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

Trabajo Práctico

2° Cuatrimestre 2018

Frba-PalcoNet

**Materia:**

* **Gestión de Datos**

**Docentes:**

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

**Fecha de entrega:**

* **30/11/2018**

**Grupo nro 31**

* **LOOPP**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**INDICE**

Contenido

[**Nuevo Modelo de Datos (DER)** 1](#_Toc531038996)

[**Proceso de Migración de Datos al Nuevo Modelo** 1](#_Toc531038998)

[**Proceso de control de ABM´s al Nuevo sistema** 7](#_Toc531038999)

# **Nuevo Modelo de Datos (DER)**

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

1. **Clientes**

Tabla con información de todas las personas que accedan a una entrada. Consideramos la unicidad del cliente tomando como campos únicos el **Tipo de Documento** y **Nro. de Documento**, como en la tabla maestra no se cuenta con el tipo de documento consideramos en un inicio que el tipo de documento es “DNI”, con la posibilidad de ser modificado.

En la migración de datos definimos que si existen dos emails iguales asociados a distintos DNIs son clientes con información inconsistente, dicha información quedara registrada en el campo **estado.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_cliente | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | Estado | nvarchar(50) | **P**or default tendra un valor **Habilitado.** Para clientes con datos dudosos se asignara **Inconsistente** |
| 3 | baja\_logica | bit | Por default tendra False |
| 4 | nombre | nvarchar(255) | Migración directa |
| 5 | apellido | nvarchar(255) | Migración directa |
| 6 | tipo\_documento | nvarchar(20) | Por default tendra el valor DNI |
| 7 | nro\_documento | numeric(18,0) | Migración directa |
| 8 | cuil | nvarchar(15) | Por default tendra un valor **No definido,** no se cuenta como dato histórico |
| 9 | mail | nvarchar(255) | Migración directa |
| 10 | teléfono | nvarchar(15) | Por default tendra un valor **No definido,** no se cuenta como dato histórico |
| 11 | direccion\_calle | nvarchar(255) | Migración directa |
| 12 | direccion\_nro | numeric(18,0) | Migración directa |
| 13 | direccion\_piso | numeric(18,0) | Migración directa |
| 14 | direccion\_depto | nvarchar(255) | Migración directa |
| 15 | direccion\_localidad | nvarchar(255) | Por default tendra un valor **No definido,** no se cuenta como dato histórico. |
| 16 | codigo\_postal | nvarchar(255) | Migración directa |
| 17 | fecha\_nacimiento | datetime | Migración directa |
| 18 | fecha creación | datetime | Fecha alta del cliente |
| 19 | id\_usuario | int | Usuario asociado al cliente. FK a la tabla Usuarios |

1. **Empresas**

Tabla con información de todas las empresas que se registraron para publicar un espectáculo en el sistema.

Consideramos como unicidad de las empresas al campo Cuit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_empresa | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | razon\_social | nvarchar(255) | Migración directa |
| 3 | cuit | nvarchar(255) | Migración directa |
| 4 | fecha creación | datetime | Migración directa |
| 5 | Mail | nvarchar(50) | Migración directa |
| 6 | teléfono | nvarchar(15) | Por default tendra **No definido,** no se posee como dato histórico |
| 7 | direccion\_calle | nvarchar(50) | Migración directa |
| 8 | direccion\_nro | numeric(18,0) | Migración directa |
| 9 | direccion\_piso | numeric(18,0) | Migración directa |
| 10 | direccion\_depto | nvarchar(50) | Migración directa |
| 11 | direccion\_localidad | nvarchar( | Por default tendra **No definido,** no se posee como dato histórico |
| 12 | cod\_postal | nvarchar(50) | Migración directa |
| 13 | ciudad | nvarchar(50) | Por default tendra **No definido,** no se posee como dato histórico |
| 14 | baja\_logica | bit | Por default tendra **False** |
| 15 | id\_usuario | int | Usuario asociado a la empresa. FK a la tabla Usuarios |

1. **Espectaculos**

Tabla con información de todos los espectáculos que se fueron registrando en el sistema anterior. Definimos publicación baja porque el porcentaje es del %10 y coincide con la comisión y facturado.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_espectaculo | int | PK de la tabla, migración directa |
| 2 | id\_usuario\_responsable | int | Usuario asociado al cliente. FK a la tabla Usuarios |
| 3 | id\_rubro | int | Rubro al cual pertenece. FK a la tabla Rubros |
| 4 | fecha\_publicacion | datetime | Migración directa |
| 5 | descripcion | nvarchar(255) | Migración directa |
| 6 | direccion | nvarchar(50) | Por default tendra **No definido,** no se posee como dato histórico |
| 7 | id\_estado\_publicacion | int | Estado por la que pasa cada publicacion. FK a la tabla Estados\_Publicacion |
| 8 | id\_grado\_publicacion | int | Prioridad de publicacion. FK a la tabla Grado\_Publicacion |
| 9 | precio\_base | numeric(18,2) | Precio que definido por los usuarios en el momento de alta de una publicación |
| 10 | fecha\_espectaculo | datetime | Migración directa |
| 11 | fecha\_venc\_espectaculo | datetime | Migración directa |
| 12 | hora\_espectaculo | time | Migración directa |

1. **Usuarios**

Tabla con información de todos los usuarios que se registran en el sistema.

Solo para el nuevo sistema. Todos los usuarios migrados tendrán como nombre de usuario: la primera parte del mail ( sin espacios) y contraseña “1234” ( clientes y empresas)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_usuario | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | username | nvarchar(255) | Se considera username la primera parte del mail de cada persona que sea migrada. (sin espacios) |
| 3 | password | nvarchar(255) | Contraseña migrada con “1234” para todos los usuarios y luego se requerida cambiar contraseña. |
| 4 | loginFallidos | int | Cantidad de intentos erróneos |
| 5 | habilitado | bit | Por default tendra **True** |
| 6 | PrimerLoginAuto | bit | Por default tendra **True -** cuando se autogenera el usuario está en True. |
|  |  |  |  |

1. **Roles**

Tabla con información de todos los roles permitidos en el sistema.

Solo para el nuevo sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rol | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nombre | nvarchar(255) | Descripción del rol |
| 3 | baja\_logica | bit | Por default tendra True |

1. **Rol X Usuario**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_usuario | int | FK a la tabla Usuarios |
| 2 | id\_rol | int | FK a la tabla Roles |
| 3 | activo | bit | Por default tendrá **True** |

1. **Funcionalidades**

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 |

1. **Func X Rol**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rol | int | FK a la tabla Roles |
| 2 | id\_funcionalidad | int | FK a la tabla Funcionalidades |
| 3 | baja\_logica | Bit | Por default tendra False |

1. **Estados Publicación**

Tabla con los diferentes estados con la que cuenta un espectáculo a la hora de ser publicada.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_estado\_publicacion | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | descripcion | nvarchar(255) | Descripción del estado |

1. **Rubros**

Tabla con los diferentes rubros con la que cuenta un espectáculo a la hora de su publicación.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_rubro | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | descripcion | nvarchar(255) | Descripción de distintos rubros de espectáculo |

1. **Grados Publicacion**

Tabla con los grados de prioridad al que se puede asociar el espectáculo a la hora de publicar.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_grado\_publicacion | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | activo | bit | Por default tendra True |
| 3 | comisión | numeric(10,2) | Valor que se cobra a la empresa por cada compra de entrada |
| 4 | descripción | nvarchar(255) | Descripción del grado de prioridad |

1. **Tipo Ubicacion**

Tabla con los distintos tipos de ubicación que se cuenta en la sala

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_tipo\_ubicacion | int | PK de la tabla, migración directa |
| 2 | descripcion | nvarchar(20) | Descripción con el detalle de ubicación |
| 3 | porcentual | numeric(10,2) | Valor que se cobra por ubicación |

1. **Ubicaciones**

Tabla con las ubicaciones con la que se cuenta en la sala.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_ubicacion | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | fila | varchar(3) | Migración directa |
| 3 | asiento | numeric(18,0) | Migración directa |
| 4 | sin\_numerar | bit | Por default tendra **False.** Migración directa |
| 5 | Id\_tipo\_ubicacion | int | FK a la tabla Tipo\_Ubicacion |

1. **Ubicac X Espectaculo**

Tabla con los datos de cada ubicación asociado al espectáculo publicado.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_espectaculo | int | FK a la tabla Espectaculos |
| 2 | Id\_ubicacion | int | FK a la tabla Ubicaciones |
| 3 | precio | numeric(18,2) | Valor total del espectaculo, migración directa |
| 4 | disponible | bit | Por default tendra True |

1. **Formas Pago Cliente**

Tabla con las formas de pago del cliente

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_forma\_pago\_cliente | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | Id\_forma\_pago | int | FK a la tabla Formas de Pago |
| 3 | nro\_tarjeta | bigint | Solo cuando corresponde a tarjeta debito/credito |
| 4 | Id\_cliente | int | FK a la tabla Clientes |

1. **Formas Pago**

Tabla con las formas de pago aceptadas por el nuevo sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_forma\_pago | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | descripcion | nvarchar(20) | Migración directa |
| 3 | marca | nvarchar(20) | Solo cuando corresponde a tarjeta debito/credito |

1. **Compras**

Tabla con las compras que realiza el cliente

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_compra | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | Id\_cliente | int | FK a la tabla Clientes |
| 3 | fecha\_compra | datetime | Migración directa |
| 4 | puntos | date | Implementada para el nuevo sistema |
| 5 | cantidad\_compra | numeric(18,0) | Migración directa |
| 6 | Importe\_total | numeric(18,0) | Valor total de la compra, migración directa |
| 7 | Id\_forma\_pago\_cliente | int | FK a la tabla Formas de pago |
| 8 | facturado | bit | Por dafault tendra False, se utilizara para las compras pendientes de rendición |

1. **Localidades Vendidas**

Tabla con las localidades que se van vendiendo por espectáculo

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_compra | int | FK a la tabla Compras |
| 2 | id\_espectaculo | int | FK a la tabla Espectaculos |
| 3 | id\_ubicacion | int | FK a la tabla Ubicaciones |

1. **Facturas**

Tabla con los datos de cada factura emitida según las rendiciones realizadas para las empresas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | nro\_factura | int | Campo PK - Migración directa |
| 2 | fecha\_factura | datetime | Migración directa |
| 3 | total\_factura | numeric(18,2) | Migración directa |
| 4 | total\_comision | numeric(18,2) | Calculado |
| 5 | Id\_empresa | int | Campo FK a la tabla Empresas |
| 6 | Id\_espectaculo | Int | Campo FK a la tabla Espectaculos |

1. **Item Factura**

Tabla con los datos detallado de lo emitido en cada factura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | nro\_item | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | nro\_factura | int | Campo FK a la tabla Facturas |
| 3 | monto\_compra | numeric(18,2) | Migración directa |
| 4 | monto\_comision | numeric(18,2) | Calculado |
| 5 | cantidad | numeric(18,0) | Migración directa |
| 6 | descripcion | nvarchar(60) | Migración directa |

1. **Canjes**

Tabla con los datos de canjes realizados por cliente. Se contempla solo para el nuevo sistema, no existe dato histórico.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_canje | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | fecha\_canje | datetime | Fecha que se canjeo los puntos |
| 3 | puntos\_canjeados | numeric(18,0) | Cantidad de puntos canjeados |
| 4 | Id\_codigo | int | FK a la tabla Catalogo Canje |
| 5 | Id\_cliente | int | FK a la tabla Clientes |

1. **Catalogo\_Canje**

Tabla con los datos de los posibles canjes a realizarse. Se contempla solo para el nuevo sistema, no existe dato histórico.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_codigo | int | PK de la tabla, autoincremental. Identity(1,1) |
| 2 | stock | numeric(18,0) |  |
| 3 | descripcion | nvarchar(30) | Descripción del los productos de canje |
| 4 | puntos\_validos | numeric(18,0) |  |

1. **Registro de puntos**

Tabla con los datos de los puntos usados del cliente.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nro.** | **Campo** | **Tipo** | **Observaciones** |
| 1 | id\_compra | int | PK de la tabla, FK de la tabla Compras |
| 2 | Id\_cliente | int | PK de la tabla, FK de la tabla Cliente |
| 3 | puntos\_usuados | int | Cantidad de puntos usados |
| 4 | fecha\_vencimiento | datetime | Fecha de vencimiento (se calcula sumando 6 meses a la fecha de la compra) |

# 

# **ABM´s al Nuevo sistema**

El ABM en general estará controlada del lado del front (funciones de admin, cliente y empresa)

La aplicación comienza con una pantalla inicial en la que todos los usuarios deben realizar un login para acceder a las funcionalidades, el cual estará limitado según sea el tipo de usuario. Tenemos 3 tipos de usuarios: cliente, empresa y administrador general (este último tiene la posibilidad de loguearse con el rol administrativo, cliente o empresa)

**Funcionalidades**

1. **ABM de Rol**

Funcionalidad que puede realizar el administrador. Inicialmente contamos con 3 roles (Administrativo, Empresa y Cliente). El usuario puede editar, crear y dar de baja un rol (se dará de baja mediante un campo habilitado de tipo booleano).

1. **Login y seguridad**

Funcionalidad que puede realizar el administrativo, empresa o cliente. Por defecto tenemos un único administrador general con usuario ‘admin’ y contraseña ‘w23e’. Cuando se ingresa la clave se compara con la clave que se encuentre hasheada por la base de datos, en caso de fallar se sumara 1 al contador de fallos de inicio de sesión. A los 3 intentos, se bloqueará el usuario y deberá comunicarse con un administrador. Todo usuario recientemente registrado tendrá por default nombre de usuario: “cuit o cuil” y contraseña: 1234, en el primer logueo el sistema solicitará el cambio de contraseña obligatoriamente.

1. **Registro de Usuario**

Funcionalidad que permite a los administradores dar de alta, baja o modificar usuarios dentro del sistema y los hoteles en los que trabajan. El administrador es el único capaz de crear nuevos administradores dentro del sistema y habilitar usuarios dados de baja.

La contraseña que se ingresa o modifica se guardara hasheado (SH256) dentro de la base de datos. Dar de baja un usuario significa que no está habilitado, contamos con un campo de tipo booleano para esto.

Hay un único administrador general del sistema. Usuario “admin” contraseña “w23e”.

Cuando el administrador registra un nuevo usuario (cliente/empresa) se le asignara como nombre de usuario: “su cuit/cuil” y contraseña: “1234” (luego el sistema solicitara el cambio de contraseña ).

1. **ABM de Clientes**

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un cliente del sistema.

Para la baja de un cliente contamos con un campo llamado “estado” que nos indica si el cliente está habilitado o es inconsistente. En el caso que se requiera dar la baja de un cliente el estado pasa a ser inconsistente hasta resolver el motivo de la baja. Volver a dar de alta un cliente implica que los datos estén ingresados correctamente (mail, numero de documento, cuil, medios de pago etc.) desde la ventana “consultas clientes- modificar”.

Los clientes son usuarios que pueden comprar localidades dentro la plataforma.

1. **ABM de Empresas**

Funcionalidad que permite a un administrativo crear, modificar y dar de baja una

empresa de espectáculos.

Para la baja de una empresa también contamos con un campo llamado “baja lógica” que nos indica si la empresa está habilitada o inhabilitada, ya sea por tener los datos mal cargados o bloqueo de login. El administrador puede volver a dar de alta desde la ventana “consultas empresas -modificar”

Las empresas solo tienen la posibilidad de venta en la plataforma, denegándoseles toda posibilidad de compra.

1. **ABM de Rubro**

Funcionalidad que permite categorizar a un espectáculo y determinar dentro de

que género se puede encontrar a dicha publicación. No dispone de ABM.

1. **ABM Grado de Publicación**

Esta funcionalidad es la que le permite al usuario determinar la prioridad

de visualización con que sus publicaciones van a mostrarse, las cuales son Alta, Media y Baja. Todos los grados de publicaciones tienen una comisión, la cual es transparente al cliente que compra una localidad.

El administrador es el único que puede dar de alta, baja o modificar dicho grado de publicación.

1. **Generar Publicación**

Esta funcionalidad es realizada únicamente por las empresas.

Una publicación se caracteriza por tener una serie de estados.

Los mismos pueden ser:

* Borrador:

Este estado permite al vendedor modificar todos los datos que sean

necesarios, ya que dicha publicación todavía no está visible para la

comunidad.

* Activa o publicada:

Este estado es aplicado por el vendedor para que dicha publicación sea

vista por toda la comunidad y esté disponible para operar con ella. se deberá chequear siempre que la fecha de compra sea menor o igual a la fecha del evento.

* Finalizada:

Estado utilizado por el vendedor para indicar que una publicación llegó a

su fin y decide no ofrecer más localidades a la venta. A fines didácticos, el

vendedor se encargará de modificar dicho estado al día siguiente de

finalizada la publicación. Una vez que

una publicación está finalizada no puede cambiarse el estado de la misma a ningún otro.

La fecha de una publicación es la que se tomas por el archivo de configuración (la fecha que se genera la publicación).

Fecha de vencimiento: suma 7 días posteriores a la fecha de publicación. Implica que después de esa fecha no se pueden comprar entradas y se finaliza dicha publicación. La fecha del espectáculo: se considera que la fecha de un espectáculo sea a partir del día posterior a la fecha de vencimiento de la publicación.

1. **Editar Publicación**

Funcionalidad realizada por las empresas que permite la modificación de una publicación contenida en la plataforma.

Para realizar modificaciones se deberá tener en cuenta el estado de la publicación, solo en estado borrador.

1. **Comprar**

Funcionalidad que solo puede realizar el cliente. Las compras podrán realizarse para publicaciones activas, la fecha de compra se encuentre dentro del rango de fecha de publicación y las publicaciones no estén finalizadas ni pausadas ni en borrador.

Si el cliente no tiene un medio de pago asociado (por algún motivo) en el momento se le dará la posibilidad de registrar uno para poder realizar la compra.

1. **Historial de Cliente**

Esta funcionalidad permite a un cliente conocer todo su historial de compras.

1. **Canje y Administración de puntos**

Esta funcionalidad permite a un cliente consultar sus puntos de usuario frecuente como así también realizar canjes con los puntos que ya tienen asociados. Un cliente suma una cantidad de puntos con cada compra que realiza, la misma tiene una fecha de vencimiento que son 6 meses a partir del día que realizo la compra.

El sistema le provee al cliente un catálogo de canje:

* 2 entradas gratis 1000 puntos
* 1 entrada gratis 800 puntos
* Una remera del espectáculo 500 puntos
* Una consumición gratis 300 puntos

Cada opción contiene una cantidad de stock y un código de canje (que es el ID de canje), este ultimo es el que se le da al cliente cada que ves que realiza un canje.

1. **Generar rendición de comisiones**

Funcionalidad que realiza el administrador, registra facturas por el cobro de comisiones de ventas de publicaciones a la empresa de espectáculos.

1. **Listado Estadístico**

Ingresando el año y trimestre se podrá consultar los siguientes listados:

* Empresas con mayor cantidad de localidades no vendidas
* Clientes con mayores puntos vencidos
* Clientes con mayor cantidad de compras