

## Examen Práctico para Desarrollador Especializado.

Nombre completo del candidato:

Realizar una agenda de contactos en Asp. Net MVC con Visual Studio , Bootstrap(Pagina Web) y utilizar como gestor base de datos SQL Server.

Se deberán cumplir los siguientes requerimientos:

1. Crear una carpeta en el escritorio con tu nombre, dentro de esta carpeta se almacenarán todos los Querys y/o Scripts de SQL que se utilizaron para la creación de la agenda, para una vez terminada mandar esa carpeta en .rar o .zip por medio de Skype.
2. Los contactos se deberán almacenar en una tabla "NORMALIZADA" con el nombre ContactosMAVI + "Tu Nombre" (ejemplo "ContactosMAVIAdrian). La tabla debe contener un campo auto incremental y secuencial de 1 en 1 de tipo INT. La creación de esta tabla deberá ser manual, es decir, no deberá crearse con ayuda del asistente para creación de tablas.
3. Los campos que deberá tener la tabla son los especificados en Tabla 1.

Campo	Control	Validaciones	Obligatorio
Nombre	Campo de texto		Si
Tipo Contacto	Lista de opciones (Amistad, Familiar, Trabajo)	No editable	Si
Teléfono Fijo	Campo de Texto	Solo Números	No
Teléfono Móvil	Campo de Texto	Solo Números	Si
Fecha Nacimiento	Calendario		Si
Sexo	Lista de opciones (Femenino, Masculino)	No editable	Si
Estado Civil	Lista de opciones (Soltero, Casado, Divorciado, Viudo, Unión Libre)	No editable	Si

Tabla 1: Lista de campos y atributos

4. Se creará un nuevo proyecto en Visual Studio dentro de la carpeta del paso 1 que tendrá por nombre ContactosMAVI.
5. Todo código escrito deberá documentarse (Métodos, Clases, etc.).
6. El desarrollo deberá contar con 2 formularios  
el primero contendrá lo siguiente:
  - Input(html) y un botón al lado derecho de este llamado "Buscar" el cual, al presionarlo, utilizando LINQ deberá filtrar por coincidencia de nombre en la Tabla(html); en caso de desconocer LINQ para filtrar la información, se podrá realizar una consulta directamente a la base de datos, pero respetando la sobrecarga del método consulta.
  - Tabla(html) que muestre los contactos almacenados en el caso de que existan, los campos a mostrar serán "Nombre", "Teléfono Fijo, Teléfono Móvil" y "Tipo Contacto"
  - Botón llamado "Eliminar Contacto" para eliminar el contacto seleccionado

- Botón llamado “Ver Contacto” que abrirá el 2do formulario con los datos pre cargados del contacto seleccionado pero con los controles deshabilitados para su edición
  - Botón llamado “Añadir Contacto” que mostrará el 2do formulario con los campos vacíos para registrar un nuevo contacto
  - Botón llamado “Editar Contacto” que abrirá el 2do formulario pero con los datos pre cargados del contacto seleccionado con el fin de poder editar la información.
  - El segundo formulario solo contara con los siguiente :
  - Campos para capturar los valores descritos en la tabla 1
  - Botón llamado “Guardar” que al ser presionado validará si es usuario es nuevo y lo guardara en la base de datos o en caso de haber sido registrado previamente existe actualizará la información, si el botón que se presionó en el primer formulario fue el “Ver Contacto” este botón deberá estar deshabilitado al mostrar el segundo formulario.
  - **Nota importante: para la consulta, Creación, Actualización y eliminación de los datos se tendrá que hacer por medio de un stored procedure de SQL server. (No utilizar EntityFramework para la creación de modelo ni de la conexión a base de datos).**
7. Para la conexión a la base de datos es necesario, crear una clase llamada Conexión que debe contener la cadena de conexión con al servidor “Proserver” a la base de datos ServicioAndroid, usuario “UsrExamen” y contraseña “U12345\*”, la clase debera contener un método para abrir la conexión y otro método para cerrarla.  
**(Nota importante: no utilizar Datasets para la conexión, ni autogenerado de cadenas de conexión, siempre realizar las conexiones a través de esta clase).**
  8. Crear una clase tipo MODELO con el nombre “Contacto” que contendrá todos los atributos que corresponden a los campos especificados en Tabla 1 y que deberá utilizarse para realizar las acciones de inserción, consulta y edición de contactos.
  9. Para la inserción, edición y consulta de todos los registros se deberá crear un controlador con el nombre de “Transacciones”. Dentro de este controlador de es necesario crear un método de consulta de registros sobrecargado, ya sea para mostrar todos los contactos o filtro por coincidencia de nombre en la Tabla(html).
  10. Al momento de insertar o editar los registros deberán respetarse las validaciones especificadas en Tabla 1; en caso de no cumplir con la validación no deberá guardar la información y será necesario notificar al usuario que hay campos sin cumplir con la validación.
  11. Al momento de añadir, editar o eliminar 1 contacto, se deberá actualizar la Tabla(html) para mostrar los datos actualizados.

Puedes utilizar métodos asíncronos para la carga de información, manejo de enumeradores.