

EJERCICIO INTEGRADOR PROG II

1. Trabajar sobre la cáscara dada
2. Lograr que la solución compile.
3. Implementar los métodos pendientes de desarrollo que se encuentran marcados como NotImplemented.
4. Al momento de iniciar el programa detectar clientes, levantando sus datos por medio de una deserialización de un archivo XML que le será adjunto a la solución. Agregar un control OpenFileDialog al formulario
5. Hacer unit test del repositorio y files .
6. Formularios:
 - a. Hacer que todos los formularios inicien en el centro de la pantalla
 - b. Quitar los controles de Min y Max del formulario y que no se pueda modificar su tamaño.
 - c. Al intentar cerrar el formulario por cualquier método debe preguntar si está seguro de querer salir. Si la respuesta es NO, deberá permanecer abierto.
 - d. El nombre del formulario debe ser personalizado
 - e. Los formularios NO pueden tener color por defecto.

SEGUNDA PARTE

1. Crear la base de datos RepasoBD.
2. Crear la tabla Customers con los campos Id (bigint), Name(varchar), LastName(varchar), Age(int)
3. El campo Id debe ser primary key y autoincremental.
4. Ejecutar el siguiente script.

```
insert into Customers (Name, LastName, Age) values ('Stacy', 'Loachhead', 33);
```

```
insert into Customers (Name, LastName, Age) values ('Eartha', 'Marthen', 87);
```

```
insert into Customers (Name, LastName, Age) values ('Mela', 'Trinkwon', 88);
```

```
insert into Customers (Name, LastName, Age) values ('Penrod', 'Spowage', 69);
```

insert into Customers (Name, LastName, Age) values ('Elbert', 'Biermatowicz', 88);

insert into Customers (Name, LastName, Age) values ('Kirsten', 'Levane', 23);

insert into Customers (Name, LastName, Age) values ('Juana', 'Goodin', 19);

insert into Customers (Name, LastName, Age) values ('Kirstyn', 'Peinton', 70);

insert into Customers (Name, LastName, Age) values ('Lisha', 'Tulleth', 83);

insert into Customers (Name, LastName, Age) values ('Dedra', 'Kinigh', 90);

insert into Customers (Name, LastName, Age) values ('Becka', 'Pahlsson', 15);

insert into Customers (Name, LastName, Age) values ('Corene', 'Moakler', 77);

insert into Customers (Name, LastName, Age) values ('Gweneth', 'Beveridge', 53);

insert into Customers (Name, LastName, Age) values ('Brendan', 'Cordle', 88);

insert into Customers (Name, LastName, Age) values ('Joceline', 'Goter', 25);

insert into Customers (Name, LastName, Age) values ('Suki', 'Cossans', 26);

insert into Customers (Name, LastName, Age) values ('Ava', 'Helix', 87);

insert into Customers (Name, LastName, Age) values ('Jo ann', 'Valintine', 62);

insert into Customers (Name, LastName, Age) values ('Pennie', 'Kelson', 45);

insert into Customers (Name, LastName, Age) values ('Erda', 'Levet', 80);

5. Crear un proyecto de Biblioteca de Clases llamado Application.DataAccess que contendrá una clase estática DataAccess. Agregar 5 métodos:

- i. GetCustomers: trae todos los clientes guardados en la BD

- ii. GetCustomerById: retornará un cliente a partir de su Id.

- iii. InsertCustomer: guardará un cliente en la BD.

- iv. UpdateCustomer: modificará un cliente a partir de su ID.

v. DeleteCustomer: eliminará un cliente de la base de datos a partir de su ID.

Refactorizar la clase CustomerRepository para mantener la funcionalidad