|  |  |
| --- | --- |
| logoutnba_2 | **Universidad Tecnológica Nacional**  **Facultad Regional Buenos Aires**  **Ingeniería en Sistemas de Información** |

Trabajo Práctico

2° Cuatrimestre 2018

Frba Ofertas

**Materia:**

* **Gestión de Datos**

**Docentes:**

* **Lacquaniti, Edgardo**
* **Moscuzza, Marcelo**

**Grupo N° 12**

* **T\_REX**

|  |  |
| --- | --- |
| **INTEGRANTES** | **LEGAJO** |
| Iván Lose | 159.622-6 |
| M. Daniela Rios Valencia | 143820-7 |
| Gabriela Galarza Salguero | 1245107 |
|  |  |

**INDICE**

**Contenido**

[**Nuevo Modelo de Datos (DER)** 4](#_Toc27265112)

[**Proceso de Migración de Datos al Nuevo Modelo** 5](#_Toc27265113)

[**ABM´s al Nuevo sistema** 10](#_Toc27265114)

[**Stored Procedures utilizados** 14](#_Toc27265115)

[**Decisiones de diseño generales.** 17](#_Toc27265116)

# Imagen que contiene texto Descripción generada automáticamente **Nuevo Modelo de Datos (DER)**

# **Proceso de Migración de Datos al Nuevo Modelo**

**1- CLIENTE**

Tabla con información de todas las personas que accedan a una entrada. Se considero la unicidad del cliente tomando como campos únicos **numero de documento y mail del cliente**, como en la tabla maestra no se cuenta con el tipo de documento consideramos que inicialmente que el tipo de documento es “DNI” con la posibilidad de cambiarlo a futuro.

En la tabla maestra nos encontramos con clientes con DNIs diferentes pero con un mismo mail por lo que tomamos la decisión de considerarlos como información inconsistente y quedo registrado en un campo **estado**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_cliente | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | estado | bit | **P**or default tendrá **1**.Para identificar clientes con mail duplicados **0** |
| 3 | baja\_logica | bit | Por default tendrá 1. Se utiliza para dar baja lógica al cliente |
| 4 | nombre | nvarchar(150) | Migración directa |
| 5 | apellido | nvarchar(150) | Migración directa |
| 6 | tipo\_documento | nvarchar(20) | Por default tendrá el valor DNI |
| 7 | nro\_documento | decimal(18,0) | Migración directa |
| 8 | fecha\_nacimiento | datetime | Migración directa |
| 9 | email | nvarchar(150) | Migración directa |
| 10 | teléfono | int | Migración directa |
| 11 | creditoTotal | decimal(18,0) | Se calcula el crédito total con el campo carga\_credito que nos proporciona la tabla maestra |
| 12 | id\_usuario | int | Usuario asociado al cliente. FK a la tabla Usuarios |
| 13 | id\_domicilio | int | Domicilio asociado al cliente. FK a la tabla Domicilio |

**2- PROVEEDOR**

Tabla con información de todos los proveedores que se registraron para publicar una oferta en el sistema. Consideramos como campos únicos el campo Cuit. En este caso no hubo datos duplicados para la migración.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_empresa | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | provee\_rs | nvarchar(150) | Migración directa |
| 3 | provee\_cuit | nvarchar(40) | Migración directa |
| 4 | email | nvarchar(100) | Por default tendrá **No definido,** no se posee como dato histórico |
| 5 | provee\_telefono | int | Por default tendrá **No definido,** no se posee como dato histórico |
| 6 | estado | bit | Por default tendrá 1. Se utiliza para dar baja lógica al proveedor |
| 7 | id\_domicilio | int | Usuario asociado al proveedor.FK a la tabla Proveedor |
| 8 | id\_rubro | int | Migración directa |
| 9 | id\_usuario | int | Usuario asociado al cliente. FK a la tabla Usuarios |

**3- DOMICILIO**

Tabla con información de todos los domicilios (clientes y proveedores) migrados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_domicilio | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | direc\_calle | nvarchar(255) | Migración directa |
| 3 | direc\_nro\_piso | nvarchar(255) | Por default tendrá “No definido”para la migración |
| 4 | direc\_localidad | nvarchar(255) | Migración directa |
| 5 | codigoPostal | nvarchar(50) | Por default tendrá “No definido”para la migración |
| 6 | direc\_nro\_depto | Nvarchar(50) | Por default tendrá “No definido”para la migración |

**4- USUARIO**

Tabla con información de todos los usuarios que se registran en el sistema. Para los clientes migrados tendrán como username el **DNI** y password “**1234**” y los proveedores username **CUIT** y password “**1234**”, en primera instancia. Cuando el usuario inicie sesión por primera vez el sistema solicitara cambiar la contraseña

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_usuario | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | username | nvarchar(255) | Para los clientes el username es su DNI y proveedores es su CUIT |
| 3 | password | nvarchar(255) | Contraseña migrada con “1234” para todos los usuarios y luego se requerirá cambiar contraseña. |
| 4 | Intentos\_login | int | Cantidad de intentos erróneos |
| 5 | estado | bit | Por default tendrá **1** y es la baja lógica del usuario |

**5- ROL**

Tabla con información de todos los roles permitidos en el sistema. Para la migración se creó 3 roles: administrativo, cliente y proveedor

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rol | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nombre | nvarchar(50) | Descripción del rol |
| 3 | estado | bit | Por default tendrá **1** |

**6- ROL USUARIO**

Tabla intermedia que relaciona a un rol con distintos usuarios para la administración de actividades dentro del sistemas. Solo para el nuevo sistema

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_usuario | int | FK a la tabla Usuario |
| 2 | id\_rol | int | FK a la tabla Rol |
| 3 | activo | bit | Por default tendrá **1 -** baja lógica |

**7- FUNCIONALIDAD**

Tabla con información de todas las funcionalidades permitidas en el sistema.

Solo para el nuevo sistema

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_funcionalidad | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nombre | nvarchar(255) | Descripción de la funcionalidad |

**8- FUNCIONALIDAD ROL**

Tabla intermedia que relaciona a una funcionalidad con un rol para la administración de actividades dentro del nuevo sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rol | int | FK a la tabla Rol |
| 2 | id\_funcionalidad | int | FK a la tabla Funcionalidad |
| 3 | estado | Bit | Por default tendrá **1** – baja lógica |

**9-CREDITO**

Tabla con los datos de “recarga de crédito” realizados por un cliente. Se migran datos históricos: carga de crédito y fecha.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_credito | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | fecha\_credito | datetime2(2) | Migración directa |
| 3 | id\_cliente | Int | Campo FK a la tabla Clientes |
| 4 | id\_forma\_pago | Int | Campo FK a la tabla Forma Pago |
| 5 | monto\_credito | decimal(20,2) | Migración directa |
| 6 | id\_tarjeta | int | Campo FK a la tabla Tarjeta |

**10- OFERTA**

Tabla con información de todas las ofertas que se fueron registrando en el sistema anterior.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_oferta | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | cod\_oferta | nvarchar(60) | Migración directa |
| 3 | descripcion | nvarchar(255) | Migración directa |
| 4 | fecha\_inicio | datetime | Migración directa |
| 5 | fecha\_fin | datetime | Migración directa |
| 6 | precio\_oferta | decimal(30,2) | Migración directa |
| 8 | precio\_lista | decimal(30,2) | Migración directa |
| 9 | cantDisponible | int | Migración directa |
| 10 | cant\_max\_porCliente | int | Migración directa |
| 11 | id\_proveedor | int | Campo FK a la tabla Proveedor |

**11- COMPRA**

Tabla con los registros de compras que realiza el cliente.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_compra | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | compra\_fecha | datetime(3) | Migración directa |
| 3 | id\_oferta | int | FK a la tabla Oferta |
| 4 | id\_cliente | int | FK a la tabla Cliente |
| 5 | cantidad | decimal(15,0) | Campo calculado |

**12- CUPON**

Tabla con los datos de cupones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_cupon | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | cupon\_codigo | nvarchar(60) | Migración directa |
| 3 | cupon\_fecha\_deconsumo | datetime2(3) | Migración directa |
| 4 | cupon\_precio\_oferta | decimal(20,2) | Migración directa |
| 5 | cupon\_precio\_lista | decimal(20,2) | Migración directa |
| 6 | id\_consumidor | int | Campo FK a la tabla Clientes |
| 7 | cupon\_estado | bit | bit=1 no consumido, bit=0 consumido |
| 8 | id\_compras | int | Campo FK a la tabla Compras |
| 9 | id\_oferta | int | Campo FK a la tabla Oferta |

**13- ITEM FACTURA**

Tabla con los datos detallado de lo emitido en cada factura

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_itemFactura | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | id\_factura | int | Campo FK a la tabla Factura |
| 3 | id\_oferta | int | Campo FK a la tabla Oferta |
| 4 | importe\_oferta | decimal(20,2) | Migración directa |
| 5 | cantidad | decimal(15,0) | calculado |

**14- FACTURA\_PROVEEDOR**

Tabla con los datos de cada factura emitida a proveedores

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_factura | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nro\_factura | decimal(20,0) | Migración directa |
| 3 | tipo\_factura | char(15) | Por default tendrá “No definido”para la migración |
| 4 | importe\_fact | decimal(20,2) | Migración directa |
| 5 | fecha\_inicio | datetime2(3) | Migración directa |
|  | fecha\_fin | datetime2(3) | Migración directa |
| 6 | id\_proveedor | Int | Campo FK a la tabla Proveedor |

**15- FORMA PAGO**

Tabla con la forma de pago aceptadas por el sistema del cliente. Para la migración es “efectivo” pero para el nuevo sistema se utilizará únicamente crédito

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_forma\_pago | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | Tipo\_pago\_desc | nvarchar(150) | Migración directa |

**16- TARJETA**

Tabla con las diferentes tarjetas asociadas del cliente para comprar crédito.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_tarjeta | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nro\_tarejeta | nvarchar(150) | Solo cuando corresponda a tarjeta crédito |
| 3 | titular\_tarjeta | nvarchar(150) | Solo cuando corresponda a datos del cliente |
| 4 | banco\_tarjeta | nvarchar(150) | Solo cuando corresponda a datos del cliente |
| 5 | tipo\_tarjeta | nvarchar(150) | Solo cuando corresponde a tarjeta crédito |

**17- RUBRO**

Tabla con los diferentes rubros de los proveedores del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rubro | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nombreDeRubro | nvarchar(100) | Descripción de distintos rubros del proveedor |

# **ABM´s al Nuevo sistema**

El ABM en general estará controlada del lado del front (funciones del admin, cliente y proveedor). La aplicación comienza con una pantalla inicial en la que todos los usuarios deben realizar un login e indicar el rol que posee para acceder a las funcionalidades del sistema.

**Funcionalidades**

**1- ABM de rol**

Funcionalidad que puede realizar el administrador. Inicialmente contamos con 3 tipo de roles: cliente, proveedor y administrador general. El usuario podrá editar, crear y dar de baja un rol, en el caso que elija dar de baja un rol será mediante un campo “estado” de tipo bit (1 habilitado, 0 deshabilitado, baja lógica)   
Se tomó también en cuenta que un usuario no pueda ni modificar ni dar de baja su propio rol, ya que en caso de hacer eso, y que no haya ningún rol con acceso a la ABM Rol, pasaría que no se podrían modificar los roles nunca más. Obviamente no se puede evitar que un usuario falle su contraseña 3 veces y deje al usuario inhabilitado. Pero al menos se evita perder acceso a esa ABM de esta manera.

**2- Login y Seguridad**

Funcionalidades que pueden realizar el administrativo, cliente y proveedor.

Tenemos un único administrador general con usuario “admin” y contraseña “w23e”.

La clave ingresa hasheada a base de datos, en caso de fallar en el password se suma 1 al contador que es el campo “intentos\_login”, luego de 3 intentos fallidos se bloquea la contraseña y deberá comunicarse con el administrador. Si el usuario fue recientemente registrado al nuevo sistema tendrá por default como: username “n° dni “si es cliente y “ n° cuit” si es proveedor, y de password será “1234” para ambos, en el primer logueo el sistema solicitará cambiar la contraseña obligatoriamente

**3- Registro de Usuario**

Hay 3 tipos de registro usuario; Proveedor, Cliente y Usuario. Además, no se permite registrar a un nuevo usuario administrador en esta ni ninguna otra pantalla.  
Los distintos logins existen ya que cada uno de esos roles (Proveedor y Cliente) usan distintos campos, los Usuarios de cualquier otro rol se registraran por el tercer tipo de Login, ya que no se indica que se necesiten más campos para su creación.

Como agregado, si se crea un usuario nuevo que no sea ni Cliente ni Proveedor, se debe esperar a que un Administrador habilite el usuario desde la ABM Usuario, que se creó con el objetivo de salvar esta dificultad.

**4- ABM de Clientes**

La ABM Clientes tiene como funcionalidad mostrar todos los clientes existentes, además de permitir su inhabilitación y su modificación. Sin embargo, no se permite que se les modifique el crédito disponible para evitar problemas judiciales.  
Los filtros disponibles son por nombre, apellido, mail y DNI. Los tres primeros buscan en función de que contengan parte del contenido de la búsqueda, pero la búsqueda por DNI es exacta.  
En la pantalla de modificación se podrá modificar cualquier campo de un cliente, incluyendo la contraseña, aunque la anterior no se muestra.  
En la de Alta, se puede crear un Cliente de 0, igual que en el caso de querer crear una cuenta Cliente desde la pantalla de login.

**5- AMB de Proveedor**

La ABM Clientes tiene como funcionalidad mostrar todos los proveedores existentes, además de permitir su inhabilitación y su modificación.  
Los filtros disponibles son por razón social, cuit, mail y rubro. Los tres primeros buscan en función de que contengan parte del contenido de la búsqueda, pero la búsqueda por rubro es exacta.  
En la pantalla de modificación se podrá modificar cualquier campo de un proveedor, incluyendo la contraseña, aunque la anterior no se muestra.  
En la de Alta, se puede crear un Proveedor de 0, igual que en el caso de querer crear una cuenta Proveedor desde la pantalla de login.

**6- Carga de crédito**

En la pantalla de Carga Crédito hay dos posibles situaciones: En la primera, un cliente está cargando crédito a su propia cuenta. En este caso deberá completar toda la información pertinente (donde inhabilitamos los pagos por efectivo ya que no tenemos una oficina física ni aceptamos rapipago o pagofacil). Para este caso se deberán llenar todos los datos sobre la tarjeta que se usará para cargar. Una vez hecho esto, cualquier uso de esa tarjeta para cargar crédito en cualquier otro cliente fallara, ya que esa tarjeta se encuentra vinculada a ese Cliente.  
En el segundo caso, es un administrador o usuario validado a tener acceso a carga de crédito. En este caso, se mantendrá el bloqueo de la tarjeta, pero se podrá elegir el cliente correspondiente.

**7- Confección y publicación de ofertas**

Para la confección y publicación de ofertas, se tiene una situación similar a con la carga crédito. Si se trata de un usuario administrativo u otro no Proveedor, se deberá elegir el proveedor por el cual se publicará la oferta.  
Para crear una oferta, se debe indicar la duración de la misma, la descripción, el precio original y el precio de oferta, el stock, y la cantidad máxima por cliente, si la cantidad máxima por cliente es 0, se toma que no existe un límite para la cantidad de la misma oferta que un cliente pueda abonar.

**8- Comprar oferta**

Como en todos los casos, si el usuario que ingresa no es un Cliente, se deberá elegir el Cliente por el cual se está comprando.

Las ofertas se pueden buscar a partir de la razón social de un proveedor o por la descripción de la misma. En la pantalla se mostrará el crédito restante del cliente seleccionado o que esté comprando. Para comprar una oferta, se indica una cantidad en el contador de cantidad de ofertas, y se toca el botón comprar en la fila de la oferta a comprar. Según si se cumple con los criterios correspondientes, donde se revisa si hay suficiente stock, si hay suficiente crédito o si el usuario no está sobrepasando su límite de compras por oferta, se acreditará la compra, y se crearan los cupones correspondientes.  
Nótese que se crea un cupón **por cada unidad comprada de la oferta**. Se llegó a ésta decisión ya que como la tabla cupón no tiene ninguna cantidad establecida, por más que esté conectada a la compra, no corresponde que cada cupón represente la totalidad.

**9- Entrega/Consumo de Oferta**

Como en todos los casos, si el usuario que ingresa no es un Proveedor, se deberá elegir el Proveedor por el cual se está entregando los cupones al entrar a esta funcionalidad.  
Una vez hecho esto (o si el usuario es un Proveedor) se muestra una pantalla donde se muestran todos los cupones pendientes por entregar por este Proveedor que no se hayan vencido aún (lo cual significa que no hayan pasado 2 semanas desde la compra de la oferta).

Hay 4 filtros, si bien no toda la información es mostrada. Se filtra por la descripción del producto del cupón, o por el nombre/apellido/mail/documento del cliente comprador. Aunque solo se muestra el documento del mismo.  
Para entregar un cupón, se hace click en el botón entregar ubicado en la misma fila que dicho cupón, lo cual abrirá un listado de Clientes similar al de ABM Clientes donde habrá un botón de entregar al lado de cada uno para poder entregar el cupón

**10- Facturación a Proveedor**

En facturación proveedor se realizan facturas para analizar el desempeño de determinado proveedor en un determinado período de tiempo.   
Para esto, se debe seleccionar el proveedor a facturar en una lista similar a la de ABM proveedor, una vez hecho esto se coloca el período entre el que se lo va a facturar.  
Se genera primero el listado de todas las ofertas que alguna vez se le compraron al Proveedor en ese período, y luego se toca al botón facturar.

Tómese en cuenta que si se toma un período de facturación demasiado grande el proceso puede llegar a tardar debido al nivel de procesamiento necesario para manejar semejante pedido.  
Una vez completada esta espera, se muestra una nueva pantalla con los resultados de la facturación, mostrando el número de factura, su tipo, el importe total, y el periodo del mismo. Todo esto se impacta en la base de datos.

**11- Listado Estadístico**

El listado estadístico es una funcionalidad que se usar para obtener a los 5 mejores proveedores dentro de un determinado año y semestre para determinados méritos: Los que ofrecieron los mejores descuentos para sus ofertas en ese semestre, y los que obtuvieron la mayor facturación registrada en dicho semestre.

Para esto se elige la opción en una lista desplegable, se indica el año, y se elige el primer o segundo semestre. Una vez hecho todo esto, se da a generar Listado, y se obtiene la información del proveedor correspondiente y o lo facturado ese semestre o el porcentaje de descuento promedio de sus ofertas.

**12- ABM Usuario**

En esta ABM se ven todos los usuarios existentes en una lista. Se los puede buscar por username. Tocando la opción de modificar, se pasa a una pantalla donde se permite modificar la password, el username, o habilitar/deshabilitar al mismo.

En alta, se crea un usuario, pero no se le asigna ningún rol. Esto deberá hacerse desde la pantalla de CrearCuenta para ese rol.

# **Stored Procedures utilizados**

*Aquí se detallarán todos los Stored Procedures usados a lo largo del programa.*

**Script\_delete\_table:**  el script\_delete\_table se usa para borrar el schema creado por el script\_inicial, borrando todos los objetos y todas las referencias de manera ordenada para asegurar un correcto reset a fábrica de la base de datos en caso de necesitar correr el script\_inicial otra vez

**1)USUARIO**

**T\_REX.LogearUsuario:** Se encarga de realizar las verificaciones y permitir el login de los usuarios

**T\_REX.BajaUsuario:** Se encarga de realizar la baja lógica de un usuario, sea el motivo que fuera.

**T\_REX.HabilitarUsuario:** Revierte la baja lógica de un usuario, restaurando sus roles.

**T\_REX.AgregarRolUsuario:** Agregar un campo a la tabla Rol\_Usuario para que el usuario pueda acceder al rol correspondiente.

**T\_REX.CambiarContrasenia:** Se usa para cambiar la contraseña cuando el usuario lo solicita

**T\_REX.AltaUsuario:** Es la función que crea a un Usuario.

**T\_REX.CambiarContraseniaModoAdmin:** Se usa para cambiar la contraseña cuando el admin lo solicita

**T\_REX.CambiarUsernameModoAdmin:** Se usa para cambiar el username cuando el admin lo solicita

**2) ROL**

**T\_REX.AgregarfuncionalidadRol:** Agrega una funcionalidad específica a un Rol.

**T\_REX.QuitarFuncionalidadRol:** Quita una funcionalidad específica a un Rol.

**T\_REX.InhabilitarRol:** Realiza una baja lógica a un rol, sacando dicho rol a todos los usuarios que lo tenían vinculado hasta el momento.

**T\_REX.ActivarRol:** Revierte una baja lógica a un rol, pero sin devolver el rol a sus anteriores Usuarios.

**T\_REX.CambiarNombreRol:**  Cmaia el nombre de un rol a pedido de un administrador.

**T\_REX.AltaRol:** Procedure usado para insertar un rol nuevo a la Base de Datos.

**3) CLIENTE**

**[T\_REX].AbmUsuario:** Procedure usado para insertar un Cliente nuevo a la Base de Datos.

**[T\_REX].DeshabilitarCliente:** Procedure usado para hacer una baja lógica a un Cliente.

**[T\_REX].HabilitarCliente:** Procedure usado para deshacer una baja lógica a un cliente.

**4) PROVEEDOR**

**[T\_REX].AbmProveedor:** Procedure usado para insertar un nuevo Proveedor a la Base de Datos.

**[T\_REX].DeshabilitarProveedor:** Procedure usado para hacer una baja lógica a un Proveedor.

**[T\_REX].HabilitarProveedor:** Procedure usado para revertir una baja lógica a un Proveedor.

**[T\_REX].ExisteUsuarioConNombre:** Procedure usado para verificar al momento de cambiar el nombre ( o crear uno nuevo) para ver si ese nombre no está en uso.

**5) OFERTAS**

**[T\_REX].ABMOferta:**  Procedure involucrado en la inserción de una nueva Oferta a la Base de Datos

**6) LISTADO ESTADISTICO**

**[T\_REX].ProveedoresConMasDescuento:**  Procedure que responde la consulta enviada por la pantalla de ListadoEstadístico

**[T\_REX].ProveedoresConMayorFacturacion:**  Procedure que responde la consulta enviada por la pantalla de ListadoEstadístico

**7) CREDITO**

**[T\_REX].CargarSaldo:** Procedure involucrado en la creación de todas las entidades involcuradas en la carga de crédito y en actualizar el crédito de un cliente.

**8) ITEM\_FACTURA**

**[T\_REX].CrearItemFactura:** Procedure involucrado en la inserción de un nuevo ItemFactura a la Base de Datos

**9) COMPRA**

**[T\_REX].CrearCompra:** Procedure involucrado en la inserción de una nueva Compra a la Base de Datos, descontar el crédito al Cliente y en la creación de todos los Cupones correspondientes

**10) CUPÓN**

**[T\_REX].EntregarCupon:** Este Procedure se usa para llenar los dos campos nulls en todo nuevo cupon, la fecha de consumo y el id\_consumidor, una vez que este es entregado a un Cliente.

**11) FACTURA**

**[T\_REX].CrearFactura:** Procedure involucrado en la inserción de una nueva Factura a la Base de Datos.

# **Decisiones de diseño generales.**

* A través de investigación del sistema de facturas en Argentina, llegamos a la conclusión que las facturas que realizamos son de tipo B.
* Un usuario no debe poder modificar su propio rol, para evitar problemas.
* Se crea un cupón por cada unidad comprada de un determinado producto, para mejor distribución y porque no está indicado el sistema en que se crean.
* Los nuevos usuarios que no sean Cliente ni Proveedor deben ser verificados por un administrador, ya que estos pueden tener acceso de nivel administrador a ciertas funcionalidades.
* Se mantienen por separados los logins para cada rol, ya que si mostrase todas las funcionalidades de todos los roles que tiene un usuario, tendría conflictos con las que sean propias de Cliente y Proveedor, ya que como se explicó arriba, depende de qué rol tenga el usuario que entra para ver el tipo de funcionamiento que tiene.
* Los roles tienen nombres únicos y sin números, por razones obvias de identificación.