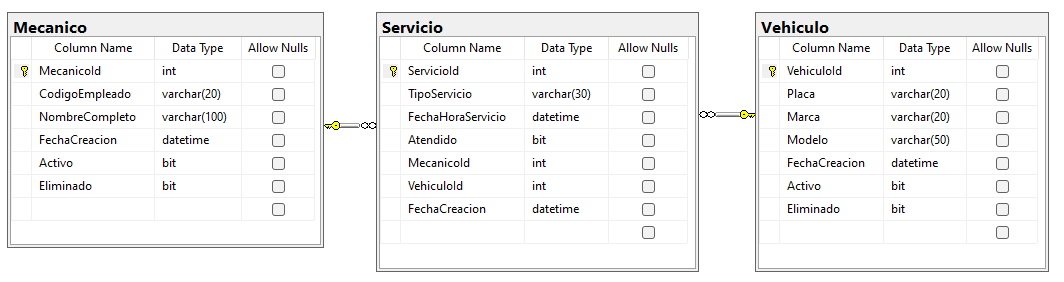
Instrucciones:

1. El examen debe de ser desarrollado exclusivamente con los temas vistos en clase. No está permitido el uso de código que no sea parte del material enseñado.
2. La solución y proyecto en Visual Studio debe ser su apellido y el archivo zipeado que contendrá el código con la solución del caso será su código de alumno. En caso de no seguir el formato indicado, se descontará 2ptos.
3. El desarrollo del examen debe ser realizado acorde a como lo plantea el profesor en su clase.
4. El examen debe ser desarrollado haciendo uso de conexión a base de datos.
5. En caso el alumno envíe un proyecto incompleto, solo se calificará lo enviado. El alumno es el único responsable de lo enviado.

Caso B

Una empresa mecánica automotriz, le solicita a Ud. que elabore una aplicación en C# que permita reservar los servicios solicitados por los clientes.

Aquí se detalla el modelo de base de datos del sistema.



El sistema considera lo siguiente:

1. La placa del vehículo no se debe de repetir.
2. El código del mecánico no se debe de repetir.
3. Un vehículo y un mecánico pueden tener más de un servicio.
4. La reserva de un servicio mecánico se realiza para un vehículo, asignando un mecánico para su atención en un día y hora específico.

**Se le solicita que implemente el mantenimiento de la tabla Servicio para lo cual debe desarrollar las siguientes funcionalidades**:

* (1pto) Solo deben de seleccionarse los vehículos no eliminados.
* (1pto) Solo deben de seleccionarse mecánicos no eliminados.
* Registrar servicio, debe considerar:
  + (1pto) El campo “FechaCreacion” de la tabla “Servicio” debe de almacenar la fecha y hora del momento en que se está registrando la reserva del servicio (fecha y hora actual). Es un campo que no puede ser editado, ni cuando se cree ni cuando se edite.
  + (2ptos) El campo “FechaHoraServicio” de la tabla “Servicio” debe de almacenar la fecha y hora de para cuando se está reservando el servicio del vehículo. No debe de ser menor a la fecha actual.
  + (2ptos) El campo “Atendido” de la tabla “Servicio” se actualizará cuando se haya realizado el servicio solicitado. True en caso sea atendido el servicio. False en caso no se haya realizado el servicio.
* (3ptos) Modificar Servicio. Sólo se puede modificar la fecha y hora del servicio, así como el mecánico asignado y vehículo.
* (2ptos) Eliminar servicio (Eliminación física)
* (3ptos) Reporte de servicios planificados dada un rango de fechas (filtros de fecha de inicio y fecha de fin) y un tipo de servicio (Filtro Tipo de Servicio). Se debe de visualizar la fecha, la placa del vehículo y el mecánico que realiza el servicio.
* (3ptos) Reporte de cantidad de servicios que un mecánico ha atendido. Se debe contabilizar aquellos servicios con valor True en el campo “Atendido” de la tabla “Servicio”. El reporte mostrará el nombre del mecánico y cantidad de servicios realizados.
* (2ptos) La aplicación debe de compilar y funcionar correctamente sin errores.

**Instrucciones:**

* Diseñar las interfaces necesarias que materialicen estas funcionalidades.
* Desarrollar el proyecto 4 capas (Entidad, Datos, Negocio y Presentación) en C# en entorno visual o haciendo uso de Entity Framework (Datos, Negocio, Presentación), que permita implementar el CRUD de la tabla Servicio.
* Cada funcionalidad solicitada debe ser desarrolla de forma correcta y tal como como se ha visto en clase.
* Los tipos de servicio de la tabla Servicio pueden ser mostrados como una lista en un combobox y son: Cambio de Aceite, Cambio de Suspensión, Revisión llantas, Revisión Freno, Cambio Amortiguadores.

Ejecutar el siguiente script que creará la base de datos.

|  |
| --- |
| use [master]  go    drop database [db\_final]  go    create database [db\_final]  go    USE [db\_final]  GO    -- tables  -- Table: Mecanico  CREATE TABLE Mecanico (      MecanicoId int  NOT NULL IDENTITY(1, 1),      CodigoEmpleado varchar(20)  NOT NULL,      NombreCompleto varchar(100)  NOT NULL,      FechaCreacion datetime  NOT NULL,      Activo bit  NOT NULL,      Eliminado bit  NOT NULL,      CONSTRAINT Mecanico\_pk PRIMARY KEY  (MecanicoId)  );  go    -- Table: Servicio  CREATE TABLE Servicio (      ServicioId int  NOT NULL IDENTITY(1, 1),      TipoServicio varchar(30)  NOT NULL,      FechaHoraServicio datetime  NOT NULL,      Atendido bit  NOT NULL,      MecanicoId int  NOT NULL,      VehiculoId int  NOT NULL,      FechaCreacion datetime  NOT NULL,      CONSTRAINT Servicio\_pk PRIMARY KEY  (ServicioId)  );  go    -- Table: Vehiculo  CREATE TABLE Vehiculo (      VehiculoId int  NOT NULL IDENTITY(1, 1),      Placa varchar(20)  NOT NULL,      Marca varchar(20)  NOT NULL,      Modelo varchar(50)  NOT NULL,      FechaCreacion datetime  NOT NULL,      Activo bit  NOT NULL,      Eliminado bit  NOT NULL,      CONSTRAINT Vehiculo\_pk PRIMARY KEY  (VehiculoId)  );  go    -- foreign keys  -- Reference: Servicio\_Mecanico (table: Servicio)  ALTER TABLE Servicio ADD CONSTRAINT Servicio\_Mecanico      FOREIGN KEY (MecanicoId)      REFERENCES Mecanico (MecanicoId);    -- Reference: Servicio\_Vehiculo (table: Servicio)  ALTER TABLE Servicio ADD CONSTRAINT Servicio\_Vehiculo      FOREIGN KEY (VehiculoId)      REFERENCES Vehiculo (VehiculoId);  go    INSERT INTO [Mecanico] ("CodigoEmpleado", "NombreCompleto", "FechaCreacion", "Activo", "Eliminado")  VALUES  ('E001', 'Roberto Jacinto', '2022-10-04 15:34:23', 1, 0),  ('E002', 'Nilton Chaparro', '2021-09-25 10:27:53', 1, 0),  ('E003', 'Franz Contreras', '2021-08-12 16:21:07', 1, 0),  ('E004', 'Octavio Flores', '2022-05-05 12:03:47', 1, 1)  GO    INSERT INTO [Vehiculo] ("Placa", "Marca", "Modelo", "FechaCreacion", "Activo", "Eliminado")  VALUES  ('BVA754', 'Kia', 'Sorrento', '2021-05-17 11:45:34', 1, 0),  ('NHV577', 'Kia', 'Cerato', '2021-06-23 11:45:34', 1, 0),  ('GBD439', 'Toyota', 'Yaris', '2021-07-09 09:53:49', 1, 1),  ('JGR143', 'Mitsubishi', 'L200', '2021-08-02 14:32:39', 1, 0),  ('LOG937', 'Mitsubishi', 'Outlander', '2021-09-25 10:29:47', 1, 0),  ('HGP752', 'Subaru', 'Forester', '2021-10-28 15:45:42', 1, 0),  ('FTD765', 'Honda', 'CRV', '2021-11-30 17:04:08', 1, 1)  GO |