Estrategia de trabajo

palco net

FECHA: 09/12/2018

integrantes:

-montanaro, daniela.

-Gomez, melisa

DESICIONES DE DISEÑO

FECHA DEL SISTEMA.

Para la fecha del sistema generamos una clase ConfigGlobal que se configurará desde App.config ubicado en el proyecto, una vez seteada la fecha el sistema utilizará esa fecha a lo largo de todo el proyecto.

GENERAR PUBLICACION.

Para la generación de la publicación decidimos que al darse el alta por default todas estén en “Borrador” para el ordenamiento de la misma.

Si un usuario quiere cambiar el estado de una publicación deberá dirigirse a la vista “Cambiar Estado” para pasarla a algún estado diferente de borrador, eso facilita la parte del enunciado que dice: “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, evitando al alumno desarrollar procedimientos automáticos de cambios de estado y complicaciones con las fechas”

Para la generación de una nueva publicación el usuario deberá ingresar todos los datos pedidos en el enunciado, y como elección del equipo pusimos como restricción que el stock sea múltiplo de diez para la mejor visión de las ubicaciones. También el precio decidimos que dependa de la categoría para que no se tenga que ingresar por cada ubicación(asiento-fila) un precio diferente.

Para la generación de una publicación dada por lotes. Se deberá importar un archivo .txt con el siguiente formato:

DD/MM/AAAA HH:MM:SS,

En la misma carpeta se encontrará un archivo PRUEBA.TXT donde se da como ejemplo una serie de fechas que podrán ser subidas por lote.

Asi sucesivamente, el sistema valida que las fechas sean posteriores a la fecha del archivo de configuración y que las mismas si pertenecen a una misma fecha que sean en horarios posteriores,

Para editar una publicación, y para la facilidad de la codificación, decidimos que el usuario pueda cambiar todos los campos que considere, MENOS las ubicaciones (sus categorías y precios) si el usuario (Empresa en este caso) decidiría cambiar alguno de estos campos (ubicación-asiento-precio de la categoría o categoría de un asiento-fila) deberá dar de baja la publicación y crearla nuevamente.

Datos Faltantes: Debido a nuevas capacidades del modelo,

fue necesario agregar datos por defecto durante la migración,

DNIs para tipos de documentos, las fechas de creaciones para las contraseñas, etc.

Gestion de Usuarios

Usuario --> Contiene los usuarios del sistema

Rol: Contiene los distintos tipos de roles.

RolXUsuario: Relaciona un usuario con uno o más roles.

Funcionalidad: Contiene las funcionalidades del sistema.

FuncionalidadXRol: Relaciona las funcionalidades con los roles.

Empresa --> Contiene las empresas del sistema

Cliente --> Contiene los clientes del sistema

Facturas: Contiene las facturas realizadas.

Items\_Factura: Contiene los ítems de las facturas realizadas.

Se crearon varios procedures a fin de servir como una interfaz del modelo de datos ante la Aplicación.

Debido a esto, por cada tabla principal, se realizaron 4 procedures básicos: get, insert, delete y update.

Todos poseen las siguientes condiciones:

 Si se ingresan campos como -1 para los numéricos y ‘’ para los varchar() se los considera como “elementos neutros” o nulos.

 Los procedures delete son lógicos (pone en 0 su CampoBaja) en su mayoría.

REVISIONES

ABM Cliente

ABM Empresa

ABM Grado

Compras filtrar por categoria, fecha y descripción

MODULOS FALTANTES A LA FECHA:

HISTORIAL DE CLIENTE

CANJE DE PUNTOS

FACTURAR PUBLICACIONES

GENERAR ESTADISTICA.