
	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 1 de 8		

TEMA: PROCEDIMIENTOS ALMACENADOS EN SQL SERVER

OBJETIVO:

- ✓ Adquirir los conocimientos necesarios para desarrollar e implementar procedimientos almacenados utilizando SQL Server y lenguaje C#

I. PROCEDIMIENTOS ALMACENADOS (STORED PROCEDURE)

Un procedimiento almacenado o Stored Procedure (En inglés) es un programa (o procedimiento) que es creado y almacenado dentro de una determinada base de datos. La ventaja de un procedimiento almacenado es que al ser invocado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos. En ese orden de ideas los procedimientos almacenados poseen acceso directo a los datos que necesitan manipular desde la base de datos donde han sido creados.

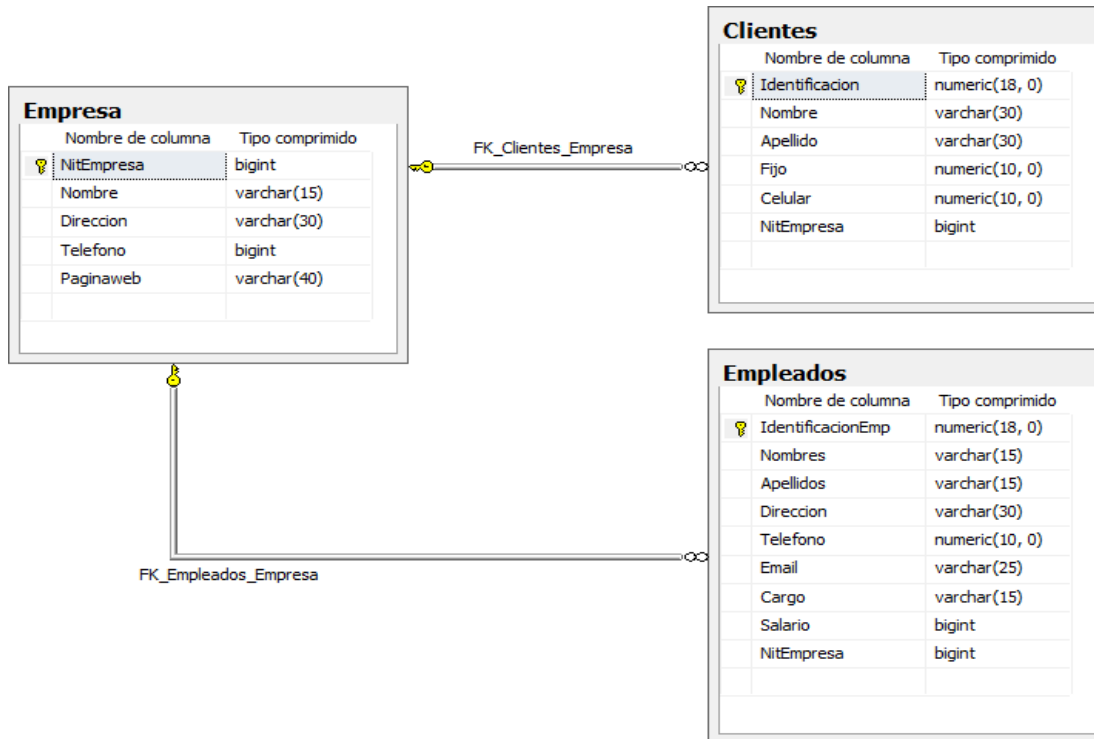
Un Procedimiento Almacenado son unidades de código compuestas por una o más sentencias SQL (Transact-SQL) que son almacenados y ejecutados directamente en el servidor de bases de datos. Prácticamente todo tipo de Sentencias SQL son permitidas dentro de un procedimiento almacenado y por ende estas sentencias pueden ser de inserción, modificación, eliminación y de consultas.

La sintaxis general para crear un procedimiento almacenado en SQL Server es:

```
CREATE PROCEDURE Nombre_Del_Prodedimiento
[ ( ) [ { @parametro Tipo_De_Dato[,][ ... n ] }
AS BEGIN
{ Sentencias SQL[;][ ... n ] }
END
```

EJEMPLOS PROCEDIMIENTOS ALMACENADOS (SQL SERVER)

A continuación encontrará una serie de de procedimientos almacenados en Sql Server, para poder crearlos y verificarlos se debe primero crear la base de datos que se muestra a continuación:



Procedimiento Almacenado N°1: Procedimiento llamado “AddEmpresa” que permite agregar un nuevo registro en una tabla Empresa

```
CREATE PROCEDURE AddEmpresa
```

```

    @NitEmpresa bigint
    ,@Nombre varchar(15)
    ,@Direccion varchar(30)
    ,@Telefono bigint
    ,@Paginaweb varchar(15)

```

```
AS
BEGIN
```

```

INSERT INTO Empresa
    (NitEmpresa
    ,Nombre
    ,Direccion
    ,Telefono
    ,Paginaweb)



```

```

VALUES
    (@NitEmpresa
    ,@Nombre
    ,@Direccion
    ,@Telefono
    ,@Paginaweb)

```

```
END
```

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 3 de 8		

Procedimiento Almacenado N°2: Procedimiento llamado “ConsultarEmpresas” que permite consultar todas las empresas almacenadas.

```
CREATE PROCEDURE ConsultarEmpresas
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Empresa
```

```
END
```

Procedimiento Almacenado N°3: Procedimiento llamado “ConsultarEmpleadoXCargo” que permite consultar empleados filtrados por su cargo. **Nota:** Para este ejemplo se debe insertar datos directamente en la tabla y luego ejecutar el procedimiento.

```
CREATE PROCEDURE ConsultarEmpleadoXCargo
```

```
    @Cargo varchar(15)
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Empleados
```

```
    WHERE Cargo=@Cargo
```

```
END
```

Procedimiento Almacenado N°4: Procedimiento llamado “AddCliente” que permite crear un nuevo registro en la tabla “Clientes”

```
CREATE PROCEDURE AddCliente
```

```
    @Identificacion numeric(18,0)
```

```
    ,@Nombre varchar(30)
```

```
    ,@Apellido varchar(30)
```

```
    ,@Fijo numeric(10,0)
```

```
    ,@Celular numeric(10,0)
```

```
    ,@NitEmpresa bigint
```

```
AS
```

```
BEGIN
```

```
INSERT INTO Clientes
```

```
    (Identificacion
```

```
    ,Nombre
```

```
    ,Apellido
```

```
    ,Fijo
```

```
    ,Celular
```

```
    ,NitEmpresa)
```

```
VALUES
```

```
    (@Identificacion
```

```
    ,@Nombre
```



```
    ,@Apellido
```

```
    ,@Fijo
```

```
    ,@Celular
```

```
    ,@NitEmpresa)
```

```
END
```

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 4 de 8		

Procedimiento Almacenado N°5: Procedimiento llamado “ConsultarClientexIdentificacion” que permite consultar un cliente por su numero de identificación



```
CREATE PROCEDURE ConsultarClientexIdentificacion
    @identificacion int
AS
BEGIN
    SELECT *
    FROM Clientes
    WHERE identificacion=@identificacion
END
```

Procedimiento Almacenado N°6: Procedimiento llamado “ConsultarClientes” que permite consultar clientes por cualquier columna (En este caso Por Identificación, Por Nombre, Por apellido y Todos)

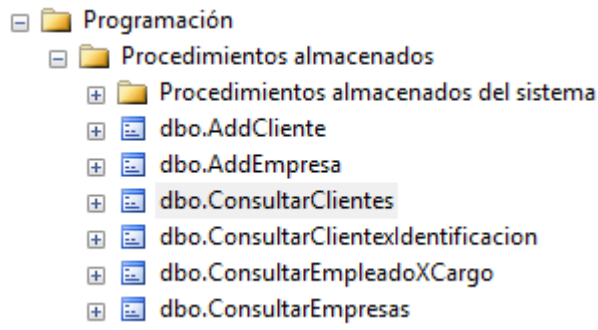
```
CREATE PROCEDURE ConsultarClientes
    @Valor varchar(15),
    @Columna varchar(15)
AS
BEGIN
    IF @Columna = 'Todas'
        SELECT * FROM Clientes
    Else IF @Columna = 'Identificacion'
        SELECT * FROM Clientes
        WHERE identificacion=@Valor
    Else IF @Columna = 'Nombre'
        SELECT * FROM Clientes
        WHERE Nombre like @Valor + '%'
    Else IF @Columna = 'Apellido'
        SELECT * FROM Clientes
        WHERE Apellido like @Valor + '%'
End
```

Nota Importante 1: Para que estos procedimientos funcionen se requiere que exista en la base de datos con una distribución de campos tal y como se muestran que se muestra en el modelo relacional al inicio de la guía.

Nota Importante 2: Para aplicar estos códigos que crean Procedimientos Almacenados se hacen igual a la ejecución de cualquier otra sentencia SQL (Insert, Select, Update..etc), es decir, abriendo una nueva consulta dentro de una Base de datos y luego dando click en ejecutar.

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 5 de 8		

Nota Importante 3: Una vez ha sido creados los procedimientos almacenados estos deben aparecer disponibles dentro de la carpeta “Programación – Procedimientos almacenados” de su base de datos. Para probar el funcionamiento de un procedimiento basta con hacer click derecho sobre el procedimiento y escoger la opción “Ejecutar Procedimiento Almacenado”





CODIGO EN LENGUAJE C# PARA INVOCAR PROCEDIMIENTOS ALMECENADOS

Un ejemplo de un posible código en lenguaje C# que permitiría ejecutar los procedimientos almacenados desarrollados en esta guía sería:

```
public class Cliente:Conexion
{
    private long identificacion;
    private string nombre;
    private string apellido;
    private string fijo;
    private string celular;
    private long nitempresa;

    public long Identificacion
    {
        get { return identificacion; }
        set { identificacion = value; }
    }
    public string Nombre
    {
        get { return nombre; }
        set { nombre = value; }
    }
    public string Apellido
    {
        get { return apellido; }
        set { apellido = value; }
    }
}
```

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 6 de 8		

```

public string Fijo
{
    get { return fijo; }
    set { fijo = value; }
}
public string Celular
{
    get { return celular; }
    set { celular = value; }
}

public int Nitempresa
{
    get { return nitempresa; }
    set { nitempresa = value; }
}

public bool InsertarCliente()
{
    string ProcedimientoInsertar =
"EXEC AddCliente @Identificacion = " + this.identificacion +
",@Nombre = '" + this.nombre + "', @Apellido = '" + this.apellido +
"', @Fijo = " + this.fijo + ", @Celular = " + this.celular + ",
@NitEmpresa = " + this.nitempresa + "'";

    bool respuestaSQL = EjecutarSQL(ProcedimientoInsertar);
    return respuestaSQL;
}



public DataSet ConsultarCliente(string identificacion)
{
    string ProcedimientoDeConsulta =
"EXEC ConsultarClientexidentificacion @Identificacion = " +
identificacion + "'";

    DataSet ConsultaResultante = ConsultarSQL(ProcedimientoDeConsulta);
    return ConsultaResultante;
}

public DataSet ConsultarCliente(string Valor, string Columna)
{
    string ProcedimientoDeConsulta =
"EXEC ConsultarClientes @Valor = '" + Valor + "', @Columna = '" +
Columna + "'";

    DataSet ConsultaResultante = ConsultarSQL(ProcedimientoDeConsulta);
    return ConsultaResultante;
}
}
}

```

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 7 de 8		

EJEMPLO DE CONSULTA POR DIFERENTES CAMPOS DE UNA MISMA TABLA USANDO PROCEDIMIENTOS ALMACENADOS

Con el código que se muestra a continuación podrá realizar consulta por varios campos (Columnas) de una tabla (ejemplo: Por Identificación, Por Nombre, Por Apellido, etc). Para este ejemplo se usa el procedimiento almacenado llamado "ConsultarClientes" cuyo código ya se mostró anteriormente.

El ejercicio requiere de un proyecto a tres capas distribuido así:

- ✓ **Capa De Datos:** Con una clase conexión y una base de datos en sql server que conste de una tabla llamada clientes con los campos (Identificación, Nombre, Apellido, Fijo, Celular, Id_Usuario y NitEmpresa).

En esta base de datos deben crearse los procedimientos Almacenados mostrados en la primera parte de esta guía.

- ✓ **Capa De Logica:** Coloque aquí la clase "Cliente" del punto anterior de esta guía
- ✓ **Capa De Presentación:** Con un formulario con las pestañas guardar, actualizar/eliminar y consultar



En la pestaña consultar debe distribuir controles de la siguiente forma:

1. Un Label que diga "Ingrese Valor a buscar"
2. Un Textbox
3. Un comboBox con las Opciones "Por Identificación - Por Nombre - Por Apellido - Ver Todos)
4. Un Boton con el nombre "Consultar"
5. Un dataGridView

El código que debe aplicar en el evento click del botón Consultar en esta pestaña debe ser:

```
private void buttonX_Click(object sender, EventArgs e)
{
    try
    {
        Cliente ObjCliente = new Cliente();
        DataSet DatosCliente = new DataSet();

        if (comboBox1.SelectedIndex == 0)
        {
            DatosCliente = ObjCliente.ConsultarCliente(textBox1.Text,
                "Identificación");
        }
        else if (comboBox1.SelectedIndex == 1)
        {
            DatosCliente = ObjCliente.ConsultarCliente(textBox1.Text,
                "Nombre");
        }
    }
}
```

	GUÍA DE TRABAJO N° 9 - LENGUAJES C# + SQL		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 8 de 8		

```

else if (comboBox1.SelectedIndex == 2)
{
    DatosCliente = ObjCliente.ConsultarCliente(textBox1.Text,
        "Apellido");
}
else
{
    DatosCliente = ObjCliente.ConsultarCliente("", "Todas");
}

DataTable DatosConsultados = DatosCliente.Tables["DatosConsultados"];

int numregistros = DatosConsultados.Rows.Count;

if (numregistros == 0)
{
    MessageBox.Show("No existe en la Base de Datos Cliente con el
        valor buscado");
}
else
{
    dataGridView1.DataSource = DatosConsultados;
}

catch (Exception ex )
{
    MessageBox.Show("No se ha podido efectuar la consulta debido al
        siguiente Error: " + ex.Message);
}
}

```

TALLER:

Completar el ejercicio de la aplicación tres capas para la tabla cliente programando las pestañas guardar y actualizar/eliminar usando procedimientos almacenados basandose en los codigos de ejemplo mostrados en esta guia.

IMPORTANTE:

- ✓ En la base de datos debe crear ademas de los procedimientos de ejemplo de esta guia (AddCliente, ConsultarClientexidentificacion y ConsultarClientes) los procedimientos "ActualizarCliente" y "EliminarCliente".
- ✓ En la clase cliente del codigo C# debe agregar los metodos que permitan Guardar, Actualizar y Eliminar Cliente en donde debe invocar los procedimientos correspondientes usando la palabra clave "EXEC"