



**Universidade Regional de Blumenau
Centro de Ciências Exatas e Naturais**

**Bacharelado em Ciências da Computação
Trabalho de Conclusão de Curso**

**Utilização da Tecnologia ActiveX Data
Objects (ADO) em um Sistema com Objetos
Distribuídos**

**Acadêmica: Suzete Teresinha Colling
Orientador: Maurício Capobianco Lopes**

Blumenau, julho de 2000

Roteiro:

- ◆ Introdução;
- ◆ ActiveX Data Objects;
 - ◆ Conceitos;
 - ◆ UDA (*Universal Data Access*);
 - ◆ Objetos Distribuídos e ActiveX;
- ◆ O Tutorial Inteligente para Delphi;
- ◆ Especificação do Protótipo;
- ◆ Estrutura de Camadas;
- ◆ Considerações Finais.

Introdução

- ◆ O Problema;
 - ◆ Importância dos dados e informações;
 - ◆ Independência da fonte de dados;
- ◆ Objetivos do Trabalho;
 - ◆ Aplicar ADO no Tutorial Inteligente para Delphi;
 - ◆ Demonstrar ADO;
 - ◆ Possibilitar o funcionamento utilizando Objetos Distribuídos ou WEB

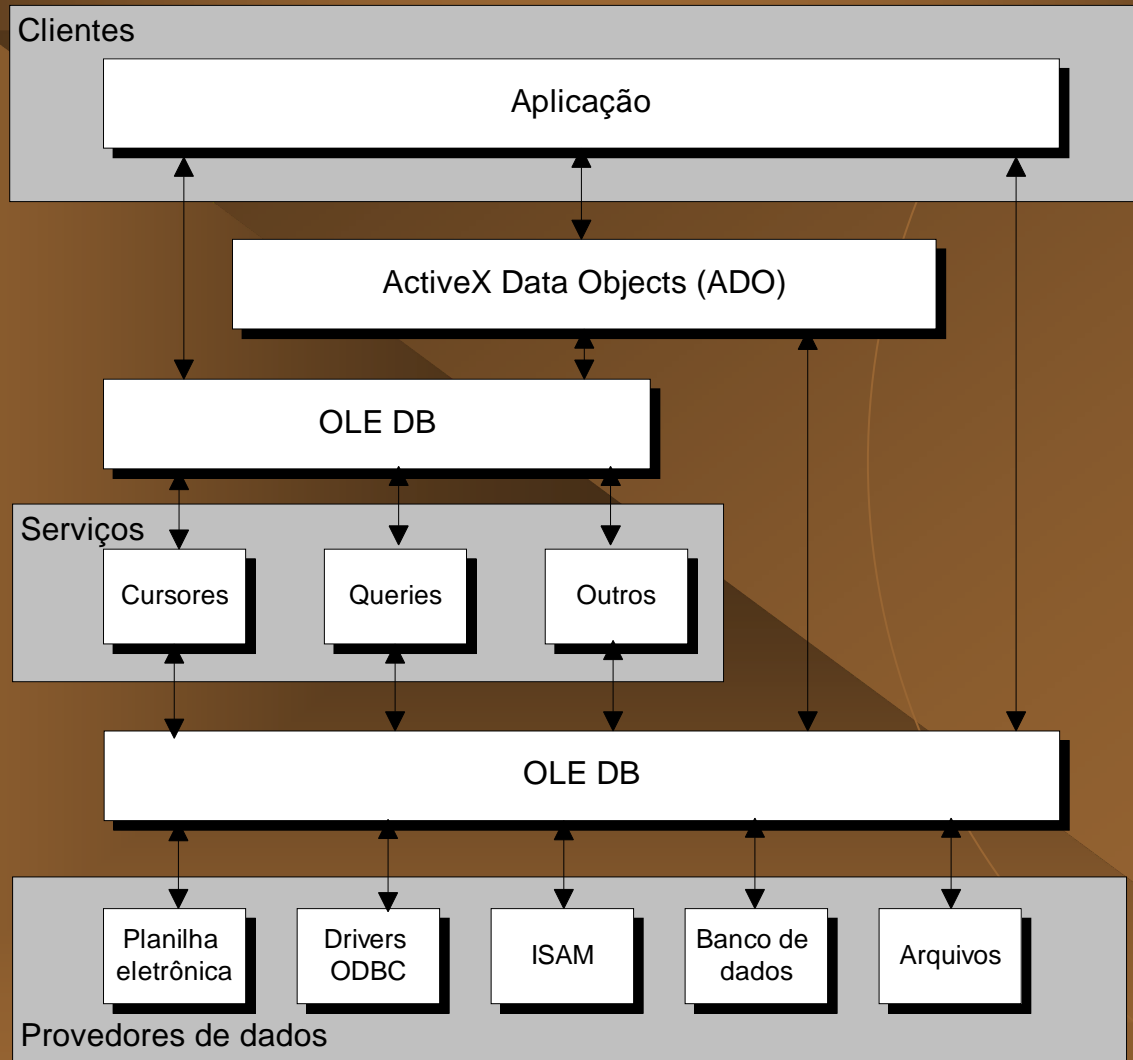
ActiveX Data Objects (ADO)

- ◆ Histórico;
 - ◆ Componentes: Partes de código reutilizáveis e independentes de aplicações e de sistemas;
- ◆ Conceito;
 - ◆ ADO: Conjunto de Componentes;
 - ◆ Uniformidade de interface;
 - ◆ Facilidade de uso;
 - ◆ Faz parte do modelo COM (Objetos que promovem interoperabilidade de software)
 - ◆ Multicamadas e WEB (ASP)

ActiveX Data Objects (ADO)

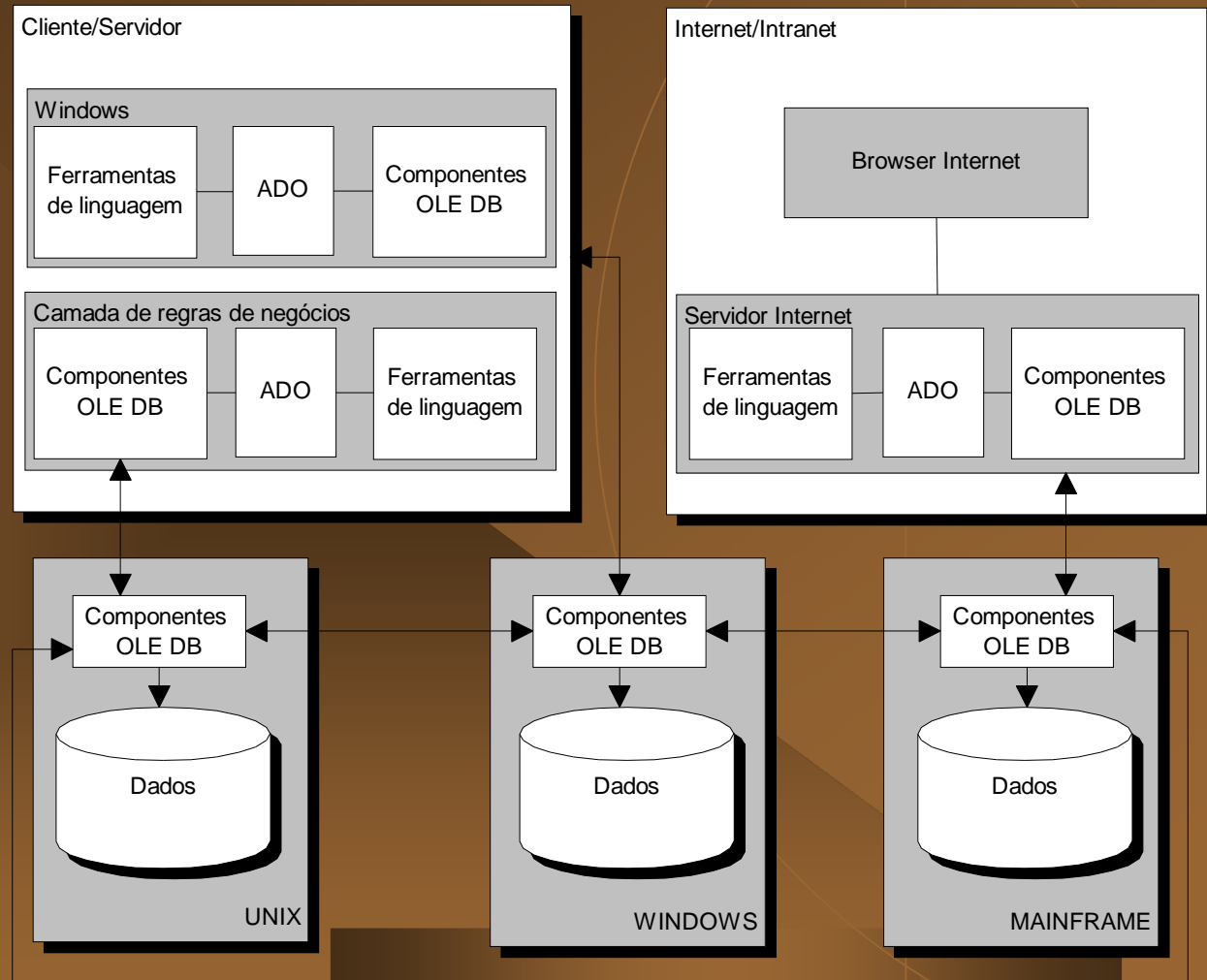
- ◆ *Universal Data Access (UDA)*: Proposta de interface única para inúmeros tipos de fontes de dados;
 - ◆ Vantagens: não requer compromisso demorado com um único banco.
- ◆ ADO e UDA;
- ◆ OLE DB (*Object Linking and Embedding - Data Base*);
 - ◆ Programação de interface COM para diversas fontes de dados.
- ◆ ADO E OLE DB:
 - ◆ Interface a nível de aplicação;
 - ◆ Interface a nível de sistema;

Arquitetura UDA com ADO



ActiveX Data Objects (ADO)

- ◆ ADO E OLE DB como integrantes de UDA Multi Plataforma
- ◆ Multiplicidade na camada de dados



ActiveX Data Objects (ADO)

- ◆ Arquitetura de camadas
 - ◆ Não é obrigatória para utilizar ADO;
 - ◆ Vantagens de usar componentes na Camada intermediária:
 - ◆ decomposição das aplicações;
 - ◆ reutilização dos componentes;
 - ◆ pode-se incorporar componentes externos.

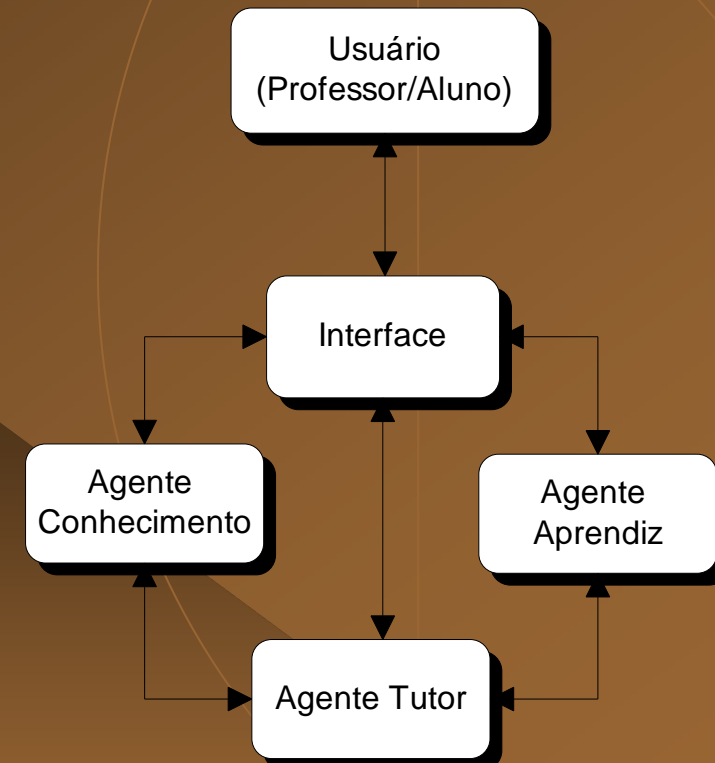
Objetos Distribuídos e ActiveX

- ◆ Aplicações Distribuídas com ADO;
 - ◆ Objetos Distribuídos:
 - ◆ Heterogeneidade e reutilização;
 - ◆ Diferença com Objeto Clássico;
 - ◆ Necessita de infra-estrutura: DCOM, CORBA.
 - ◆ ActiveX: Conjunto de tecnologias
 - ◆ Controles ActiveX: objetos COM, com interface padronizada;
 - ◆ Permite utilizar OLE através da WEB.

O Tutorial Inteligente para Delphi

- ◆ O Tutor Inteligente para Delphi;
 - ◆ Cadastro do Exercício;
 - ◆ Aprendizagem do Exercício;
 - ◆ Correção do Exercício;

Relação entre os agentes

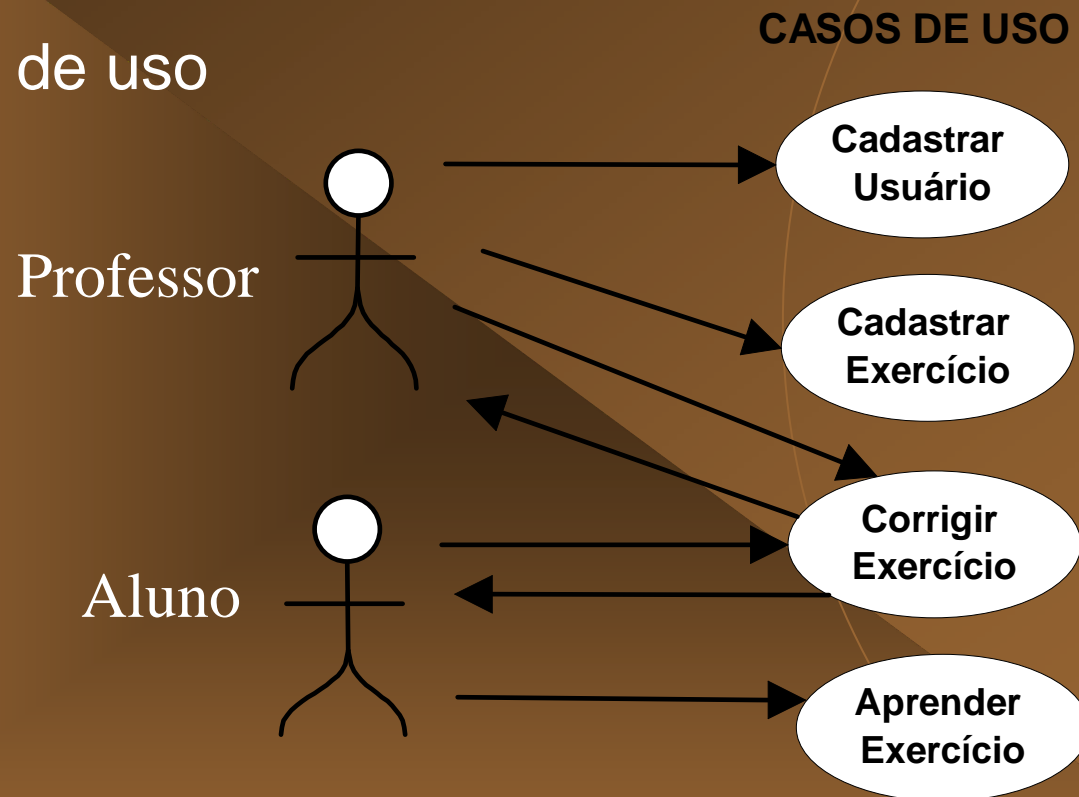


Mudanças Básicas no Tutor Delphi

Assunto	Versão Antiga	Versão Atual
Execução	<i>Standalone</i> (cada agente era executado separada e independentemente).	Cliente/Servidor (cada agente é um objeto, chamado através da interface, que está no programa cliente. O Cliente e o Servidor comunicam-se remotamente através de DCOM.
Informações	Informações armazenadas parte em tabelas Paradox e parte em arquivos .INI, mantidos na própria estação.	As informações são todas gravadas em tabelas. Optou-se por utilizar para demonstração o banco de dados Sybase SQLAnywhere.
Acesso aos dados	Através de componente TTable do Delphi.	Através de componentes ADOTable, encapsulando toda a tecnologia ADO.
Conhecimento	Todo o conhecimento adquirido ficava armazenado nos arquivos .INI na estação local.	O conhecimento fica disponível na base de dados do servidor, na camada de dados.
Segurança	Qualquer aluno pode cadastrar outros usuários.	Somente professores podem cadastrar novos usuários.

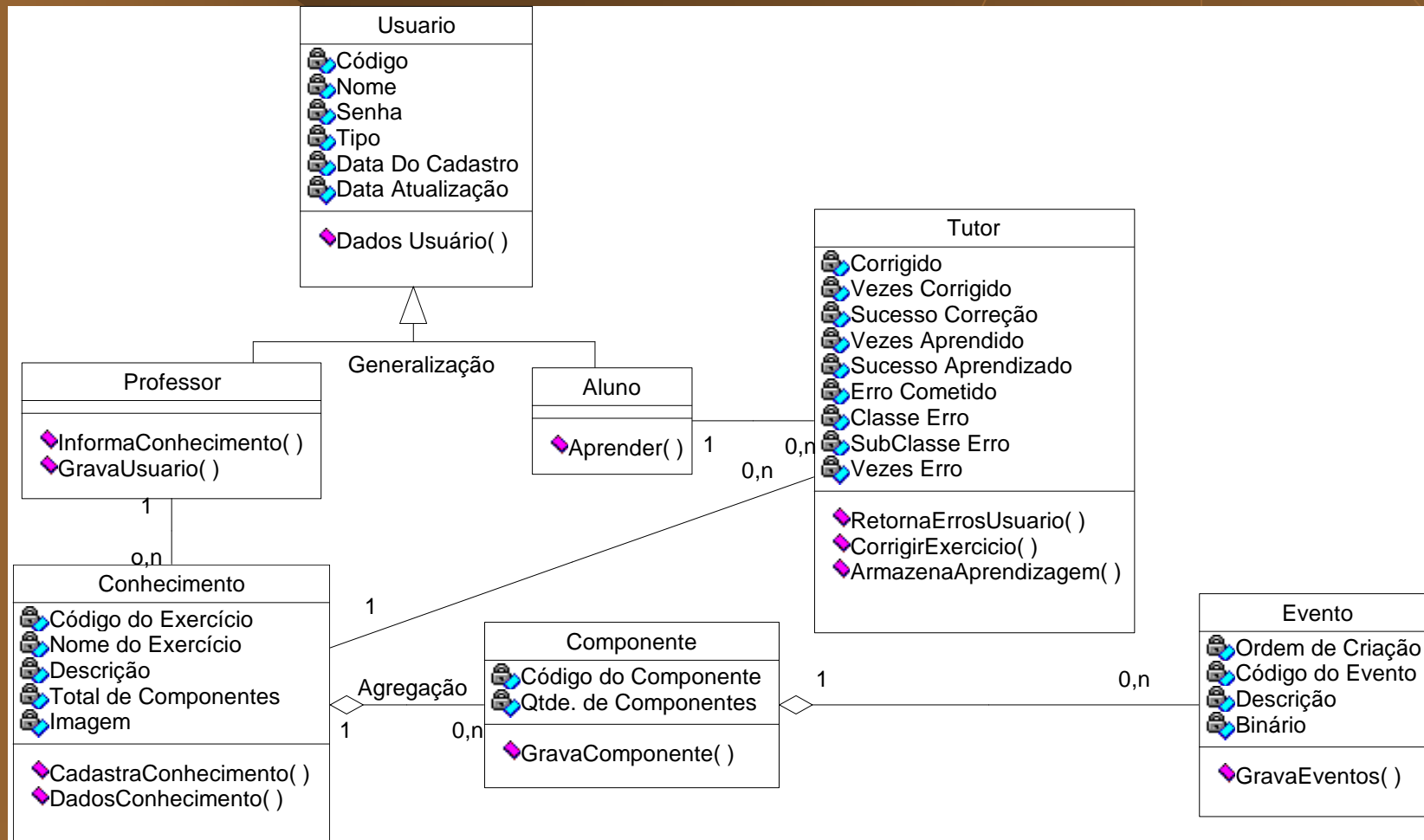
Especificação do Protótipo

◆ Casos de uso



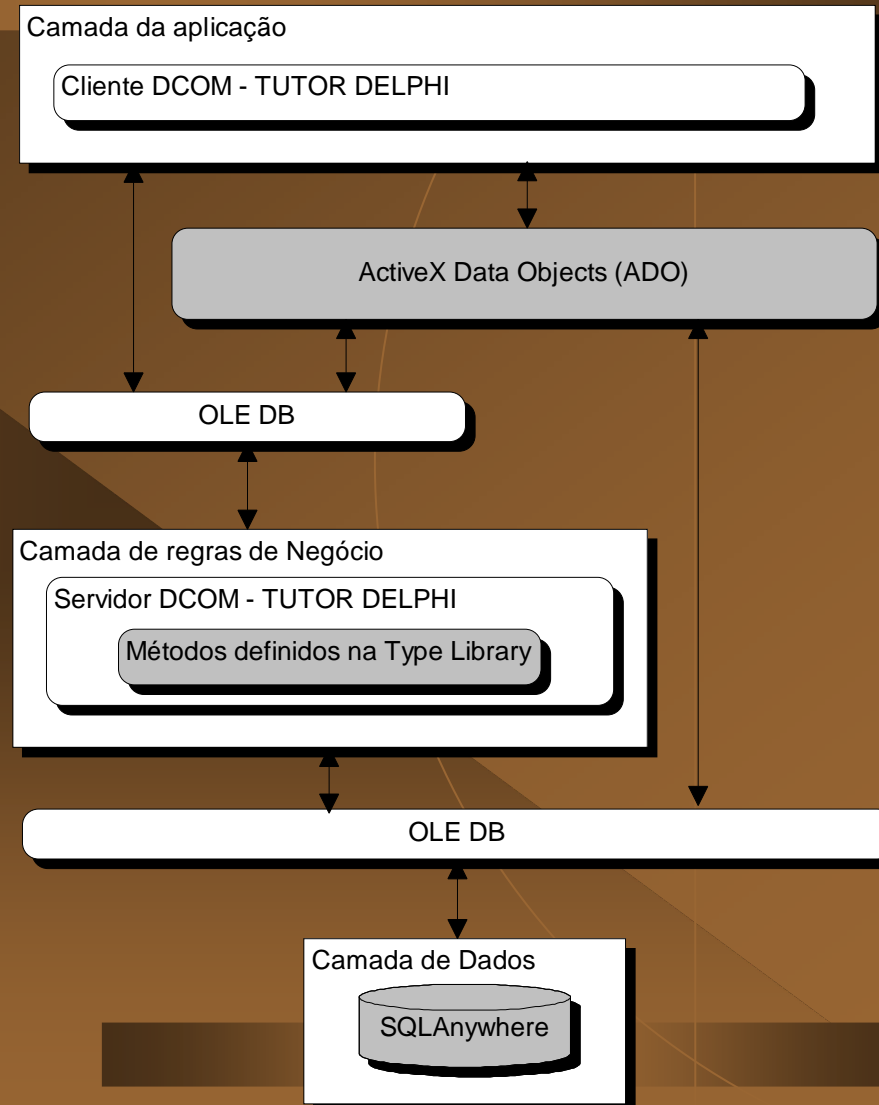
Especificação do Protótipo

◆ Diagrama de classes



Especificação do Protótipo

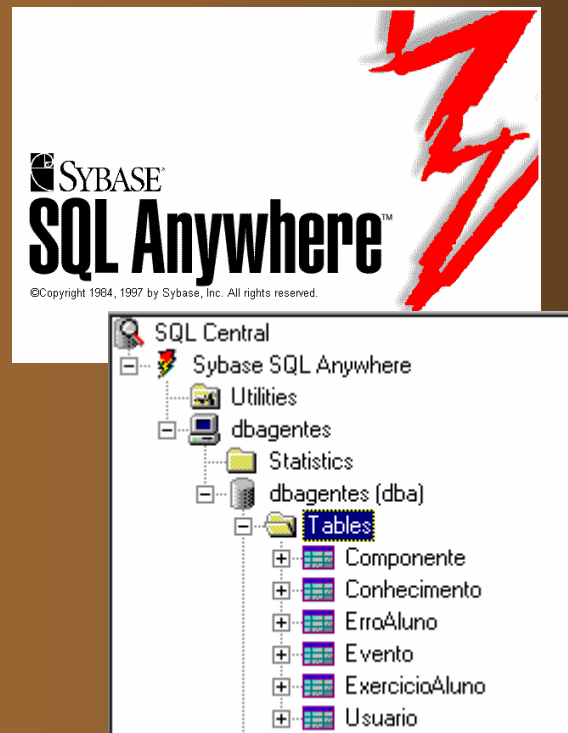
- ◆ Visão Distribuída, considerando conceitos de ADO e UDA
- ◆ Estrutura de camadas



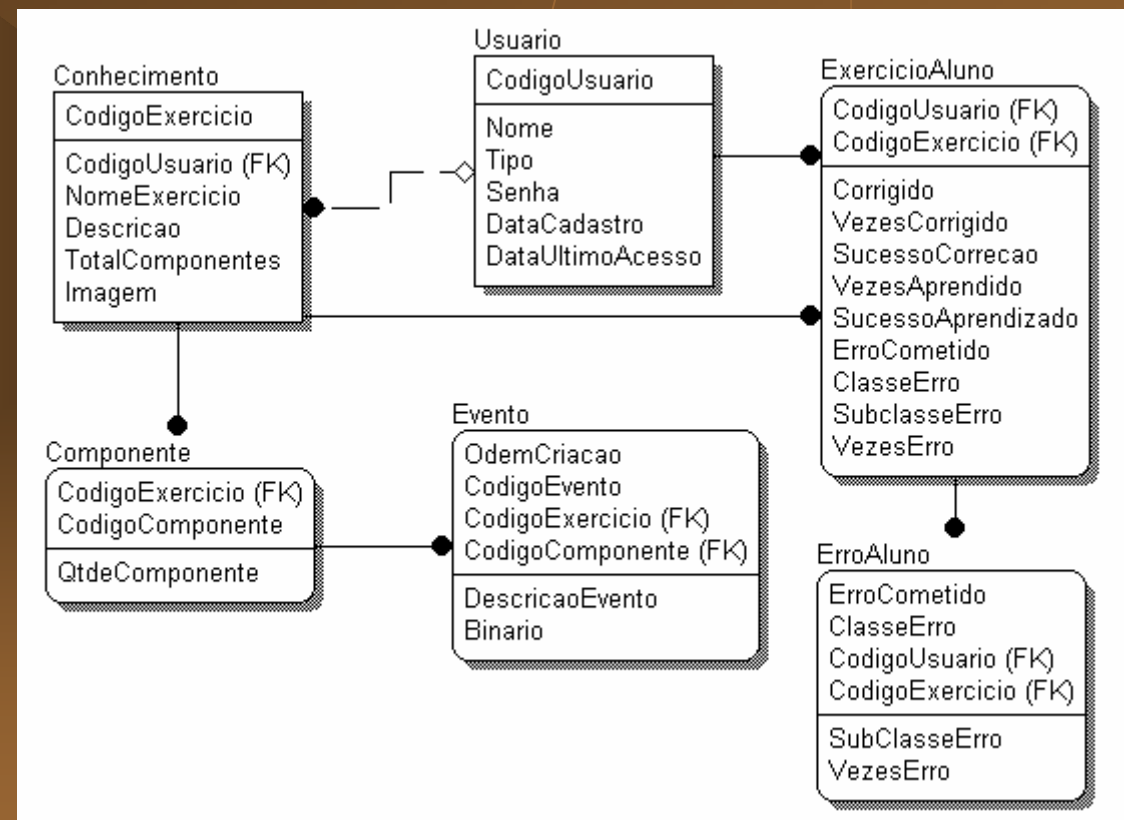
Especificação do Protótipo

- ◆ Camada de Dados: Responsável pela Integridade dos dados

Banco de Dados utilizado:

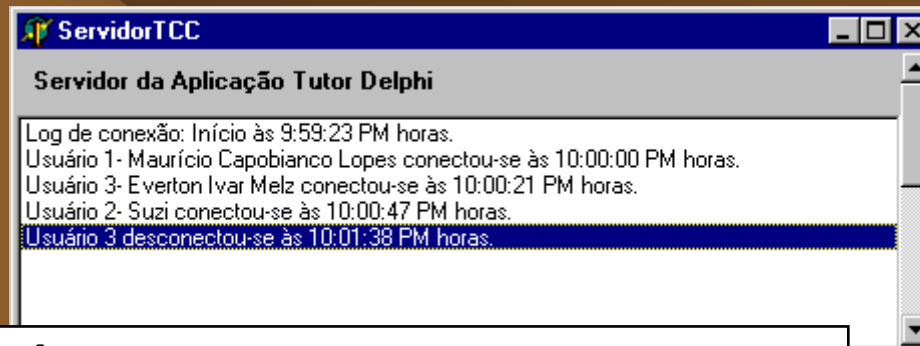


Modelo de Dados



Especificação do Protótipo

- ◆ Camada de regras de Negócios: Servidor, métodos



```
IUnknown = interface  
    ['{00000000-0000-0000-C000-000000000046}']  
  
function QueryInterface(const IID: TGUID; out Obj): HRESULT;  
stdcall;  
  
    function _AddRef: Integer; stdcall;  
    function _Release: Integer; stdcall;  
  
end;
```

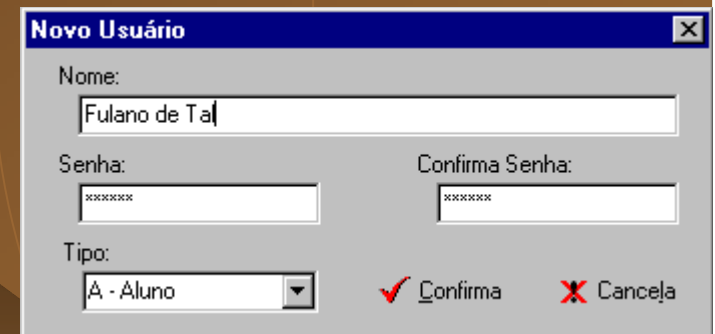
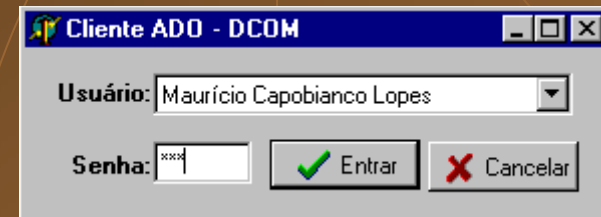
```
type  
    IAppServerTcc = interface;  
AppServerTcc = IAppServerTcc;  
end;  
uses ComObj;  
class function CoAppServerTcc.Create: IAppServerTcc;  
begin  
    Result := CreateComObject(CLASS_AppServerTcc) as  
IAppServerTcc;  
end;  
class function CoAppServerTcc.CreateRemote(const MachineName:  
string): IAppServerTcc;  
begin  
    Result := CreateRemoteComObject(MachineName,  
CLASS_AppServerTcc) as IAppServerTcc;  
end;
```

- ◆ Responsável por políticas de negócio, regras e segurança

Especificação do Protótipo

- ◆ Camada da Aplicação: Interface com usuário

- ◆ Responsável pela apresentação e navegação



Considerações Finais

- ◆ Conclusões:
 - ◆ Orientação a Objetos sempre mais forte, ainda mais se aliada a Internet, como o Comércio Eletrônico;
 - ◆ Objetos distribuídos: Cresce em virtude das necessidades do mercado;
 - ◆ As vantagens do conceito de UDA;
 - ◆ Unindo ADO a isso tudo, tem-se um conjunto promissor, seja nível comercial ou à aprendizagem;
 - ◆ Objetivos do trabalho alcançados e ADO corresponde as exigências de mercado.

Considerações Finais

- ◆ Dificuldades Encontradas:

- ◆ Escassez de material;
- ◆ Tecnologia “proprietária”;

Considerações Finais

- ◆ Sugestões:
 - ◆ Implementações para o funcionamento do Tutor Delphi pela WEB e aprofundar estudos de ActiveX e ASP;
 - ◆ Técnicas de segurança, criptografia, ataque de hackers, controle multiusuário;
 - ◆ Testes com relação à independência de plataforma

Apresentação do Protótipo

Muito Obrigada

Suzi - 10/07/2000