DiWork

Challenge Recruiting

Taller mecánico

Elaborar una aplicación que permita llevar delante de manera simple un sistema de presupuestos para reparaciones de un taller mecánico.

La clase **Vehículo** será abstracta. De esa clase heredará **Automóvil** y **Moto**. Los automóviles podrán ser del tipo: *compacto, sedan, monovolumen, utilitario y lujo*. Implementarlo en un enumerador público.

El sistema permitirá:

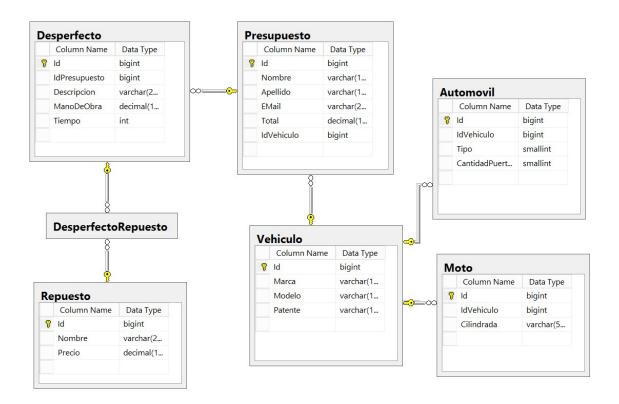
- Ingresar motos y automóviles al taller y se les diagnostique uno o más desperfectos.
- Cada desperfecto tendrá una descripción, un costo de mano de obra y un tiempo estimado de trabajo. Así mismo los desperfectos podrán tener asignados uno o más repuestos.
- Emitir un presupuesto para cada vehículo.
 - El presupuesto indicará el costo total de reparación (calculado en base a los repuestos necesarios, la mano de obra, \$130 por día de trabajo para cubrir costos de estacionamiento, los descuentos o recargos que correspondan y un recargo del 10% como ganancia del taller).
 - El presupuesto estará asignado a un cliente y un vehículo (moto o automóvil), con una lista de desperfectos.
 - o La sumatoria del costo total debe estar reflejada en el presupuesto.
- Consultas de Salida
 - o Desarrollar un (1) Stored Procedure por cada consigna que se especifica a continuación:
 - Repuesto más utilizado por Marca/Modelo en las reparaciones realizadas (Mostrar Descripción del Repuesto y cantidad de veces usado)
 - Promedio del Monto Total de Presupuestos por Marca/Modelo
 - Sumatoria del Monto Total de Presupuestos para Autos y para Motos
- Cargar Repuestos y precios
 - Modificar el Stored Procedure dbo.MassiveCharge adjunto (...el cual contiene dentro una Tabla Temporal cargada con registros de Repuestos y sus precios...), para que permita hacer lo siguiente:
 - Se necesita que se carguen en la tabla Definitiva de Repuestos, cada registro que exista en la temporal, siempre y cuando:
 - El valor del Repuesto sea menor a 20\$
 - Aquellos Repuestos que no se inserten/actualicen, pues su valor es mayor o igual a 20\$, deberá reportarse en un Query de salida.
 - Si un repuesto existiere más de una vez, se deberá acumular el valor del mismo, para el mismo Respuesto.
 - Se recomienda usar un Cursor para la evaluación de los registros.



Challenge Recruiting

Condiciones

- El código debe estar en 4 capas (Presentación, Lógica, Persistencia, Modelos).
- La conexión a la base de datos se debe establecer por "ADO Desconectado" e invocar "Stored Procedures" para obtener y almacenar datos.
- Los proyectos deben ser creados en lenguaje C# o VB.NET.
- La capa de presentación puede estar desarrollada en Windows Forms, Web ASP.NET, Web API o Razor.
- Puede incluir algún patrón de diseño de software: singleton, observer, state, builder, etc. (ideal, al menos uno).
- Organice el código en regiones y las clases en namespaces.
- La BD debe ser de tipo Relacional "SQL Server".
- Las tablas creadas deben encontrarse en 3ra forma normal.



^{*} El diagrama es ilustrativo, agregue las columnas que considere necesarias para mantener una estructura que responda a las buenas prácticas.

¡Buena suerte!