftcards (Origen) con los que las reciben (Destino). Esta relación se ve en la entidad Giftcard.

Hubo que separar en tablas diferentes a los clientes emisores de los receptores ya que sino SQL SERVER no permitía usar un mismo campo para 2 campos diferentes en la misma tabla (Giftcard). Dentro de Origen y Destino se relaciona un dni con un índice, y este índice se usa luego dentro de Giftcard para relacionar clientes y el valor de las giftcard.

- **Compras / Devoluciones / Consumos**

Estas entidades, como sus nombres indican, registran las compras, devoluciones y consumos realizados por los clientes.

Cada renglón de estas tablas corresponde a un único cupón. Al realizarse una compra por varios cupones, por detrás se ingresa una fila nueva a la tabla compras.

El código original provisto por la base de datos a migrar (Groupon_Codigo), lo consideramos como código de la promoción, ya que varios usuarios diferentes adquirirían cupones con el mismo código, por lo que al no tener unicidad, no se podía saber con certeza quién había comprado qué. Por estas razones, implementamos nuestro propio id_compra, el cual es un valor autonúmerico que se incrementa cada vez que se inserta una nueva compra.

Por otro lado, los proveedores, vendedores dentro del sistema, tienen asociados otras entidades, las cuales se detallan a continuación:

- **Datos_Proveedores**

En esta entidad, al igual que Datos_Clientes para los clientes, se guardan los datos personales de los proveedores, excepto la dirección, que se guarda aparte.

- **Factura / Factura_Renglon**

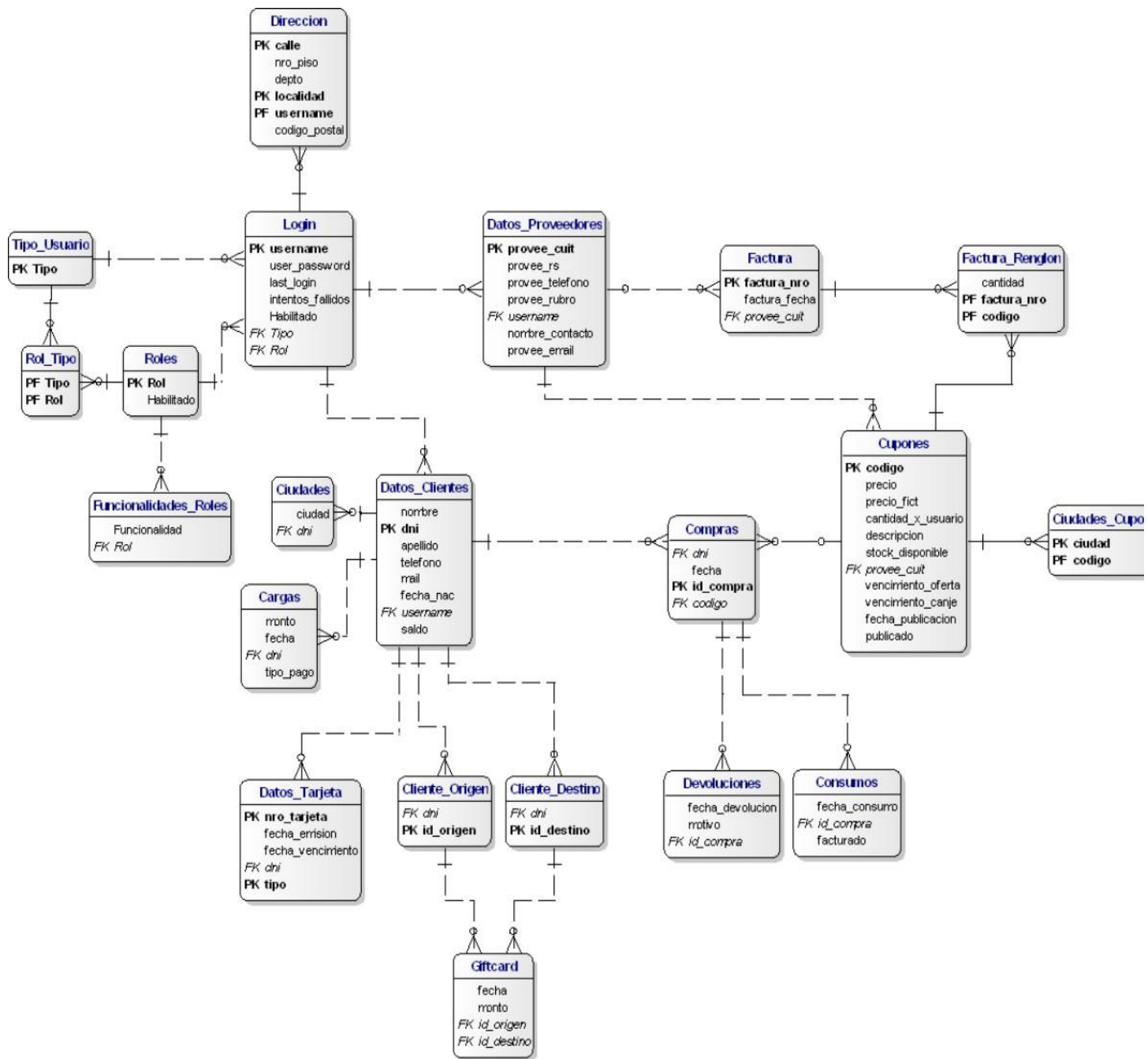
Como sus nombres lo indican, estas entidades guardan todos los elementos que se deben facturar a los proveedores.

- **Cupones / Ciudades_Cupon**

La entidad Cupones contiene toda la información relativa a un producto X, su precio, código, fechas de publicación, vencimiento, stock, cantidad máxima por usuario, etc. A su vez, la entidad Ciudades_Cupon es la que determina en que ciudad aplica la promoción dada por el cupón.

Finalmente, la tabla de login contiene toda la información relevante al logueo de un usuario al sistema, ya sea su username, password (se guarda el hash de la clave del usuario, el cual está hecho con sha256), si está habilitado, cuando fue la última vez que se logueo, su rol y su tipo de usuario. Si un usuario está inhabilitado, este no podrá logearse al sistema hasta que un administrador lo habilite nuevamente.

Modelo entidad-relación



Aclaraciones: FK = Foreign Key, PK = Primary Key, PF = Primary Key y Foreign Key

Stored Procedures:

- **sp_new_rol** : Agrega un nuevo rol a la tabla de roles.
- **sp_del_func_rol** : Borra una funcionalidad asociada a un rol.
- **sp_add_func_rol** : Agrega una funcionalidad a un rol.
- **sp_change_status_rol** : Habilita/Deshabilita un rol. Todos los usuarios que tengan este rol también son deshabilitados.
- **sp_login** : Realiza la validación de un usuario para permitirle acceder al sistema.
- **sp_insert_login** : Agrega un nuevo usuario al sistema.
- **sp_insert_cliente** : Agrega un nuevo cliente a la tabla Datos_Clientes, habiendo este creado un usuario previamente.
- **sp_insert_proveedor** : Idem sp anterior, salvo que el proveedor es agregado a la tabla Datos_proveedores.
- **sp_insert_direccion** : Inserta una nueva dirección en la tabla de direcciones.
- **sp_modify_client** : Permite la modificación de los datos de un cliente.
- **sp_change_status_user** : Permite habilitar/deshabilitar un usuario.
- **sp_modify_proveedor** : Permite la modificación de los datos de un proveedor.
- **fn_total_factura** : Dado un número de factura, esta función calcula el total facturado.
- **sp_agregar_ciudad** : Agrega una ciudad de preferencia de un cliente a la tabla ciudades.
- **sp_agregar_tarjeta** : Agrega los datos de una tarjeta de crédito/débito de un cliente a la tabla Datos_Tarjeta.
- **sp_cargar_credito** : Permite la carga de crédito a un cliente. El monto mínimo a cargar debe ser 15\$, y los tipos de pago pueden ser en efectivo, o tarjeta (crédito/débito). Si es mediante tarjeta, se deben ingresar los datos de la misma. Si la tarjeta no existe en la base de datos, esta se agrega y asocia al cliente correspondiente.
- **sp_compra_giftcard** : Permite que un cliente le regale una giftcard a otro cliente, lo cual significa un aumento de saldo en el cliente receptor.
- **sp_ajusta_saldo_devoluciones** : Parte de la migración, se usa para la actualización de saldos.
- **sp_ajusta_saldo_compras** : Idem anterior.
- **sp_nombre_usuario_no_existente** : Sirve para confirmar que un usuario no existe.
- **historial_compra** : Muestra el historial de compra/devoluciones de un usuario.
- **sp_ver_cupones_habilitados** : Muestra los cupones habilitados a un usuario. Estos cupones a su vez están filtrados de acuerdo a las ciudades de preferencia del usuario.
- **sp_comprar_cupon** : Permite realizar la compra de un cupón.
- **sp_pedir_devolucion** : Permite realizar la devolución de un cupón.
- **sp_registra_consumo_cupon** : Registra que un cupón fue consumido.
- **sp_modify_direccion** : Permite modificar los datos de una dirección.
- **sp_get_direccion** : Devuelve la dirección de un usuario.
- **sp_delete_ciudad** : Borra una ciudad de las preferencias de un usuario.
- **sp_add_ciudad_user** : Agrega una ciudad a las preferencias del usuario.
- **sp_get_ciudades** : Devuelve los nombres de las ciudades de la tabla ciudades, sin repetidos.
- **sp_get_datos_cliente** : Devuelve los datos de un cliente.

- **sp_get_datos_proveedor** : Devuelve los datos de un proveedor.
- **sp_agregar_cupon** : Agrega un cupón a la tabla de cupones.
- **sp_cupones_a_publicar** : Devuelve todos los cupones que no están publicados.
- **sp_publicar_cupon** : Publica un cupón.
- **sp_facturar_proveedor** : Crea una nueva factura en la cual están los cupones vendidos y consumidos de un proveedor.
- **sp_facturar_proveedor_nfactura** : Genera el numero de la factura.
- **sp_factura_proveedor_importe** : Genera el importe de la factura.
- **sp_buscador_clientes** : Permite a un administrador realizar una búsqueda de clientes.
- **sp_buscador_proveedores** : Permite a un administrador buscar a un proveedor.
- **sp_existe_cuit** : Permite determinar si existe un cuit.
- **get_datos_rol** : Devuelve los datos de un rol, si esta habilitado, sus funcionalidades, y el tipo de usuario que lo puede usar.
- **get_all_rols** : Devuelve todos los roles existentes.
- **get_nombre_rol_de_usuario** : Devuelve el rol que tiene un usuario.
- **sp_get_rols_para_tipo** : Devuelve los roles correspondientes a un tipo de usuario.
- **sp_modificar_rol_usr** : Permite modificar el rol de un usuario.
- **sp_get_all_usrs** : Devuelve todos los usuarios de un tipo dado.
- **sp_get_usrs_filtrados** : Devuelve usuarios que cumplan con una serie de filtros.
- **sp_get_ciudades_para_usr** : Devuelve las ciudades correspondientes a un usuario.
- **sp_usuario_habilitado** : Determina si un usuario está habilitado o no.
- **sp_existe_usuario** : Determina si existe un usuario.
- **sp_cambiar_password** : Permite crear una contraseña para un usuario del sistema viejo, ya que estos en el sistema nuevo no tienen contraseña registrada.
- **fn_in_semester** : Determina si una fecha dada pertenece a un semestre dado.
- **sp_estadistico_devoluciones** : Cumple con la parte estadística del punto 15 para las devoluciones.
- **sp_estadistico_usuarios** : Cumple con la parte estadística del punto 15 para los usuarios.
- **sp_guardar_ciudades_cupon** : Registra la ciudad a la cual aplica un cupón.