UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Gestión de Datos

Trabajo Práctico

2º Cuatrimestre 2017

PAGO AGIL

Legajo	Nombre	Apellido	Curso	E-mail
153.511-0	Aldo Gabriel	Silvestre	K3572	aldo_soga@hotmail.com
149.280-9	Juan Manuel	Crespo	K3572	juanmacrespo@gmail.com
131.972-3	Gabriel Nicolás	Figueroa	K3572	gabrieln.figueroa@yahoo.com
150.139-2	Matías	Lanneponders Yonadi	K3572	mlyonadi@gmail.com

INDICE

ESTRATEGIAS DE SOLUCION

Configuración

Se utilizó para realizar la conexión con la base de datos el archivo nombrado en la carpeta de AplicacionDesktop/App.config la que posee la misma por ConnectionString donde se especifica el nombre de la conexión, el host donde esta alojada la BD y donde tendrá que ir a buscar los datos requerido, el nombre de la BD y los certificados para acceder a la misma; en nuestro caso se solicito que se ingrese como USUARIO= gd y CONTRASEÑA= gd2017.

DIAGRAMA ENTIDAD RELACION (DER)

CLIENTES

(Aquellas personas que se dirigen a la ventanilla de cobro a efectuar el pago)

La tabla Clientes está conformada por las siguientes columnas:

- **CLIE_ID:** de tipo int , es la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **CLIE_NOMBRE:** de tipo nvarchar(255), almacena el nombre del cliente.
- **CLIE_APELLIDO:** de tipo nvarchar(255), almacena el apellido del cliente.
- **CLIE_FECHA_NACIMIENTO:** de tipo datetime, contiene la fecha de nacimiento del cliente.
- **CLIE_DIR_ID:** de tipo int, es una clave foránea del id de la tabla DIRECCIONES, como consecuencia de esta acción un cliente solo puede tener una dirección en el sistema.
- **CLIE_DNI:** de tipo int, contiene el documento único de identidad del cliente en cuestión.
- **CLIE_MAIL:** de tipo nvarchar(255), deberá tener identificado un mail dicho cliente. Se agregó a este campo un constraint de tipo Unique.
- **CLIE_ACTIVO:** de tipo bit, acepta dos valores: 0, en caso de que el cliente se encuentre inhabilitado en el sistema, o 1, si el cliente está habilitado.

ROLES

La tabla Roles está conformada por las siguientes columnas:

- **ROL_ID:** de tipo int, es la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1) , ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **ROL_NOMBRE:** de tipo varchar(45). Contiene un nombre descriptivo del rol en cuestión.

- **ROL_ACTIVO:** de tipo bit, sólo acepta dos valores posibles: 0, para indicar que el rol está inactivo, o 1, si el rol está activo.

USUARIO_X_ROL

La tabla Usuario_x_rol está conformada por las siguientes columnas:

- **USERX_ID:** de tipo int, es parte de la clave primaria (junto con la columna user_id) de la tabla. Además, actúa como clave foránea al campo rolx_id de la tabla ROLES.
- **ROLX_ID:** de tipo int, es parte de la clave primaria (junto con la columna rol_id) de la tabla. Además, actúa como clave foránea al campo user_id de la tabla USUARIOS.

USUARIOS

La tabla Usuarios está conformada por las siguientes columnas:

- **USER_ID:** de tipo int, es la clave primaria de la tabla.
- **USER_USUARIO:** de tipo varchar(255). Almacena el nombre de usuario que se utiliza para ingresar a la aplicación desktop.
- **USER_PASSWORD:** de tipo varchar(150). Contiene la contraseña del usuario en cuestión, encriptada bajo el algoritmo de encriptación SHA256.
- **USER_ACTIVO:** de tipo bit. Solo almacena dos tipos posibles: 0, en caso de que el usuario este inactivo en el sistema, no pudiendo ingresar, o 1, el usuario se encuentra activo para la aplicación.
- **USER_INTENTOS:** de tipo tynint. Se podrá observar la cantidad de intentos erróneos de ingreso a la aplicación desktop que realizó el usuario. Si llega a tres intentos erróneos, el usuario es inhabilitado automáticamente por el sistema; si el usuario ingresa satisfactoriamente antes de llegar a ese punto, los intentos erróneos se borran y este campo vuelve a 0.

EMPRESAS

(Entidades que tienen convenio y se registra pagos de servicio):

La tabla Empresa está conformada por las siguientes columnas:

- **EMP_ID:** de tipo int, es la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1) , ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **EMP_CUIT:** de tipo nvarchar(50), es un campo obligatorio, donde se encontrará el cuit de la empresa.
- **EMP_NOMBRE:** de tipo nvarchar(255), donde figurara el nombre de la empresa. Tener en cuenta que es un campo obligatorio.
- **EMP_DIR_ID:** de tipo int, actúa como clave foránea con el campo DIR_ID de la tabla DIRECCIONES.

- **EMP_RUB_ID:** de tipo int, actúa como clave foránea con el campo RUB_ID de la tabla RUBRO.
- **EMP_DIA_REND:** de tipo int, donde se informa el día de rendición.
- **EMP_ACTIVA:** de tipo bit. Solo almacena dos tipos posibles: 0, en caso de que la empresa es inhabilitada para el sistema y por lo tanto no se podrán hacer rendiciones, o 1, donde la empresa se encuentra habilitada.

  Tener en cuenta que todos estos campos son obligatorios.

SUCURSALES

(Corresponden a las sucursales donde se realizan cobro de facturas)
La tabla Sucursal está conformada por las siguientes columnas:

- **SUC_ID:** de tipo int, corresponde a la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **SUC_DIR_ID:** de tipo int, actúa como clave foránea con el campo DIR_ID de la tabla DIRECCIONES. Identificamos la dirección física de la sucursal.
- **SUC_NOMBRE:** de tipo nvarchar(45), corresponde al nombre de la sucursal.
- **SUC_HABILITADA:** de tipo bit, el 0 indicara que se encuentra inhabilitada para recibir cobros de facturas, y 1 para identificar que se encuentra activa.

FACTURAS

(Contendrán las facturas que serán procesadas por el sistema)
La tabla Facturas está conformada por las siguientes columnas:

- **FACT_ID:** de tipo numeric(18,0), corresponde a la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **FACT_CLIE_ID:** de tipo int, actúa como clave foránea con el campo CLIE_ID de la tabla Clientes.
- **FACT_EMP_ID:** de tipo int, actúa como clave foránea con el campo EMP_ID de la tabla Empresas, ya que es la empresa que proviene esa factura.
- **FACT_VENCIMIENTO**: de tipo datetime, corresponde al vencimiento de la factura.
- **FACT_ALTA:** de tipo datetime, corresponde a la fecha de alta de la factura en el sistema.
- **FACT_REND_ID:** de tipo numeric(18,0), actúa como clave foránea con el campo REND_ID de la tabla Rendiciones.
- **FACT_PAGO_ID:** de tipo numeric(18,0), actúa como clave foránea con el campo PAGO_ID de la tabla Pagos.

- **FACT_TOTAL:** de tipo numeric(18,2), donde contendrá el total del importe total en pesos de la factura que se almacena en el sistema.
- **FACT_ACTIVA:** de tipo bit, posee dos opciones: 0, cuando la factura se encuentra inactiva, o 1 para la factura se encuentra activa de cara al sistema.

PAGOS

(Registra el pago del cliente para las facturas que está presentando)

La tabla Pagos está conformada por las siguientes columnas:

- **PAGO_ID:** de tipo numeric(18,0), corresponde a la tabla primaria de la tabla.
- **PAGO_FECHA_PAGO:** de tipo datetime, fecha que se autogenera por el sistema informando la fecha del pago de las facturas del cliente.
- **PAGO_TOTAL:** de tipo numeric(18,2), informa el importe total en pesos de todas las facturas que el cliente pago en dicho momento.
- **PAGO_FORMA_PAGO:** de tipo nvarchar(255), forma de pago en que se realizó las facturas del cliente.
- **PAGO_CLIE_ID:** de tipo int, actúa como clave foránea con el campo CLIE_ID de la tabla Clientes, donde relaciona el cliente que fue a pagar esas facturas.
- **PAGO_USER_ID:** de tipo int, actúa como clave foránea con el campo USER_ID de la tabla Usuarios, para identificar ese usuario que registro el pago de las facturas presentadas por el cliente.
- **PAGO_SUC_ID:** de tipo int, actúa como clave foránea con el campo SUC_ID de la tabla Sucursales. Donde se registra la sucursal en donde se realizó la transacción.

FUNCIONALIDADES

La tabla Funcionalidades está conformada por las siguientes columnas:

- **FUNC_ID:** de tipo int, es la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **FUNC_DESCRIPCION:** de tipo varchar(100). Contiene un nombre descriptivo de la funcionalidad en cuestión.

ROL_X_FUNCIONALIDAD

La tabla Rol_x_funcionalidad está conformada por las siguientes columnas:

- **ROL_ID:** de tipo int, es parte de la clave primaria (junto con la columna Roles) de la tabla. Además, actúa como clave foránea al campo id de la tabla Funcionalidades.
- **FUNC_ID:** de tipo int, es parte de la clave primaria (junto con la columna Funcionalidades) de la tabla. Además, actúa como clave foránea al campo id de la tabla Roles.

RENDICIONES

(Rendiciones de las facturas cobradas a los clientes que se acercan a las sucursales)

La tabla Rendiciones está conformada por las siguientes columnas:

- **REND_ID:** de tipo numeric(18,0), es la clave primaria de la tabla. Es un número que identifica unívocamente la rendición en cuestión.
- **REND_EMP_ID:** de tipo int, es la clave foránea con la tabla empresas para identificar la relación.
- **REND_TOTAL_RENDICION:** de tipo numeric(18,2), contiene el monto total de la rendición en cuestión.
- **REND_PORCENTAJE_COMISION:** de tipo tinyint, es la ganancia por cobro el cual se descuenta del total rendido.
- **REND_FECHA:** de tipo datetime, fecha en que ocurrió la rendición.

DEVOLUCIONES

(Muestra aquellas devoluciones de las facturas cobradas que no fueron rendidas)

La tabla Devoluciones contiene los siguientes campos:

- **DEV_FACT_ID:** de tipo numeric(18,0), es la clave primaria de la tabla.
- **DEV_DESCRIPCION:** de tipo nvarchar(255), informa el motivo de la devolución.
- **DEV_USER_ID:** de tipo int, es una clave foránea de la tabla Usuarios relacionando al usuario que realizo la devolución.
- **DEV_CLIE_ID:** de tipo int, es una clave foránea de la tabla Clientes relacionando a quien se devolverá las facturas cobradas.
- **DEV_TIPO:** de tipo varchar(45), corresponde al tipo de devolución.

DIRECCIONES

(Muestra las direcciones de distintas entidades como clientes, empresas y sucursales)

La tabla Direcciones contiene los siguientes campos:

- **DIR_ID:** de tipo int, corresponde a la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **DIR_DIRECCION:** de tipo nvarchar(255), es la dirección física de distintas entidades.
- **DIR_CODIGO_POSTAL:** de tipo char(8), indica el código postal relacionada con la dirección mostrada.
- **DIR_PISO:** de tipo char(2), corresponde al piso en caso de ser departamento.
- **DIR_DEPARTAMENTO:** de tipo char(2), indica la ubicación del departamento.
- **DIR_LOCALIDAD:** de tipo nvarchar(22), nos muestra la localidad que condice con el código postal y la dirección.

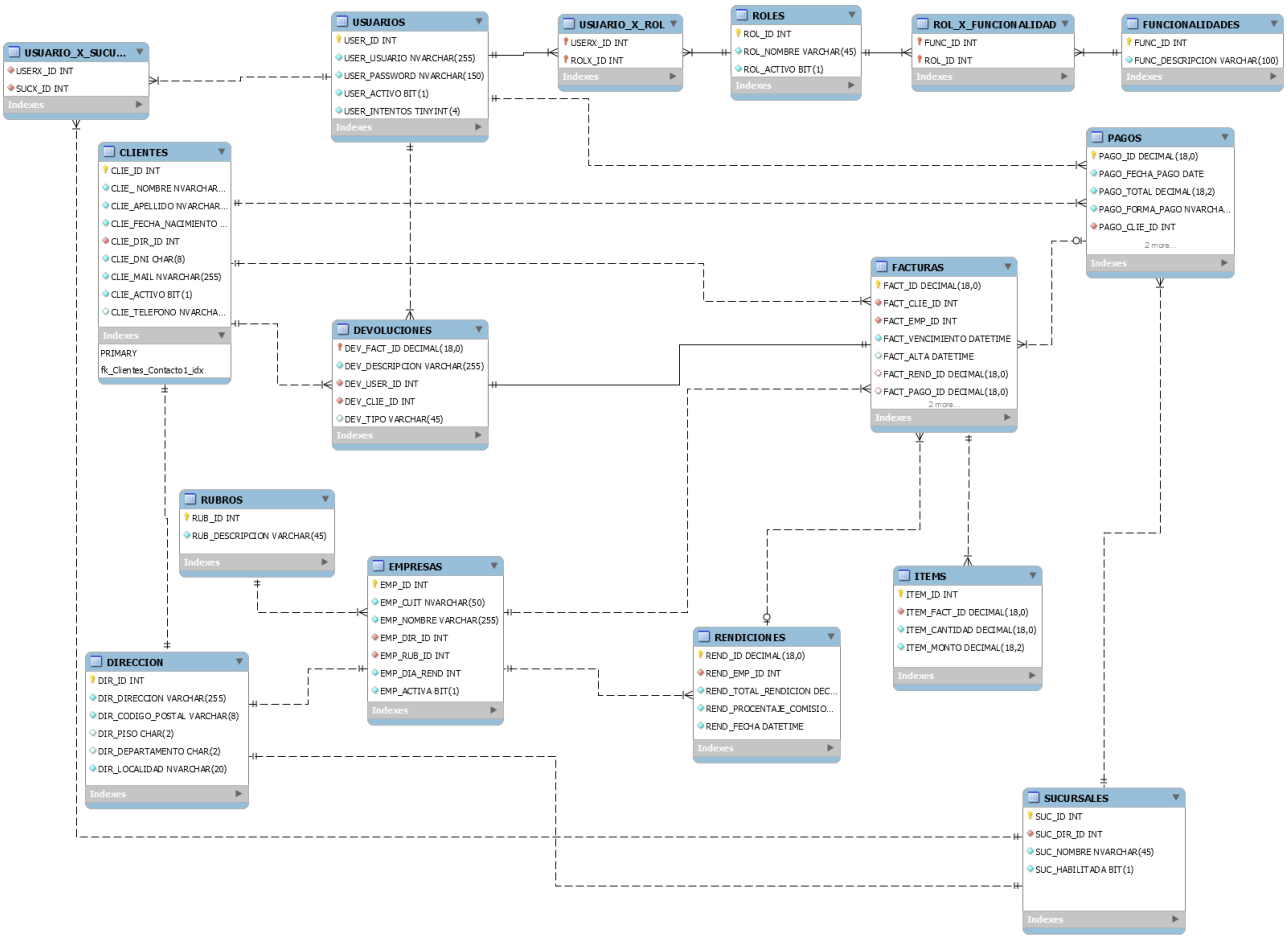
ITEMS

(Proviene del detalle de las facturas en cuestión que van a ser cobradas)

La tabla Ítems contiene los siguientes campos:

- **ITEM_ID:** de tipo int, corresponde a la clave primaria de la tabla. Además, tiene la propiedad Identity(1,1), ya que el valor del campo es autoincremental para cada nuevo registro que se agregue a la tabla.
- **ITEM_FACT_ID:** de tipo numeric(18,0), al ser una clave foránea relaciona al concepto de la facture que proviene este ítem.
- **ITEM_CANTIDAD:** de tipo numeric(18,0), corresponde a la cantidad del ítem que se está presentando.
- **ITEM_MONTO:** de tipo numeric(18,2), es el importe total del monto de la cantidad del ítem.

DER



MODIFICACIONES DIAGRAMA ENTIDAD-RELACION (DER)

Se nombraran a continuación los cambios realizados al modelo de datos entregado, con la agregación y/o modificación de los datos con respecto a la base de datos otorgada por la catedra. Las tablas llevaran el símbolo “#” para ser identificadas.

#Usuario

- ➔ Se agregaron los certificados de username y password correspondientes para que el usuario pueda realizar su logueo.
- ➔ Se agrego el campo sucursal Id ya que el Usuario que se loguea debe pertenecer minimamente a una sucursal.
- ➔ Se contiene una columna intentos la cual determina las veces que el usuario intento acceder al sistema por password errónea, donde la cantidad se ve reflejada en la misma.
- ➔ La columna activo representa la habilitación para realizar operaciones, un usuario deshabilitado no puede ingresar por logueo a la aplicación.

Aclaraciones:

1) Como se cita en el enunciado no se realizo el abm correspondiente para la gestión de Usuarios, los mismos deben ser modificados y/o creados por BD, aunque la estructura está construida para soportar dicha funcionalidad.
2) Para que un Usuario pueda poseer varios roles se creo una tabla intermedia en #USUARIO_X_ROL la cual permitirá al usuario poseer uno o mas roles con el cual accederá al sistema.

#Rol , # Funcionalidades

- ➔ Cada rol ingresado al Sistema poseerá su propia id y descripción. Con la cual el primer campo identificatorio será relacionado con una tabla intermedia que será #ROL_X_FUNCIONALIDAD donde se posee en esta ultima el identificador del rol junto con el identificador de la funcionalidad.
- ➔ Cada rol cuenta con un campo ACTIVO que identifica si dicho rol se encuentra activado en el sistema o no.
- ➔ La tabla #FUNCIONALIDADES posee los campos id y descripción en donde se puede detallar en el segundo el nombre de la funcionalidad, con la cual se podrá mostrar la descripción de la misma al usuario.

#Clientes

- ➔ Campo Activo, es para verificar que cuando una sucursal se encuentra deshabilitada o en si se desea realizar un alta de un cliente pero por algún motivo este no estará habilitado para realizar pagos.

#Empresas

- ➔ Campo Activo, para verificar si dicha empresa puede solicitar el cobro de facturas, la deshabilitación de este campo será condicionado porque no tenga facturas pendientes de rendición.
- ➔ Campo Día de Rendición, con este dato se podrá realizar las rendiciones de las facturas en un determinado día del mes establecido por la empresa, por defecto todas las rendiciones se realizan el primer día de cada mes.

#Sucursales

- ➔ Campo Habilitada, se lo utiliza para la verificación de que dicha sucursal se encuentra se manera habilitada para poder recibir pagos de los clientes. Y a su vez impide el acceso a los Usuarios pertenecientes a esta sucursal.

#Direcciones

- ➔ Campos piso y departamento para proporcionar una información mas detallada del mismo.

MIGRACION BASE DE DATOS

Objetos de Base de Datos

Migración (script_creeacion_inicial.sql)
En primer momento se depuran todas las tablas para comenzar la migración.

Direcciones

En la tabla maestra ubicamos ciertos campos correspondientes a esta tabla:
[Cliente_Direccion, Cliente_Codigo_Postal]
En el campo localidad ingresamos como defecto la asignación 'CABA'. Este caso de migración es para los Clientes. Se tuvo en cuenta que dicha tabla tiene además a la información de las direcciones de las empresas y tomamos de referencia los siguientes campos:
[Empresa_Direccion]
Mientras que los demás campos tomamos como convención en el código postal 1702, y en la localidad 'CABA', mientras que los demás campos fueron ingresados en NULL.
Las sucursales también van a tener sus direcciones en esta tabla a través de los siguientes campos de la tabla maestra:
[Sucursal_Dirección, Sucursal_Codigo_Postal]
Se tuvo en cuenta las direcciones de sucursales que contengan el código postal.

Rubros:

Notamos que existen campos que se utilizan para identificar a los rubros:
[Rubro_descripcion]
Cada diferente descripción encontrada representa a un rubro.

Sucursales:

Para identificar las sucursales encontramos los siguientes campos de la tabla maestra: [Sucursal_Nombre].
En este caso ya tenemos una diferencia, porque cargamos la dirección ya a través de la tabla DIRECCIONES previamente cargada. Optamos que todas las sucursales se encuentren habilitadas.

Clientes:

Analizando la tabla maestra, notamos que existen campos que se utilizan para identificar a los clientes:
[Cliente-Dni, Cliente-Apellido, Cliente-Nombre, Cliente-Fecha_Nac, Cliente_Mail, Cliente_Direccion, Cliente_Codigo_Postal]
Por lo tanto, cada diferente combinación encontrada de estos campos representaría a un cliente distinto. Entonces, si por ejemplo existieran 5 registros donde estos campos fueran totalmente iguales entre sí, estos 10 registros harían referencia a un mismo cliente.

Por lo tanto, para normalizar los datos, este cliente se registra una sola vez en la tabla Clientes. Y luego se harán las referencias correspondientes, como la dirección que se obtendrá de la información de la tabla Direcciones anteriormente migrada.

Empresas:

Existen campos en la base original que se utilizan para identificar los diferentes Empresas:
[Empresa_Cuit, Empresa_Nombre, Empresa_Direccion, Rubro_Descripcion]
Después se tuvo en cuenta para cargar las claves foráneas del EMP_RUB_ID y EMP_DIR_ID ya con la información cargada de las tablas Rubros y Direcciones migradas. Para este caso tuvimos consideramos a todas las empresas como activas.

Rendiciones:

Encontramos los siguientes campos referidos a las rendiciones:
[Rendicion_Nro, Rendicion_Fecha, ItemRendicion_Importe]
Por cada número de rendición encontrado existirá un único registro en nuestra tabla Rendiciónes con los datos correspondientes (Fecha e importe total de la misma).
El ID de la empresa lo obtenemos consultando nuestra tabla de Empresas.

Pagos:

Notamos que existen campos que se utilizan para identificar los Pagos:
[Pago_nro, Pago_Fecha, FormaPagoDescripcion, Total]
Para dicho caso el Id del cliente lo obtenemos consultando a nuestra tabla de Clientes, y para el caso del Id de sucursal de la tabla Sucursales previamente ya cargada. En el caso se usuario se consideró uno solo que corresponde al único usuario cargado por ahora en el sistema.

Facturas:

Encontramos ciertos campos referidos específicamente a las facturas:
[Nro_Factura, Factura_Fecha, Factura_Fecha_Vencimiento, Rendicion_Nro, Pago_nro, Factura_Total]
Aquellos registros con número de factura similares fueron agrupados ya que consideramos que hacen referencia a una misma factura.
Luego, a partir del DNI del cliente podemos consultar la tabla clientes y así obtener su ID para mantener la referencia. De esta forma también hicimos algo similar para el ID de empresas, pero en este caso con el Cuit y el nombre de la empresa.
Finalmente, para cada registro (correspondiente a esta factura) de la rendición y el pago se tomó en consideración el más elevado.

Items:

Notamos que existen campos que se utilizan para identificar a los ítems correspondientes a la factura:
[Nro_Factura, ItemFactura_Cantidad, ItemFactura_Monto]
Dicha información se encuentra ordenada previamente de manera ascendente para la carga de la tabla.

APLICACIÓN DE ESCRITORIO

Login y Seguridad

Al acceder a la aplicación en primer instancia se mostrara el form correspondiente al logueo del usuario, el cual se solicitara su nombre de usuario seguido de una contraseña alojada en la BD con el hash256; como se la funcionalidad no se encuentra agregada los usuarios deben registrarse mediante la base de datos. El usuario al acceder al mismo si posee mas de un rol y/o una sucursal se lanzara otro form en el cual escojera a que sucursal desea ingresar y con que rol desea hacerlo, los datos obtenidos serán almacenados en una clase del tipo Singleton que contendrá su identificador univoco, el identificador del rol con el cual accedió, su nombre de usuario, la sucursal a la cual pertenece y las funcionalidades del rol con el cual accedió. Cuando un Usuario realiza 3 intentos de logueo fallido por contraseña el mismo queda deshabilitado automáticamente, solo pudiendo deshabilitarse por BD.

Rol de Usuario

Esta funcionalidad permite crear, dar de baja y modificar roles. Para crear un rol, el sistema carga todas las funcionalidades existentes junto a un checkBox para cada una y un campo libre para que el Administrador ingrese el nombre del nuevo rol. Al momento de grabar los datos de este rol, se graba también su relación con las funcionalidades cuyo checkBox se encuentra marcado. Para dar de baja un rol, el sistema carga todos los roles que se encuentren habilitados en un comboBox y al momento de grabar el cambio deseado se da de baja de forma lógica; es decir, cambiando el campo rol_activo de la tabla Rol por 0; el rol que se haya seleccionado con el comboBox.

Para la modificación de los roles primero se obtienen los roles al cual el usuario puede acceder y si tiene los permisos necesrios para modificar los roles y se los almacena en un comboBox listos para seleccionar, una vez seleccionado se pasa a otro form donde se cargan todas funcionalidades del sistema que se verán reflejados en un comboBox junto con un checkBox que mostrara el estado de las mismas para dicho rol elegido pudiendo activar y desactivar las funcionalidades incluso fuera del comboBox existe un checkBox con la funcionalidad de inhabilitar el rol con el cual estamos accediendo.

Abm Clientes

Esta funcionalidad permite crear, dar de baja y editar clientes. El crear cliente muestr a una serie de campos a completar por el usuario que quiera crear el cliente. Como re stricción obliga a que todos los campos estén completos a excepción del teléfono, piso y departamento. Al ingresar un nuevo cliente el sistema verifica que el campo email sea único en la tabla de clientes.

Al dar de baja y editar clientes se llegan por el mismo lugar: el filtrar cliente. Este filtro de cliente es un form donde se muestran todos los clientes disponibles a modificar, una vez seleccionada la fila donde se encuentra el cliente al cual queremos dar de baja se procede a presionar el botón editar. Las búsquedas se pueden realizar mediante el nombre del cliente, su apellido o su DNI. Recordar que esta funcionalidad dependerá del rol ingresado y si el usuario posee la funcionalidad para editar los clientes.

Abm Empresas

Posee las mismas funcionalidades que cliente a excepción que una empresa el botón para modificar la misma esta contenido dentro de la fila en la última columna, esta se mostrara si el usuario del sistema tiene los permisos suficientes para modificarla.
La modificación de una empresa se podrá realizar siempre y cuando los campos obligatorios estén completos, al igual que la misma form posee un check que determina si la empresa está habilitada o deshabilitada. Cuando una empresa quiere ser pasada a deshabilitada se verifican que todas las facturas que fueron pagadas hayan sido rendidas, en caso contrario se mostrara un MessageBox informando al usuario que la empresa no puede ser inhabilitada por poseer facturas pendientes de rendición.

Abm Sucursales

Posee las mismas características también que empresas salvo que no verifica si las facturas que posee están rendidas. Solo si al editar se ve modificado el campo código postal que no exista otra sucursal con el mismo.

Abm Facturas

Al dirigirse a dicha funcionalidad se le mostrara un form en el cual se listaran las facturas. En el caso de querer agregar alguna se presionara el botón agregar contenido dentro del mismo form. Dentro de agregar se pueden visualizar los campos para buscar el cliente al cual se le desea agregar una factura, en el caso de no existir debe ser creado de la sección clientes antes de poder realizar una alta de una factura a su nombre. Se puede visualizar un dataGrid en donde se guardaran los Items cargados para dicha factura, colocando el monto y la cantidad.
La baja lógica de una factura estará limitada si la misma no ha sido pagada, ya que según el enunciado propuesto solo se pueden dar de baja aquellas que no están rendidas y/o pagadas.

Registro de Pago

Cuando un cliente solicita el pago de una factura el Usuario accede a la sección de registrar un pago en el cual se ingresara la factura que desee pagar, buscando por

filtros el cliente que desea ingresar el pago y la factura a la cual desea pagar, adjuntando mediante un comboBox el medio de pago en el cual desea realizarlo.

Rendiciones

A la hora de realizar dicha funcionalidad se deben tener los permisos apropiados para poder realizarla, si se lo posee se podrá visualizar el botón que le permitirá acceder a dicha funcionalidad, en caso contrario el botón estará desactivado.

Se accede al form correspondiente que muestra comboBox en el cual seleccionaremos la empresa a la cual queremos rendir las facturas. Cuando se realiza la rendición de las facturas serán todas aquellas sin cobrar anteriores a un dia del mes especificado de cada empresa, exceptuando las facturas que estes inhabilitadas. Es decir que si la empresa solicita que se le rindan las facturas el 5 dia de cada mes, la aplicación buscara las facturas pendientes de rendición anteriores al dia del mes en cuestión.

Devoluciones

Al realizar una devolución el usuario deberá tener permisos administrativos para realizarla, se comprobaran que el tipo de devolución denotado por 2 radiobuttom que al ser seleccionado se efectúan sobre facturas que no hayan sido rendidas, en el caso de que la factura este rendida se deberán poseer permisos especiales el cual le permitan realizar dicha operación.

Listado estadístico

Esta funcionalidad permite realizar 4 consultas estadísticas. Para realizar una consulta se deben ingresar los datos año y trimestre, seguido a esto hay que elegir la consulta a realizar.
Esta funcionalidad está diseñada de manera tal que al ingresar una sola vez la información del año y el trimestre, estos datos se pueden utilizar para las 4 consultas. Cada vez que se realiza una consulta se puede volver al menú de estadísticas conservando la información de año y trimestre.

Interfaz con la base de datos

Se utiliza para la misma una clase llamada conexión que nos proporciona una clase del tipo facade que por detrás levanta una conexión del tipo SqConnection que nos permite relizar las operación con una interfaz mas amigable y comprensible para el codificador.

Se disponen de los siguientes métodos

1) Conexión: El contructor que recibe es vacio. Ya que por detrás levanta una SqlConeccection con la localización donde se encuentra la StringConnection.

2) Commandear: Es el encargado de levanter de levanter la consulta realizada a la BD junto con su conexion.

3) AbrirConexion y CerrarConexion: Son los encargados de avisarle a la BD cuando se realizara una operación con ella y cuando finaliza la misma.

4) Ejecutar: Se utiliza especialmente cuando no queremos obtener parámetros de retorno, como por ejemplo en los UPDATE o INSERT. El cual abre una conexión, la comandea, la ejectuta mediante ExecuteNonQuery utilizados para avisarle a la BD que queremos realizar una transacción.

5) Leer: Utilizado para cuando queremos obtener datos de la BD, donde abre una conexión, la comandea y ejecuta el ExecuteReader del SqlCommand que nos crea un objeto del tipo SqlDataReader y lo setea al lector (atributo donde guardamos lo que lee).

6) LeerReader: es la interfaz que utilizamos para decir si tenemos aun datos para seguir leyendo de la base de datos. Cuando esta contiene el valor falso significara que no contenemos mas datos para leer.

Objetos extras de Base de datos

Las vistas se crearon para evitar realizar joins hacia la tabla utilizando un querys extensos cada ves que se realiza una consulta. También se implementaron para limitar la cantidad de campos que recibe la aplicación, tanto en las sucursales obteniendo la dirección como en las facturas.

Decisiones del grupo

En El ABM de clientes, el enunciado especifica que: *“El mail es un dato único, por ende no pueden existir 2 clientes con el mismo mail” ,* habiendo leído esto automáticamente la decisión del grupo fue utilizar el constraint UNIQUE para validar el mail, pero al momento de migrar datos a la tabla de clientes nos encontramos con un problema, dos clientes tenían el mismo correo electrónico en la tabla maestra. Esto hizo que no podamos utilizar la validación UNIQUE ya que no se podían dejar datos afuera de la base. La manera de solucionarlo fue generando una validación en la aplicación. Antes de agregar un cliente verifica que el correo a cargar no exista.

En el ABM de Rendiciones habíamos decidido según el enunciado que había que buscar las facturas sin rendir del ultimo mes. Esto no se pudo realizar debido que en la

base no esta esa información. Por lo cual la decisión que tomamos finalmente fue poner todas las facturas sin rendir previas al día de la fecha.
Con respecto a la fecha de rendición interpretamos que solo se puede realizar la rendición de una fecha fija del mes, entonces generamos el campo emp_dia_rend en la tabla EMPRESAS para que cada empresa tenga su día de rendición configurado en su registro base.