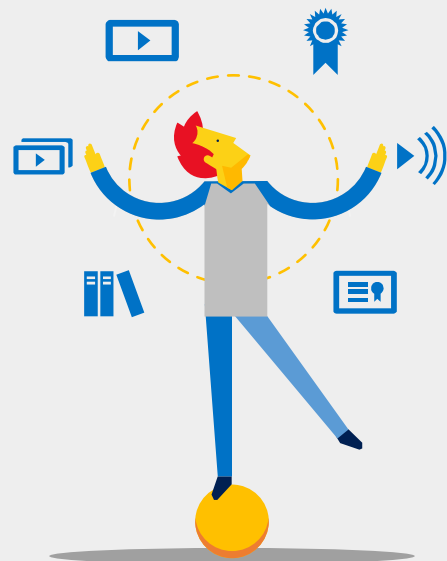


Microsoft Virtual Academy

Conectando los datos con la aplicación

Ejecutar consultas, recuperar y guardar información

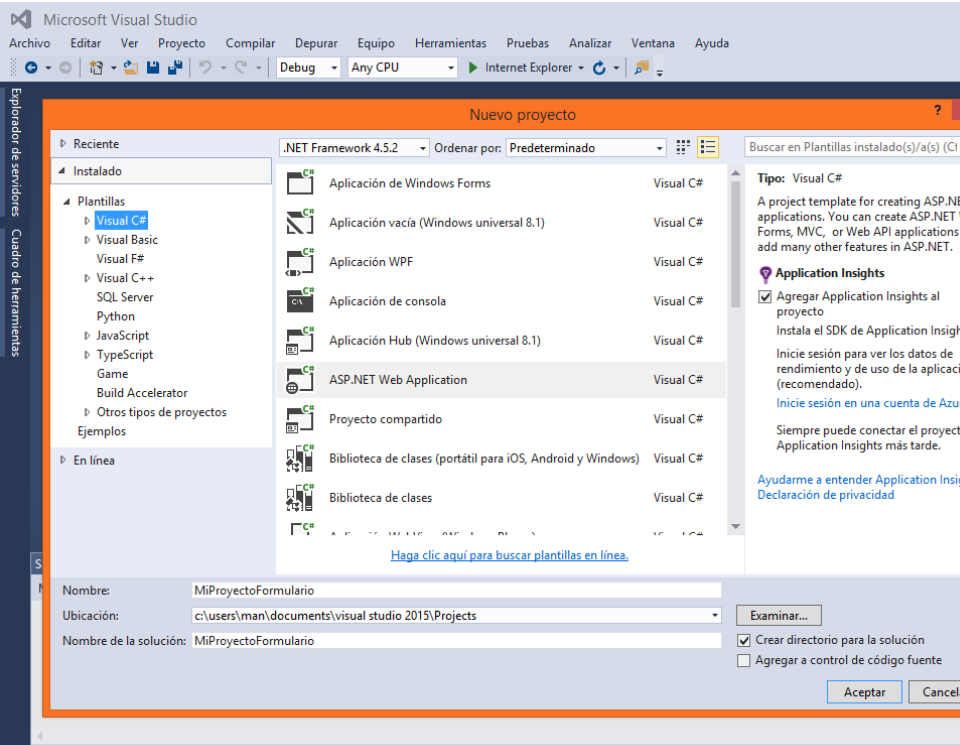
Aprendiendo a programar. Capitulo 9. Tutorial



Objetivo

Guardar y/o recuperar información almacenadas en una base de datos desde nuestra aplicación.





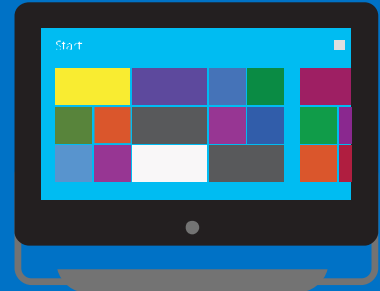
Comencemos

Abra un proyecto existente, sino, cree uno nuevo del tipo Visual C# - Aplicación web ASP.NET y utiliza la plantilla simple MVC

Crear o usar una base creada

Siguiendo el ejercicio del capítulo anterior crear la tabla Contactos y Localidades, y cargue datos.

Si se trata del mismo proyecto, omite este paso. La base de datos ya está creada



FormasController.cs

MiProyectoFormulario

MiProyectoFormulario.Controllers.FormasController

GuardarContacto(string nombre, string apellido, string

```
6 namespace MiProyectoFormulario.Controllers
7 {
8     public class FormasController : Controller
9     {
10         // GET: Formas
11         public ActionResult Contacto()
12         {
13             return View("/Views/Formularios/Contacto.cshtml");
14         }
15     }
16
17     public ActionResult GuardarContacto(string nombre, string apellido, string email, string mensaje)
18     {
19         string nombreCompleto = nombre.ToUpper() + " " + apellido.ToUpper();
20
21         // Armo el mensaje
22         System.Net.Mail.MailMessage mailMessage = new System.Net.Mail.MailMessage();
23         mailMessage.To.Add(email);
24         mailMessage.Subject = "Tu solicitud ha sido procesada";
25         mailMessage.Body = "Hemos recibido tu consulta. Responderemos a la brevedad.";
26         mailMessage.IsBodyHtml = true;
27
28         // me conecto al servidor smtp
29         System.Net.Mail.SmtpClient smtp = new System.Net.Mail.SmtpClient();
30         smtp.Host = "smtp.live.com";
31         smtp.Port = 587;
32         smtp.Credentials = new System.Net.NetworkCredential("miusuario@live.com", "micontraseña@live");
33         smtp.EnableSsl = true;
34
35         // envío el mensaje
36         smtp.Send(mailMessage);
37
38         ViewBag.nombreentero = nombreCompleto;
39         ViewBag.texto = "Te responderemos a la brevedad a " + email;
40         return View("/Views/Formularios/ContactoRespuesta.cshtml");
41     }
42 }
43 }
```

Explorador de soluciones

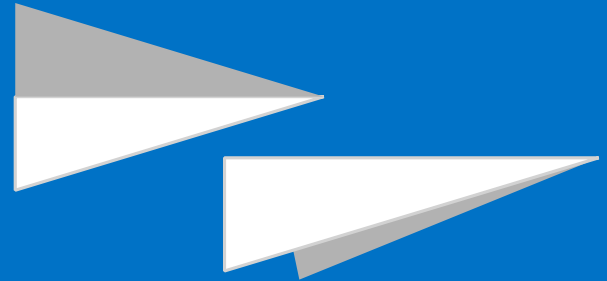
Buscar en Explorador de soluciones (Ctrl+)

- Solución 'MiProyectoFormulario' (1 proyecto)
 - MiProyectoFormulario
 - Properties
 - Referencias
 - App_Data
 - App_Start
 - Areas
 - Content
 - Controllers
 - fonts
 - Models
 - Providers
 - Results
 - Scripts
 - Views
 - ApplicationInsights.config
 - favicon.ico
 - Global.asax
 - packages.config
 - Project_Readme.html
 - Startup.cs
 - Web.config

Explorador de soluciones Team Explorer Vista de clases

Sobre el formulario de contacto creado en los ejercicios anteriores

Además de enviar un email, vamos a proceder a guardar (también puede guardarse sin enviar el correo, para esto habría que reemplazar el código)



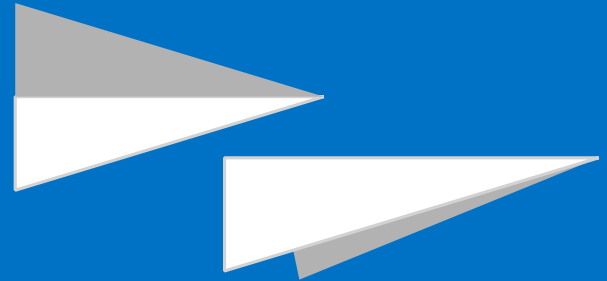
Los parámetros básicos para interactuar con una base de datos son

Dirección IP del Servidor de Base de Datos

Mecanismo de Autenticación

Credenciales de Autenticación

Nombre de la Base de Datos



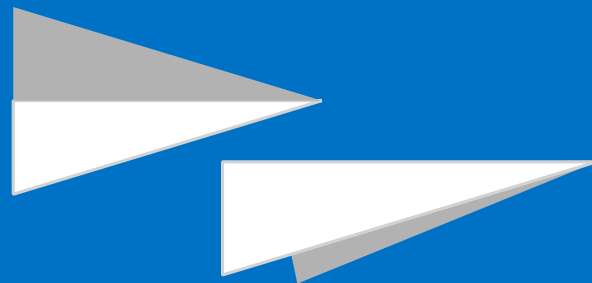
En nuestro caso los parámetros son

Dirección IP: localhost –significa que está instalado en nuestra máquina-

Mecanismo de autenticación:

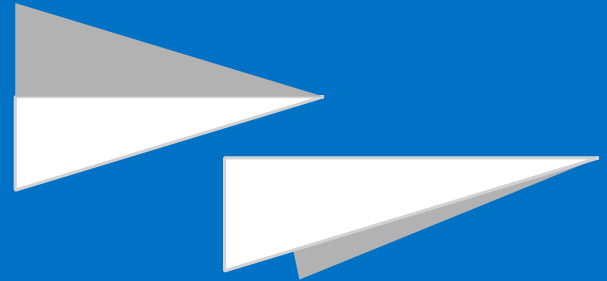
Credenciales de autenticación: no es necesario

Nombre de la Base de Datos: en mi ejemplo, proyecto.



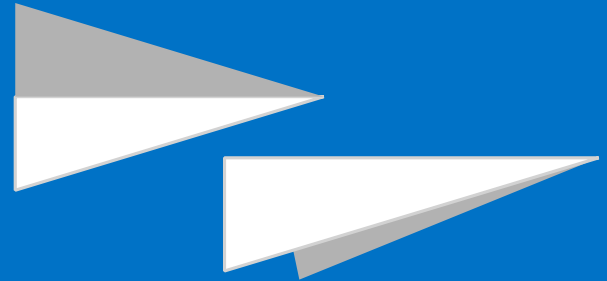
Para conectar los datos con nuestra aplicación bastan 6 pasos

1. Crear la conexión
2. Crear la sentencia
3. Asociar la sentencia a la conexión
4. Ejecutar la sentencia
5. Leer el resultado de la sentencia
6. Cerrar la conexión



Para conectar los datos con nuestra aplicación bastan 6 pasos

1. Crear la conexión
2. Crear la sentencia
3. Asociar la sentencia a la conexión
4. Ejecutar la sentencia
5. Leer el resultado de la sentencia
6. Cerrar la conexión

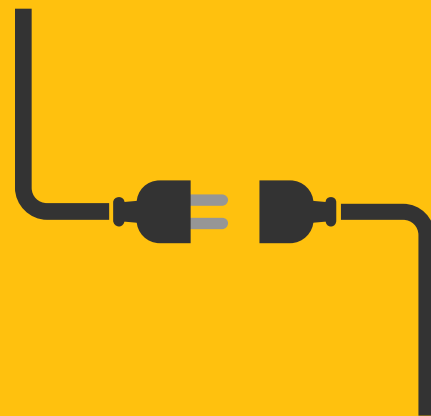
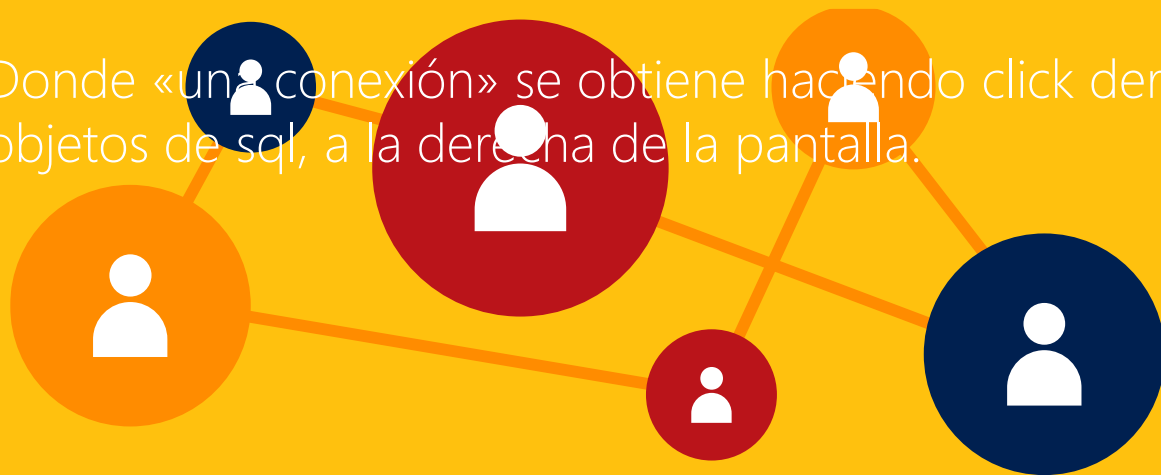


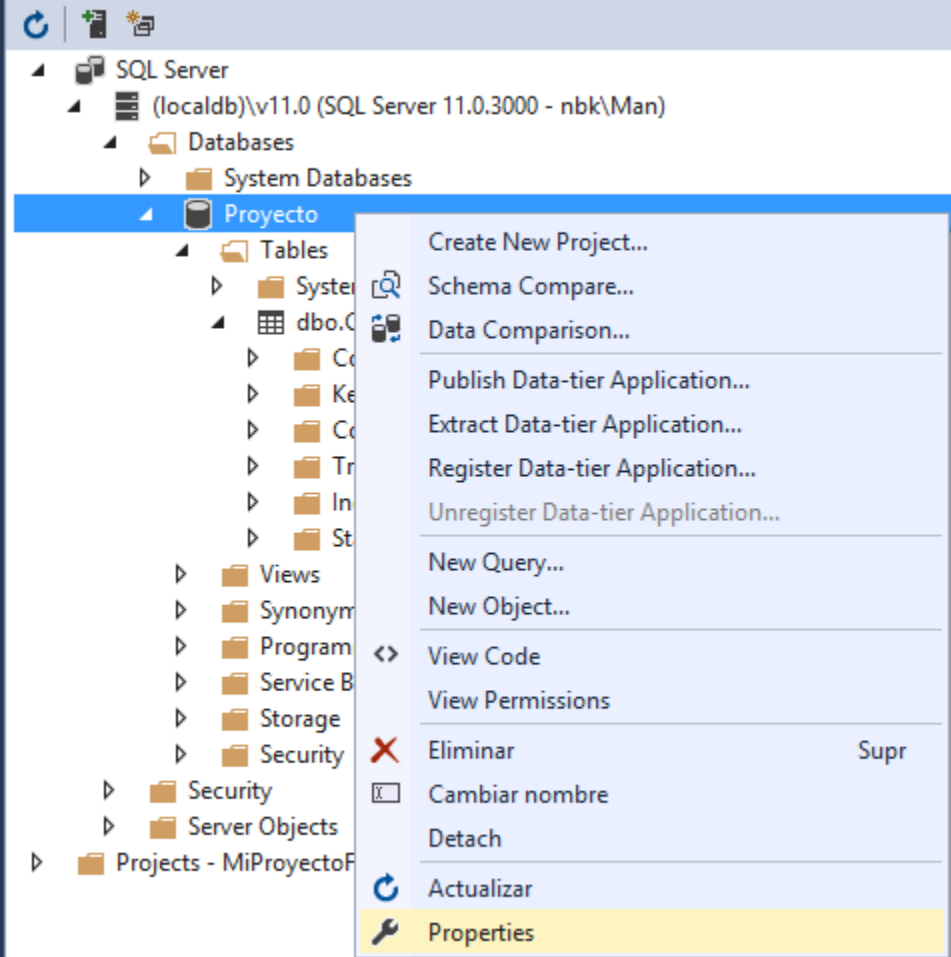
1

Crear la conexión

```
var connection = new System.Data.SqlClient.SqlConnection();  
connection.ConnectionString = "una conexión";  
connection.Open();
```

Donde «una conexión» se obtiene haciendo click derecho en el explorador de objetos de sql, a la derecha de la pantalla.

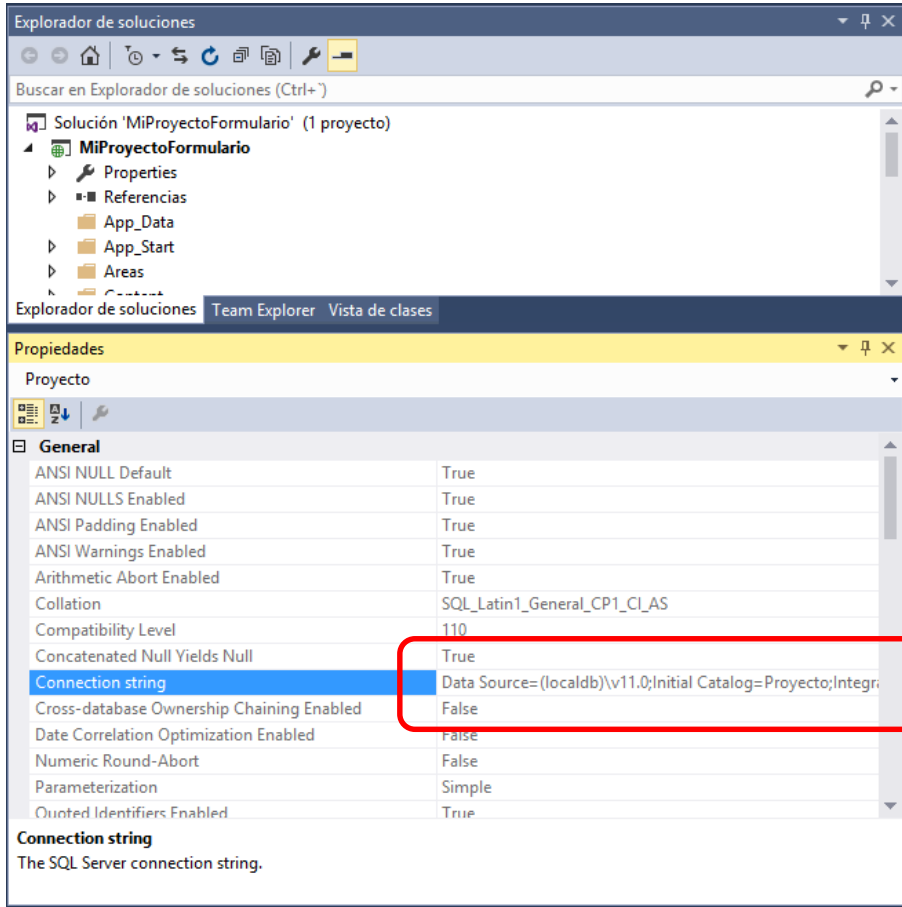




Para extraer la cadena de conexión

Haga clic en el explorador de Objetos de SQL.

Elija la Base de Datos. Clic derecho, propiedades



Copie la cadena de conexión

Es un texto largo, algo parecido a esto


Data

```
Source=(localdb)\Projects;Initial  
Catalog=proyecto;Integrated  
Security=True;Connect  
Timeout=30;Encrypt=False;TrustS  
erverCertificate=False
```

2

Crear la sentencia

```
var sentence = new System.Data.SqlClient.SqlCommand();  
sentence.CommandType = System.Data.CommandType.Text;  
sentence.CommandText = "Insert into Contacto (nombre,apellido,correo),  
values (@pnombre, @papellido, @pcorreo)";  
sentence.Parameters.Add( new  
System.Data.SqlClient.SqlParameter("pnombre", nombre ));  
sentence.Parameters.Add( new  
System.Data.SqlClient.SqlParameter("papellido", apellido ));  
sentence.Parameters.Add( new System.Data.SqlClient.SqlParameter("pcorreo",  
correo ));
```

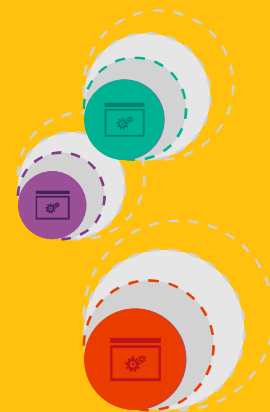


3

Asociar sentencia a conexión

```
sentence.Connection = connection;
```

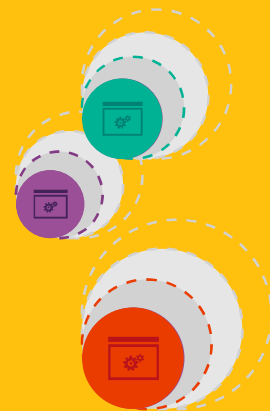
Es la sentencia menos compleja



4

Ejecutar la sentencia

```
var result = sentence.ExecuteNonQuery();
```



5

Leer el resultado de la sentencia

```
var feedbackMessage = "";  
if (result == 0)  
    feedbackMessage = "Exitoso!";  
else  
    feedbackMessage = "Fallo";
```

Según la sentencia podrá entregarse uno u otro mensaje

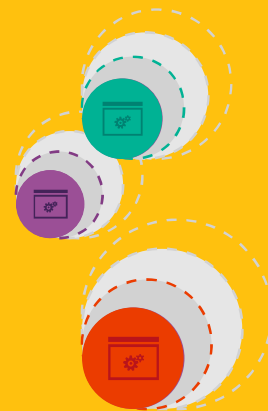


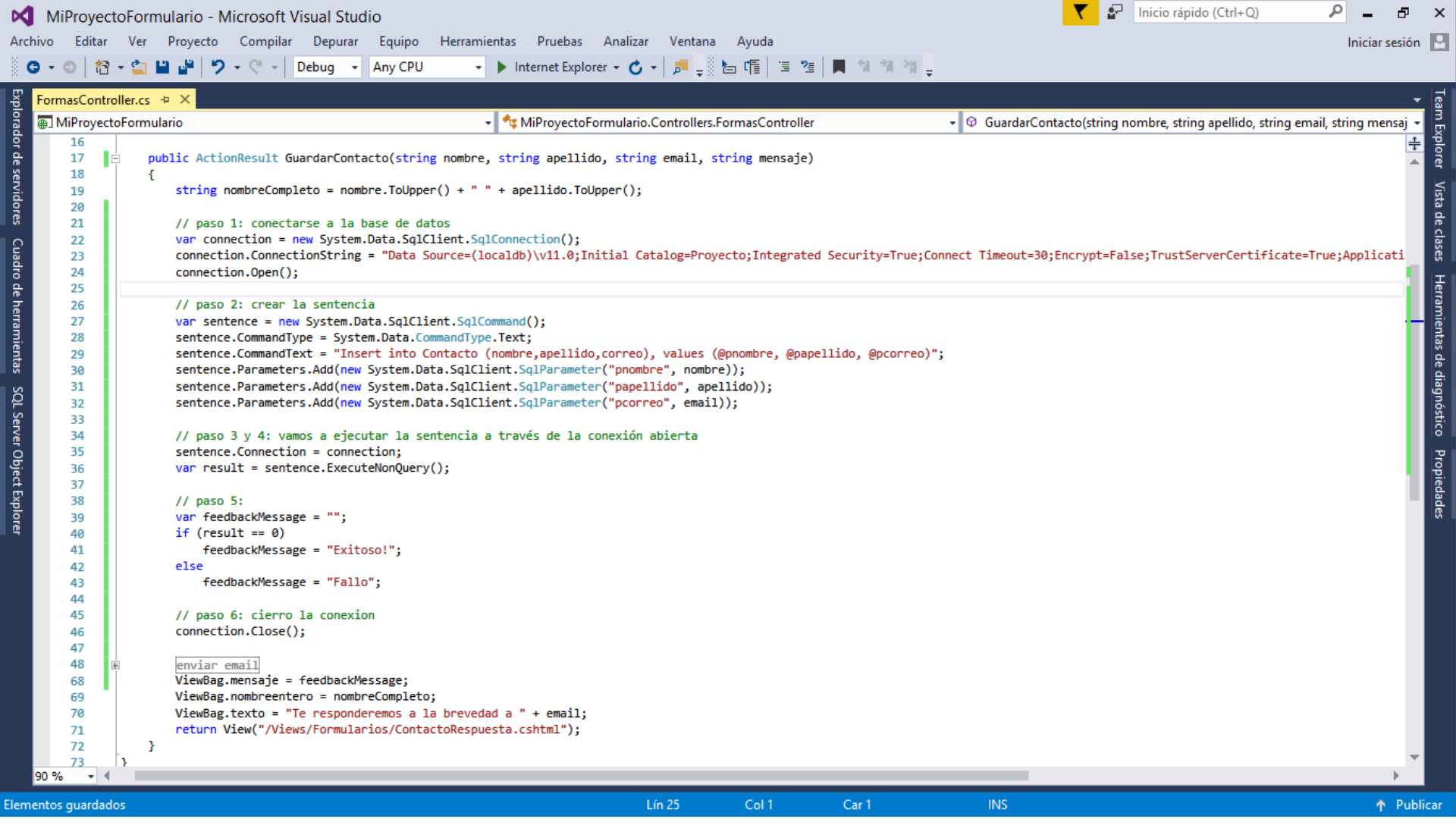
6

Cerrar la conexión

```
connection.Close();
```

Si no se cierra la conexión, se corre el riesgo de no poder conectarse más tarde desde nuestra aplicación





FormasController.cs

MiProyectoFormulario MiProyectoFormulario.Controllers.FormasController GuardarContacto(string nombre, string apellido, string email, string mensaje)

```

16
17 public ActionResult GuardarContacto(string nombre, string apellido, string email, string mensaje)
18 {
19     string nombreCompleto = nombre.ToUpper() + " " + apellido.ToUpper();
20
21     // paso 1: conectarse a la base de datos
22     var connection = new System.Data.SqlClient.SqlConnection();
23     connection.ConnectionString = "Data Source=(localdb)\v11.0;Initial Catalog=Proyecto;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=True;Applicati
24     connection.Open();
25
26     // paso 2: crear la sentencia
27     var sentence = new System.Data.SqlClient.SqlCommand();
28     sentence.CommandType = System.Data.CommandType.Text;
29     sentence.CommandText = "Insert into Contacto (nombre,apellido,correo), values (@pnombre, @papellido, @pcorreo)";
30     sentence.Parameters.Add(new System.Data.SqlClient.SqlParameter("pnombre", nombre));
31     sentence.Parameters.Add(new System.Data.SqlClient.SqlParameter("papellido", apellido));
32     sentence.Parameters.Add(new System.Data.SqlClient.SqlParameter("pcorreo", email));
33
34     // paso 3 y 4: vamos a ejecutar la sentencia a través de la conexión abierta
35     sentence.Connection = connection;
36     var result = sentence.ExecuteNonQuery();
37
38     // paso 5:
39     var feedbackMessage = "";
40     if (result == 0)
41         feedbackMessage = "Exitoso!";
42     else
43         feedbackMessage = "Fallo";
44
45     // paso 6: cierro la conexión
46     connection.Close();
47
48     enviar_email
49     ViewBag.mensaje = feedbackMessage;
50     ViewBag.nombreentero = nombreCompleto;
51     ViewBag.texto = "Te responderemos a la brevedad a " + email;
52     return View("/Views/Formularios/ContactoRespuesta.cshtml");
53 }
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

```

Explorador de servidores Cuadro de herramientas SQL Server Object Explorer

Team Explorer Vista de clases Herramientas de diagnóstico Propiedades

Continué ejercitando

Investiga las variantes para poder efectuar una consulta y mostrar los datos. Ten en cuenta que una vez que obtengas los datos de la base de datos puedes enviarlos al controlador utilizando el ViewBag y luego imprimirlos en la vista



Continué ejercitando

Investiga las variantes para borrar un datos.

Usa el buscador web para buscar ejemplos de escenarios CRUDL (Create Read Update Delete y List) bajo ASP.NET MVC, intenta implementarlos.

Coloca las claves «CRUDL Asp.NET»





© 2016 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.