

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET

Prof.

Samuel Ribeiro (Samuka)

PROCESSWARE

Consultoria e Treinamento

E-MAIL: professor.samuka@gmail.com 2004

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET

Com a necessidade de grandes empresas precisarem integrar todo o banco de dados corporativo com a Internet, tornou-se indispensável facilitar o acesso à base de dados com uma boa performance, para que os aplicativos consigam uma importância significativa.

A Plataforma .Net inova também nesta parte. O velho ADO necessitava de mudanças, ficou obsoleto principalmente com relação às aplicações baseadas na Web, então a Microsoft criou o ADO.NET e toda uma arquitetura baseada em XML. Com isso podemos afirmar que esta tecnologia trabalha com dados desconectados.

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

Vantagens do ADO.NET

Escalabilidade – pelo fato de o DataSet ser baseado em acesso a dados desconectados, por ter uma arquitetura baseada no XML, o tempo de manipulação dos dados no banco de dados torna-se mínimo. Portanto mesmo com um número simultâneo de acesso maior, a aplicação consegue garantir uma boa escalabilidade;

Performance – no ADO.NET a transmissão de dados é feita em XML, com isso pode se comunicar com diversas plataformas e aplicativos;

Segurança – um firewall não consegue bloquear um arquivo texto. Portanto como o ADO.NET é baseado em XML, as portas dos firewalls não são mais problemas.

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET

O ADO.NET disponibiliza classes para a manipulação dos dados. Portanto depende da classe System.Data que contem os seguintes namespaces:

- System.Data
- System.Data.OleDb
- System.Data.SqlTypes
- System.Data.SqlClient



CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

controles de validação

ADO.NET - SQL

Este provedor é para o acesso exclusiva do SQL Server. Pertence a classe System.Data.SqlClient e seus principais objetos estão representados na tabela 7.2.

Objeto	Descrição
OleDbConnection	Define a abertura da conexão
OleDbCommand	Define a instrução Sql a ser executada
OleDbDataReader	Define somente para leitura um conjunto de dados
OleDbDataAdapter	Define a conexão a ser usada para preencher um DataSet, e Representa um conjunto de comandos de dados

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

controles de validação

ADO.NET - DataSet

O ADO.NET baseia-se no DataSet. Esse objeto é um conceito completamente novo para substituir o tradicional Recordset do ADO. O DataSet é um armazenamento de dados simples residente na memória, que fornece um modelo de programação consistente de acesso a dados, independentemente do tipo de dados.

Podemos afirmar também que, os componentes do DataSet foram desenvolvidos para manipular dados com mais rapidez, sendo possível executar comandos de leitura, alteração de dados, stored procedures e enviar ou receber informações parametrizadas.

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

controles de validação

ADO.NET - Conectando com um Banco de Dados

Para toda manipulação de dados realizada via ADO.NET, é preciso estabelecer uma conexão para montar um DataSet.

```
using System.Data.OleDb;  
OleDb Connection conn;  
conn = new OleDbConnection ("Data Source=localhost;Integrated  
Security=SSPI;Initial Catalog=northwind");  
OleDbCommand cmd = new OleDbCommand(INSTRUÇÃO SQL);  
cmd.Connection = conn;  
conn.Open( );
```

Para fechar a conexão, utilize a sintaxe:
conn.Close();

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET - DataReader

Este é o objeto que armazena o resultado de uma consulta ou stored procedure executada. Dependendo da forma de acesso (SQL ou OleDb), o desenvolvedor precisa respeitar os métodos existentes em cada objeto.



CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET - DataReader

```
String InstrucaoSql = "SELECT * FROM Products";  
OleDbConnection conn;  
OleDbCommand cmd;  
OleDbDataReader Dr;  
conn = new OleDbConnection (@"Provider=Microsoft.Jet.OLEDB.4.0;  
Persist Security Info=False;Data Source=f:\BANCO\samudb.mdb");  
cmd = New OleDbCommand(InstrucaoSql, conn);  
cmd.Connection.Open( );  
DR = cmd.ExecuteReader(CommandBehavior.CloseConnection);  
While (Dr.Reader() )  
{  
    TextBox1.Text += Dr["Matricula"] + " - " + Dr["Nome"] + "\n";  
}  
Dr.Close( );  
conn.Close( );
```

OBS: não se esqueça de importar a classe: System.Data.OleDb;

CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

ADO.NET - DataReader

Podemos realizar uma consulta com a utilização do objeto SqlDataAdapter. No exemplo abaixo, temos um método chamado Consultar() que retorna um DataSet.

```
public void DataSet consultar(string InstrucaoSql)
{
    OleDbConnection objconn = new
    OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0; " +
    "Persist Security Info=False;Data Source=f:\BANCO\samudb.mdb");
    OleDbDataAdapter objda = new OleDbDataAdapter(InstrucaoSql, objconn);
    DataSet objds = new DataSet();
    objda.Fill(objds, "Listar");
    return objds;
}
```

OBS: não se esqueça de importar a classe: System.Data.OleDb