

Delphi 2005 e .NET

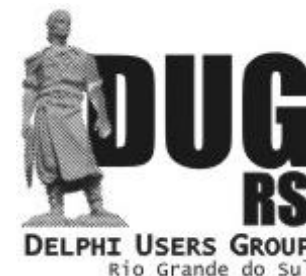
Benefícios e Novidades em Exemplos Práticos

André Sandri

Analista/Programador

www.sandri.cjb.net

andresandri@hotmail.com



Objetivos

- O que é .NET?
- Características Principais
- Benefícios
- Novidades
- Diferenças e Similaridades
- Exemplos Práticos

Microsoft® .NET Framework Versão 1.1

- **O que é?**
- **Os componentes chaves são:**
 - **CLR - Common Language Runtime**
 - **.NET Framework Class Library**
- **Este framework proporciona um ambiente de execução gerenciado, simplificando o desenvolvimento, a publicação e a integração com várias linguagens de programação.**

O Framework

- Framework foi planejado para atender os seguintes objetivos:
 - Oferecer um ambiente de programação orientado a objeto onde o código pode ser armazenado e executado localmente, executado localmente e distribuído pela Internet, ou executado remotamente.
 - Oferecer um ambiente de execução de código que minimiza o desenvolvimento de software e conflitos de versões.
 - Oferecer um ambiente de execução seguro de código, incluindo código criado por autores desconhecidos ou por empresas não confiáveis.
 - Oferecer um ambiente de execução de código que elimina problemas de performance de scripts e códigos interpretados.
 - Aproveitar a experiência do desenvolvedor entre vários tipos de aplicações, como aplicativos Windows, aplicativos para dispositivos móveis e aplicativos WEB.

Instalação

.NET Framework Version 1.1 Redistributable Package

23.698 KB

.NET Framework SDK Version 1.1

108.757 KB

.NET Framework 2.0 Redistributable Package Beta 2

22.989 KB

.NET Framework 2.0 SDK Beta 2 (x86)

317.018 KB

Requisitos

- Cliente
 - Microsoft® Windows® 98 a Microsoft® Windows® Server 2003
 - Pentium 90 MHz - 96 MB RAM
- Servidor
 - Microsoft® Windows® 2000 Professional (SP2) a Microsoft® Windows® Server 2003
 - Pentium 133 MHz – 256 MB RAM

Runtime

É um agente que controla a execução de código, oferecendo:

- Serviços de gerência de memória
- Serviços de gerência de threads
- Segurança
- Remoting
- Outros

Runtime

Código que pode ser executado pelo Runtime pode ser:

- Código Gerenciado (managed code)
- Código não-gerenciado (unmanaged code)

Segurança

- Conforme o local de execução e a política de segurança, o Runtime pode negar acesso a funções (acesso a disco, acesso ao registro, etc).
- Usuários e administradores da rede podem configurar qual código é seguro ou não para execução.

Gerência de Memória

- A Gerência Automática de Memória evita "memory leaks" e referências inválidas.
- Garbage Collector (coleta de lixo)
- Código com ponteiros devem ser alterados (safe code).

Linguagens

- Delphi (Object Pascal)
- C#
- C++
- Visual Basic
- J# (similar a Java)
- Eiffel, Perl, COBOL, outros (third-party)

Performance

- O código não é interpretado, e sim executado!
Just-in-time (JIT) compiling
- Boa performance!
- Performance Counters
- Caches (principalmente em ASP.NET)

Compilar / Executar

- **Compilar:** Código fonte para Microsoft Intermediate Language (MSIL)
- MSIL é gerado respeitando a Common Language Specification (CLS)
- **Executar:** O compilador JIT (Just-in-time) automaticamente traduz o código MSIL em código nativo
- O código nativo é gerado conforme o sistema e hardware de execução

Class Library

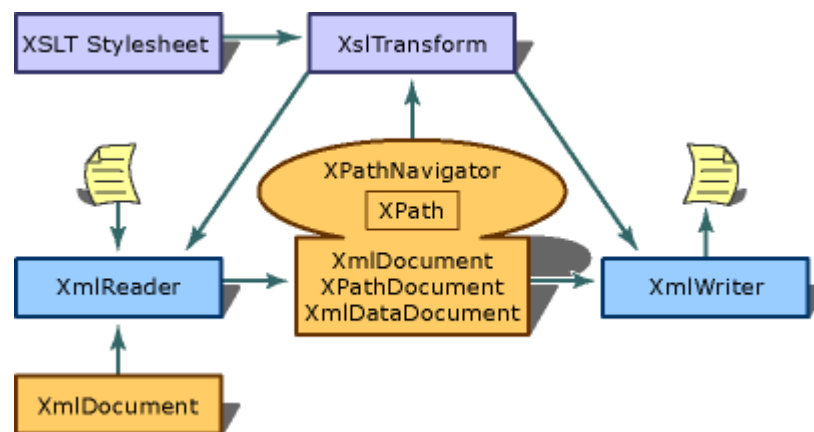
- O que há de novo?
 - XML Web Services
 - Integração de XML com vários componentes
 - Classes para acessar praticamente todos os serviços disponibilizados pelo S.O.
 - Integração com Active Directory
 - Integração fácil com tecnologias COM, ActiveX
 - Windows Management Instrumentation (WMI)

Aplicativos WEB

- A arquitetura da Microsoft mais utilizada era ASP (código interpretado).
 - Linguagens Script: VBScript, JScript
 - Acessava-se componentes externos via ActiveX
- A tendência agora é ASP.NET
 - Pode-se utilizar todos os recursos do Framework de forma direta
 - Classes específicas: WebControls, WebForms, etc.
 - Todas as linguagens suportadas pelo .NET
 - O código pode ser executado (JIT Compiling)

XML e Web Services

- Suporte total a XML e Web Services (classes e ferramentas)
- Interoperabilidade com soluções não-Microsoft (na grande maioria das vezes)
- Novo modelo de criação e disponibilização de Web Services



Side-by-Side

- Desde a versão 1.1
- É a possibilidade de armazenar e executar múltiplas versões de um aplicativo ou componente no mesmo computador, ao mesmo tempo.
- Você pode manter várias versões diferentes do Runtime instalados. Futuras instalações não afetarão o funcionamento dos componentes e aplicativos previamente instalados.
- Se você não decidir, o runtime decide.

Assemblies

- São os arquivos (.exe, .dll, outros) que mantêm código .NET
- Designados para simplificar o desenvolvimento e resolver problemas de versões.
- Fim de "DLL Hell"
- Aposentadoria do IDL (Interface Definition Language)
- Auto Descritíveis (Metadados)

Global Assembly Cache

- Todo computador que tem o Runtime instalado tem um cache de componentes chamado de Global Assembly Cache (GAC)
- É o repositório de "assemblies" projetado para compartilhar componentes para todos os aplicativos deste computador
- Normalmente, os componentes de um aplicativo são mantidos na pasta do aplicativo (pasta "bin").
- Formas de instalação:
 - Preparar um instalador
 - Global Assembly Cache Tool (Gacutil.exe)
 - Arrastar ou Copiar/Colar

Tipos Comuns (Classes)

- * Object (classe raiz da hierarquia de objetos)
- * String (Imutável, tamanho fixo, Unicode)
- * Char (Caractere Unicode 16-bit)
- * Boolean (true ou false)
- * Byte (inteiro sem sinal de 8-bits)
- * Int16 (inteiro com sinal de 16-bits)
- * Int32 (inteiro com sinal de 32-bits)
- * Int64 (inteiro com sinal de 64-bits)
- * UInt64 (inteiro sem sinal de 64-bits)
- * Single (ponto-flutuante de 32-bit)
- * Double (ponto-flutuante de 64-bit)
- * Outros...

Outras Classes do Namespace System

- * Array
- * Buffer
- * Console
- * Environment
- * Exception
- * GC
- * Math
- * OperatingSystem
- * Uri
- * Version

Conversões de Tipos

- Classe de conversão para tipos **Boolean**, **Char**, **SByte**, **Byte**, **Int16**, **Int32**, **Int64**, **UInt16**, **UInt32**, **UInt64**, **Single**, **Double**, **Decimal**, **DateTime** e **String**.
- Falha com exception "InvalidCastException" em poucos casos.
Exemplo: `Num := Convert.ToInt32(DataAtual)`
- Conversões com perda de precisão são permitidas, desde que o resultado não seja maior (não possa ser representado).
- Conversões automáticas (operadores sobrecarregados)

Sobrecarga de Operadores

- Sobrecarga do operador ponto (.)
- Exemplo:

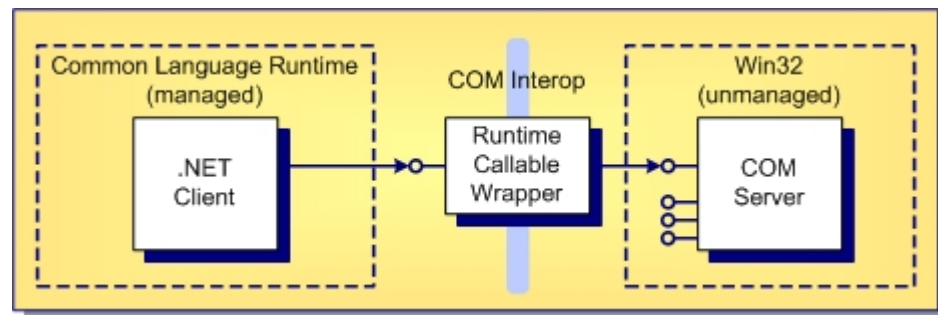
```
CodigoClienteFormatado =  
    CodigoCliente.Trim.PadLeft(10, "0")
```

System.Reflection

- Assemblies – Modules – Types – Members
- GetType
- ConstructorInfo
- MethodInfo
- FieldInfo
- EventInfo
- PropertyInfo
- ParameterInfo

COM Interop

- Integração entre código gerenciado e não-gerenciado
- Permite utilizar ou expor componentes
- Runtime Callable Wrappers e COM Callable Wrappers
- Primary Interop Assembly





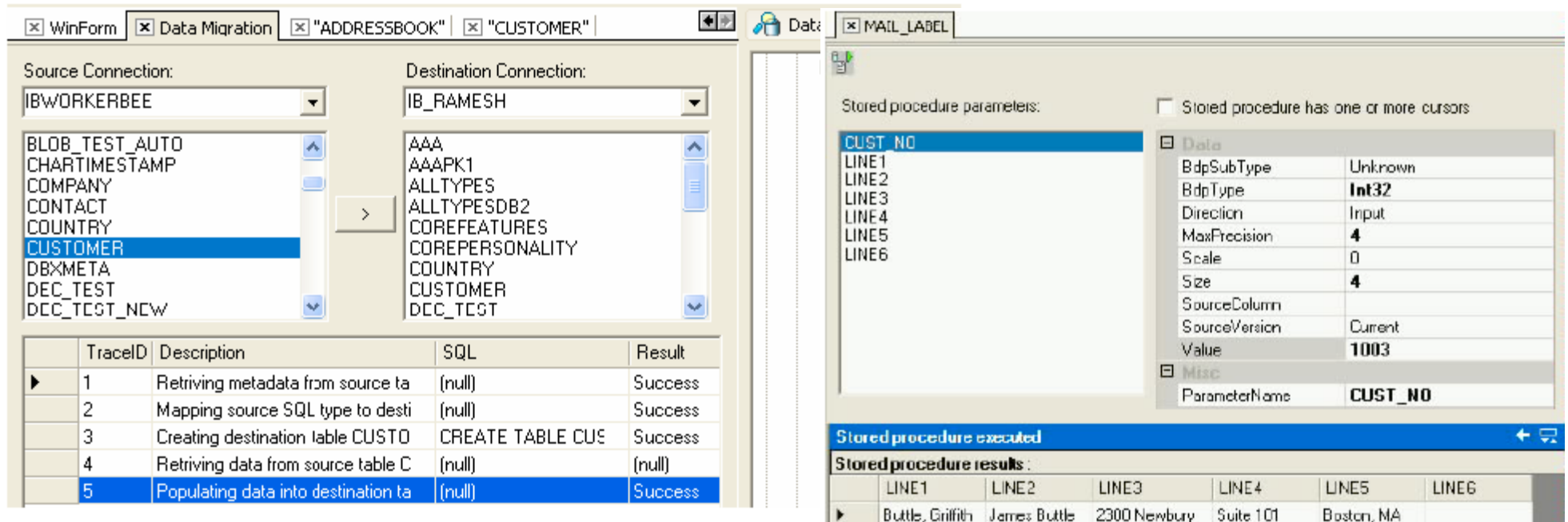
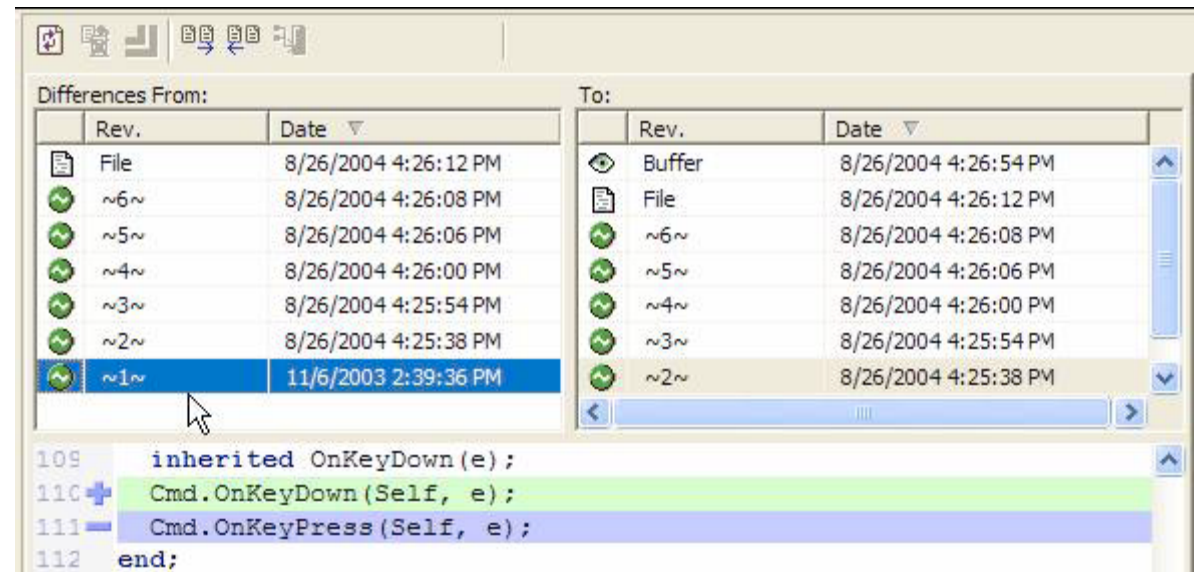
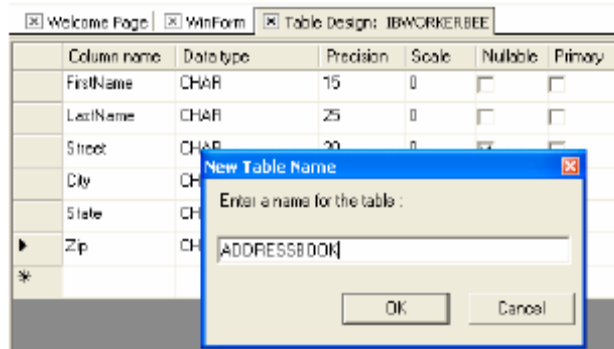
Novidades Delphi 2005

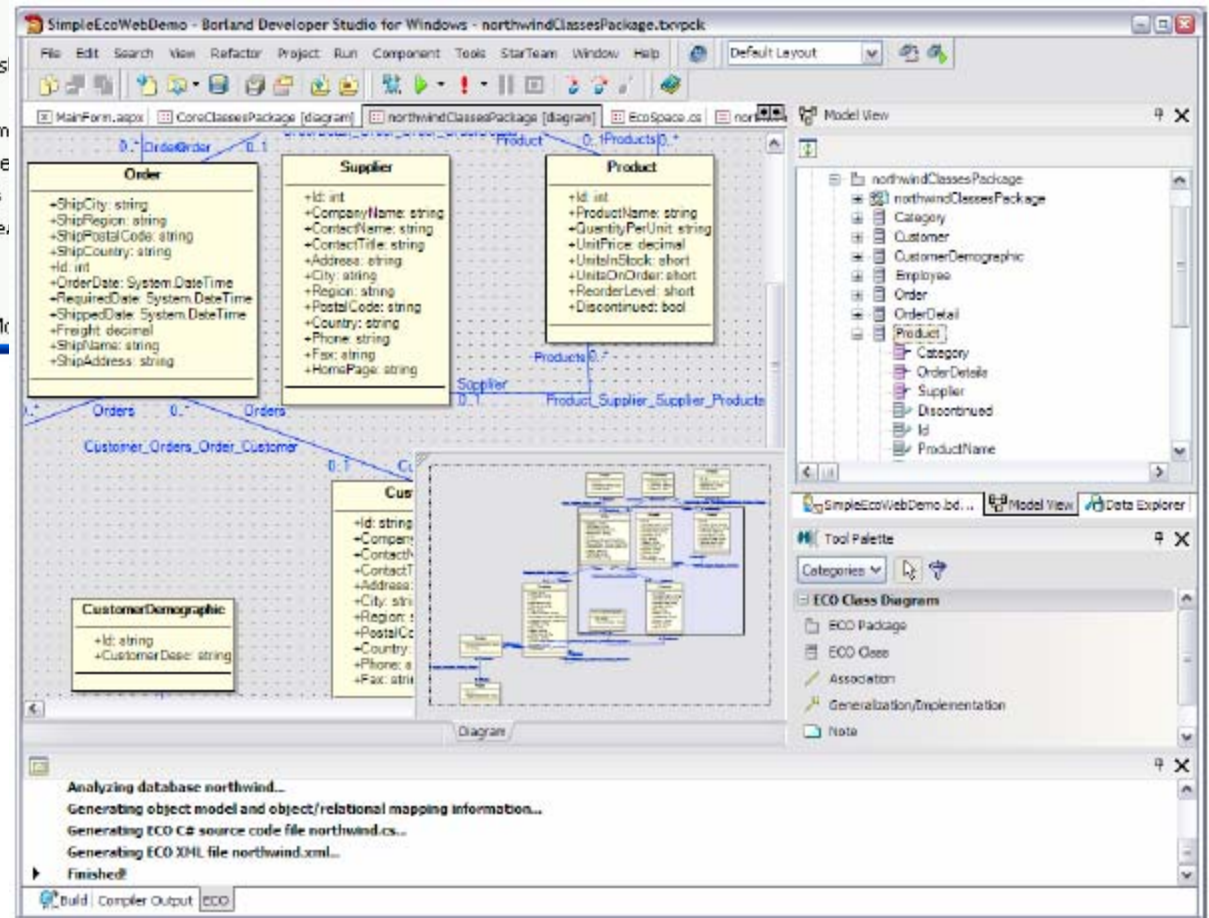
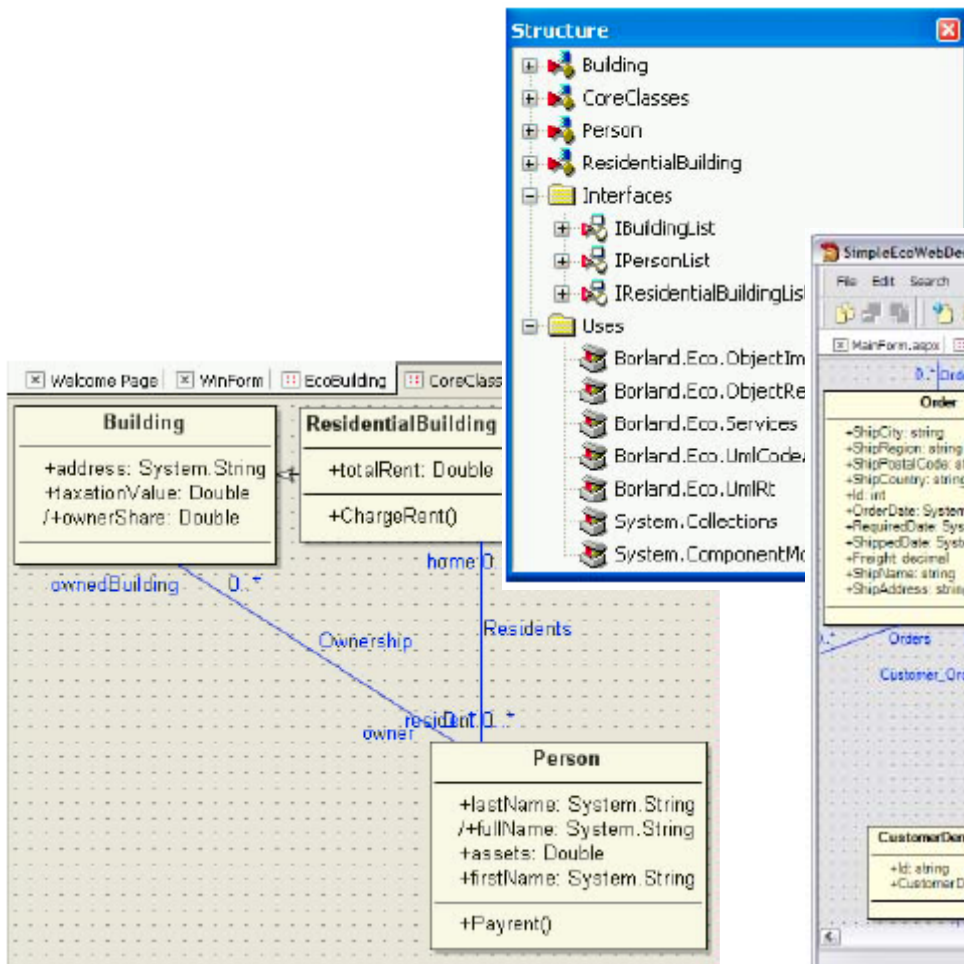
- IDE Único e Integrado
- Integração com CaliberRM e StarTeam
- UML e Enterprise Core Objects (ECO)
- VCL Forms, Windows Forms, Web Forms, páginas HTML
- BDE, dbGO, WebSnap, IntraWeb e Indy foram portados!
- Integração com HTML, CSS, XML, Dados...
- Integração Janeva (CORBA)



Editor de Código

- Refactoring
- Synchronized editing
- Code completion
- Reusable code snippets
- Recorded keystroke macros
- Custom key mappings
- Syntax highlighting
- Code folding





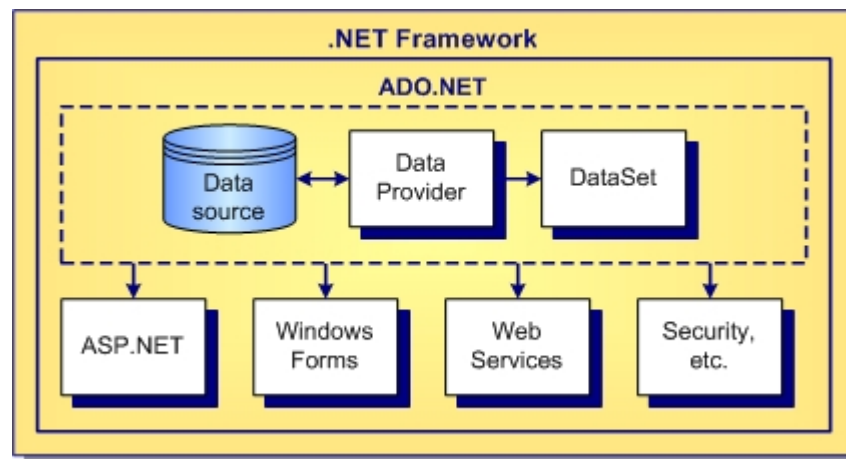


Novos Componentes .NET

- Borland Data Providers for .NET
 - Suporte a InterBase, Oracle, DB2, Microsoft SQL Server, Sybase, outros...
- WebSnap, IntraWeb, Indy, Crystal Reports
- VCL, BDE
- dbGo.Net
- DB Web Controls

ADO.NET

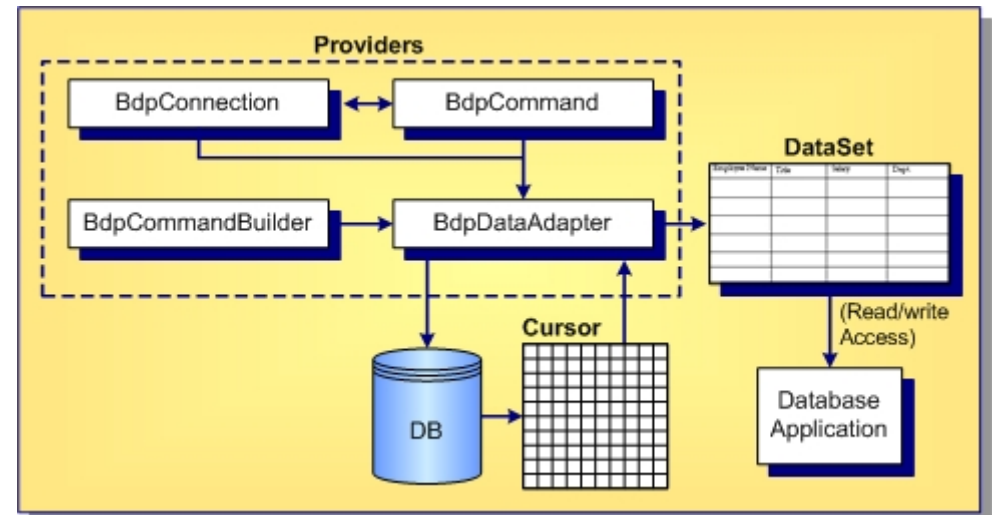
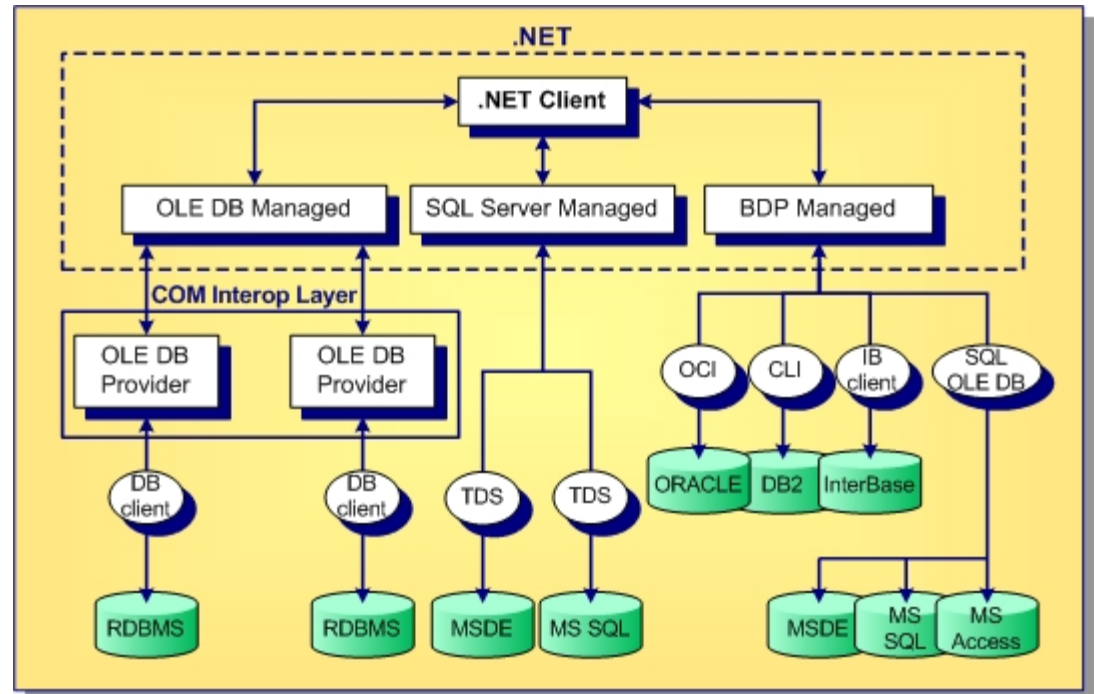
- Nova solução da Microsoft para acessar dados (SGBDs, XML ou outros tipos).
- DataProvider e DataSet



Borland®

BDP.NET

- Componentes genéricos
- Maior performance
- Não exige camada COM
- Live Data Design
- Mais fácil (sem DataBinding)





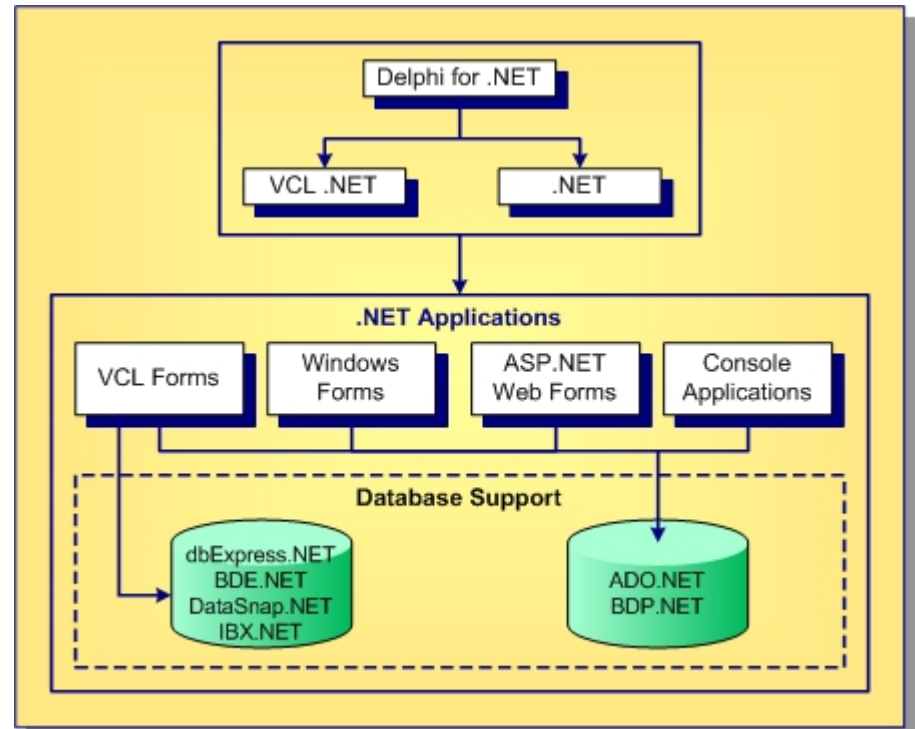
BDP - Borland Data Provider

- Os drivers ADO.NET (mantidos pela Microsoft, Oracle, outros) usam seus próprios tipos de dados.
- Exemplo: SQL Server utiliza a classe **SqlDbType**, enquanto Oracle utiliza a classe **OracleData**.
- BDP elimina estas dificuldades pois o driver é genérico, além de ser uma especificação aberta.
- Para migrar para outra base de dados, basta mudar a string de conexão!

VCL for .NET

- Permite continuar utilizando a VCL em .NET
- Permite migrar aplicativos Win32 para .NET

Namespace
Borland.VCL



Borland®

Muito Mais Possibilidades

- Apenas um exemplo...

INDY – Internet Direct

<http://www.indyproject.org/Features.iwp>



Próximas Palestras

- **Delphi Meeting 2005**

27 de junho - Porto Alegre

<http://info.borland.com.br/delphimeeting/>

- **ClubeDelphi Tech Weekend**

15 e 16 de julho - São Paulo

<http://www.devmedia.com.br/tw/>

Certificação

- Menina de 9 anos recebe certificação da Microsoft

www.terra.com.br

03/Maio/2005

Arfa Karim Randhawa

Faisalabad, Paquistão



Delphi 2005 na Prática

Perguntas?

Muito Obrigado!

andresandri@hotmail.com

Referências:

- www.dug-rs.org
- www.borland.com.br
- www.microsoft.com.br