

UniSALESIANO

The Delphi logo is a stylized blue wordmark with a white outline, set against a white diamond background. This diamond is positioned over a blue square, which is itself on a black background.

Banco de Dados Delphi

Alexandre Marcelino Silva

alexandre@salesiano-ata.br

alexandre@vidadura.org (Facebook/Skype)



☐ Tecnologia de acesso a dados

- O que é?
- Porque utilizar
- BDE
- DBExpress
- ADO
- IBX
- UIB
- FIREDAC



UniSALESIANO

P.O.O. Cliente-Servidor

- **BDE Borland Database Engine**
- A BDE fornece a capacidade de acesso padronizado a banco de dados para Delphi, C++ Builder e outros ambientes de programação da Borland, oferecendo um grande conjunto de funções para auxiliar no desenvolvimento de aplicações Desktop e Cliente/Servidor.

P.O.O. Cliente-Servidor

- Os controladores da BDE podem ser usados para acessar bases de dados dBase, Paradox, Firebird, Access, FoxPro, Interbase, Oracle, Sybase e MSSQL Server, DB2, Informix, além de um controlador de acesso a arquivos texto. Você também pode utilizar fontes de dados ODBC, podendo acessar qualquer base de dados compatível.



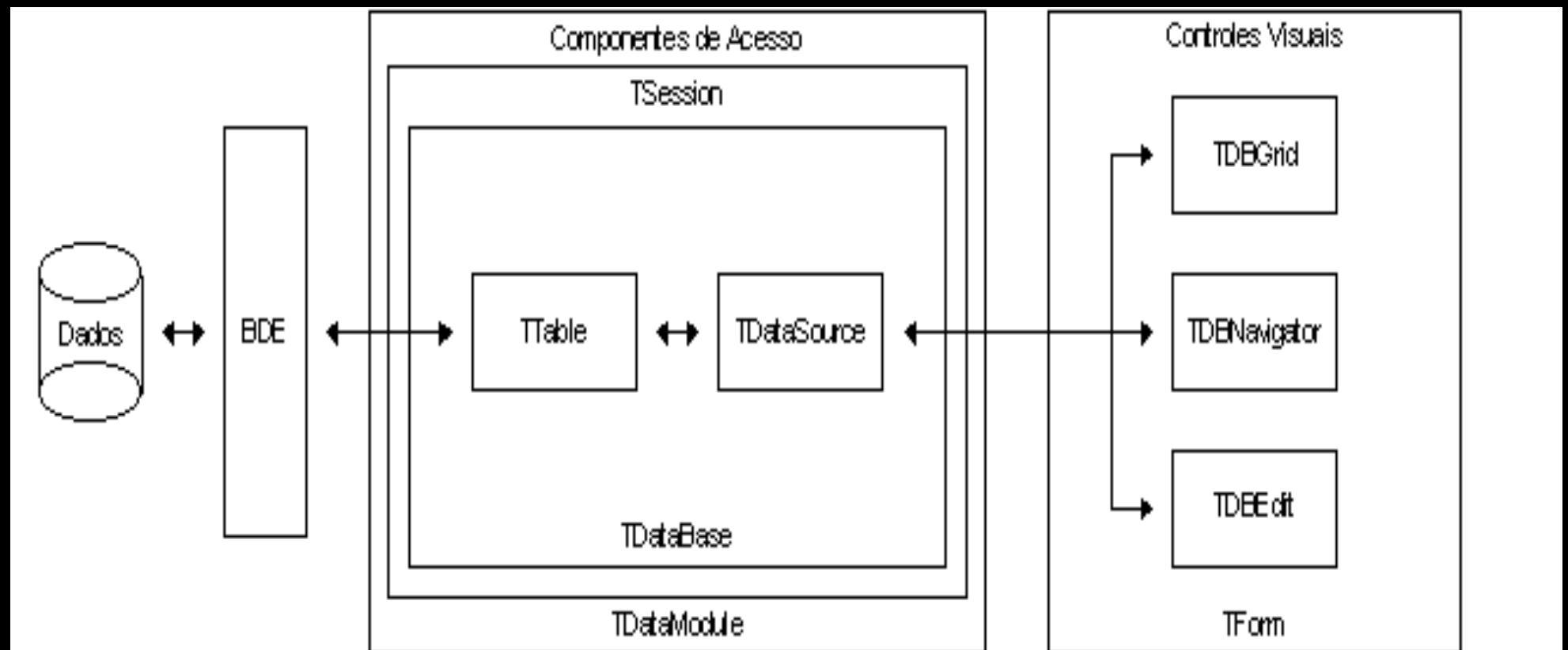
P.O.O. Cliente-Servidor

- **Arquitetura de Acesso**
- O acesso e manipulação de um banco de dados por um programa Delphi é realizado como mostrado abaixo, note que a aplicação não acessa os dados diretamente, mas usa sempre a BDE.



UniSALESIANO

P.O.O. Cliente-Servidor



P.O.O. Cliente-Servidor

- Assim, para uma aplicação de bancos de dados funcionar, é preciso que a BDE esteja instalada na máquina, não bastando apenas o arquivo executável.



P.O.O. Cliente-Servidor

- As principais propriedades dos componentes de Banco de Dados (BDE)
- TTable
 - DatabaseName => Nome da Aliases criada no BDE
 - TableName => Nome da tabela que pretende trabalhar
 - Active => Abre a tabela para poder inserir, editar e excluir
 - Obs.: *Duplo clique na TABLE para adicionar os campos no “editor de campos”.*
 - *Pode também arrastar os campos diretamente para o Formulário!*



P.O.O. Cliente-Servidor

- **TDataSource**
 - DataSet => Nome da Table que deseja "visualizar" os dados
- **TDBGrid**
 - DataSource => Nome do DataSource criado anteriormente
 - *Obs.: Duplo clique na GRID para adicionar os campos na GRID.*
- **TDBNavigator**
 - DataSource => Nome do DataSource criado anteriormente

P.O.O. Cliente-Servidor

- **TDBEdit**
 - DataSource => Nome do DataSource criado anteriormente
 - DataField => Nome do campo que deseja visualizar no DBEdit
- Outros Campos de Formulário praticamente repetem as funções do TDBEdit, mudando somente as características de Obtenção/navegação de informações.



UniSALESIANO

P.O.O. Cliente-Servidor

- Criação de Aliases (Apelido)
 - Define o caminho do Banco de Dados
- Através do painel de controle do windows
 - Ferramentas Administrativas;
 - Fontes de Dados (ODBC);
 - Driver do banco de dados.



UniSALESIANO

P.O.O. Cliente-Servidor

- DBExpress – Pode-se considerá-la a evolução dos Componentes BDE
- Composta dos seguintes componentes:
 - SQLConnection;
 - SQLDataSet;
 - SQLQuery
 - SQLStoreProc;
 - SQLTable;
 - SQLMonitor
 - SQLSimpleDataSet.

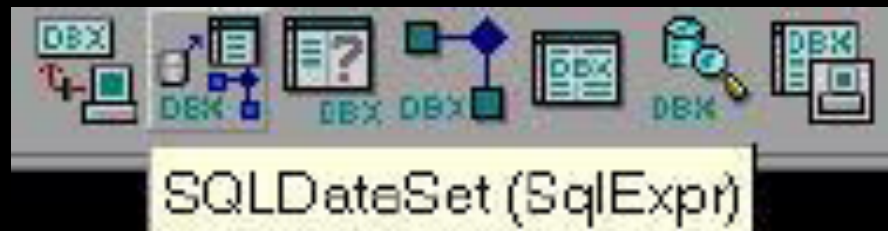


P.O.O. Cliente-Servidor

- 1 – O SqlConnection
 - Esse é o componente responsável pela conexão física do banco de dados. É através dele que obtemos o acesso ao banco de dados, mas não acesso direto ao conteúdo das tabelas.



- 2 – O SQLDataSet
 - Um DataSet é uma estrutura onde são armazenados listas de registros do banco. O SQLDataSet nos permite mostrar o resultado de uma consulta ou de uma Stored Procedure, executar Stored Procedures que não retornem dados, ou obter os parâmetros disponíveis no banco. Ele é unidirecional, ou seja, ele não armazena na memória muitos registros simultâneos e também não suporta atualizações direto na memória.





UniSALESIANO

P.O.O. Cliente-Servidor

- 3 – O SQLQuery
 - Ele executa comandos SQL no banco de dados, retornando resultados de consultas, inserções, exclusões, etc.





UniSALESIANO

P.O.O. Cliente-Servidor

- 4 – O SQLStoredProc
 - É um componente específico para a execução de Stored Procedures armazenadas no banco. Pode armazenar o resultado de uma Stored Procedure que retorne um valor.





P.O.O. Cliente-Servidor

- 5 – O SQLTable
 - O SQLTable representa uma tabela no banco de dados. Ele traz todas as colunas e linhas da tabela especificada, mas também é um componente unidirecional, não permitindo a movimentação entre os registros. Pode realizar inclusões, alterações, etc.



- 6 – O SQLMonitor
 - É um componente utilizado para fazer a depuração da comunicação entre a aplicação e o servidor de banco de dados. Ele grava em logs os comando SQL de uma conexão, adicionando em uma StringList.



- 7 – O O SimpleDataSet
 - É um conjunto dos componentes TSQLDataSet e TDataSetProvider (Provedor de Acesso ao Banco). Ele combina o acesso de um DataSet com a possibilidade de habilitar edições e navegações entre os dados. O componente armazena todo o conteúdo em memória, permitindo salvar as alterações realizadas pela aplicação. É o típico componente utilizado com o TDBGrid.

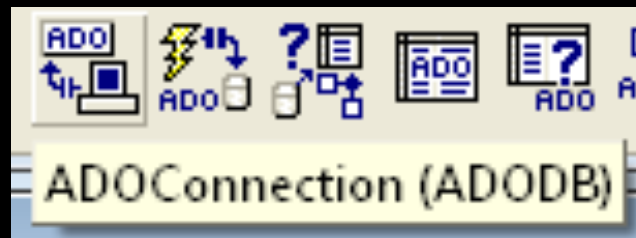


P.O.O. Cliente-Servidor

- ADO (ou *ActiveX Data Objects*), é um mecanismo de acesso criado pela Microsoft, pelo qual o aplicativo se comunica com o gerenciador de banco de dados através de um provedor em uma conexão de rede, mesmo que a aplicação e o SGBD estejam rodando localmente. As operações de inclusão, exclusão, alteração e seleção de dados são executadas a partir de comandos SQL enviados pelo aplicativo ao SGBD através do provedor
- Os componentes de acesso possuem as mesmas funcionalidades do DBExpress

P.O.O. Cliente-Servidor

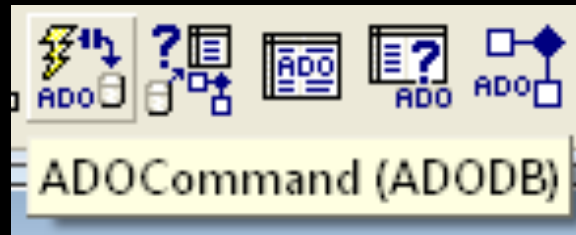
- - TADOConnection: engloba o objeto de conexão ADO, fornecendo strings de conexão, login e transações. Este componente se assemelha ao componente TdataBase, usado para a conexão BDE e ODBC.





P.O.O. Cliente-Servidor

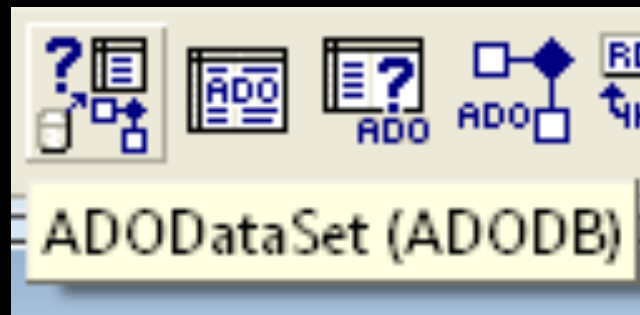
- - TADOCommand: engloba os objetos ADO Command, fornecendo um modo de executar uma consulta que não retorna um conjunto de dados.





P.O.O. Cliente-Servidor

- - TADODataSet: engloba os objetos Command e RecordSet ao mesmo tempo. Assim como acontece em qualquer outro conjunto de dados, é executado um comando sql com uma ou mais tabelas ou view e resulta em um conjunto de registros.





P.O.O. Cliente-Servidor

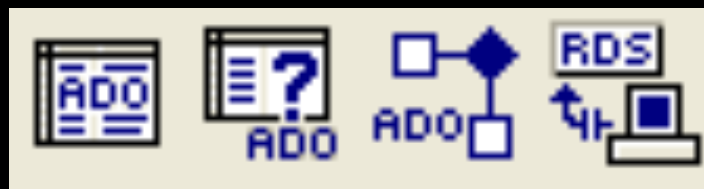
- Com esse componentes é possível criar qualquer aplicação de Banco de Dados usando ADO no Delphi.
- Contudo a forma de usá-los é bem diferente da programação tradicional a banco de dados no Delphi. Sendo assim a migração dos seus sistemas atuais, que usam os componentes de acesso a banco de dados tradicionais, para ADO seria muito complicada e, muitas vezes, até inviável, tendo em vista que suas propriedades disponíveis são diferentes.



UniSALESIANO

P.O.O. Cliente-Servidor

- Vendo tal dificuldade o Delphi traz também componentes de acesso ADO especializados, com recursos, propriedades e eventos correspondentes aos componentes tradicionais TTable, TQuery e TstoreProc, sendo eles o ADOTable, ADOQuery e ADOstoredProc respectivamente.



P.O.O. Cliente-Servidor

- IBX InterBase Express – Fornece acesso nativo entre a sua aplicação e os componentes de Acesso Nativo para acessar a base de dados. Neste meio, não existe nada “BDE/ODBC/OLE DB” e ou outro driver. Isto é, e a sua Aplicação e a base de dados, nada mais envolve na conversação dos dados. E isto é feito através de Funções da API do SGDB InterBase.

P.O.O. Cliente-Servidor

- **TIBDataBase**

É o responsável pela conexão entre a sua aplicação e o Banco de Dados InterBase.

- **TIBTransaction**

É o responsável pelo controle de transações da sua aplicação. Você pode controlar transações concorrentes, ou em threads independentes. Para tratarmos de transações, precisaríamos de um artigo a parte.



P.O.O. Cliente-Servidor

- **TIBQuery**

Faz a conexão SQL com a sua Base de Dados InterBase. Este componente aceita quase todas as instruções DDL, DML e DQL. Para utilizar o TIBQuery como um componente de edição, é necessário o uso do TIBUpdateSQL. As propriedades e eventos, são praticamente as mesmas do componente TQuery de Data Access do Delphi. Explicarei algumas que estão diretamente ligada ao InterBase.



P.O.O. Cliente-Servidor

- **TIBStoredProc**

Executa uma procedure armazenada no Servidor InterBase. Este componente é equivalente ao componente TStoreProc de Data Access do Delphi.

- **TIBUpdateSQL**

Permite definir instruções DML para cada método Insert, Edit e Delete.

TIBUpdateSQL+TIBQuery representa toda a funcionalidade SQL de manipulação de Dados e Live Result Set.

- **TIBDataSet**

Engloba toda a funcionalidade de TIBQuery+TIBUpdateSQL, e ainda é mais rápido. É o recomendado pelo Autor e pelos criadores dos componentes Interbase® Express "IBX".

- **TIBSQL**

Executa instruções SQL. Recomendável para o uso de instruções DML. Este componente não oferece condições de controles de Data Control, desta forma, um SELECT não poderá ser recuperado.

- **TIBDataBaseInfo**

Componente que retorna várias informações do seu Banco de Dados especificado. Para capturar as informações, apenas coloque o componente no formulário e sete DataBase para o objeto TIBDataBase correspondente.

- **TIBSQLMonitor**

Cria um LOG para acompanhamento de todas as instruções enviadas para o servidor. Este componente trabalha em conjunto com a propriedade TraceFlags de TIBDataBase. Para utilizar o componente, basta coloca-lo no formulário. Para capturar as informações enviadas pelo servidor, utilize um List Box e ou outro componente de Lista de String e configure o evento OnSQL.

- **TIBEvents**

Componente que captura eventos do Banco de Dados InterBase. O SGDB Interbase® pode emitir eventos e a sua aplicação capturar. Este eventos são programados dentro de Trigger e Stored Procedure.

P.O.O. Cliente-Servidor

- **TIBExtract.**
-
- Componente responsável por extrair informações das Tabelas de Sistemas no InterBase. Tem o mesmo papel do Extract MetaData do IBConsole. Existem dois tipos relacionados ao IBExtract responsáveis pela Extração das informações do Banco de Dados :
 - TExtractObjectTypes
 - TExtractType

P.O.O. Cliente-Servidor

- Foi citado mais acima, a recomendação do uso do IBDataSet para manutenção dos Dados. Mas, não posso deixar de citar a dobradinha “IBQuery+IBSQL”, pois, estes dois componentes tem uma melhor performance sobre os demais componentes. Vale ressaltar também, que se for apenas fazer pesquisa e mostrar num GRID “por exemplo” os dados, use sempre o IBQuery !.

- **Unified InterBase**

O UIB (Unified InterBase) que produzia somente componentes de acesso a dados, agora possui um driver free para a tecnologia dbExpress, podendo acessar os bancos Firebird, InterBase e Yaffil.

- Funciona em conjunto com os componentes DBExpress