Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico 1° Cuatrimestre 2016

Mercado-Envío!

Grupo : “CLAVE_MOTOR”

Integrantes:

- VERA, Jonathan Mirco (134.511-4)

Curso K3151

Índice de contenido

Introducción.....3
Decisiones tomadas.....3
Modelo de Datos.....4
Diseño de la aplicación.....4
Apéndice I: Modelo de Datos.....5

Introducción

La resolución del Trabajo Práctico consistió en crear una aplicación en C# que se conecte a la base de datos en SQL Server.

El primer paso del trabajo consistió en cruzar las reglas de negocio con los datos de la base de datos desnormalizada provista a fin de resolver ambiguedades y poder tomar decisiones fundamentales de diseño.
Al contar con un bosquejo del dominio del problema, se pasó a realizar un trabajo de normalización, modelado y extracción de datos sobre la tabla inicial, a fin de tener un modelo más consistente de datos.

Una vez solucionado el modelado de los datos, se realizó el modelado y construcción de la aplicación, en la que se trataron de diferenciar las capas de Modelo, Acceso a Datos y Presentación, usando diferentes clases para cada aspecto. En esta etapa surgieron también reformas al modelo de datos previamente planteados, y se generaron nuevos objetos de base de datos auxiliares.

A lo largo de todo el proceso, el replanteo y rectificación de interpretaciones y decisiones previamente tomadas fueron constantes, aunque siempre manteniendo la línea de pensamiento inicial.

Datos de usuario administrador:
Username:admin
Password:admin

Decisiones tomadas

Considerando el relevamiento realizado y las respuestas de consultas hechas se tomaron algunas decisiones de diseño:

- Se han respetado los identificadores de entidades existentes en la base para no generar confusiones a los usuarios que utilicen la app nueva.
  También teniendo en cuenta que alguna app exterior que se comunicaba con la app vieja ahora quiera hacerlo con la nueva.
- Las publicaciones que ya habian cumplido su fecha de vencimiento, fueron migradas como publicaciones ya finalizadas.
- Se han encontrado algunas publicaciones que no poseían un vendedor asociado

en el modelado viejo. Se ha optado por dejar a esas publicaciones sin vendedor.
- Como se mencionó antes, la línea de pensamiento fue separar el acceso de datos del Modelo y la presentación al usuario.
  Se utilizaron DAOs para acceder a los datos de todas las entidades posibles.
- El usuario **admin** generado por el sistema no está asociado a ningun Cliente o Empresa.
- Cada item de factura tiene asociado una Publicación. Dentro de la app, particularmente se factura de forma que no se mezclan entre si las facturas de diferentes publicaciones a una misma Empresa. Pero con este modelo se deja la opción de optar por otra forma de facturación como por ejemplo semanal o diaria a una empresa por todos los gastos generados por sus publicaciones en la semana o dia según corresponda.
- Como el login es una funcionalidad que no puede ser asignada a ningun rol, no se la trato como una Funcionalidad dentro del sistema.
- Para resolver el requerimiento de la fecha de sistema se decidió usar un stored procedure que se encarga de revisar las publicaciones vencidas, que se ejecuta en el inicio de la aplicación automáticamente.
- Para poder modificar las entidades Usuario, Empresa, Cliente es necesario utilizar los buscadores dentro de la función de ABM de Usuario.

Modelado de Datos

El DER del modelo de datos se encuentra al final del documento

Diseño de la aplicación

La aplicación implementada en C# fue diseñada tratando de separar el modelo, de la vista y del acceso a datos.

El namespace MercadoEnvio.Model contiene las clases del Modelo de toda la aplicacion.

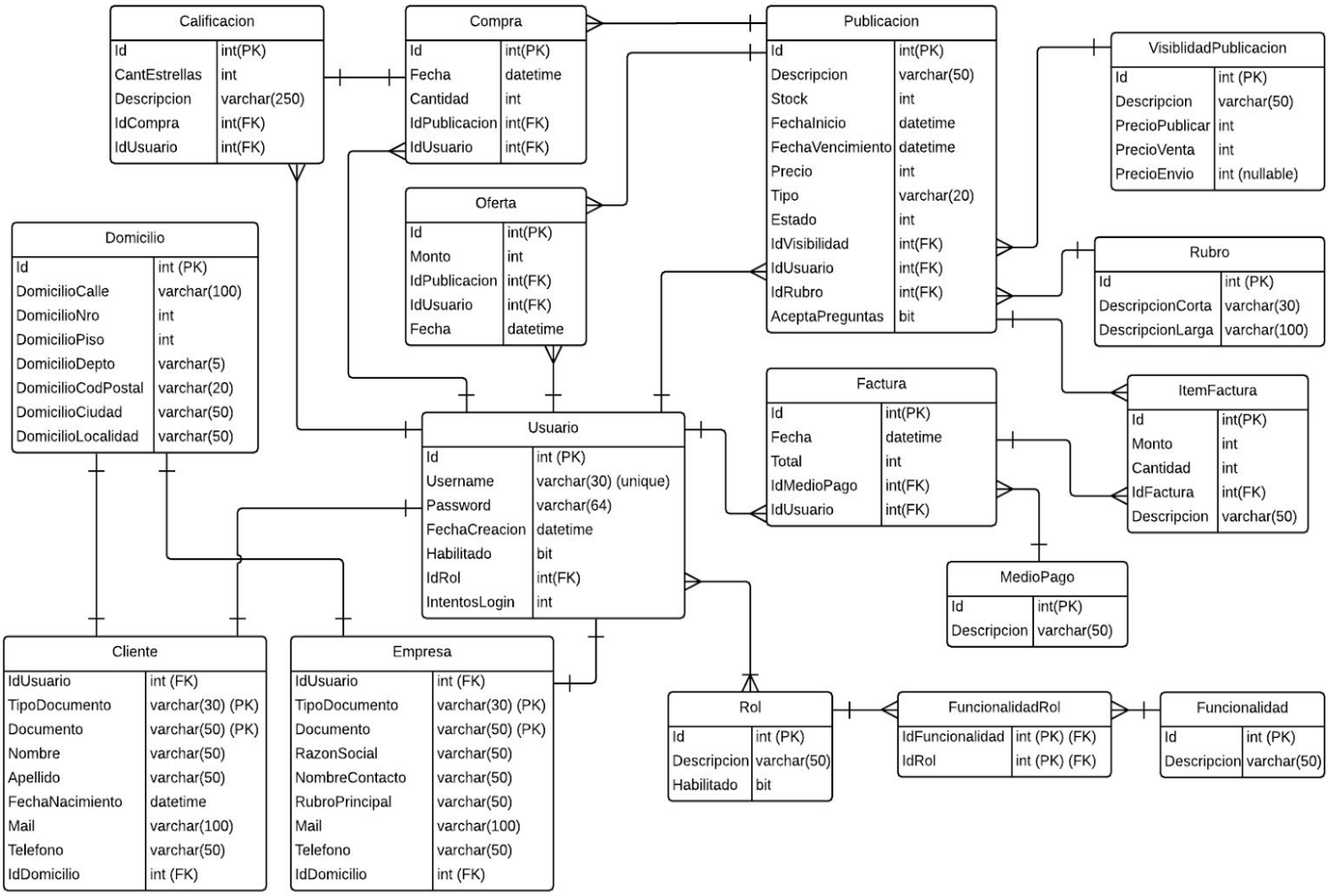
El namespace MercadoEnvio.DAO contiene las clases para el acceso a datos de las entidades del sistema, tratando de dar un nexo más transparente y desacoplado a la vista y el modelo.

Dentro del namespace MercadoEnvio.Utils se encuentra una clase hasher, usada para hashear la password usando SHA256.

El namespace MercadoEnvio.Buscadores agrupa los formularios de busqueda de entidades del sistema, como son Empresa, Clientes,etc.
Cada buscador solo muestra sus correspondientes entidades con los filtros que a estos incunban.

La clase ClaseSQL funciona como una interfaz para utilizar los objetos y métodos provistos por el lenguaje para la conexión con la base de datos. Para mayor facilidad esta clase implementa el patrón Singleton para el uso de una única conexión a la base de datos, minimizando los costos y reduciendo los riesgos de un mal manejo de multiples conexiones.

Los parametros tanto de la base de datos como la fecha de sistema se encuentran todos en el archivo de configuración de la aplicación, tal como lo solicitaba el enunciado.


Calificacion
Id int(PK)
CantEstrellas int
Descripcion varchar(250)
IdCompra int(FK)
IdUsuario int(FK)
Compra
Id int(PK)
Fecha datetime
Cantidad int
IdPublicacion int(FK)
IdUsuario int(FK)
Publicacion
Id int(PK)
Descripcion varchar(50)
Stock int
FechaInicio datetime
FechaVencimiento datetime
Precio int
Tipo varchar(20)
Estado int
IdVisibilidad int(FK)
IdUsuario int(FK)
IdRubro int(FK)
AceptaPreguntas bit
VisiblidadPublicacion
Id int (PK)
Descripcion varchar(50)
PrecioPublicar int
PrecioVenta int
PrecioEnvio int (nullable)
Oferta
Id int(PK)
Monto int
IdPublicacion int(FK)
IdUsuario int(FK)
Fecha datetime
Domicilio
Id int (PK)
DomicilioCalle varchar(100)
DomicilioNro int
DomicilioPiso int
DomicilioDepto varchar(5)
DomicilioCodPostal varchar(20)
DomicilioCiudad varchar(50)
DomicilioLocalidad varchar(50)
Rubro
Id int (PK)
DescripcionCorta varchar(30)
DescripcionLarga varchar(100)
Factura
Id int(PK)
Fecha datetime
Total int
IdMedioPago int(FK)
IdUsuario int(FK)
ItemFactura
Id int(PK)
Monto int
Cantidad int
IdFactura int(FK)
Descripcion varchar(50)
Usuario
Id int (PK)
Username varchar(30) (unique)
Password varchar(64)
FechaCreacion datetime
Habilitado bit
IdRol int(FK)
IntentosLogin int
MedioPago
Id int(PK)
Descripcion varchar(50)
Cliente
IdUsuario int (FK)
TipoDocumento varchar(30) (PK)
Documento varchar(50) (PK)
Nombre varchar(50)
Apellido varchar(50)
FechaNacimiento datetime
Mail varchar(100)
Telefono varchar(50)
IdDomicilio int (FK)
Empresa
IdUsuario int (FK)
TipoDocumento varchar(30) (PK)
Documento varchar(50) (PK)
RazonSocial varchar(50)
NombreContacto varchar(50)
RubroPrincipal varchar(50)
Mail varchar(100)
Telefono varchar(50)
IdDomicilio int (FK)
Rol
Id int (PK)
Descripcion varchar(50)
Habilitado bit
FuncionalidadRol
IdFuncionalidad int (PK) (FK)
IdRol int (PK) (FK)
Funcionalidad
Id int (PK)
Descripcion varchar(50)