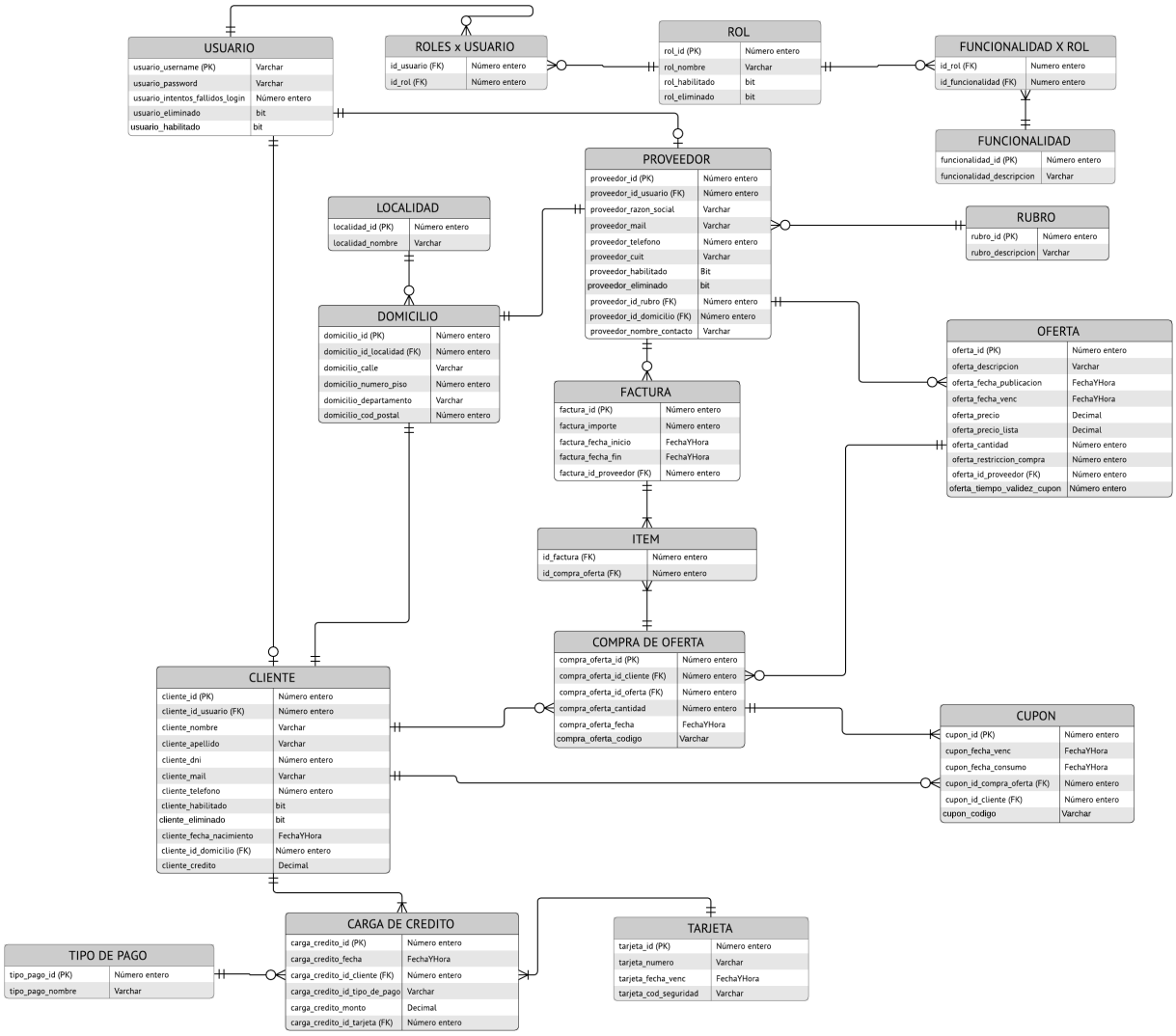
Gestión de Datos
Trabajo Práctico
2º Cuatrimestre 2019
FRBA Ofertas

Grupo: NO_LO_TESTEAMOS_NI_UN_POCO

Nombre	Legajo
Szwimer, Daiana Kalanit	1632553
Peccia, Diego	1633156
Posru, Marina	1590133

Índice

DER


USUARIO
usuario_username (PK) Varchar
usuario_password Varchar
usuario_intentos_fallidos_login Número entero
usuario_eliminado bit
usuario_habilitado bit
ROLES x USUARIO
id_usuario (FK) Número entero
id_rol (FK) Número entero
ROL
rol_id (PK) Número entero
rol_nombre Varchar
rol_habilitado bit
rol_eliminado bit
FUNCIONALIDAD X ROL
id_rol (FK) Numero entero
id_funcionalidad (FK) Numero entero
FUNCIONALIDAD
funcionalidad_id (PK) Número entero
funcionalidad_descripcion Varchar
PROVEEDOR
proveedor_id (PK) Número entero
proveedor_id_usuario (FK) Número entero
proveedor_razon_social Varchar
proveedor_mail Varchar
proveedor_telefono Número entero
proveedor_cuit Varchar
proveedor_habilitado Bit
proveedor_eliminado bit
proveedor_id_rubro (FK) Número entero
proveedor_id_domicilio (FK) Número entero
proveedor_nombre_contacto Varchar
LOCALIDAD
localidad_id (PK) Número entero
localidad_nombre Varchar
RUBRO
rubro_id (PK) Número entero
rubro_descripcion Varchar
DOMICILIO
domicilio_id (PK) Número entero
domicilio_id_localidad (FK) Número entero
domicilio_calle Varchar
domicilio_numero_piso Número entero
domicilio_departamento Varchar
domicilio_cod_postal Número entero
OFERTA
oferta_id (PK) Número entero
oferta_descripcion Varchar
oferta_fecha_publicacion FechaYHora
oferta_fecha_venc FechaYHora
oferta_precio Decimal
oferta_precio_lista Decimal
oferta_cantidad Número entero
oferta_restriccion_compra Número entero
oferta_id_proveedor (FK) Número entero
oferta_tiempo_validez_cupon Número entero
FACTURA
factura_id (PK) Número entero
factura_importe Número entero
factura_fecha_inicio FechaYHora
factura_fecha_fin FechaYHora
factura_id_proveedor (FK) Número entero
ITEM
id_factura (FK) Número entero
id_compra_oferta (FK) Número entero
COMPRA DE OFERTA
compra_oferta_id (PK) Número entero
compra_oferta_id_cliente (FK) Número entero
compra_oferta_id_oferta (FK) Número entero
compra_oferta_cantidad Número entero
compra_oferta_fecha FechaYHora
compra_oferta_codigo Varchar
CLIENTE
cliente_id (PK) Número entero
cliente_id_usuario (FK) Número entero
cliente_nombre Varchar
cliente_apellido Varchar
cliente_dni Número entero
cliente_mail Varchar
cliente_telefono Número entero
cliente_habilitado bit
cliente_eliminado bit
cliente_fecha_nacimiento FechaYHora
cliente_id_domicilio (FK) Número entero
cliente_credito Decimal
CUPON
cupon_id (PK) Número entero
cupon_fecha_venc FechaYHora
cupon_fecha_consumo FechaYHora
cupon_id_compra_oferta (FK) Número entero
cupon_id_cliente (FK) Número entero
cupon_codigo Varchar
TIPO DE PAGO
tipo_pago_id (PK) Número entero
tipo_pago_nombre Varchar
CARGA DE CREDITO
carga_credito_id (PK) Número entero
carga_credito_fecha FechaYHora
carga_credito_id_cliente (FK) Número entero
carga_credito_id_tipo_de_pago Varchar
carga_credito_monto Decimal
carga_credito_id_tarjeta (FK) Número entero
TARJETA
tarjeta_id (PK) Número entero
tarjeta_numero Varchar
tarjeta_fecha_venc FechaYHora
tarjeta_cod_seguridad Varchar

Modelo de datos + Decisiones

Tablas

Usuario:

Tabla con todos los usuarios del sistema. En la migración, dada que los datos del sistema viejo no contaban con username y contraseña, decidimos setear como username CUIT en caso de los proveedores y DNI en caso de los clientes, la contraseña es la misma que es username

Columnas:

usuario_username varchar(64) - PK: es el username del usuario y debe ser único, es el identificador

usuario_password varchar(500): es la contraseña del usuario encriptada

usuario_habilitado bit: por default es 1 dado que todo usuario inicialmente está habilitado. Cuando supera los intentos fallidos del login (3), pasa a valer 0.

usuario_intentos_fallidos_login int: por default vale 0 y es la cantidad de intentos fallidos en el login

usuario_eliminado bit: por default vale 0. Cuando se setea en 1 es como eliminarlo, es la baja lógica

Rol:

Tabla con todos los roles del sistema. Cada rol tiene muchas funcionalidades. En la migración, creamos los roles que dice el enunciado: administrativo, proveedor, cliente y administrador general.

Columnas:

rol_id int - PK: es el identificador del rol. Se autoincrementa.

rol_nombre varchar(64): es el nombre del rol. No pueden haber dos roles con el mismo nombre.

rol_habilitado bit: por default es 1, ya que todo rol inicialmente está habilitado

rol_eliminado bit: por default es 0, ya que todo rol inicialmente no está borrado.

Funcionalidad:

Tabla con todas las funcionalidades del sistema. En la migración creamos las funcionalidades que indica el enunciado: ABM Rol, ABM Cliente, ABM Proveedor, Baja y Modificación Usuario, Carga de Crédito, Confección y Publicación de Ofertas Compra de Ofertas, Entrega/Consumo de Ofertas, Facturación a Proveedor y

Listado Estadístico. Dado que todo usuario tiene acceso al login, alta de usuario y a cambiar la contraseña, no lo tomamos como funcionalidades ya que todos tienen acceso y no tiene sentido validar si un usuario dado tiene acceso a eso.

Columnas:

funcionalidad_id int - PK: es el identificador de la funcionalidad. Se autoincrementa.

funcionalidad_descripcion varchar(64): es la descripción de la funcionalidad.

RolesxUsuario:

Tabla intermedia entre Usuario y Rol porque la relación entre estos es de muchos a muchos. Dado que en el enunciado menciona la posibilidad de que un usuario tenga más de un rol en un futuro, decidimos modelarlo así para que, cuando pase eso, no haya que hacer ningún cambio a nivel base de datos. Las dos columnas que tiene forman la PK.

Columnas:

rolesxusuario_id_usuario varchar(64) - FK: es el id del usuario

rolesxusuario_id_rol int - FK: es el id del rol

FuncionalidadxRol:

Tabla intermedia entre Funcionalidad y Rol porque la relación entre estos es de muchos a muchos. Las dos columnas que tiene forman la PK. El administrador general tiene acceso a todas las funcionalidades, el proveedor tiene acceso a Confección y Publicación de Ofertas y Entrega/Consumo de Ofertas, el cliente tiene acceso a Carga de Crédito y Compra de Ofertas, y el administrativo tiene acceso a todas las funcionalidades menos ABM Rol, Compra de Ofertas y Entrega/Consumo de Ofertas.

Columnas:

funcionalidadxrol_id_rol int - FK: es el id del rol

funcionalidadxrol_id_funcionalidad int FK - es el id de la funcionalidad

Localidad:

Tabla que contiene todas las localidades de los clientes y proveedores.

Columnas:

localidad_id int - PK: es el identificador y es autoincrementable

localidad_nombre varchar(64): es el nombre de la localidad

Domicilio:

Tabla que contiene todas las direcciones de los clientes y proveedores. En la migración, como no conocemos el piso, departamento ni el código postal, ingresamos 0 o ‘’.

Columnas:

domicilio_id int - PK: es el identificador y es autoincrementable

domicilio_id_localidad int - FK: es el id de la localidad del domicilio

domicilio_calle varchar(64): es la calle del domicilio

domicilio_numero_piso int: es el piso, si no tiene, es NULL

domicilio_departamento varchar(20): es el departamento, si no tiene, es NULL

domicilio_codigo_postal int: es el código postal del domicilio

Cliente:

Tabla que contiene todos los clientes del sistema. En la columna cliente_credito, en la migración, le ponemos como valor 0 y luego calculamos todas las cargas de crédito que hizo cada cliente y luego le restamos lo gastado en cada compra. En caso de que un cliente no haya hecho ninguna carga, lo dejamos en 0. Cada vez que un cliente realiza una compra, se le resta el monto de la misma a la columna recién mencionada. Asumimos que si dos clientes tienen mismo DNI y mail son "gemelos", por eso pusimos la constraint de unique. Para no perder los datos del sistema anterior, decidimos que si hay clientes con mismo mail, lo dejamos así, pero, si se da de alta un cliente, se valida que no haya otro cliente con el mismo mail. Cuando creamos un cliente nuevo, en el campo cliente_credito guardamos el valor 200 ya que a todo cliente nuevo se le debitan $200. A estos usuarios se les asigna el rol de cliente.

Columnas:

cliente_id int - PK: es el identificador y es autoincremental

cliente_id_usuario varchar(64) - FK: es el id del usuario que corresponde al cliente

cliente_nombre varchar(64): es el nombre del cliente

cliente_apellido varchar(64): es el apellido del cliente

cliente_dni int: es el DNI del cliente. Agregamos una constraint de unique sobre esta columna ya que no existen dos personas distintas con el mismo DNI

cliente_mail varchar(64): es el mail del cliente.

cliente_telefono int: es el teléfono del cliente.

cliente_habilitado bit: por default vale 1 ya que todo cliente inicialmente está habilitado. Cuando se deshabilita, vale 0

cliente_eliminado bit: por default vale 0 ya que todo cliente inicialmente no está borrado. Cuando vale 1 es borrarlo lógicamente

cliente_fecha_nacimiento datetime: es la fecha de nacimiento del cliente

cliente_id_domicilio int - FK: es el id del domicilio del cliente

cliente_credito int: es el crédito actual del cliente

Tarjeta:

Tabla que contiene todas las tarjetas que fueron usadas para realizar una carga de crédito.

Columnas:

tarjeta_id int - PK: es el identificador de la tarjeta y es autoincrementable

tarjeta_numero varchar(64): es el número de tarjeta

tarjeta_fecha_venc datetime: es la fecha de vencimiento

tarjeta_cod_seguridad varchar(64): es el código de seguridad

Tipo_Pago:

Tabla que contiene los tipos de pagos mencionados en el enunciado: crédito, efectivo y débito.

Columnas:

tipo_pago_id int - PK: es el identificador y es autoincrementable

tipo_pago_nombre varchar(64): es el nombre del tipo de pago

Carga_Credito:

Tabla que contiene todas las cargas de crédito realizadas por clientes en el sistema. En la migración, como no tenemos la información de la tarjeta, en carga_credito_id_tarjeta ponemos NULL.

Columnas:

carga_credito_id int - PK: es el identificador y es autoincrementable

carga_credito_id_cliente int - FK: es el id que identifica al cliente que hizo la carga

carga_credito_id_tipo_pago int - FK: es el id que identifica el tipo de pago con el que se hizo la carga

carga_credito_id_tarjeta int - FK: es el id que identifica la tarjeta con la que se hizo la carga

carga_credito_fecha datetime: es la fecha en la que se hizo la carga (según el archivo de configuración)

carga_credito_monto int: es el monto de la carga

Rubro:

Tabla que contiene los rubros de los proveedores del sistema.

Columnas:

rubro_id int - PK: es el identificador y es autoincrementable

rubro_descripcion varchar(64): es la descripción del rubro

Proveedor:

Tabla que contiene los proveedores del sistema. En la migración, como no tengo el dato de mail y nombre de contacto, inserto ‘’. A estos usuarios se les asigna el rol de proveedor.

Columnas:

proveedor_id int - PK: es el identificador, autoincrementable

proveedor_id_usuario varchar(64) - FK: es el id que corresponde al usuario del proveedor

proveedor_mail varchar(64): es el mail

proveedor_telefono int: es el teléfono

proveedor_cuit varchar(64): es el CUIT. Tiene la constraint de unique ya que no existen dos proveedores distintos con un mismo CUIT.

proveedor_habilitado bit: por default vale 1 ya que todo proveedor inicialmente está habilitado.

proveedor_eliminado bit: por default vale 0 ya que todo proveedor inicialmente no está borrado.

proveedor_id_rubro int - FK: es el id del rubro del proveedor.

proveedor_id_domicilio int - FK: es el id del domicilio.

proveedor_nombre_contacto varchar(64): es el nombre de contacto

proveedor_razon_social varchar(64): es la razón social. Debe ser única, por lo tanto tiene una constraint de unique

Oferta:

Tabla que contiene todas las ofertas creadas por proveedores. En la migración, dado que no tenemos el dato de oferta_restriccion_compra, lo seteamos con el mismo valor que oferta_cantidad, entonces es lo mismo que no haya restricción, ya que no se puede comprar más del stock disponible. Como no sabemos el oferta_tiempo_validez_cupon, lo seteamos en 0.

Columnas:

oferta_id int - PK: es el identificador de la oferta, autoincrementable

oferta_descripcion varchar(64): es la descripción de la oferta

oferta_fecha_publicacion datetime: es la fecha de publicación de la oferta (según el archivo de configuración)

oferta_fecha_venc datetime: es la fecha de vencimiento de la oferta

oferta_precio decimal(12, 2): es el precio de la oferta (el precio rebajado)

oferta_precio_lista decimal(12, 2): es el precio original de la oferta

oferta_cantidad int: es el stock disponible. Cada vez que se realiza una compra, se resta la cantidad de ofertas compradas

oferta_restriccion_compra int: es la cantidad límite que puede comprar un cliente por oferta

oferta_id_proveedor int - FK: es el id del proveedor dueño de la oferta

oferta_tiempo_validez_cupon int: es el tiempo que es válido el cupón de una oferta una vez generado en días. Por ejemplo, si vale 10 y se hace una compra de la oferta, entonces el cupón de la compra de esa oferta se vencerá en 10 días.

Compra_Oferta:

Tabla que contiene todas las compras de ofertas del sistema. En la migración, asumimos que cada compra es de 1 unidad, por lo tanto, en compra_oferta_cantidad guardamos 1. Dado que el código de compra oferta no es único en cada compra, a los códigos duplicados le concatenamos “-1” o “-2”

Columnas:

compra_oferta_id int - PK: es el identificador de la compra, autoincrementable

compra_oferta_id_cliente int - FK: es el id del cliente que realizó la compra

compra_oferta_id_oferta int - FK: es el id de la oferta la cual se compró

compra_oferta_cantidad int: es la cantidad de ofertas que el cliente compra

compra_oferta_fecha datetime: es la fecha en la que se realiza la compra (según el archivo de configuración)

compra_oferta_codigo varchar(64): es el código de compra

Cupon:

Tabla que contiene todos los cupones generados para cada compra oferta. Si el valor de cupon_fecha_consumo es NULL y el de cupon_id_cliente es NULL, entonces el cupón aún no se consumió. En la migración, asumimos que el código de compra es igual al código de cupón. Ya que no tenemos la fecha de vencimiento del cupón, elegimos usar la fecha de vencimiento de la oferta. También asumimos que el cliente que compró la oferta es el mismo que la retiró. Aquellas compras que no fueron retiradas, le generamos el cupón correspondiente para que puedan ser retiradas.

Columnas:

cupon_id int - PK: es el identificador del cupón, autoincrementable

cupon_id_cliente int - FK: es el id del cliente que usó el cupón, es decir, al que le fue entregado la oferta de la compra correspondiente.

cupon_id_compra_oferta int - FK: es el id de la compra

cupon_fecha_venc datetime: es la fecha de vencimiento del cupón. Se calcula en base a la fecha que fue hecha la compra y se le suma el valor de oferta_tiempo_validez_cupon de la oferta correspondiente

cupon_fecha_consumo datetime: es la fecha en la que se consume el cupón (según el archivo de configuración)

cupon_codigo varchar(64): es el código del cupón. Es único, para garantizarnos esto, usamos un stored procedure para generarlo.

Factura:

Tabla que contiene todas las facturas generadas en el sistema. En la migración, para cada factura, tomamos al número de factura como id, la fecha de inicio como la fecha menor de compra de oferta que aparece en el sistema viejo con ese número de factura. Mirando los datos, nos dimos cuenta que la fecha que figura como fecha de factura en el sistema viejo es la fecha más grande de cada factura siempre (en comparación con las fechas de compra), entonces usamos esa columna. El importe de la factura lo calculamos sumando todos los importes de cada factura en particular.

Columnas:

factura_id int - PK: es el identificador de la factura. En la migración, como mencionamos recientemente, es el número de factura. Una vez finalizada la migración, es autoincrementable

factura_importe decimal(12, 2): es el importe total de la factura

factura_fecha_inicio datetime: es la fecha de inicio de la factura

factura_fecha_fin datetime: es la fecha de finalización de la factura

factura_id_proveedor int - FK: es el id del proveedor al cual se factura

Item:

Tabla intermedia entre Factura y Compra_Oferta, ya que la relación entre ellos es de muchos a muchos. Las dos columnas que tiene forman la PK.

Columnas:

item_id_factura int - FK: es el id de la factura

item_id_compra_oferta int - FK: es el id de la compra de la oferta

Stored Procedures

cliente_comprar_oferta:

Valida que un cliente tenga crédito suficiente para realizar una compra, que el cliente no esté deshabilitado, que la cantidad que desea comprar no supere la restricción de cantidad de ofertas por compra por cliente, y que haya suficiente cantidad de ofertas en stock. Si se pasan todas estas validaciones, genera el código de compra -único, no puede haber otra compra con ese mismo código- y el código de cupón -único, no puede haber otro cupón con ese mismo código-. Luego inserta en Compra_Oferta la compra realizada con los datos correspondientes, actualiza el stock de la oferta y lo disminuye en la cantidad que el cliente compró, le resta el importe al crédito del cliente y genera el cupón correspondiente a la compra.

Parámetros:

@id_cliente int - es el id del cliente que realizó la compra

@id_oferta int - es el id de la oferta que se compra

@fecha datetime - es la fecha en la que se realiza la compra

@cantidad int - es la cantidad de ofertas que se compra

@codigo varchar(64) output - es el código de compra generado

proveedor_entrega_oferta:

Valida que exista un cliente con el DNI que se ingresa, que el proveedor que está entregando la oferta sea el mismo que es dueño de la oferta, que el cupón no haya sido canjeado, y que el cupón no se haya vencido. Si se pasan todas estas validaciones, actualiza el cupón correspondiente (según el código que le llega por parámetro) y le setea fecha de consumo y id de cliente.

Parámetros:

@fecha_consumo datetime - es la fecha de consumo del cupón

@id_proveedor int - es el proveedor que desea entregar la oferta

@dni_cliente int - es el DNI del cliente que va a retirar la oferta

@codigo_cup varchar(64) - es el código del cupón

facturacion:

Inserta una fila en la tabla Facturacion y las correspondientes en la tabla Item. Calcula el importe de todas las ofertas compradas de determinado proveedor en un rango determinado de fechas y devuelve el id de la factura. Si no hay compras en ese período o hay compras que ya fueron facturadas, lanza un error.

Parámetros:

@fecha_inicio datetime - fecha de inicio de facturación

@fecha_fin datetime - fecha de finalización de facturación

@id_proveedor int - id del proveedor al cual se le quiere facturar

@id_factura int output - id de la factura generada

Funciones

obtener_ofertas_factura:

Devuelve todas las ofertas que pertenecen a una factura basándose en las compras que están en la factura.

Parámetros:

@id_factura int - el id de la factura de la cual se desea obtener las ofertas cuyas compras fueron facturadas en esa factura

obtener_codigos_cupones:

Devuelve todos los códigos de cupones que fueron generados para las compras de las ofertas de determinado proveedor.

Parámetros:

@id_proveedor int - el proveedor del cual se quieren obtener los códigos de los cupones

top_5_mayor_porcentaje:

Devuelve el CUIT, razón social, nombre de contacto, y descuento de los 5 proveedores que ofrecieron más descuento -en promedio- en sus ofertas en el año y semestre correspondiente.

Parámetros:

@anio int - el año del cual se quieren ver las estadísticas

@semestre int - si vale 1 toma el primer semestre, si vale 2 toma el segundo

top_5_mayor_facturacion:

Devuelve el CUIT, razón social, nombre de contacto, y facturación en el año y semestre correspondiente de los 5 proveedores que más facturaron en el año y semestre correspondiente.

Parámetros:

@anio int - el año del cual se quieren ver las estadísticas

@semestre int - si vale 1 toma el primer semestre, si vale 2 toma el segundo

Aplicación

Archivo de configuración

Ubicación: src\FrbaOfertas\FrbaOfertas\App.config

Detalle:

- La fecha del sistema con el formato:

<add key="MyDate" value="13-11-2019 00:00:00.000"/>

Disponible para ser editada reemplazando los valores destacados en naranja respetando el formato dd-MM-yyyy HH:mm:ss.fff .

- Los parámetros de conexión a la base de datos con el formato:

<add name="dbOfertas"connectionString="DataSource=localhost\SQLSERVER2012; Initial Catalog=GD2C2019;uid=gdCupon2019;password=gd2019"/>

Aplicación Desktop

Decidimos aprovechar las características de la tecnología ofrecida por la cátedra orientada a objetos para organizar el sistema en diferentes clases que representan el negocio.

Cada form tiene un modelo asociado, dentro de la cual se encuentran los distintos elementos. Utilizamos **textBox** para el ingreso de cadenas de caracteres, **botones** para llevar a cabo una acción, **comboBox** para seleccionar un solo elemento de una lista reducidas de valores, **checkedListBox** que permite tener un elemento, varios o ninguno seleccionado, **checkBox** para determinar si una condición determinada está activada o desactivada, **dateTimePicker** para seleccionar una fecha, **DomainUpDown** para seleccionar dentro de un rango de valores, **errorProvider** para identificar campos obligatorios o violaciones en los formatos de datos que debian ingresar, **MessageBox** para mostrar información destacada al usuario, entre otros.

En el caso de los listados, se obtienen los datos a través de **SqlDataAdapter** que actúa como puente entre DataSet y la base de datos. El **DataSet** es una representación de datos en memoria que proporciona una modelo independientemente del origen de datos. Y por último desde el DataSet se vuelcan los datos a un **DataGridView** para poder buscar, mostrar y editar datos originarios de distintas tablas dentro de una única.

A la hora de persistir o recuperar datos persistidos utilizamos **SqlCommands** para representar store procedures, functions o instrucción de Transact-Sql de tipo Select, Insert, Update o Delete. Con la utilización del método ExecuteReader del SQLCommand se crear

un **SQLDataReader** que este objeto mantienen la conexión SQLConnection ocupada y abierta mientras realiza la lectura.

- Form1

El form1 representa al Login. Es la primer pantalla con la que se encuentra el usuario al ingresar al sistema. Puede seguir dos caminos posibles: ingresar username y password que ya se encuentren dentro del sistema o registrarse como usuario.

- Registrar usuario

El registro de usuario debe ser utilizado solo cuando es el primer ingreso al sistema de un usuario de tipo cliente o proveedor.

Establecemos como default rol cliente para usuarios de tipo cliente y rol proveedor para usuarios de tipo proveedor.

Para cargar los datos correspondientes a cada tipo de usuario, agregamos a través de un panel el form Alta de abm cliente o el form Alta de abm proveedor según corresponda.

- Menú

El menú es un form dinámico. Su estructura es un TableLayoutPanel que se reconfigura dependiendo de las funcionalidades correspondientes con el rol de usuario logueado.

Para todos los usuarios independientemente de su rol, dispone de la funcionalidad de cambiar contraseña y cerrar sesión en el menú.

Para el rol administrador general se agrega además de las funcionalidades específicas del enunciado, la funcionalidad de baja y modificación de usuario.

- Barra de opciones

La barra de opciones es MenuStrip que permite volver al menu principal o cerrar sesion desde todas pantallas exceptuando el Login, el Menu y el Registro de usuario.

- ABM Rol

Permite dar de alta un rol y mostrar el listado de roles. Dentro del listado se puede: modificar, eliminar o asignar un rol.

Sobre asignar un rol, decidimos que un usuario cuyo rol original fue inhabilitado se pueda asignar alguno de los roles habilitados, salvo rol cliente y rol proveedor que se tratan de forma particular en otra instancia. También permite crear un nuevo usuario con username , password y rol desde esta pantalla.

- ABM Cliente

Permite dar de alta un cliente y mostrar el listado de clientes. Dentro del listado se puede: filtrar, modificar o eliminar un cliente.

Sobre clientes gemelos, decidimos validar que no pueden existir dos clientes con diferente username pero igual dni, ya que es un número único de identidad para residentes de nuestro país.

- ABM Proveedor

Permite dar de alta un proveedor y mostrar el listado de proveedores. Dentro del listado se puede: filtrar, modificar o eliminar un proveedor.

- Carga de crédito

Permite la carga de crédito a la cuenta de un cliente.

- Crear oferta

Permite a proveedores armar y publicar ofertas dentro del sistema.

En caso que un administrativo ingrese una oferta para un proveedor, se habilita en el form un campo donde, a través de un listado, puede seleccionar a qué proveedor le corresponde la oferta.

Sobre la tiempo de validez de un cupón, decidimos agregar un campo para definir el periodo de validez del cupón antes de su vencimiento.

- Compra oferta

Permite a clientes comprar una oferta publicadas por los diferentes proveedores.

En caso que un administrativo compre una oferta para un cliente, se habilita en el form un campo donde, a través de un listado, puede seleccionar a qué cliente comprara esa oferta.

En el listado de ofertas es dinámico, irán apareciendo aquellas ofertas con fecha de publicación y fecha de vencimiento mayor o igual a la fecha del sistema.

El selector de cantidad de oferta a comprar tiene como límite máximo la restricción de compra propia de la oferta seleccionada.

Al momento de confirmar la compra, aparece un mensaje informativo con el código de compra.

- Entrega oferta

Permite a un proveedor dar de baja un cupón y asignarle el cliente que lo retira.

En el listado de cupones, aparecerán aquellos cupones del del proveedor logueado que aún no fueron canjeados.

En caso de un administrativo, se habilita en el form un campo donde, a través de un listado, puede seleccionar cual es el proveedor al cual se le dará de baja una cantidad de oferta.

- Facturación a proveedor

Permite a un administrativo facturar a un proveedor.

En el listado de ofertar, se mostrará aquellas que fueron adquiridas durante el período especificado anteriormente. De no existir ofertas adquiridas en ese periodo, aparecerá el nro de factura y el monto en cero junto con un mensaje informativo.

- Listado Estadístico

Permite consultar estadísticas sobre los porcentajes de descuento ofrecidos en sus ofertas y sobre facturación.

Se debe elegir entre entre 1er semestre (de Enero a Junio) o 2do semestre (Julio a Diciembre)