

Estrategia

Archivo de configuración

El archivo se encuentra en la raíz del proyecto y se llama app.config.

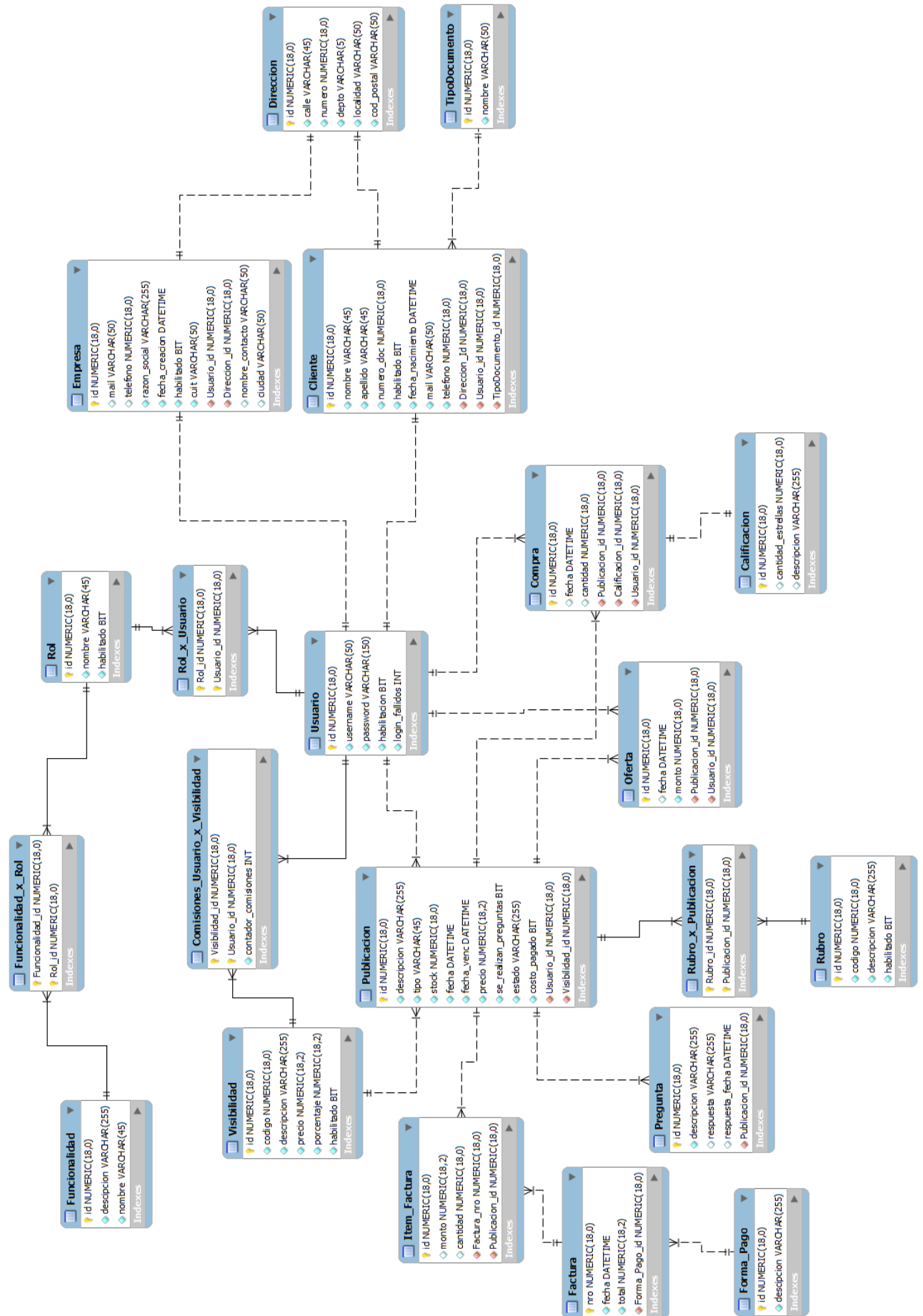
En el mismo se indica como fecha de inicio del sistema el 17/06/2014.

Por esa razón, como todas las publicaciones tienen fecha de vencimiento en el 2013, decidimos migrarlas como publicaciones finalizadas.

Der

Campos no exigidos en los requerimientos pero que son necesarios para simplificar la solución:

- #Usuario:login_fallidos: se usa para llevar un contador de cuántas veces se quiso registrar un usuario ingresando la contraseña de forma incorrecta.
- #Visibilidad:duración: este campo no está explícito en la tabla maestra pero se puede obtener a partir de otros datos (fecha_inicio y fecha_vencimiento de las publicaciones). Las fechas de vencimiento de las publicaciones se toman a partir de este dato.
- #Publicacion:costo_pagado: se usa para llevar un control de si fue o no pagado el costo por publicar. Sirve para saber si facturar o no el costo de publicación de esa publicación de forma sencilla.
- #Compra:facturada: se usa para llevar un control de si fue o no facturada la factura. Sirve para simplificar las consultas de saber que facturas faltan facturar.
- #Comisiones_Usuario_x_Visibilidad: esta tabla se usa para llevar una cuenta de todas las veces que un usuario facturó por determinada visibilidad. Se usa para saber cuando se le bonifica una factura a un usuario. Por cada usuario en el sistema y por cada visibilidad del sistema existe un campo.
- #Cliente#Empresa#Visibilidad:dado_de_baja: es un campo de un bit que permite saber si fue o no dado de baja en el sistema. Si fue dado de baja no existe posibilidad de volverlo a cambiar.
- Se diferencia el concepto de habilitado y dado de baja. La diferencia radica en que si se fue deshabilitado (poner la contraseña mal muchas veces, no facturar) se puede volver a habilitar mediante la cuenta del admin. En cambio, si se da de baja, el admin no puede volver a darlo de alta, es definitivo.



Migración

Consideraciones:

- No se dropa el esquema si existe porque no nos interesa que se borren todos los datos, sino que solamente los que vamos a utilizar. Además para eliminar el esquema necesitaríamos eliminar todo lo que tenía cargado previamente, cosa que es muy complicado distinguir.
- En los crear_bla bla se usa un SCOPE_IDENTITY() que lo que hace es recuperar el último id que fue agregado. Esto nos libera de tener que hacer un select top 1...
- En varios lugares se usa SET IDENTITY_INSERT LOS_SUPER_AMIGOS.NombreTabla ON. Lo que hace esto es permitir insertar valores en un campo identity. Cuando se inserte un nuevo campo el identity va a seguir funcionando a partir del último valor
- Decidimos que el telefono de todas lo clientes y todas las empresas sea 0 para simplificar las consultas.
- Decidimos que la localidad de todas las direcciones sea 'localidadMigrada' porque decidimos que no sea un valor null para simplificar las consultas.
- Decidimos que la ciudad de todas las empresas sea 'ciudadMigrada' porque decidimos que no sea un valor null para simplificar las consultas.
- Decidimos que a todos los usuarios se les setea el tipo de documento como 'DNI' para simplificar.
- Cuando se migra una visibilidad se le setea como duracion 7. Todas las visibilidades de la tabla migrada tienen de duración 7, por eso las seteamos como 7.
- Después de migrar todas las publicaciones, se checkea si su fecha de vencimiento es menor a la fecha actual del programa que es '17/06/2014'. Si lo es, se la setea como 'Finalizada'

Aplicación desktop

ABM Rol

Esta funcionalidad permite crear, dar de baja y editar 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 encuentre habilitados en un comboBox y al momento de grabar el cambio deseado se da de baja de forma lógica; es decir, cambiando el campo habilitado de la tabla Rol por 0; el rol que se haya seleccionado con el comboBox.

Para editar un rol, primero se accede a un buscador que permite filtrar en un listado los roles por habilitados y por deshabilitados. Desde este mismo listado se selecciona el rol que se desea editar y se nos abre un nuevo form. Este nuevo form es casi igual al de agregar rol, solo que al cargar la funcionalidades, algunas van a tener su checkBox marcado; esto se debe a que son las funcionalidades que ya posee el rol. También contiene un checkBox que no se habilita si el rol está habilitado; y en caso de que el rol este deshabilitado, si se habilita para que el Administrador pueda marcarlo y rehabilitar el rol.

La modificación del rol permite:

- cambiarle el nombre al rol
- agregar o quitar funcionalidades
- rehabilitar el rol

Estas 3 modificaciones pueden realizarse todas a la vez o también de forma individual. Esto dependerá de los datos modificados que ingrese el Administrador.

Login y seguridad

Mediante el login se puede iniciar sesion o ir hacia el form para crear un nuevo usuario.

Utilizamos una clase singleton para guardar la información del usuario mientras que tiene sesión iniciada.

Usamos el algoritmo de encriptación SHA256 para proteger la contraseña del usuario. Por defecto los usuarios migrados desde la tabla maestra tienen:

- username = 'USER' + id
- contraseña = "A"

Si un usuario inicia sesión con contraseña "A", entonces se asume que es el primer inicio de sesión y se le pide cambiar su contraseña. Esto funciona ya que tanto al cambiar la contraseña como al registrar un nuevo usuario, se solicita que la contraseña tenga por lo menos 8 caracteres, por lo que nunca podrá ser "A" a menos que se trate de un usuario migrado.

En caso de que un usuario ingrese 3 veces mal su contraseña, el mismo se deshabilita automáticamente y no podrá iniciar sesión.

En caso de que el usuario que inicia sesión tenga más de 1 rol asignado, entonces se debe elegir con que rol iniciar.

Luego se ingresa al menú principal.

Registro usuario

Hay 2 caminos por los cuales se puede llegar a registrar un usuario:

- En primer lugar cuando el mismo usuario se registra para poder empezar a usar la aplicación. En este caso se registra el usuario habiendo ingresado toda la información requerida y el mismo accede al

menú principal, donde posee todas las funcionalidades correspondientes a su rol.

- En segundo lugar cuando el administrador quiere agregar un usuario. En este caso se registra el usuario habiendo ingresado toda la información requerida, y luego se vuelve al menú principal estando iniciada la sesión obviamente del administrador con todas las funcionalidades.

ABM Cliente

Esta funcionalidad permite crear, dar de baja y editar clientes.

El crear cliente muestra una serie de campos a completar por el usuario que quiera crear el cliente. Como restricción obliga a que todos los campos estén completos.

Existen dos formas de llegar al crear cliente: una mediante el registrar usuario y otra mediante el menú principal si estas loggeado como admin. Parar ambos casos, además de crear el cliente, crea su usuario el cual se le pasa por parámetro a este form. Si es el admin el que crea el usuario, el username es el username default. En cambio, si fue un nuevo usuario, lo crea con el nombre que este eligió.

Al crear el usuario, si todos los campos estan correctos, navega hacia el menú principal.

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 (no incluye los dados de baja) y dos columnas más que son modificar y eliminar. Se agregaron textbox para que el admin ingrese ciertos criterios para filtrar. Se pueden hacer la combinación que quieran de ellos.

Para dar de baja solamente hace falta clicar el boton eliminar.

Para modificar hay que clicar el boton modificar que dirige al form modificar cliente, que carga todos los datos del cliente seleccionado. Es necesario que todos los datos a modificar estén completos.

ABM Empresa

Hace exactamente lo mismo que ABM Cliente pero con diferentes campos a completar.

ABM Rubro

Decidimos no hacerlo ya que no era un requerimiento del trabajo práctico.

ABM Visibilidad

Al ser un ABM también tiene los form crear, filtrar y editar.

Los campos presentados para completar en crear visibilidad deben completarse todos para poder crear una visibilidad ya que son indispensables para las siguientes funcionalidad.

Para dar de baja y modificar una visibilidad se actúa de manera similar que el ABM Cliente. En el filtro de visibilidad (que te permite filtrar por descripción) hay una columna con el botón dar de baja y uno con modificar.

Si clickeas en dar de baja se da de baja la visibilidad y no se puede volver a dar de alta.

Si se clickea en modificar se abre el form de editar donde se cargan los datos de la visibilidad seleccionada. Todos los datos deben estar completos antes de guardar la modificación.

Generar publicación

La forma de crear es la misma que en todos los ABM de crear.

Solamente se puede generar publicaciones con estado borrador o con estado publicada.

Lo importante para destacar acá es como maneja la fecha de inicio y la de vencimiento.

Todos los tipos de publicación tiene fecha de inicio y de vencimiento, incluso estando en estado borrador.

La fecha de inicio al generar la publicación es el día de hoy, indistintamente para el estado borrador y publicada.

Editar publicación

La forma de filtrar y modificar publicaciones es la misma que en todos los ABM.

Lo único valioso a distinguir en este caso es que hay varias restricciones a la hora de completar datos.

Si el estado es borrador, al editar, podemos cambiarla a publicada.

Si el estado es publicada, podemos cambiarle el estado a pausada o finalizada. Además se nos restringen los cambios a solamente poder editar el stock y la descripción.

Si el estado es pausada, podemos cambiarle el estado a publicada o finalizada. Sin embargo no podemos modificarle ningún otro dato.

Si el estado es finalizada, no podemos modificar nada.

Para inhabilitar la posibilidad de modificar datos se deshabilitan los textbox.

Por último, la fecha de inicio y de vencimiento se maneja de la siguiente forma. Comprueba antes de crear el objeto publicación si el estado inicial de la publicación era borrador. Si lo era quiere decir que solamente puede ir a Borrador o Publicada, es decir, hay que setearle la fecha de inicio como hoy (la de duración es automática siempre). En el caso de que no sea borrador, la publicación ya tiene una fecha de inicio no modificable, por lo tanto, la busca en la base de datos.

Gestionar preguntas

Trabajamos por separado las preguntas y las respuestas en cuanto a la creación de cada una, así como la forma de poder verlas. Aun así comparten ciertos elementos. Para ambos casos, se decidió establecer un límite de 255 caracteres y el diseño para interactuar con ambas es muy similar, pero se trabajó en forms independientes ya que se trata de 2 funcionalidades que no pertenecen por igual a los mismo roles.

Para los Vendedores, el sistema brinda un listado con todas las preguntas que hayan recibido sus publicaciones y aun no se han respondido; mostrando en cambio un mensaje de aviso si es que el listado no tiene datos que ofrecerle. El listado muestra parte de cada pregunta solamente, pero si el Vendedor lo desea, el sistema le brindará los datos de la publicación sobre la cual se realizó la pregunta que le interese. Luego, al momento de responder una de las preguntas, accediendo desde el listado, se muestra la pregunta en su total extensión.

Para los Clientes, el sistema brinda un listado que muestra todas las preguntas que realizo que fueron respondidas y sus respectivas respuestas. Este listado también muestra un aviso si no tiene datos que mostrar y se le pueden pedir los datos de la publicación sobre la que se preguntó. Al igual que el listado de los Vendedores, solamente muestra parte de la pregunta y la respuesta, pero es posible observar la extensión total de ambas accediendo a un nuevo form desde el listado.

Comprar/Ofertar

El filtro de publicaciones es dinámico en cuanto a la cantidad de parámetros por los cuales se puede buscar. Esto varía según los datos que aporte un Cliente para realizar la búsqueda; pudiendo buscar publicaciones por algún rubro, por alguna coincidencia con la descripción de las publicaciones, por ambos filtros o por ninguno; devolviendo este último todas las publicaciones activas.

Debido a que la cantidad de resultados de la búsqueda puede ser muy grande, los resultados los organizamos por páginas que muestran solo 10 publicaciones cada una; siendo posible retroceder o avanzar de a una página a la vez, o ir directamente a la primera o última página. Para poder llevar esto a cabo, se consulta la base de datos y los resultados obtenidos se almacenan en un DataTable de uso temporal en Visual. Luego a medida que se navega por las diferentes páginas, según en que página nos encontremos, el sistema le pedirá a este DataTable que muestre por pantalla las filas (indexes) correspondientes a la página activa. El propósito principal por el que lo desarrollamos de esta manera es para no consultar la base de datos cada vez que se cambia de página y en cambio hacerlo una sola vez por búsqueda.

Por defecto, la búsqueda impide que se muestren las publicaciones que le pertenecen al Cliente que está realizando la búsqueda y consideramos publicaciones activas a todas aquellas cuyo estado sea Publicada o Pausada.

En el caso de la Pausadas, un Cliente puede observar los datos pertenecientes a dicha publicación y realizar preguntas al Vendedor en caso de que la misma lo permita; pero no se podrán realizar compras u ofertas. Para impedir la compra de dichas publicaciones, no solo bloqueamos el botón de Comprar/Ofertar, sino que además el sistema informa al Cliente que no puede comprar u ofertar debido al estado de la publicación.

Tanto publicaciones de compra como de oferta utilizan el mismo form para mostrar su información relevante; existiendo un botón que ajusta su leyenda y el evento que acciona según el tipo de publicación que se trate. También hay otro botón, el cual se habilita o deshabilita según las propiedades de la publicación, para que el Cliente puede enviarle una pregunta al Vendedor.

Al momento de realizar una compra, el sistema muestra al Cliente los datos del Vendedor y le pide que ingrese la cantidad de productos que comprará. En cambio para ofertar, el sistema solo pedirá al Cliente el monto que quiere ofertar. En ambos casos el sistema restringe los datos que se pueden ingresar exclusivamente a números enteros; rechazando los negativos, textos y símbolos. Del mismo modo tampoco se permite realizar compras que superen el stock actual de la publicación ni ofertar por un monto menor al que este ganando en el momento. En caso de que estas restricciones sean violadas por el Cliente, el sistema le informa cual ha violado y que datos son los que se le permiten ingresar.

Al sistema también se le agregó un Trigger “**finalizar_x_fin_stock**” para que controle el stock de una publicación cada vez que se realiza una compra. Esto lo hicimos con el objetivo de que en el caso de que con una compra se termine el stock, la publicación cambia de estado automáticamente a Finalizada.

Hay 2 propiedades de las publicaciones que decidimos manejarlos de forma lógica; el stock de las compras y el precio de las ofertas. Para llevarlo a cabo utilizamos 2 vistas: “**VistaCantidadVendida**” y “**VistaOfertaMax**”.

Para el stock, la vista revisa las compras que tiene cada publicación y calcula el total que se vendió de cada publicación. Luego al stock inicial de la publicación, que se mantiene almacenado en la tabla Publicaciones, se le resta el valor obtenido de la vista.

Para las ofertas, la vista revisa todas las ofertas que recibió cada publicación y calcula cual tiene el monto máximo, mientras que el precio inicial de la publicación se mantiene almacenado en la tabla Publicaciones. Luego el sistema para mostrarle el precio de la publicación al Cliente utilizará el valor de la vista, en caso de que ya haya recibido ofertas anteriormente, y en caso contrario usará el precio de la tabla Publicaciones.

Historial del cliente

El historial del cliente es un form con el diseño de un filtrado. Este te da la posibilidad de filtrar (en realidad es elegir) el tipo de historial que desees ver. Existen tres tipos de historiales: compras, ofertas, calificación.

Lo que se hace es según la elección, se selecciona un select distinto con el que se le pega a la base de datos. El resultado de ese select es el que se muestra en el datagridview.

A veces las consultas tardan un poco.

Calificar vendedor

Se muestra el listado de las compras que el usuario ha realizado y no calificó.

Allí el usuario puede calificar a los vendedores de las compras que hizo. Se registra la calificación en dicha compra y por lo tanto la misma deja de aparecer como pendiente de calificar al vendedor.

La calificación tiene una cantidad de estrellas y una descripción.

Facturar publicaciones

Al ingresar al menú “Generar factura”, se puede ver:

1. La cantidad de costos por publicación que deben facturarse (aquellos donde la publicación ha pasado a estar finalizada)
2. La cantidad mínima y máxima de ventas por facturar. La cantidad máxima es el total y la cantidad mínima está restringida por aquellas ventas que son de publicaciones ya finalizadas.

A su vez también al realizar una factura debe seleccionarse la forma de pago, que podrá ser en efectivo o por tarjeta de crédito.

Al seleccionar la cantidad de ventas y costos de publicación que se facturarán, se calcula el costo que tendrá la factura sin contar posibles bonificaciones por cantidad de comisiones de ventas de publicaciones de una determinada visibilidad.

Luego al realizar la factura se tiene en cuenta que si se llega a la décima venta de una determinada visibilidad de publicación, esta décima venta tiene costo 0.

Se le informa al usuario que genera la factura la cantidad de dinero que se le descontó por bonificaciones.

Para llevar la cuenta de la cantidad de comisiones de publicaciones de un usuario según cada tipo de visibilidad distinta, se tiene la tabla **“LOS_SUPER_AMIGOS.Comisiones_Usuario_x_Visibilidad”**.

Al crear un nuevo usuario o al crear una nueva visibilidad, deberían agregarse estos a dicha tabla. Para esto se tienen 2 triggers:

- agregar_valor_default_de_nueva_visibilidad_en_comisiones
- agregar_valor_default_de_nuevo_usuario_en_comisiones.

Listado estadístico

El diseño de este form es el de un filtrar. Los campos por los que se pueden filtrar son: año, trimestre y tipo de estadística.

El form esta diseñado para que si no se escribe un año, no te deja ingresar un trimestre. Si no elegis un trimestre, no te deja seleccionar un tipo de estadística. Si no elegis un tipo de estadística, no te deja clicar en buscar.

El form tiene definido los distintos tipos de select para cada una de las posibilidad. Mediante ifs y switch se va construyendo la consulta con la que se le pega a la base datos. El resultado es mostrado en el datagridview.

Coneccion DB

Esta clase lo que permite es tener una interfaz sencilla para abrir y cerrar conexiones. Los datos necesarios para conectarse con la base de datos esta preseteados.

Interfaz disponible:

- AbrirConexion()
- CerrarConexion()

BuilderDeComandos

Permite ejecutar consultas a la base de datos de una forma más simple y sencilla. Le pasamos por parámetro la consulta y una lista con los parámetros que esta consulta necesite, y nos crea un comando listo para ejecutar. Nos abstrae de tener que tener en cuenta la conexión a la base de datos cada vez que mandamos una consulta.

Interfaz disponible:

- Crear(query, parametros): SqlCommand

ComunicadorConBaseDeDatos

Interfaz disponible:

- crear_usuario(): idUsuario
- crear_usuario_con_valores(usuario, password): idUsuario
- crearObjeto(objeto): idObjetoCreado
- modificar(idObjeto, objeto): Boolean
- obtenerObjeto(idObjeto): Objeto
- eliminar(idObjeto, tabla): Boolean

- selectFromWhere(queCosaBuscar, enQueTabla, valorDeTabla, valorMandado): Dato
- selectDataSet(select, from): DataSet
- selectDataSet(select, from, where): DataSet

Diagrama de secuencia de crear:

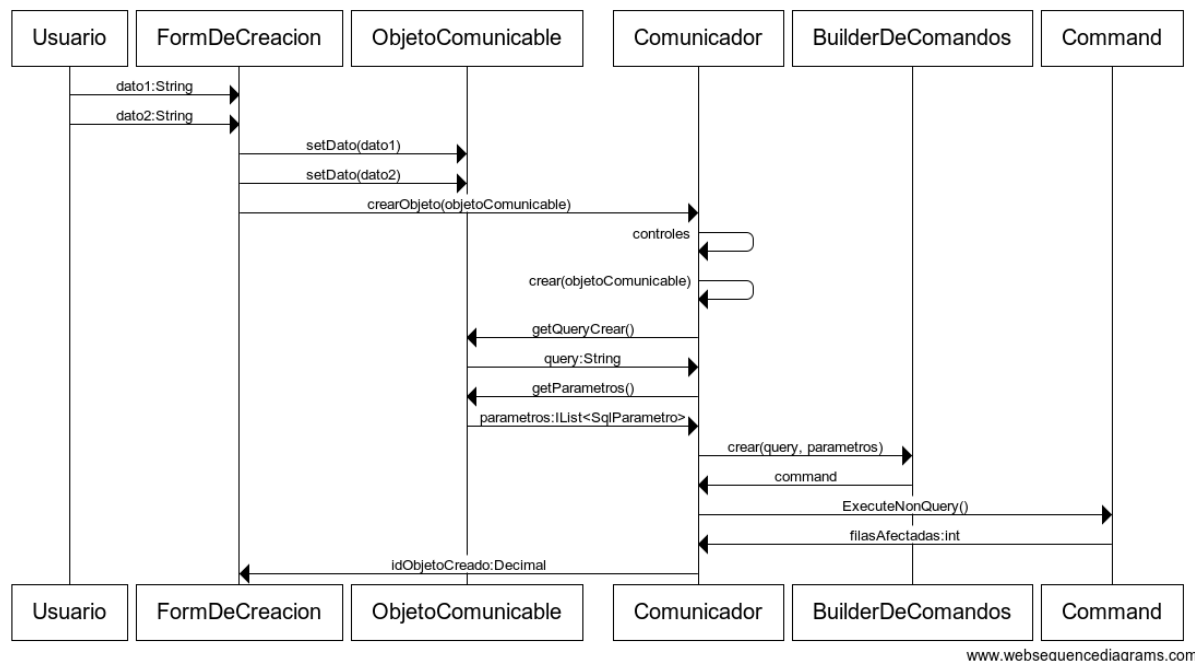


Diagrama de secuencia de modificar:

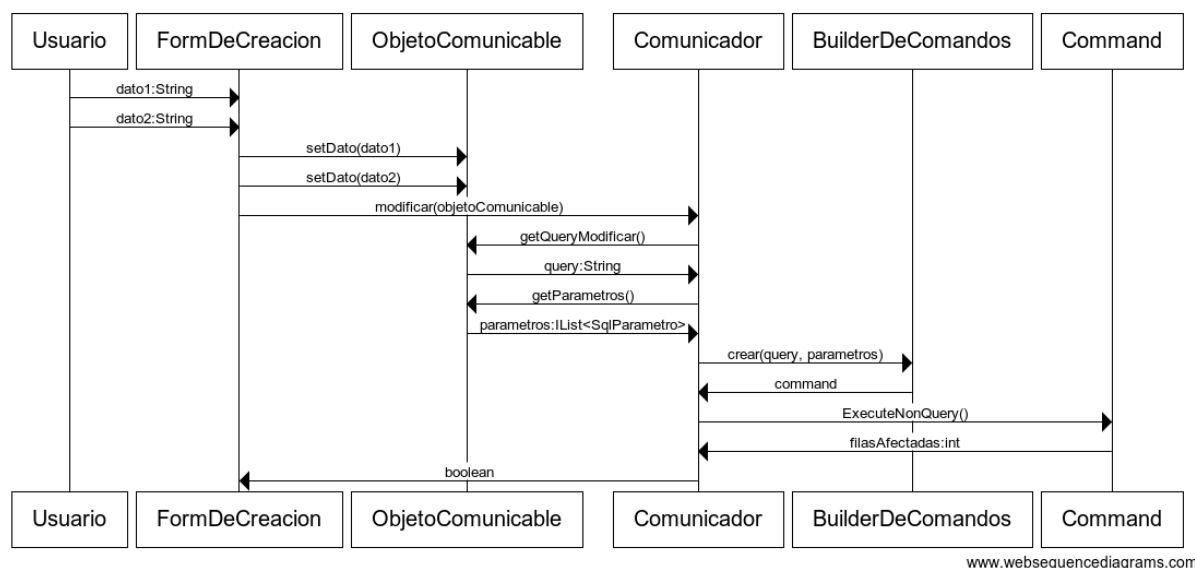


Diagrama de secuencia de obtener:

