

2. Tecnologias de acesso a BD

Tecnologia DAO (Data Access Objects) apareceu com VB 3 (otimizado para acesso local)

Tecnologia RDO (Remote Data Objects)

Tecnologia ADO (ActiveX Data Objects)

Cliente/ Servidor
Provider/ Consumer

Tecnologia **ADO.NET**  ambiente *n-camadas*

2.1. ADO

☞ é um *consumer* do *provider* OLE DB.

OLE DB – tecnologia universal de acesso a dados: Excel, Access, dBase, Active Directory, DB2, SQL Server, Oracle,

☞ camada intermédia entre o OLE DB e o VB, ASP,

☞ implementa as facilidades disponibilizadas pelos *providers*, Base de dados e outros

☞ permite o uso de cursores cliente ou servidor

☞ permite a execução de queries parametrizados

☞ suporta transacções

☞ mantém-se permanentemente ligado ao *provider*

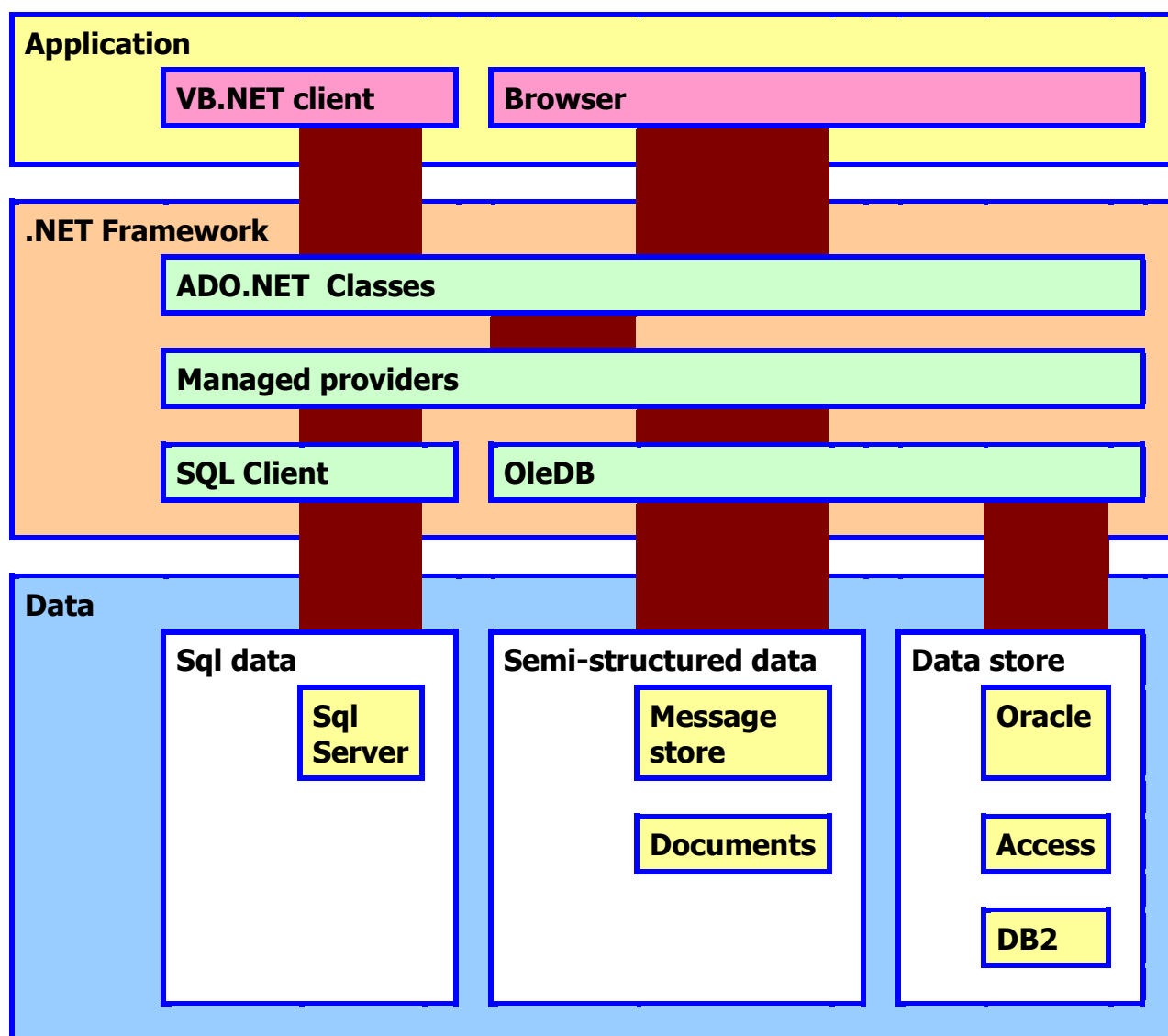
2.2. ADO.NET

☞ deriva do ADO mas está desenvolvido sobre a plataforma .NET com código controlado

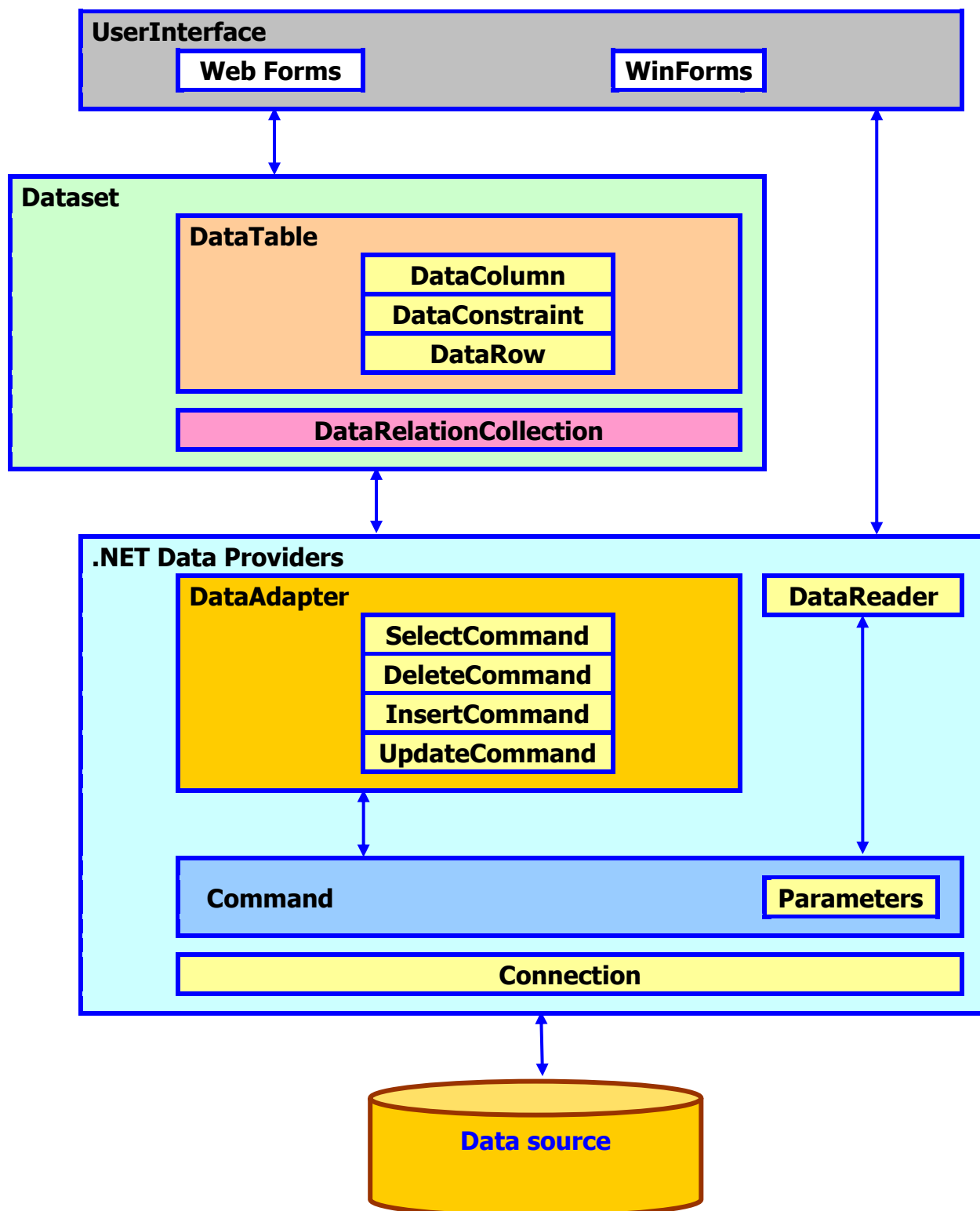
☞ suporta arquitectura de dados desconectados própria das aplicações multicamada, n-tier, para a Web

☞ baseado no **Dataset** que consiste num armazenamento local de registos obtidos da base de dados; espécie de **mini base de dados**, contém tabelas, colunas, restrições, registos e relações. A conexão à origem dos dados é aberta para obter os dados (em formato XML) e terminada; será de novo aberta para enviar as alterações.

2.3. Como as aplicações cliente comunicam com as base de dados



2.4. Visão global da arquitectura ADO.NET



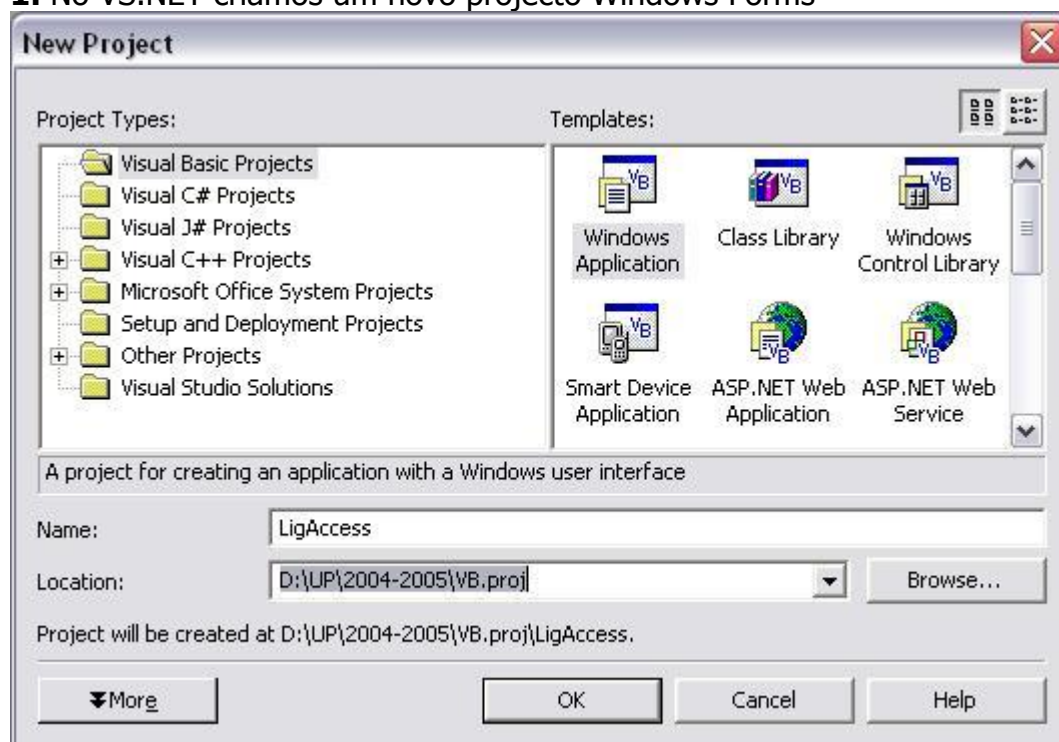
2.5. Como utilizar o ADO.NET num projecto VB.NET – versão 2003

No .NET a manutenção de uma ligação a uma base de dados e o trabalhar com a sua informação é tratada por código automático.

Aceder à informação é tão simples no .NET que uma única linha de código é suficiente.

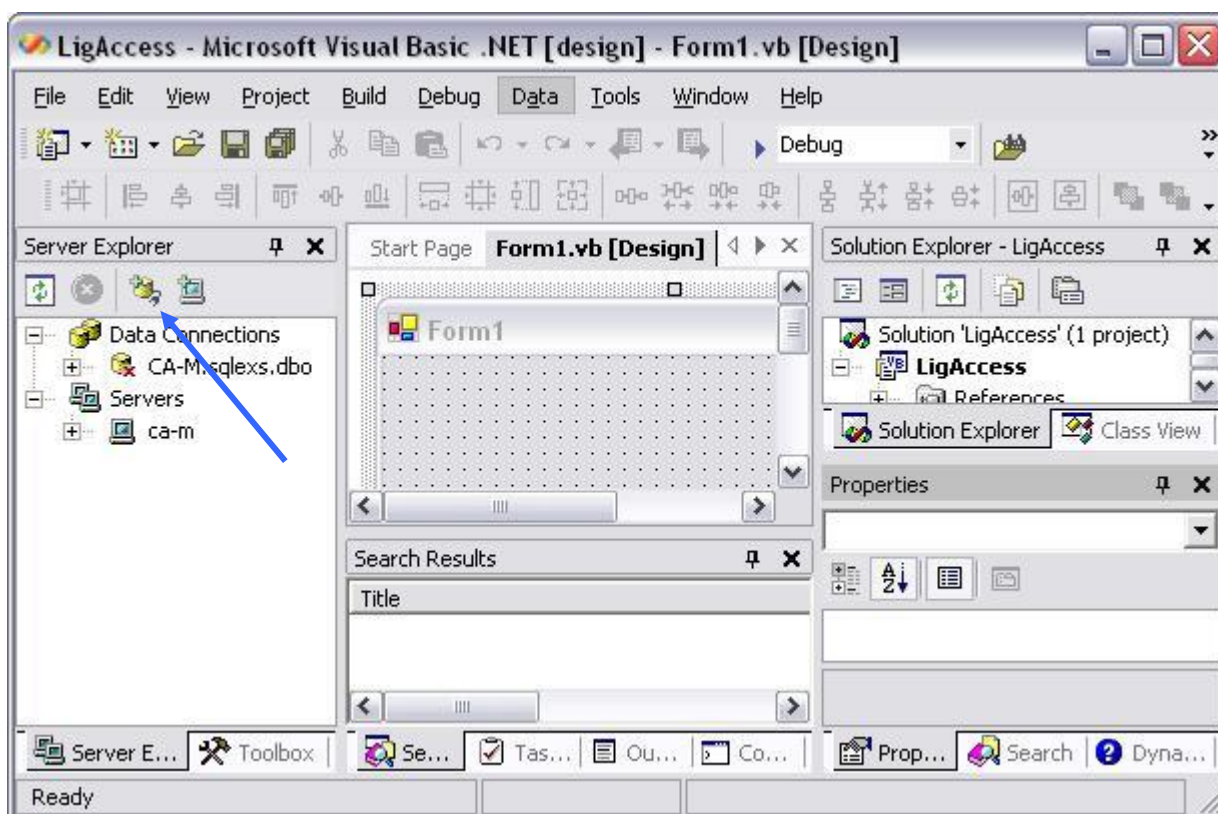
Efectuemos os seguintes passos:

1. No VS.NET criamos um novo projecto Windows Forms



e aparece a Form1.

2. Activamos a janela Server Explorer para podermos encontrar o nosso provider; neste caso a base de dados em Access com o nome INSCRIÇÕES.mdb.
Com um clique no botão assinalado na figura seguinte [Connect to Database] vai estabelecer-se uma nova ligação à base de dados.



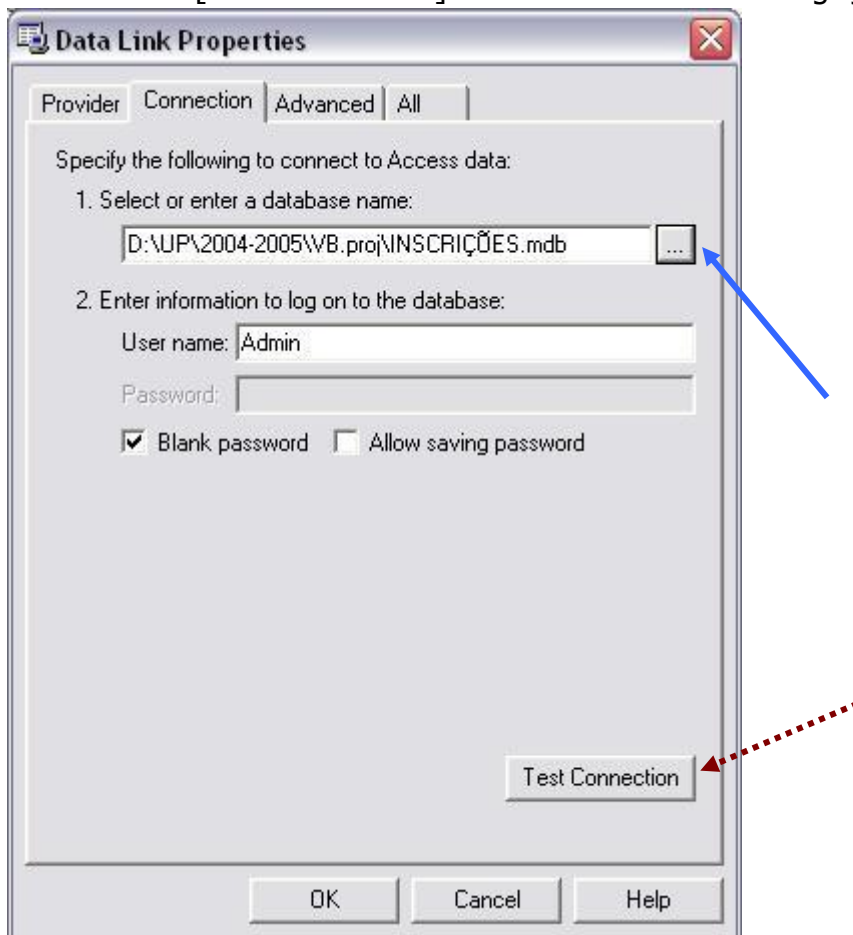
Aparece uma janela onde se deverá escolher o tipo de provider, no caso iremos escolher o Microsoft Jet 4.0 OLE DB Provider.



Aparecem outras possibilidades de ligações, mas que dependerão do que estiver instalado no computador. Poderemos encontrar, por exemplo:

- Microsoft OLE DB Provider for Sql Server,
- Microsoft OLE DB Provider for Oracle e
- OLE DB Provider for Microsoft Directory Services

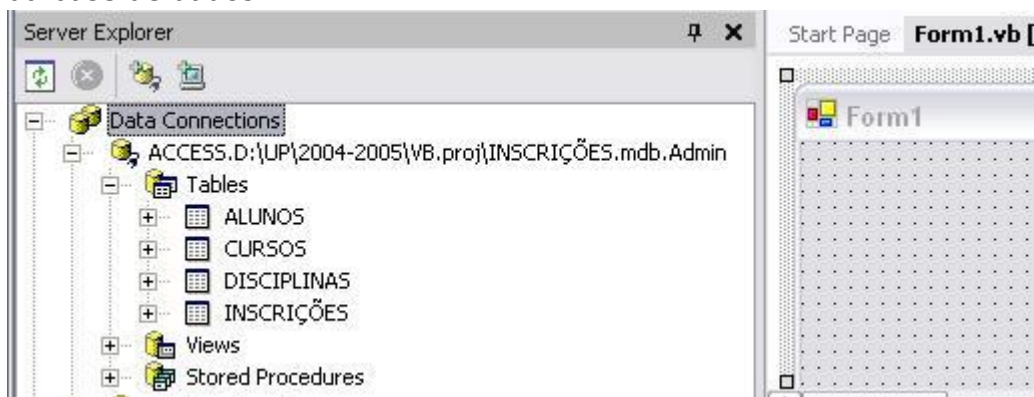
No separador seguinte teremos de especificar qual a base de dados. Podemos usar o botão assinalado para usar a janela habitual do Windows de localização de ficheiros, e devemos usar o botão [Test Connection] para verificarmos a boa ligação à base de dados:



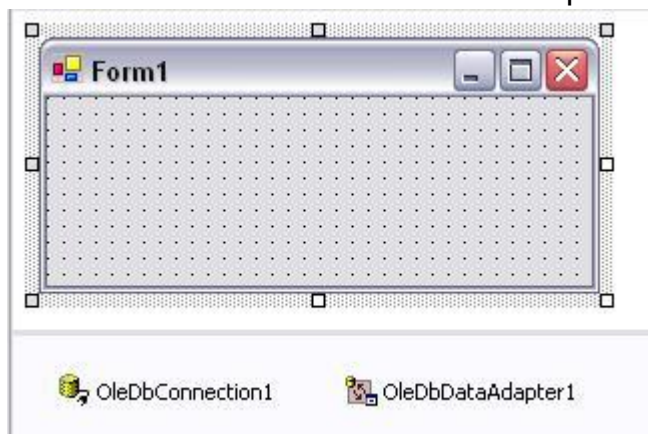
Pode agora observar-se na janela do Server Explorer a existência da nova ligação:



Na janela do Server Explorer, terá de se expandir a nova ligação e encontramos as tabelas da base de dados:



3. Por arrasto levamos a tabela Cursos para a Form1. Aparecem dois novos objectos:



OleDbConnection1 – cria a conexão à base de dados

OleDbDataAdapter1 – objecto responsável por obter os dados

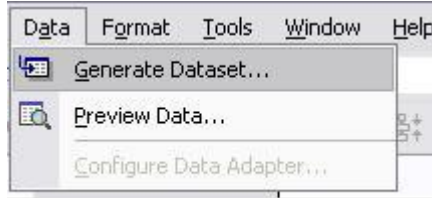
Podemos observar na janela das propriedades a ConnectionString da ligação que contém a seguinte informação:

```
Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Registry Path=;Jet OLEDB:Database Locking Mode=1;Jet OLEDB:Database Password=;Data Source="D:\UP\2004-2006\VB.proj\INSCRIÇÕES.mdb";Password=;Jet OLEDB:Engine Type=5;Jet OLEDB:Global Bulk Transactions=1;Provider="Microsoft.Jet.OLEDB.4.0";Jet OLEDB:System database=;Jet OLEDB:SFP=False;Extended Properties=;Mode=Share Deny None;Jet OLEDB:New Database Password=;Jet OLEDB:Create System Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;User ID=Admin;Jet OLEDB:Encrypt Database=False
```

Esta string é complexa mas poderá ser construída programaticamente.

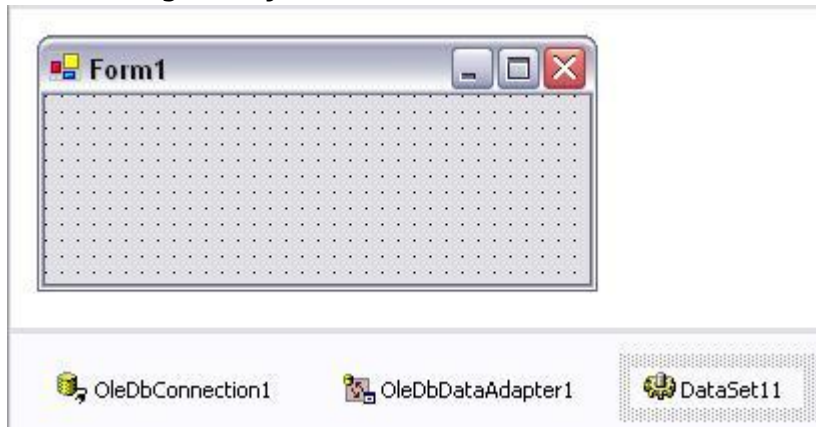
4. É necessário um outro objecto, um DataSet que irá armazenar os dados obtidos pelo DataAdapter.

No Menu Data seleccionamos: Generate Dataset



aparece uma outra janela onde devemos aceitar os valores apresentados e fazer [OK].

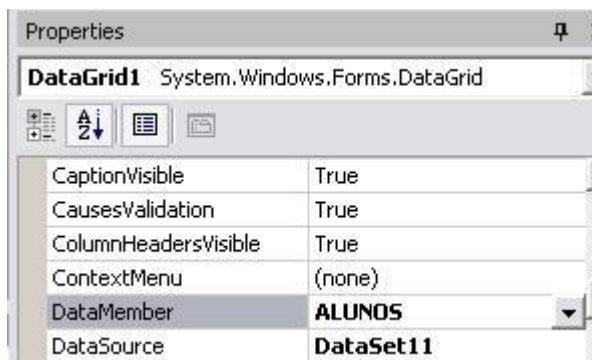
No IDE surge o objecto DataSet11:



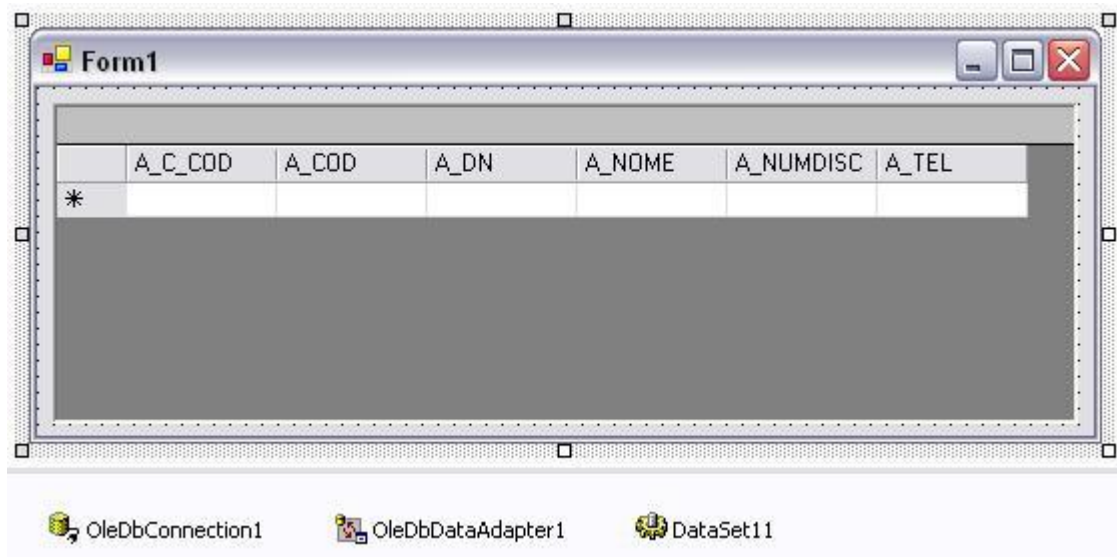
5. Para vermos a informação no quadro é necessário criar um controlo no Form1 e ligá-lo ao Dataset.

No IDE abrimos a Toolbox, seleccionamos DataGrid e arrastamos para cima de Form1, criando assim uma instância.

6. Com a DataGrid seleccionada, nas janelas das propriedades atribuímos a **DataSource** o valor DataSet11 e a **DataMember** o valor ALUNOS.



7. O aspecto da DataGridView altera-se e passa a mostrar os campos da tabela ALUNOS



8. Agora é necessário criar uma linha de código para obter os dados da base de dados e popular o Dataset. Para isso, com um clique duplo em Form1, abre-se o quadro do código no event: `Form1_Load` e escreve-se o seguinte:

```
Me.OleDbDataAdapter1.Fill(Me.DataSet1)
```

9. Com [F5] ou no menu Debug/ Start pode iniciar-se a aplicação e aparecerá a Form1:

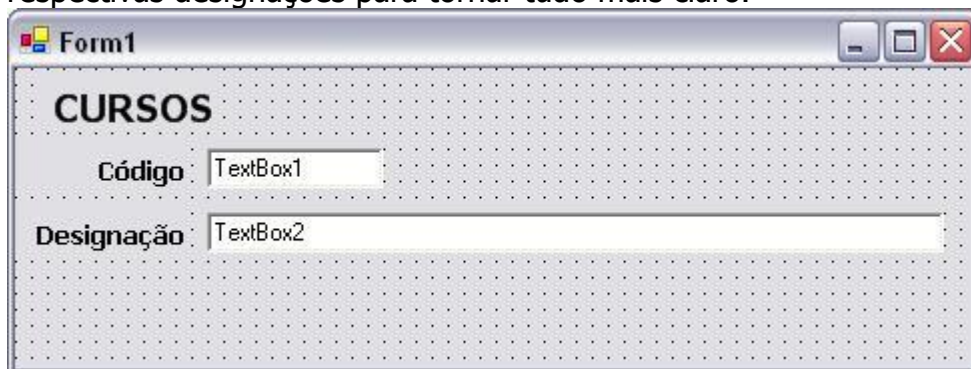


De notar que embora a DataGridView permita alterações aos valores eles não serão aplicados à base de dados.

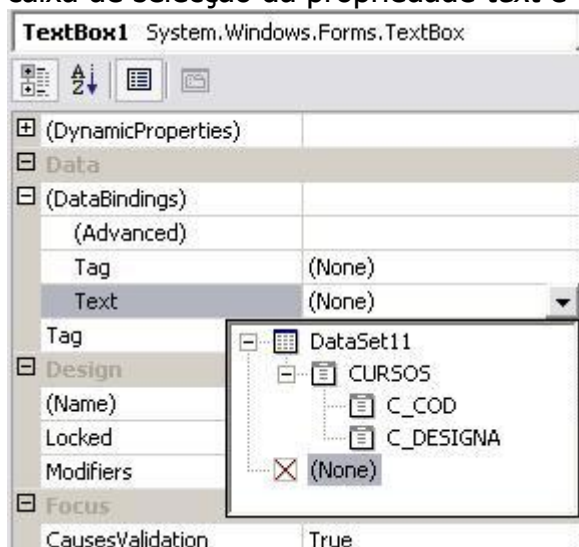
2.6. Criação de uma aplicação para visualizar registos num quadro

A primeira tarefa será criar um novo projecto e repetir todos os passos do 1 ao 4, mas usando a tabela Cursos, e o passo 8 para carregamento do Dataset. E em seguida:

1. No Form1 vamos criar 2 caixas de texto para receberem os dados da tabela com as respectivas designações para tornar tudo mais claro.



2. O passo seguinte será relacionar os campos do quadro com os campos da tabela e para isso e para cada campo deverá seleccionar-se nas propriedades: **Data Binding**, abrir a caixa de selecção da propriedade *text* e escolher o campo respectivo.



Pode correr-se a aplicação e verifica-se que aparecem os dados relativos ao primeiro departamento.

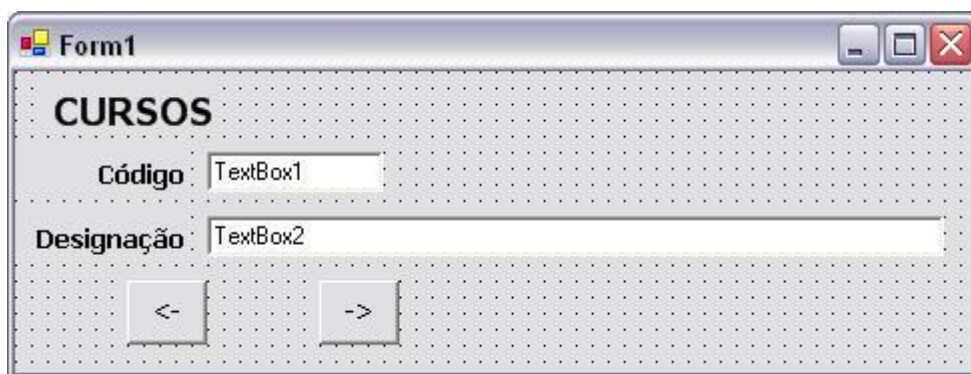


3. Para que se possa passar para o registo seguinte ou anterior vamos adicionar dois botões ao quadro: btSEGUINTE e btANTERIOR. Fazemos um clique duplo no btSEGUINTE e acedemos ao código do evento click onde escrevemos:

```
Me.BindingContext(Me.DataSet11, "CURSOS").Position += 1
```

e no botão btANTERIOR escrevemos:

```
Me.BindingContext(Me.DataSet11, "CURSOS").Position -= 1
```



4. Para criarmos uma solução mais completa vamos ainda adicionar um botão btPRI para saltar para o primeiro registo e um outro botão btULT para saltar para o último registo. Em vez de +=1 e -=1 usamos, para ir directamente para o primeiro:

```
Me.BindingContext(Me.DataSet11, "CURSOS").Position = 0
```

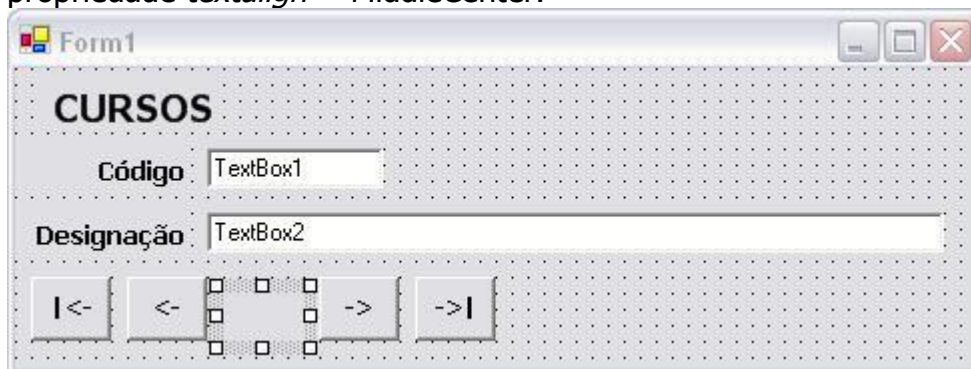
e para o último:

```
Me.BindingContext(Me.DataSet11, "CURSOS").Position =  
Me.BindingContext(Me.DataSet11, "CURSOS").Count - 1
```

O objecto BindingContext permite a navegação pelos registos do Form.

O objecto BindingContext é membro no namespace *System.Windows.Forms* e tem uma colecção de métodos e propriedades muito úteis. Pode por exemplo ser usado para determinar a posição do registo corrente na colecção total de registos.

5. Vamos criar um novo texto no quadro: lblPOS, com a propriedade *text* limpa, a propriedade *textalign* = MiddleCenter.



6. Vamos criar uma nova subrotina

```
Private Sub MostraPos()  
    With Me.BindingContext(Me.DataSet11, "CURSOS")  
        Me.lblPOS.Text = .Position + 1 & "/" & .Count  
    End With  
End Sub
```

e fazer-lhe chamadas em todas as rotinas que alteram a posição do registo corrente: Form1_Load e de todos os botões de navegação.

O '+1' aparece pelo facto de todas as colecções no .NET começarem em Zero, mas fazer mais sentido o primeiro registo ser designado por 1.

Ao correr a aplicação é sempre mostrada a posição do registo corrente e o total de registos da tabela.



A ligação dos campos do quadro aos campos da tabela pode ser feita por código durante a execução através de:

```
Me.TextBox1.DataBindings.Add("text", DataSet11, "CURSOS.C_COD")
```

Logicamente que ao ser usado o método Add, a propriedade não pode estar definida, mas se por acaso estiver, quer tenha sido durante a fase de desenho do quadro, quer por outro código executado previamente, poderá ser limpa com o método Clear:

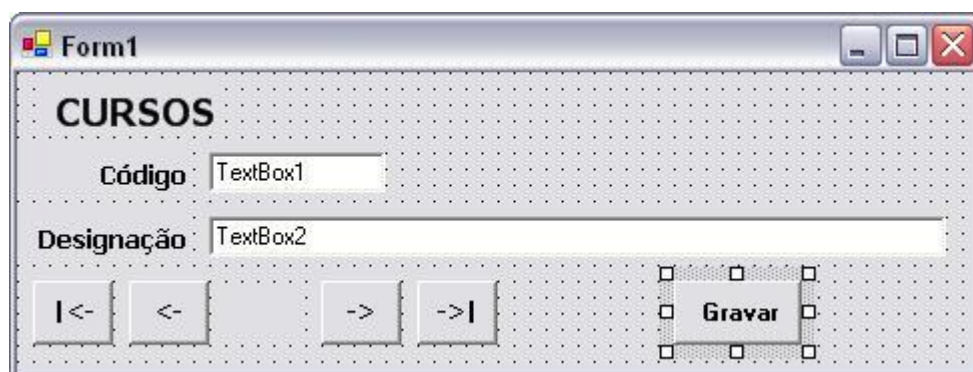
```
Me.TextBox1.DataBindings.Clear()
```

2.7. Como actualizar os dados

Os dados estão originalmente na base de dados. Na aplicação são copiados para o Dataset em memória. O quadro com os campos ligados ao dataset mostra essa informação e qualquer alteração que seja feita é passada para o Dataset, mas só são tornadas efectivas na base de dados se houver uma ordem explícita com o método Update.

Esta característica é importante e é ela que permite que o Dataset esteja desligado da origem dos dados enquanto o utilizador faz as consultas e alterações.

São várias as formas de conseguir gravar as alterações. Vamos fazê-lo de forma simplificada com um botão próprio: btGRAVAR.



E no código do clique escrevemos:

```
Me.BindingContext(DataSet11, "CURSOS").EndCurrentEdit()  
Me.OleDbDataAdapter1.Update(Me.DataSet11)
```

Ao correr a aplicação podemos alterar qualquer registo e ao pressionarmos o botão gravar tornamo-las efectivas na origem. Note-se a chamada ao método `EndCurrentEdit()` sem o qual as alterações não são passadas dos campos ligados ao Dataset. A chamada a `Update` faz a passagem do Dataset para a origem de dados.

Numa aplicação mais completa deverá ser possível cancelar as alterações e para isso deverá usar-se o método `CancelCurrentEdit()`.

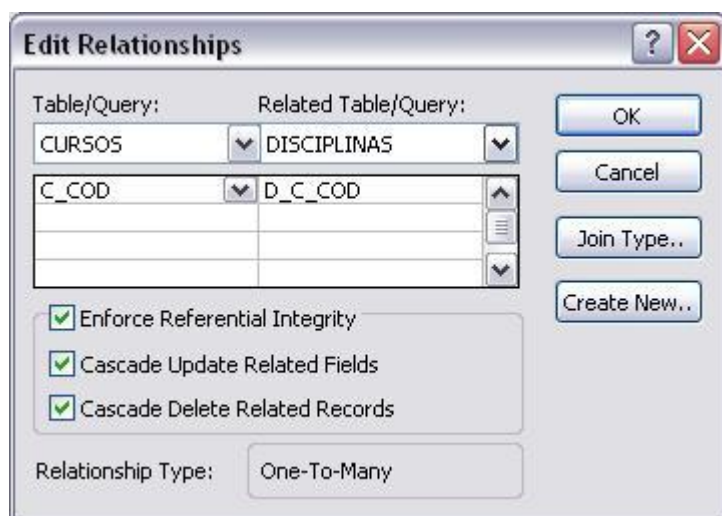
Curiosamente este código não vai funcionar correctamente se a base de dados Access estiver conforme a criamos.

Há no entanto duas formas de ultrapassar o problema.

Se as ligações entre as tabelas não tiverem a verificação de integridade referencial activa o código passa a funcionar, mas obviamente com riscos elevados para os dados.

A outra forma, mais efectiva e portanto a implementar será a de activar a integridade referencial e também as opções de "Cascade Update Related Fields" e "Cascade Delete Related Records".

Temos portanto de voltar ao quadro de relações do Access e para cada uma modificá-la nesse sentido. Com um clique com o botão direito do rato, sobre cada uma das relações existentes, escolhe-se a opção "Edit Relationship" e activam-se as duas opções adicionais de integridade:



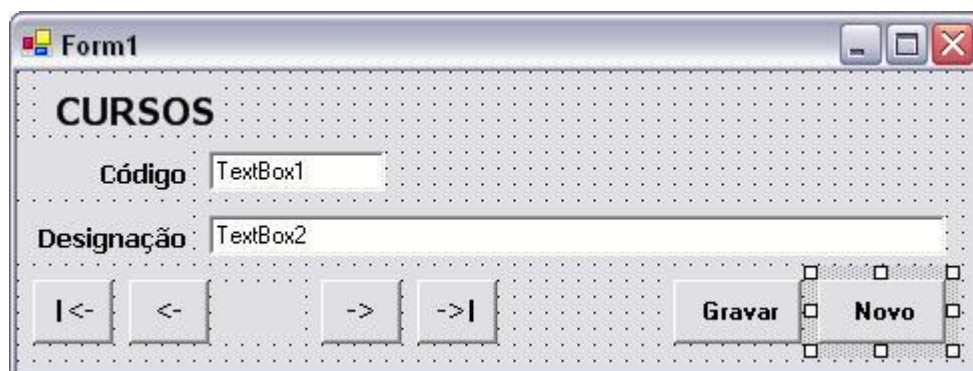
O resultado de "Cascade Update Related Fields" é o de forçar a alteração de todas as chaves estrangeiras (nas tabelas relacionadas) se se alterar a chave primária.

O resultado de "Cascade Delete Related Records" é o de suprimir conjuntamente com o registo da chave primária, todos os registos relacionados onde a chave ocorra.

Estas duas situações são desejáveis e devem ser utilizadas mas como provocam alterações para além da tabela em que estamos a trabalhar, deve ser objecto de alguns cuidados e eventuais avisos aos utilizadores.

2.8. Como criar novos registos através de um quadro

Para criar um registo usa-se o método **Addnew()** do objecto **BindingContext**, que ao ser chamado limpa todos os campos ligados, permitindo a entrada de novos valores, depois com o método **update** do **DataAdapter** faz-se a gravação. Podemos usar o botão Gravar para o efeito.



Criamos o botão **btNOVO** e associamos-lhe o seguinte código:

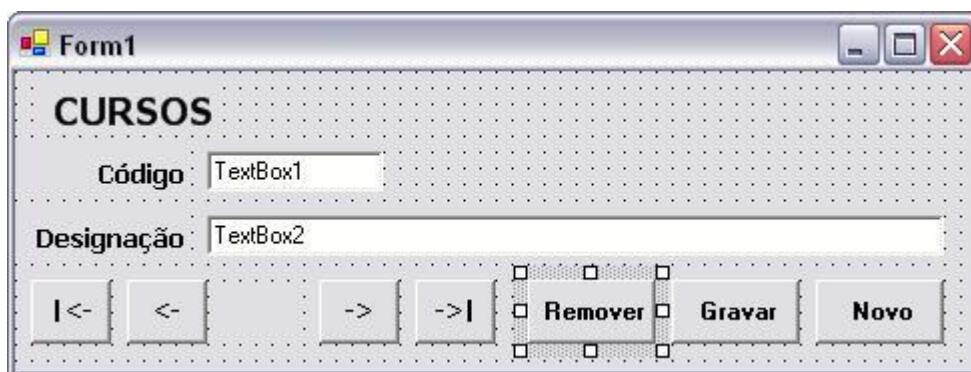
```
Me.BindingContext(Me.DataSet11, "CURSOS").AddNew()
```



```
Me.TextBox1.Focus()
MostraPos()
```

2.9. Como remover registos através de um quadro

Para remover um registo iremos usar o método RemoveAt do objecto BindingContext que recebe como parâmetro o número do registo a eliminar que normalmente será o registo corrente.

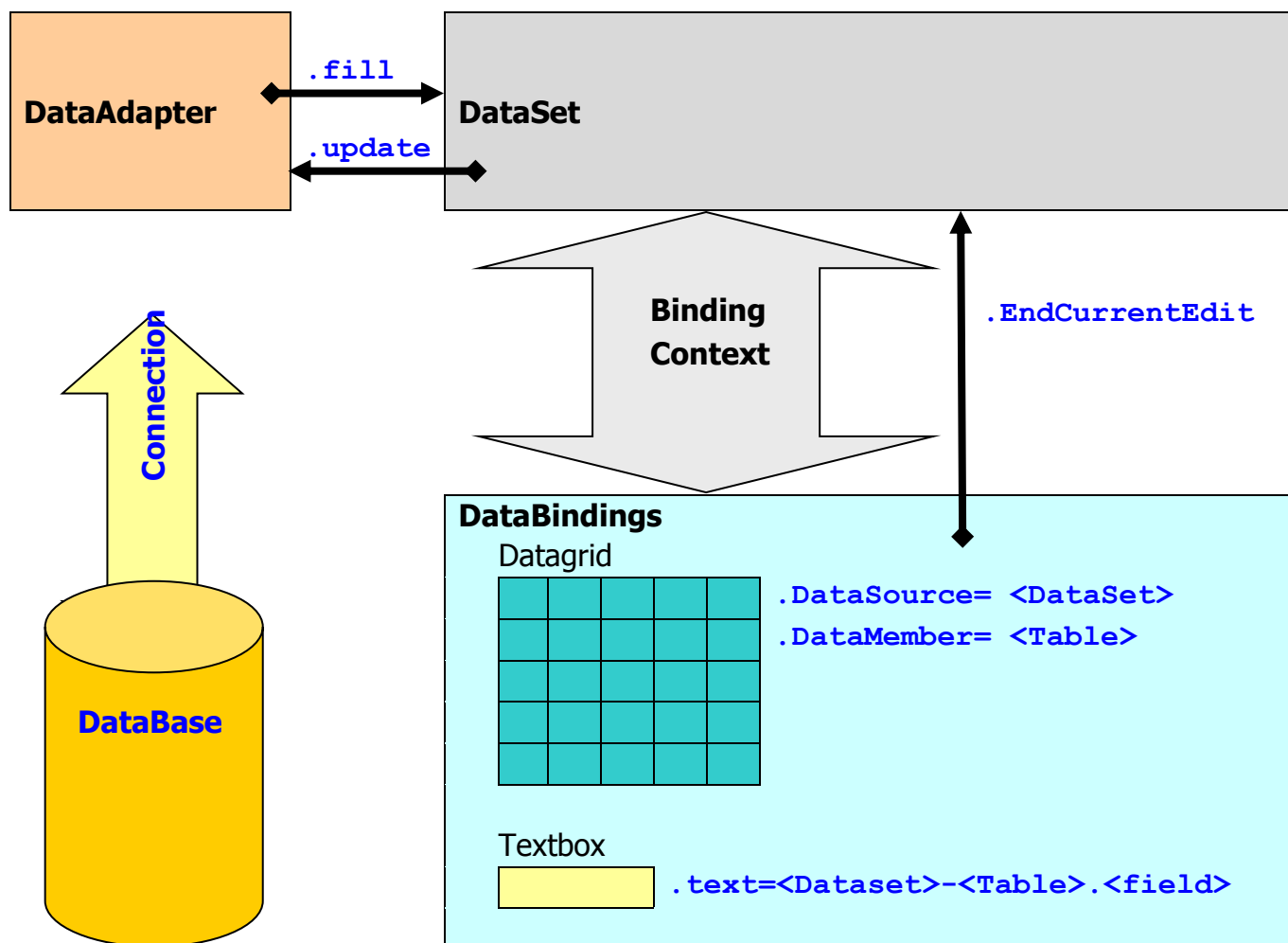


Criamos então o botão btREMOVED e associamos-lhe o código seguinte:

```
Try
    If MsgBox("Confirma REMOVER registo?", MsgBoxStyle.OKCancel) Then
        With Me.BindingContext(DataSet11, "CURSOS")
            .RemoveAt(.Position)
        End With
        Me.OleDbDataAdapter1.Update(Me.DataSet11)
        MostraPos()
    End If
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

O uso de Try é para prevenir o facto de se tentar apagar algum registo de CURSOS que esteja a ser referenciado em ALUNOS ou DISCIPLINAS.

Interacções entre os diversos componentes ADO.NET



3. Como utilizar o ADO.NET num projecto VB.NET – versão 2005

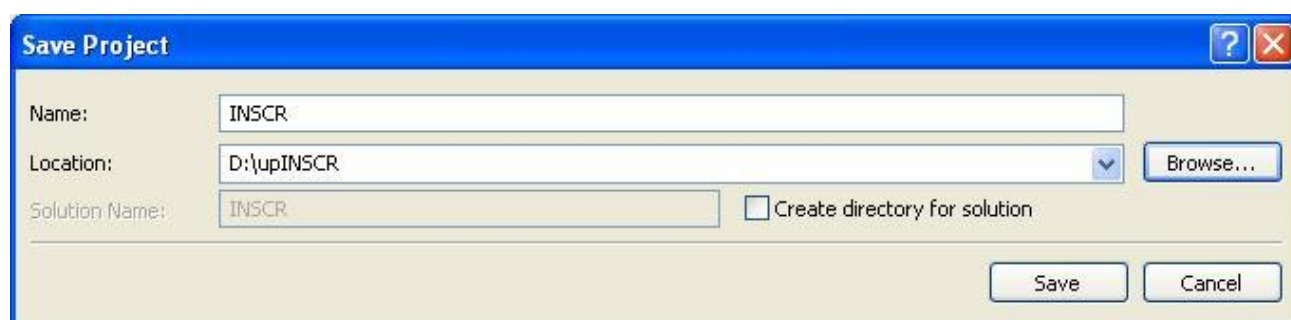
Vejamos um exemplo de utilização de uma base de dados com o VB.NET 2005. Vamos notar algumas alterações nos componentes, mas verificar que as ideias base são as mesmas. Aparecem novos componentes e outros redesenhados e temos a vida mais facilitada.

Vamos então criar um novo projecto sem grandes preocupações com questões visuais que poderão ser ajustadas mais tarde. Como para quase todas as situações, são muitas as alternativas para obter o mesmo fim. O caminho seguido parece-nos ser o mais curto e de grande economia em termos de programação. Recomenda-se, no entanto, que se observe todo o código gerado automaticamente e o código escondido. Recorde-se que tudo o que é feito com o IDE serve apenas para gerar código que pode também ser escrito à mão. O bom domínio dessa escrita fará alguma diferença no caso de se querer sofisticar um pouco mais os programas.

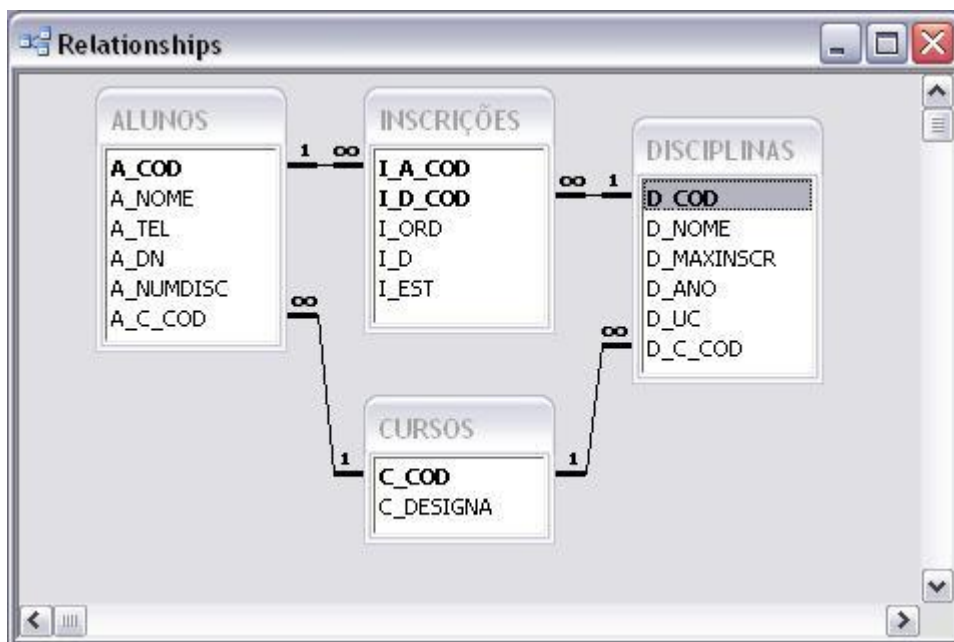
Dado que o nosso negócio é o das inscrições em disciplinas teremos um quadro para as **INSCRIÇÕES** a partir do qual é possível aceder também aos **ALUNOS, CURSOS** e **DISCIPLINAS**.

Para tornar as coisas simples e claras, comecemos por criar um directório, p.ex: **D:\upINSCR** para onde copiamos a nossa base de dados **INSCR.mdb** (versão Access 2003) ou **INSCR.accdb** (versão Access 2007).

Depois iniciamos um **novo projecto** VB a que damos o nome **INSCR** e que vamos guardar no mesmo directório. Uma possibilidade é logo que entremos no projecto, fazer a sua gravação, através do menu **File/ Save All** e na caixa de diálogo indicar:

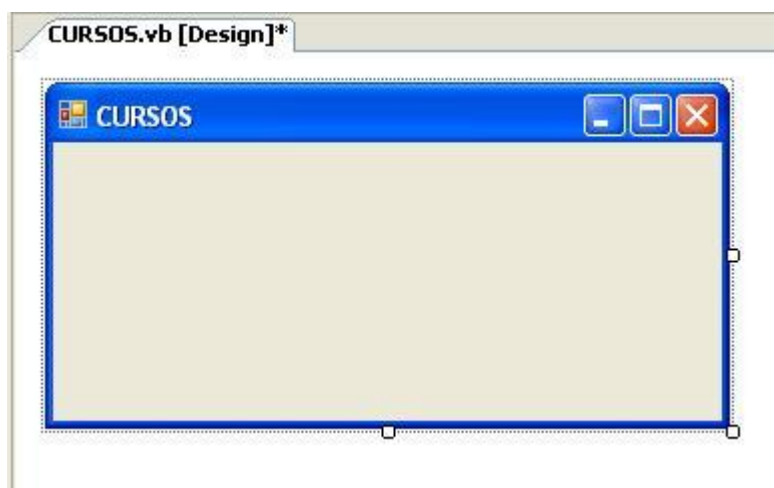


Recordemos o nosso modelo de dados, representado no diagrama:



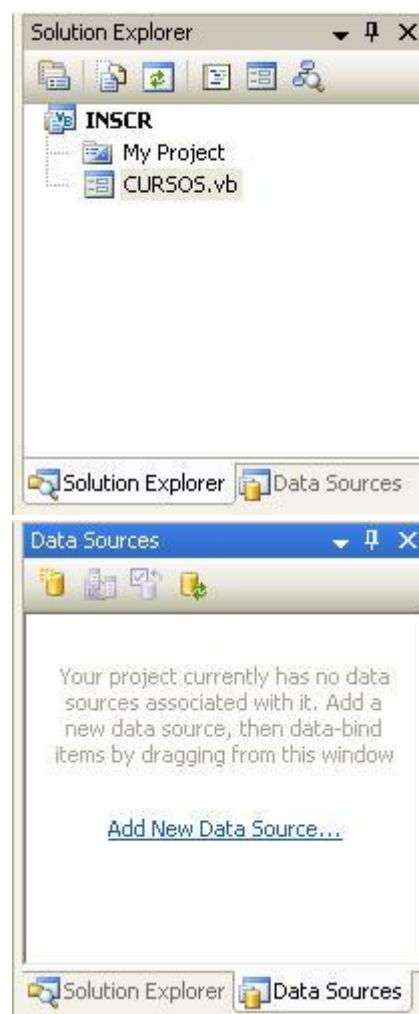
Uma vez que a tabela Cursos não tem dependências, vamos começar por criar um quadro para fazer a sua manutenção.

Como temos no projecto um **Form1** proposto pelo IDE, vamos alterar-lhe o nome para **CURSOS** e altermos também a sua propriedade **text** igualmente para **CURSOS**. Ficamos então com o seguinte aspecto inicial:

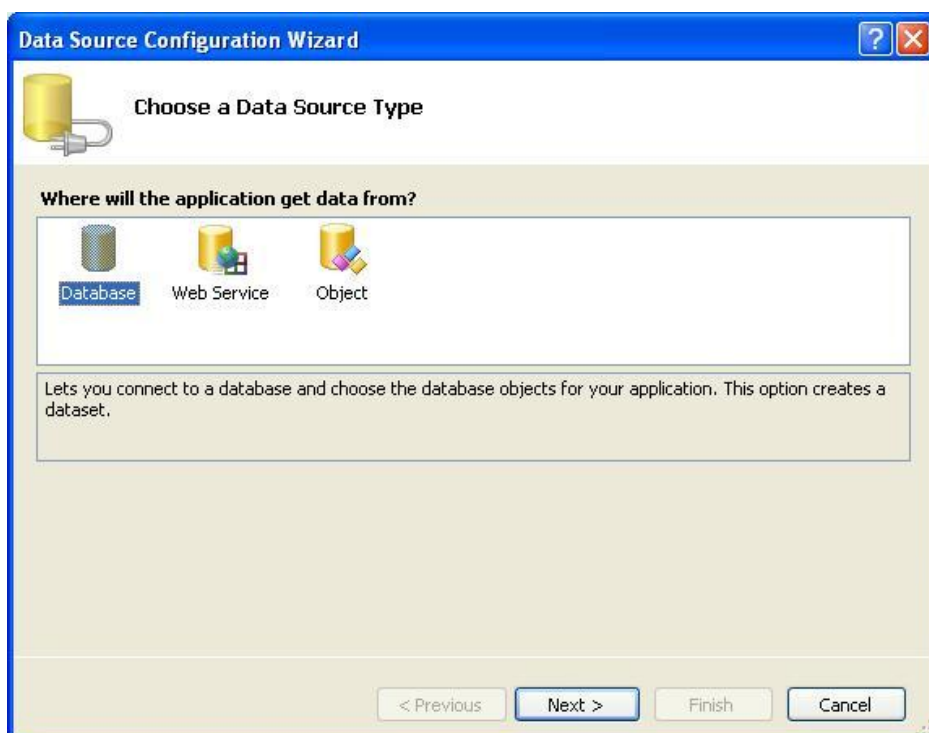


O passo seguinte é estabelecer a **ligação à base de dados**, e para isso vamos começar por notar que num separador junto ao ***Solution Explorer*** temos um outro designado por ***Data Sources***:

Vamos activá-lo com um clique em cima do nome e aparece uma mensagem para adicionarmos uma origem de dados e um atalho para continuarmos:



Levando-nos a um quadro onde iremos escolher o tipo de ligação a efectuar.



Dado que o nosso objectivo é a ligação à nossa base de dados das inscrições iremos aceitar a opção mostrada (*database*) e com um clique no botão **Next** avançamos para um quadro onde teremos de criar uma nova ligação com um clique em **New Connection** aparece um quadro onde o tipo de provider já está escolhido como sendo o Access, mas poderemos alterar para outro, e onde iremos com o botão **Browse** escolher a nossa base de dados em: **D:\upINSCR\INSCR.mdb**

Podem observar-se algumas das opções de acesso à base de dados e o botão **Advanced** para situações mais elaboradas.



Há igualmente um botão para testar: **Test Connection**, que deve ser a nossa próxima escolha para verificarmos a disponibilidade dos dados, e que depois de pressionado deve fazer aparecer a mensagem de tudo **OK**.



Fazendo **OK** nesta mensagem e no quadro de escolha da base de dados voltamos ao quadro de configuração onde deve aparecer o tipo de provider, local e nome da nossa base de dados, onde podemos ver também a string de conexão que foi criada e que poderá mais tarde ser usada no código:



Choose Your Data Connection

Which data connection should your application use to connect to the database?

ACCESS.D:\upINSCR\INSCR.mdb New Connection...

This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set this information in my application code.

☐ Yes, include sensitive data in the connection string.

☒ Connection string

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\upINSCR\INSCR.mdb

Com um clique em **Next** recebemos a seguinte mensagem:

Microsoft Visual Studio

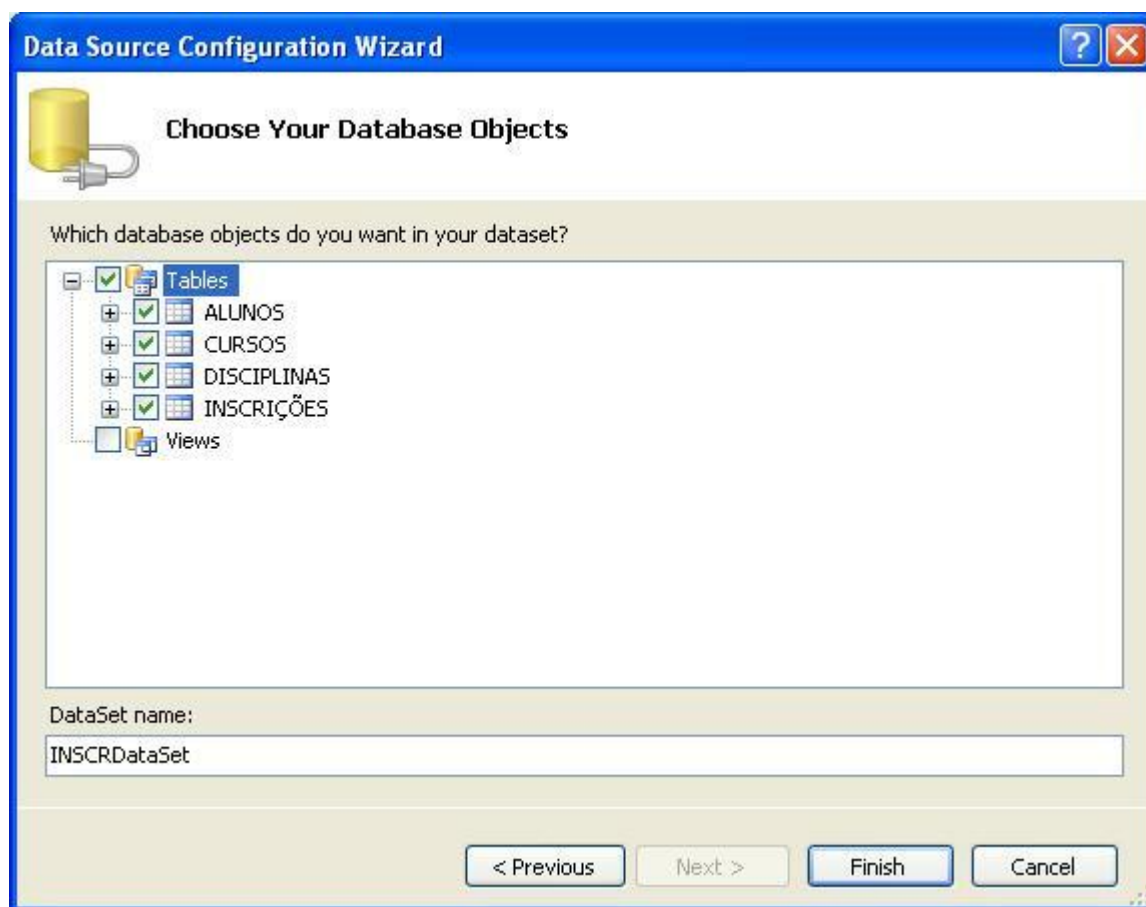
The connection you selected uses a local data file that is not in the current project. Would you like to copy the file to your project and modify the connection?

If you copy the data file to your project, it will be copied to the project's output directory each time you run the application. Press F1 for information on controlling this behavior.

Yes No

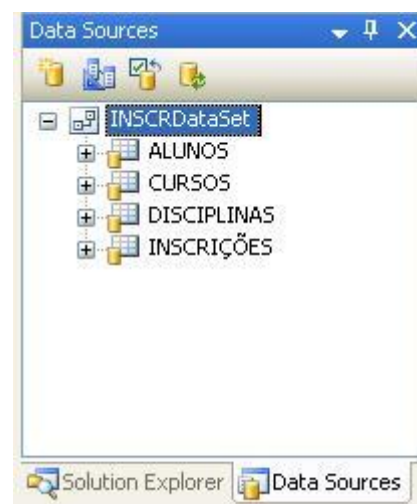
Temos de usar o botão **[No]** para usarmos efectivamente a base de dados que pretendemos e não uma cópia.

Num quadro a seguir aceitamos a gravação da string de conexão no ficheiro de configuração da aplicação e aparece então um outro quadro onde iremos especificar a que objectos da base de dados nos pretendemos ligar. Vamos seleccionar as **tabelas** (aliás os únicos objectos com que sabemos tratar, de momento) e podemos com um clique no quadrado **[+]** expandir a colecção e vê-las todas seleccionadas:



Com um clique no botão **Finish** terminámos este passo de ligação à base de dados e voltámos ao IDE onde devemos notar o aparecimento dos objectos da base de dados no separador **Data Sources**.

Note-se a criação automática pelo IDE do **DataSet** de acesso aos dados, contendo todas as tabelas.



Como temos por objectivo criar um quadro para manusear os Cursos vamos fazer uma coisa tão simples como **arrastar o nome Cursos** que aparece nos **Data Sources**, para o quadro. Apenas com um pequeno ajuste de dimensões vamos obter o seguinte aspecto:



Comecemos por notar os controlos: **CURSOSTableAdapter**, o **INSCRDataSet** eo **CURSOSBindingSource**, com os significados já conhecidos.

Aparece ainda um objecto, o **CURSOSBindingNavigator**, que como o nome sugere é um controlo que permite percorrer a tabela, e que corresponde à barra que temos de momento no topo do quadro. E que vamos mover para o fundo, apenas por uma questão de hábito...

Analisemos o código que foi adicionado ao quadro:

```
Public Class CURSOS

    Private Sub CURSOSBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CURSOSBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.CURSOSBindingSource.EndEdit()
        Me.CURSOSTableAdapter.Update(Me.INSCRDataSet.CURSOS)
    End Sub

    Private Sub CURSOS_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.CURSOSTableAdapter.Fill(Me.INSCRDataSet.CURSOS)
    End Sub
End Class
```

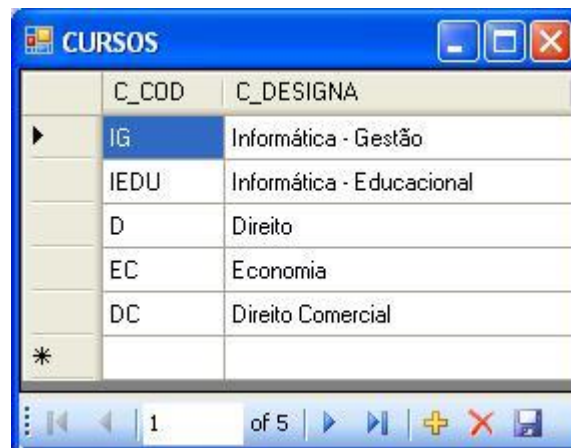
Note-se:

- No **Load** do quadro o método **Fill** sobre o **TableAdapter** para carregamento da **DataGrid** com os dados existentes.

- No botão **Save** do **Navigator** o código para a passagem dos dados do quadro para o **TableAdapter** e a sua actualização na base de dados.

Com uns pequenos ajustes de dimensões podemos correr a aplicação e vemos:

Com os botões do navegador podemos avançar e recuar nos registos, ir para o primeiro e para o último, avançar para um novo, apagar e gravar as alterações para a base de dados.



Vamos então alterar o código de **D** para **DIR**, eliminar o registo de código **DC** e inserir um novo curso **Iqf** – **Informática que funciona**.

Depois de todas estas manipulações, ou uma a uma, teremos de as gravar e podemos no final ir ver à própria base de dados no Access qual a situação em que ficamos.

Encontramos então:



ou seja, as nossas alterações gravadas, como seria de esperar.

Vamos acrescentar ao nosso projecto um quadro para os alunos.

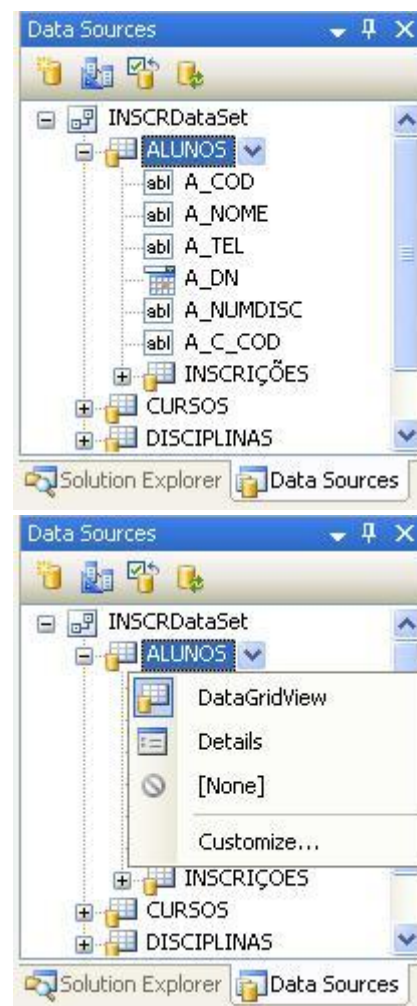
É normal e desejável, nas aplicações que a informação possa ser vista na forma de listas onde o utilizador pode filtrar e ordenar os dados, e também na forma de ficha, onde, para as tabelas com muitos campos, é possível ver toda a informação de cada entidade num só quadro. Vamos então criar um quadro tipo ficha de aluno.

Devemos ter em atenção o facto de o código estar a ser executado fora do nosso controlo de erros, o que será sempre desejável e necessário para prevenir situações inesperadas.

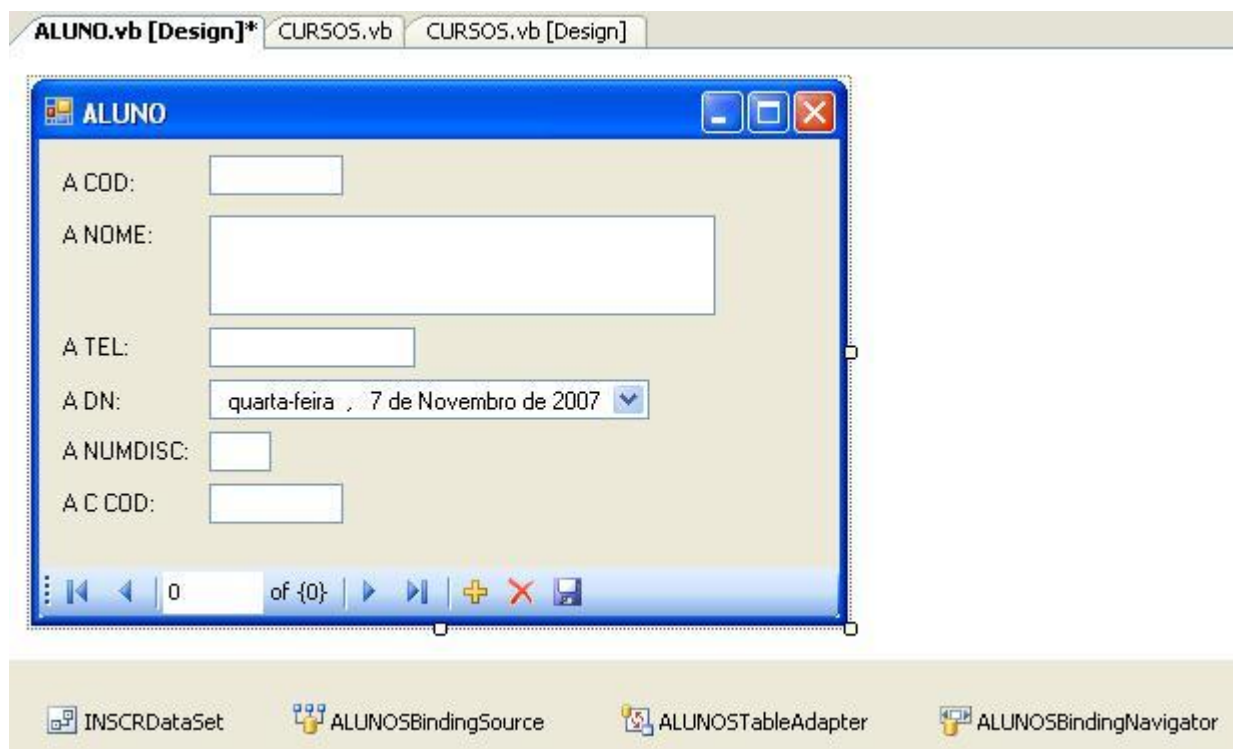
Adicionamos um novo quadro **ALUNO** e vamos ao **Data Sources** para nos ligarmos à base de dados.

Com um pouco mais de curiosidade expandimos no **DataSet**, a tabela dos alunos, e passamos a ver todos os campos.

Note-se que temos um botão de opções à direita do nome **ALUNOS**, e que abrindo as opções encontramos as opções de **DataGrid** e **Details**.




Selecionando **Details**, arrastando para o quadro e compondo ligeiramente vamos poder ter o seguinte aspecto:



onde todos os controlos que aparecem são nossos conhecidos.

O código gerado é o equivalente ao do quadro anterior.

Alterando nas propriedades do projecto o quadro de arranque para ALUNO, podemos ver:



Uma vez que o campo do **Curso**, deve ser preenchido com valores que existam na tabela, seria interessante que em vez de ser uma caixa de texto pudesse ser uma **ComboBox**.

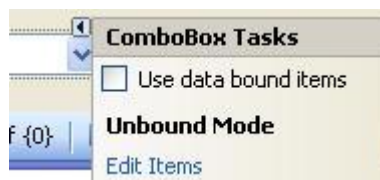
Poderíamos ter tomado essa decisão anteriormente. Quando expandimos a tabela **Alunos** e vimos os seus campos, para cada um poderia ser possível definir algumas das propriedades dos controlos que vão ser usados. Repare-se na figura anterior e veja-se como o campo data aparece com o **DateTimePicker**. Como há sempre muitas alternativas vamos adicionar a **Combobox** manualmente e ligá-la à tabela.

Arrastamos da **ToolBox** para o quadro uma **Combobox** e retiramos a caixa de texto com o código do curso.

Com a **ComboBox** seleccionada podemos ver que tem um botão para opções:

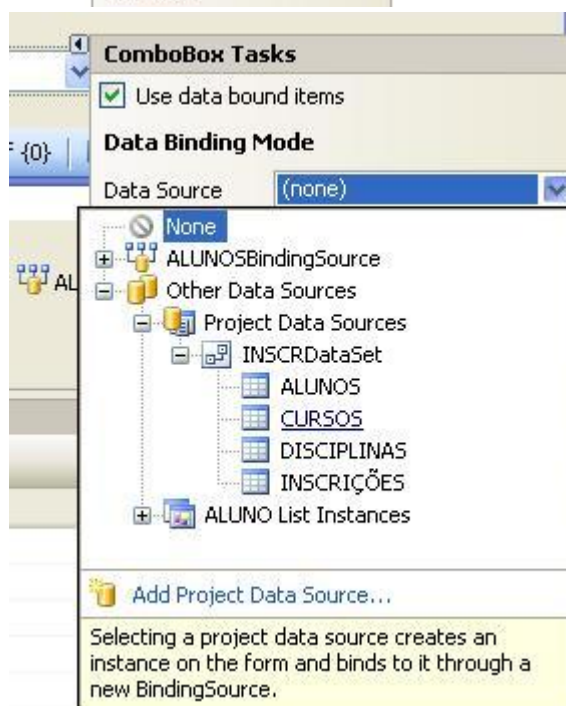


que ao ser pressionado abre uma lista de opções onde encontramos a possibilidade de fazer a ligação a campos do **DataSet**. "**Use data bound items**"



Repare-se nas opções escolhidas para o **Data Source**:

- **Other Data Sources**
- **Project Data Sources**
- **INSCRDataSet**
- **CURSOS**



O que se fez, foi ir ao **DataSet** buscar a tabela dos **Cursos**.

Para finalizar vamos indicar:

- **Display Value:** **C_COD**
- **Value Member:** **C_COD**
- **Selected Value:** **A_C_COD** do **BindingAlunos**.

Ao finalizarmos esta operação temos de reparar que foram adicionados mais controlos ao quadro:

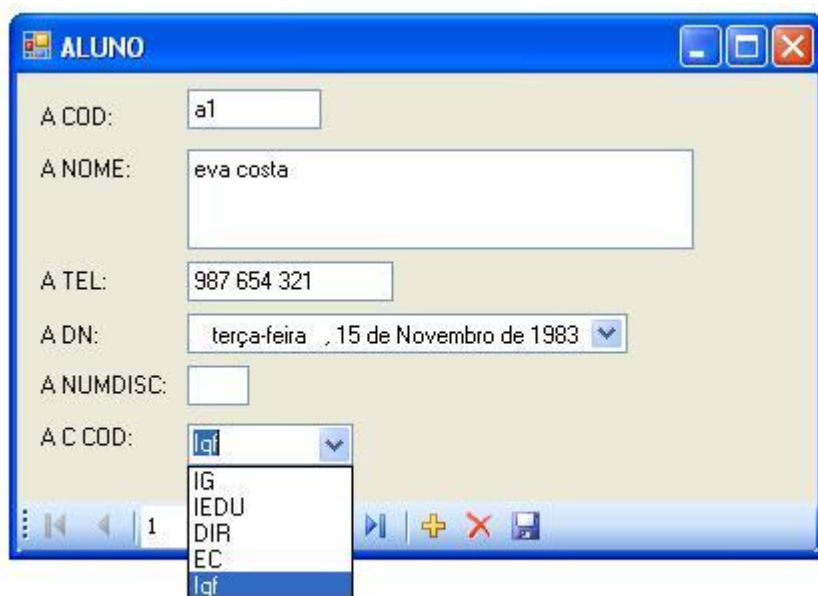



ou seja, um **DataAdapter** para a tabela dos **Cursos** e o respectivo **BindingSource**.

Devemos igualmente reparar que no **Load** do quadro ocorre também o carregamento da origem de dados para a **ComboBox**:

```
Private Sub ALUNO_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.CURSOSTableAdapter.Fill(Me.INSCRDataSet.CURSOS)
    Me.ALUNOSTableAdapter.Fill(Me.INSCRDataSet.ALUNOS)
End Sub
```

Correndo a aplicação aparece o quadro com a ficha do aluno e podemos verificar o funcionamento da **ComboBox**:

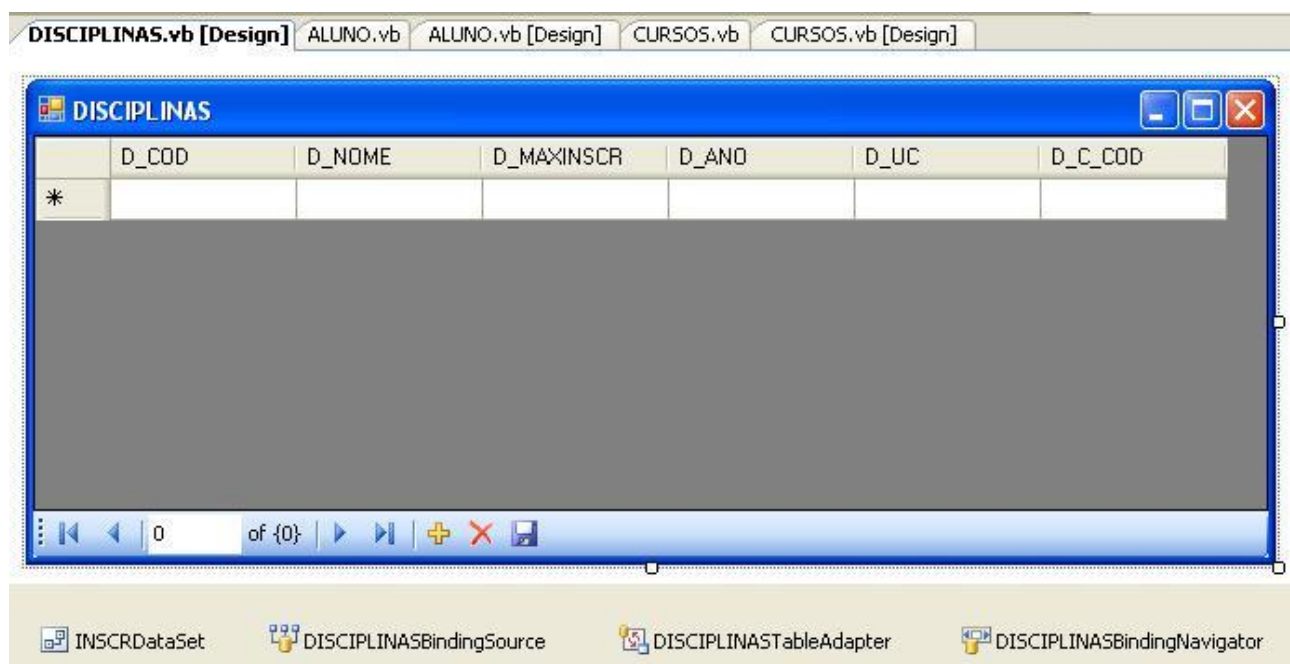


Precisamos agora de um quadro para as **Disciplinas** e temos várias hipóteses.

- ▶ Uma hipótese seria um quadro em que para cada **Curso** pudessemos indicar as **Disciplinas** que contempla;
- ▶ Outra; seria uma lista de **Disciplinas** onde em cada uma se escolheria o **Curso**.
Vamos optar por esta segunda hipótese para demonstrarmos mais uma variante no uso de controlos no VB.

Criamos então um novo quadro a que chamaremos **DISCIPLINAS** e que como **Data Source** terá a tabela com o mesmo nome, e vamos usar mais uma vez uma vista de dados na forma de lista, com o uso de uma **DataGrid**.

Seria este o aspecto inicial do quadro:



onde nada aparece de novo, e que podemos visualizar correndo o programa e vendo alguns dados já introduzidos na base de dados:

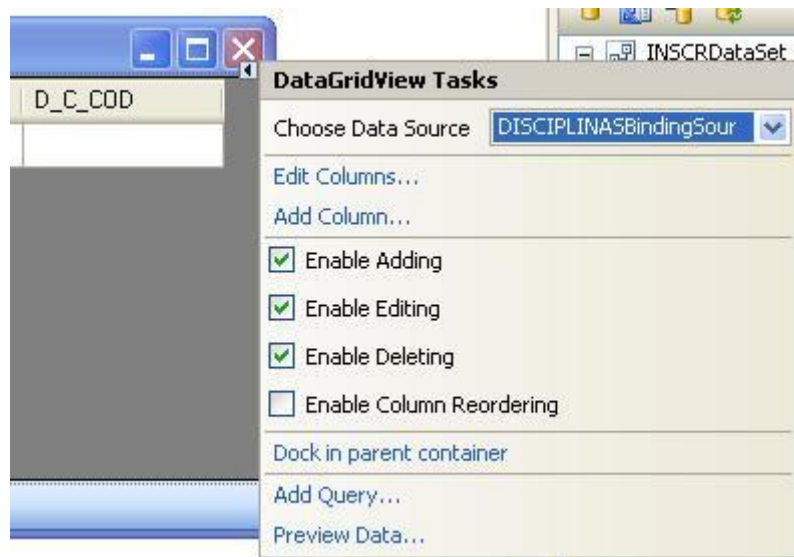


Uma vez que cada disciplina é de um curso e que já sabemos que cursos temos, seria interessante poder usar uma **Combobox** como campo da grelha na coluna do curso: **D_C_COD**. Para isso, vamos alterar algumas propriedades na **DataGrid**.

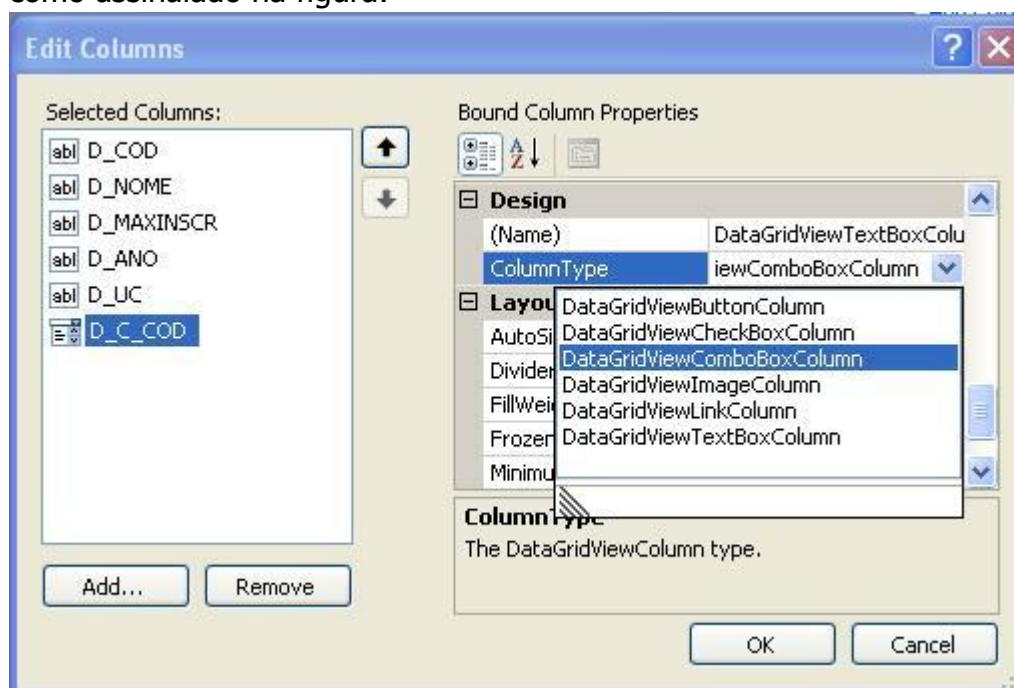
Com um clique na grelha aparece o botão de opções da grelha:



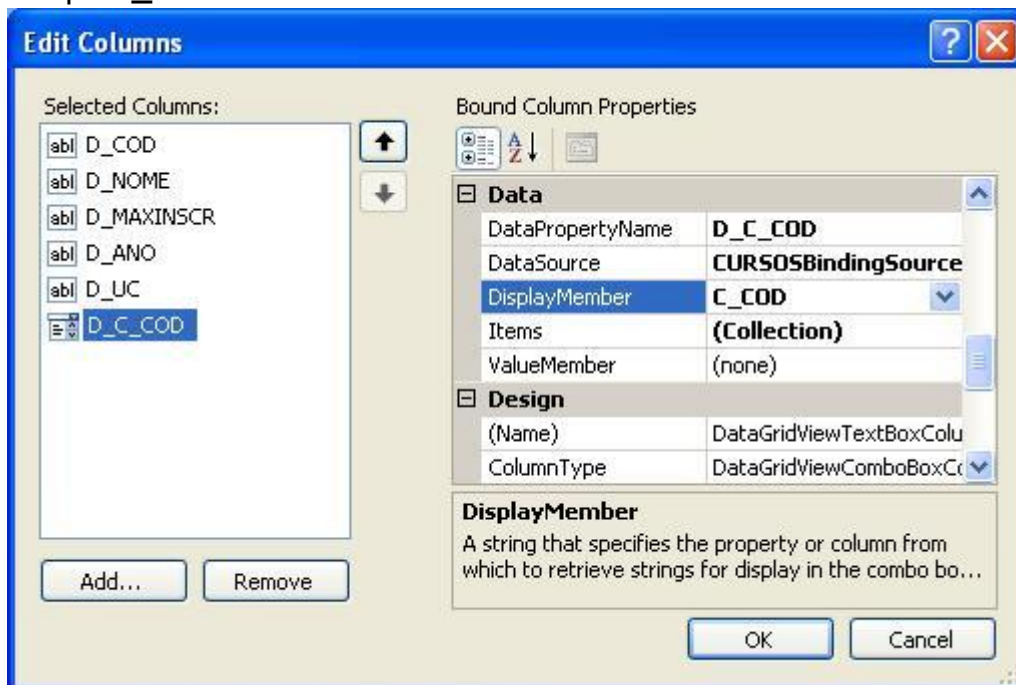
Onde poderemos alterar as colunas.



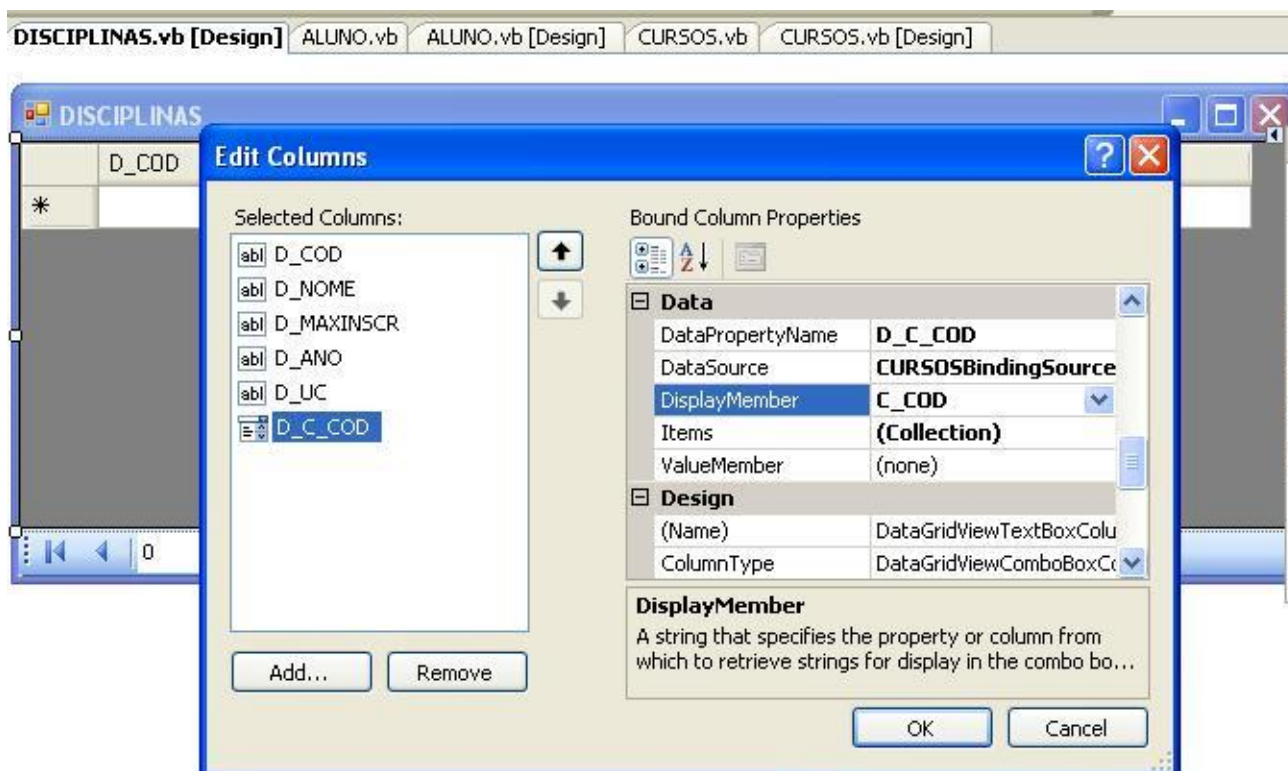
Fazendo então um clique em **Edit Columns** e escolhendo o campo do **Curso**, procuramos na secção **Design** das propriedades, o tipo de coluna e vamos passá-lo para **ComboBox**, como assinalado na figura:



Nas propriedades **Data**, vamos poder indicar, da mesma forma que procedemos no caso anterior, qual o **Data Source** e o **Display Member**, que serão a tabela dos **Cursos** e o campo **C_COD**.



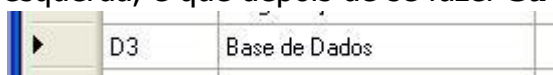
Com esta selecção efectuada, podemos observar que no quadro já apareceu o **DataAdapter** para os **Cursos** e o respectivo **Binding Source**:



Também no código deverá ter sido feita a inclusão do carregamento dos cursos. Podemos executar o programa e observar o quadro com as caixas de escolha:



Aproveitemos para notar o símbolo de edição de registos que aparece no selector de registos à esquerda, e que depois de se fazer **Save** no botão do navegador passa a ser uma seta:



Por curiosidade podemos ir ver ao **Access**, esta alteração que acabamos de efectuar, e encontramos:



Microsoft Access

File Edit View Insert Format Records Tools Window Help

INSCR : Database (Access 2002 - 2003 file format)

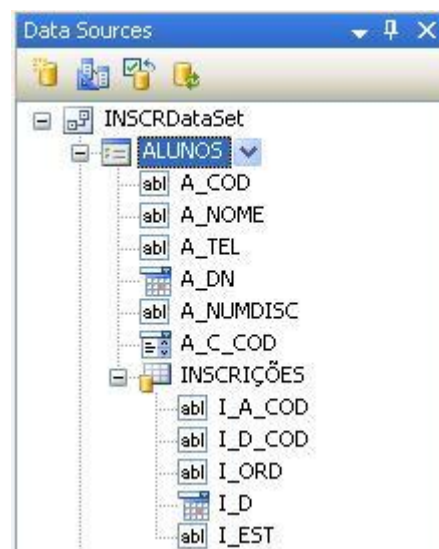
DISCIPLINAS : Table

	D_COD	D_NOME	D_MAXINSCR	D_ANO	D_UC	D_C_COD
+	D1	Algoritmia				
+	D2	Programação				
+	D3	Base de Dados				lqf
+	D4	Álgebra				
+	D5	Arquitectura de Computadores				
+	D6	Estruturas de Dados				
+	D7	Teoria da Computação				
+	D8	Sistemas Operativos				
*						

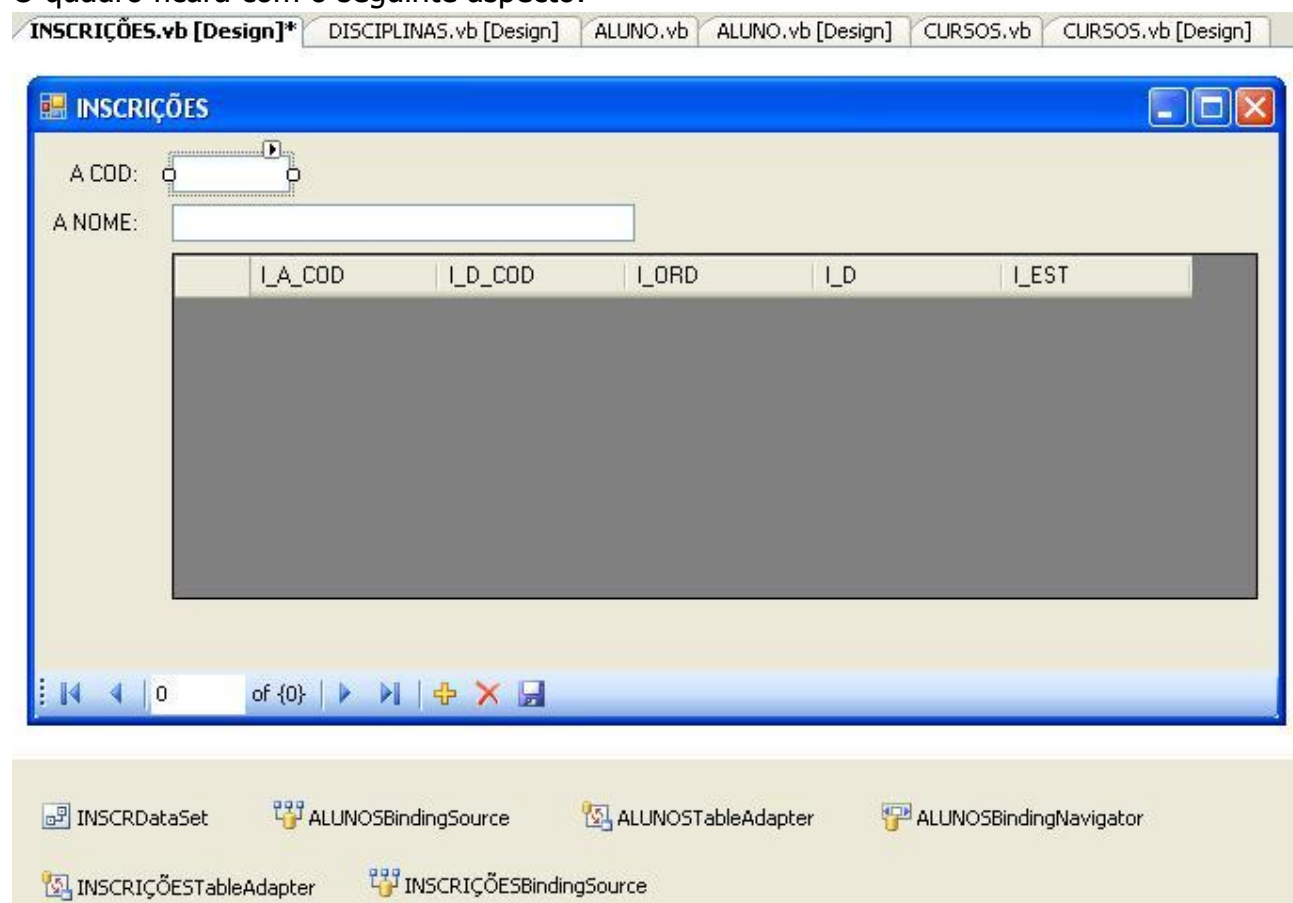
Estamos agora em condições de construir o quadro das **INSCRIÇÕES**, onde para cada aluno queremos manusear as disciplinas em que se inscreve.

Criamos o quadro e vamos definir o **DataSource**, expandindo a tabela dos **ALUNOS**, notamos que as **Inscrições** aparecem como uma grelha na lista de campo dos alunos. O IDE vai buscar esta informação à base de dados, procurando as relações que estão definidas.

Assim, e pelo processo habitual vamos arrastar para o quadro criado os campos **A_COD** e **A_NOME** e a tabela das **INSCRIÇÕES** que aparece na lista de expansão dos **ALUNOS**.



O quadro ficará com o seguinte aspecto:



O programa pode ser executado tendo o quadro **INSCRIÇÕES** como inicial e havendo dados na base de dados, eles iriam aparecer.

Acontece porém que não temos forma de gravar as inscrições que efectuarmos, uma vez que o navegador e os seus botões estão ligados ao **BindingSource** do quadro que é a dos **ALUNOS**. Por outro lado também não devemos querer andar à procura dos alunos com o navegador pelo que a melhor opção será removê-lo, e juntar um novo ligado às **Inscrições**.

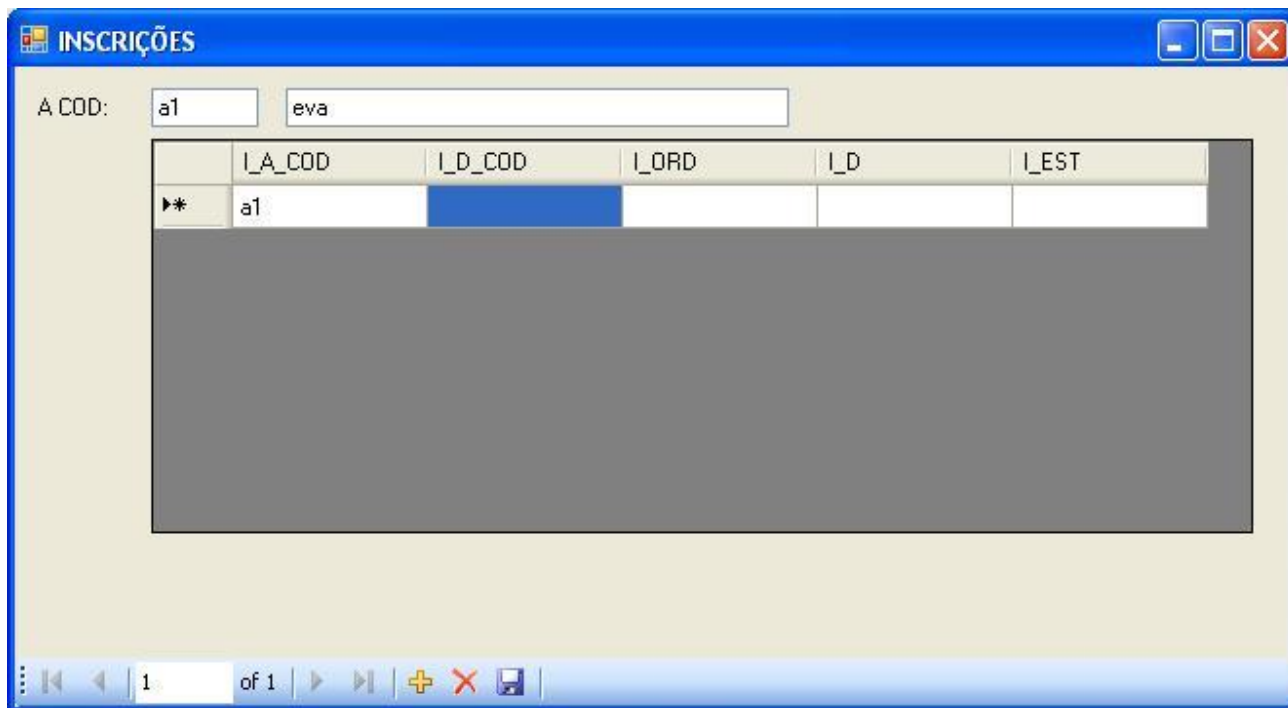
Ao adicionarmos o novo navegador observamos que não tem o botão de gravação, pelo que teremos de ir às suas propriedades e adicioná-lo. Teremos igualmente de o associar às **Inscrições**.

Abrimos então as propriedades do **Navigator** (como visto anteriormente) e fazemos um clique em **Insert Standard Items**, teremos depois de remover tudo o que não queremos.

Nas suas propriedades (**Edit Items**) vamos também alterar a ligação de modo a relacionar-se com a grelha de **Inscrições**. Temos então de colocar o **BindingSource**, com o valor **INSCRIÇÕESBindingSource**.

Se executarmos o programa verificamos que aparece o primeiro registo da tabela dos alunos, e que não temos modo de sair dele.

Podemos verificar que se levamos o cursor até ao campo **I_D_COD** o campo **I_A_COD** aparece o código do aluno que está em cima. Isto vem do facto de termos usado a tabela das **Inscrições** que o IDE detectou como relacionada com a dos **ALUNOS** e que por isso considerou no **BindingSource**.



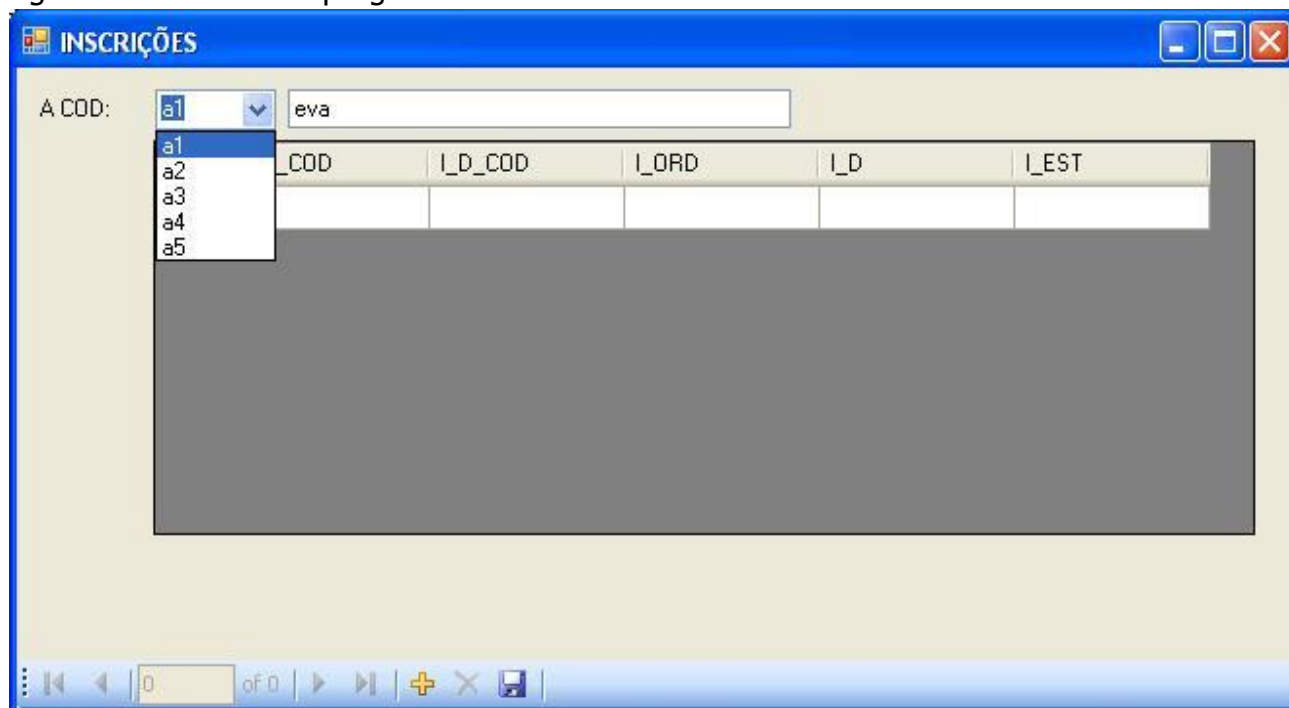
	I_A_COD	I_D_COD	I_ORD	I_D	I_EST
▶*	a1				

Vamos resolver a questão de nos colocarmos no aluno para o qual pretendemos fazer as inscrições, substituindo a caixa de texto por uma **Combobox**, da forma que já fizemos nos outros casos.

Podemos remover o campo **A_COD** e adicionar a **Combobox** no seu lugar.

Temos de passá-la a campo **bounded**, e vamos usar como **data source** **AlunosBindingSource**, que é o suporte do quadro, e como **Display Member** e **Value Member** usamos **A_COD**.

Agora se correremos o programa vamos encontrar:



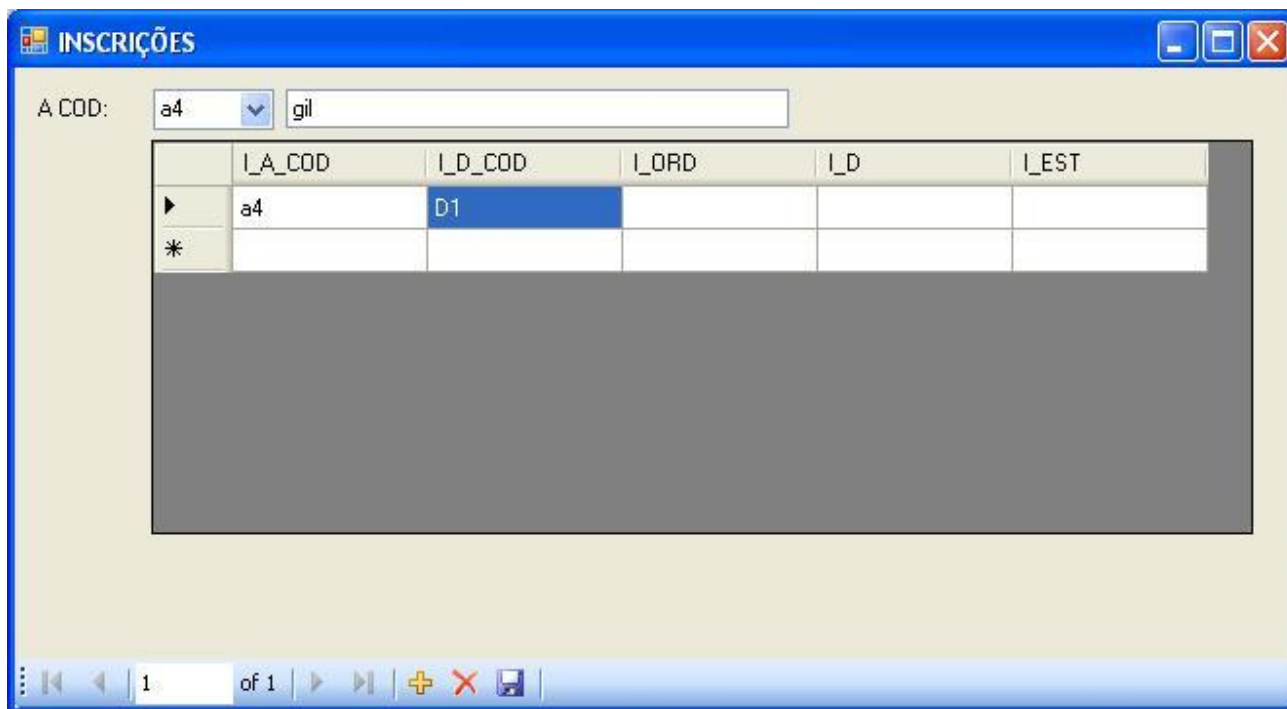
e seleccionando qualquer alunos iremos ver aparecer o seu nome e as suas inscrições.

Uma vez que estes campos (código e nome) não serão alterados aqui, vamos passar a sua propriedade **Readonly** a **True**.

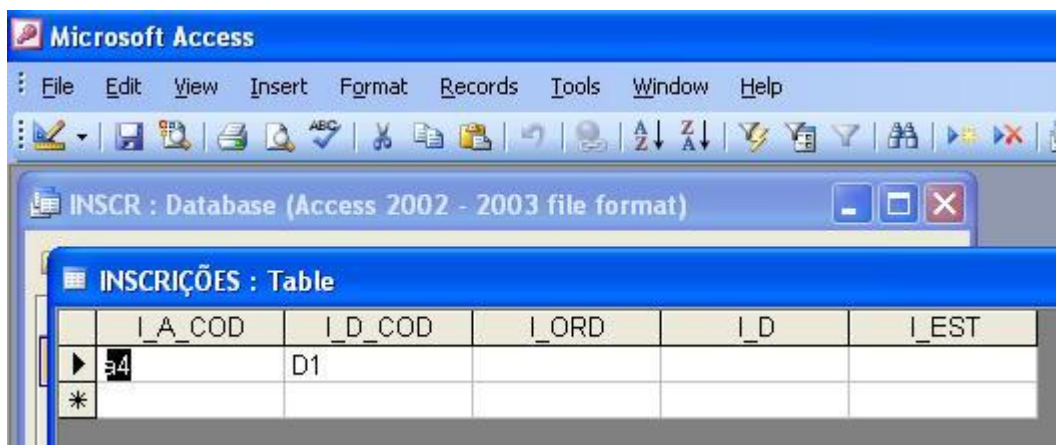
Estaríamos em condições de criar inscrições, mas não temos código associado ao botão gravar no nosso novo navegador. Para isso fazemos um clique duplo em cima do botão e passamos ao painel de código.

Vamos notar que, como de costume, pelo facto de termos retirado o controlo anterior de navegação, o seu código permanece lá, embora desligado da aplicação, pelo que podemos copiá-lo para o nosso navegador substituindo a palavra **ALUNOS** por **INSCRIÇÕES**.

Podemos voltar a executar a aplicação, escolher na combobox o aluno **A4** e inscrevê-lo na disciplina **D1**, observamos o registo em alteração (símbolo à esquerda), gravamos com o botão do navegador:



e podemos ir ao Access verificar o que aconteceu.



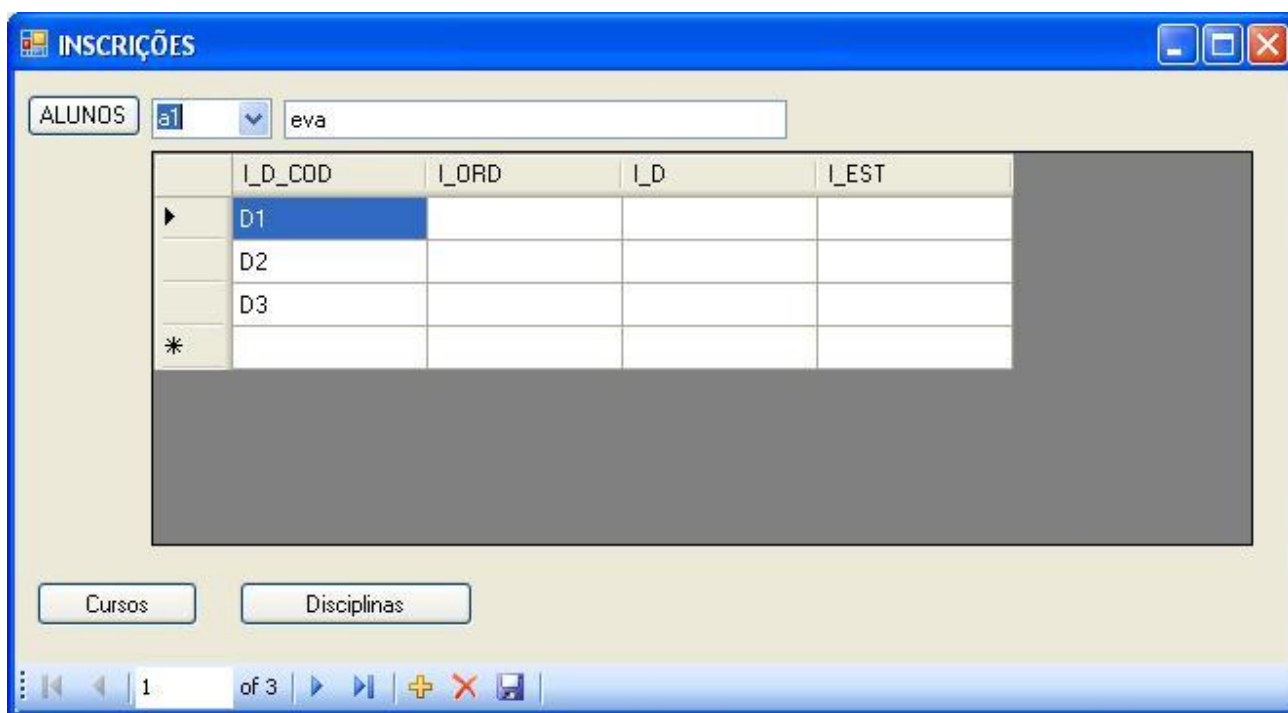
Temos a nossa aplicação quase pronta.

Vamos trocar o label **A_COD** por um botão que abra a ficha do aluno para podermos acrescentar novos alunos, com o seguinte código:

```
ALUNO.ShowDialog()
Me.ALUNOSTableAdapter.Fill(Me.INSCRDataSet.ALUNOS)
```

Deste modo abrimos o quadro dos alunos, e no regresso garantimos que todas as alterações que fizemos são vistas pelo quadro **INSCRIÇÕES**.

Para finalizar vamos colocar um botão para abrir o quadro dos **Cursos**, outro para abrir o quadro das **Disciplinas** e finalmente esconder a coluna **I_A_COD** da **Datagrid**. Ficaremos com o seguinte aspecto:



Melhoria necessária na aplicação:

- **Controlo de erros** em todo o código... que é quase nenhum. Podem ocorrer falhas por questões relacionadas com a base de dados: campos obrigatórios, relações de integridade, etc...

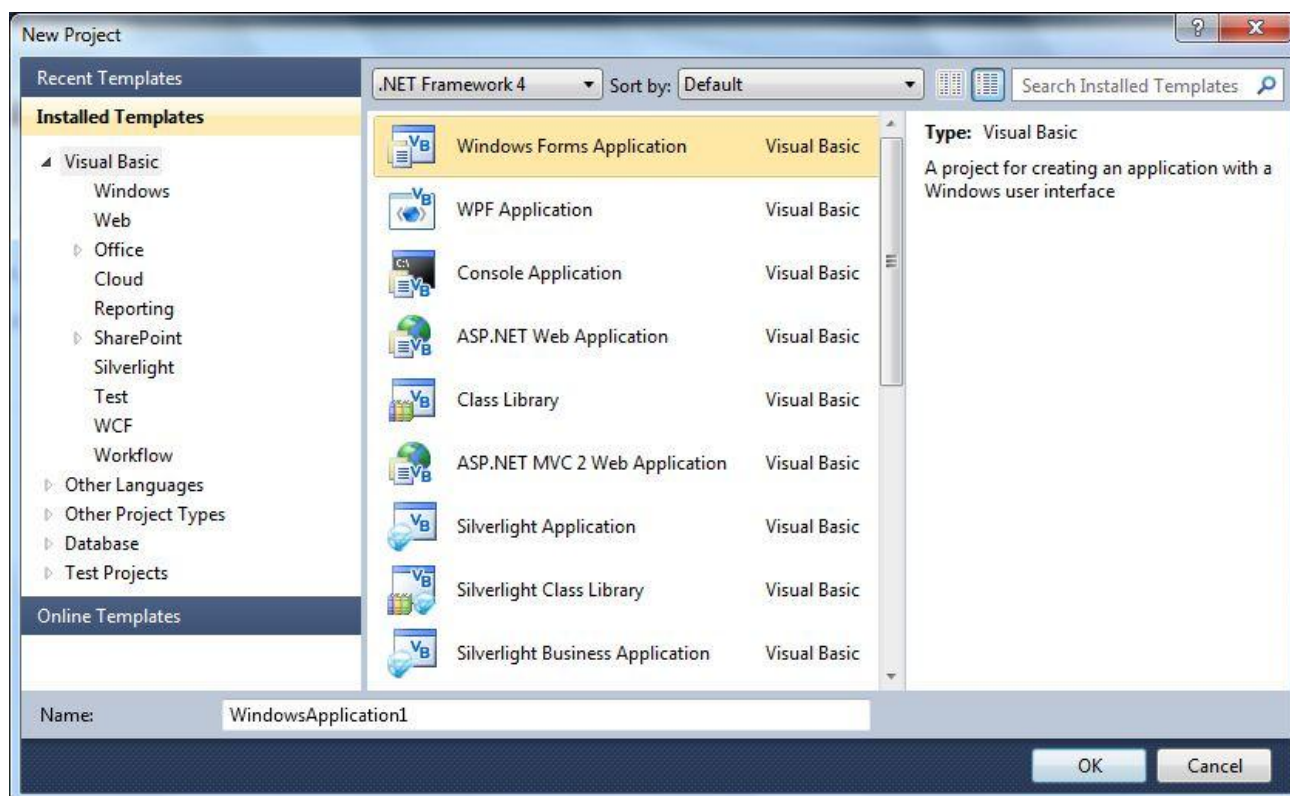
Algumas melhorias interessantes a fazer:

- alterar os labels dos campos de texto e colunas
- alterar a cor do contorno (ou fundo) dos campos que são **Read only**
- passar o campo **I_D_COD** a **Combobox**
- controlar se há alteração nos campos ou registos e mostrar ao utilizador que há dados para gravar, ou até tornar a gravação automática
- criar um ficheiro de controlo para alguns parâmetros da aplicação, como por exemplo o local da base de dados e a situação anterior: a opção de gravar automaticamente os dados

4. Visual Studio 2010

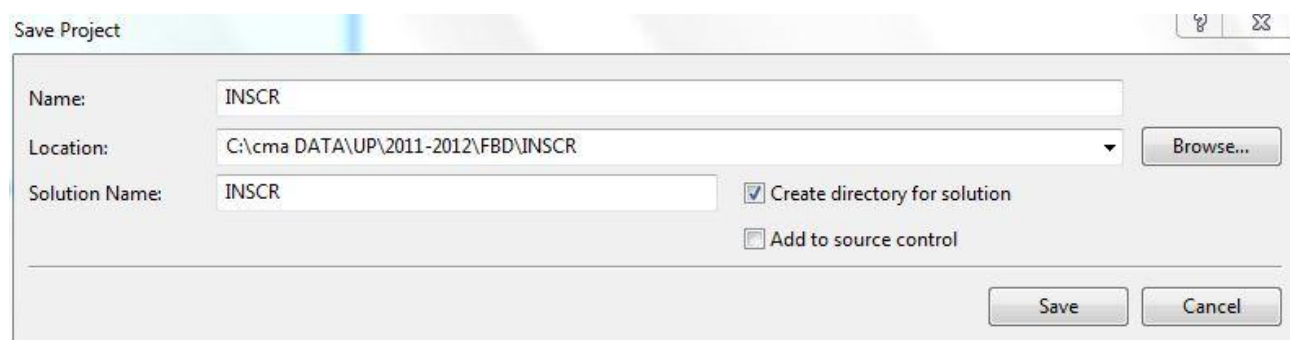
Nesta versão do VS as diferenças aparecem na forma de estabelecer a ligação do projeto VB à base de dados. A utilização dos componentes básicos é a mesma do VS 2005.

Será necessário criar um novo projeto



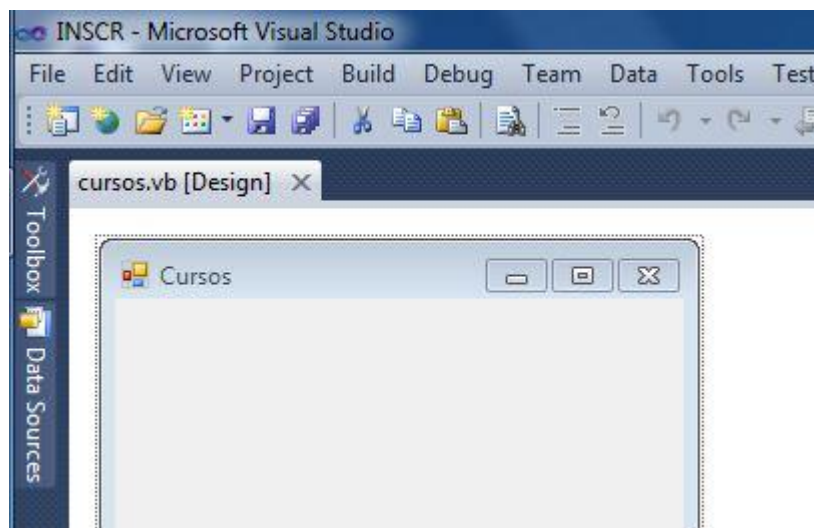
O nome do projeto muda-se para INSCR, por exemplo.

Recomenda-se efetuar File/ Save All com a criação de um directório para os componentes do projeto:

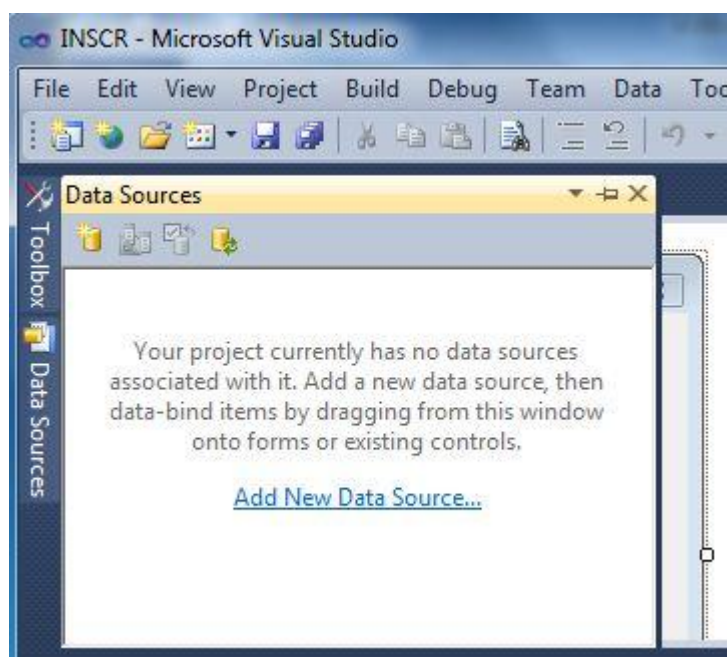


Mudar o nome do form1, do ficheiro e o título para: Cursos, ou apagar o Form1 e fazer Add de um novo form.

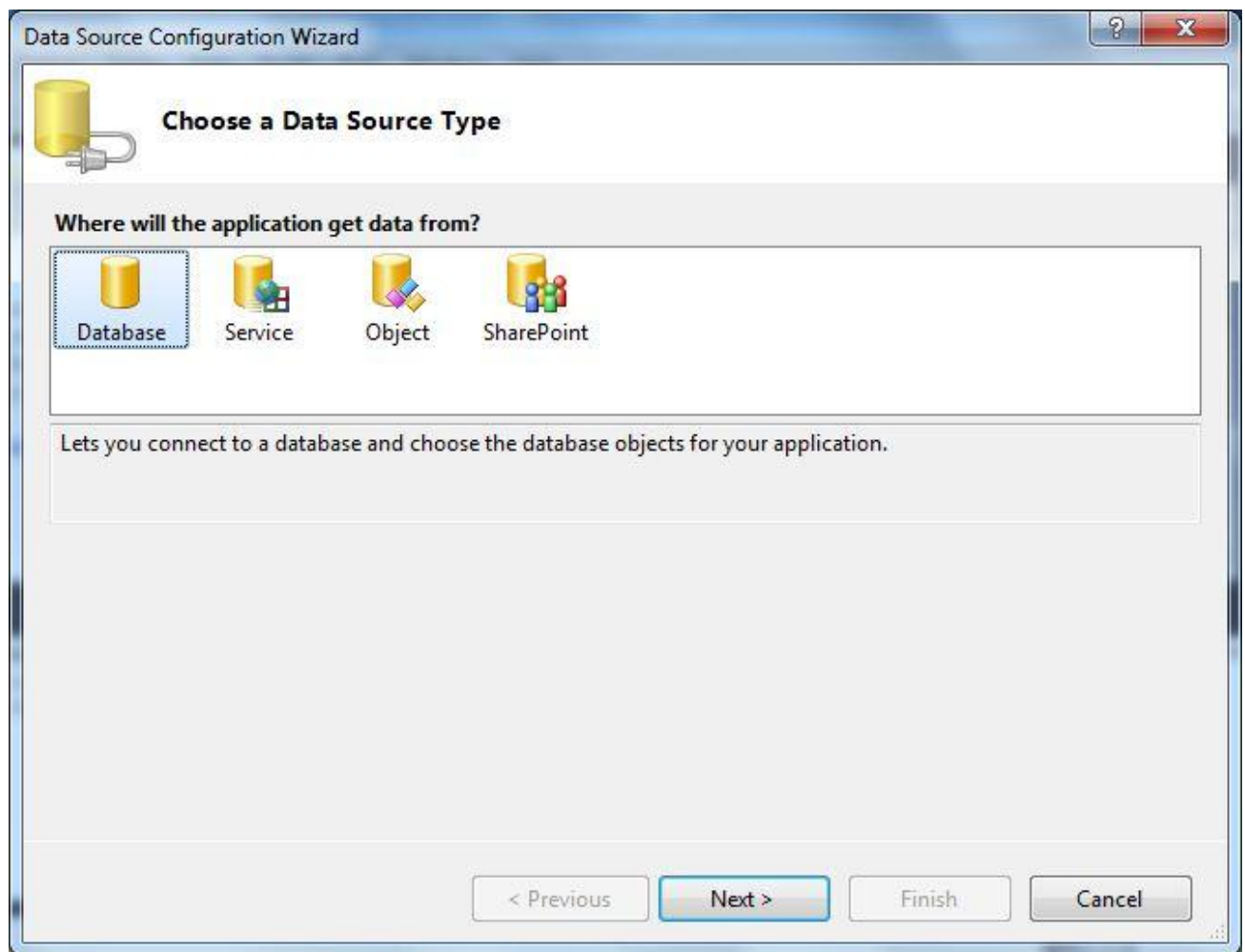
Abrir "Data Sources"

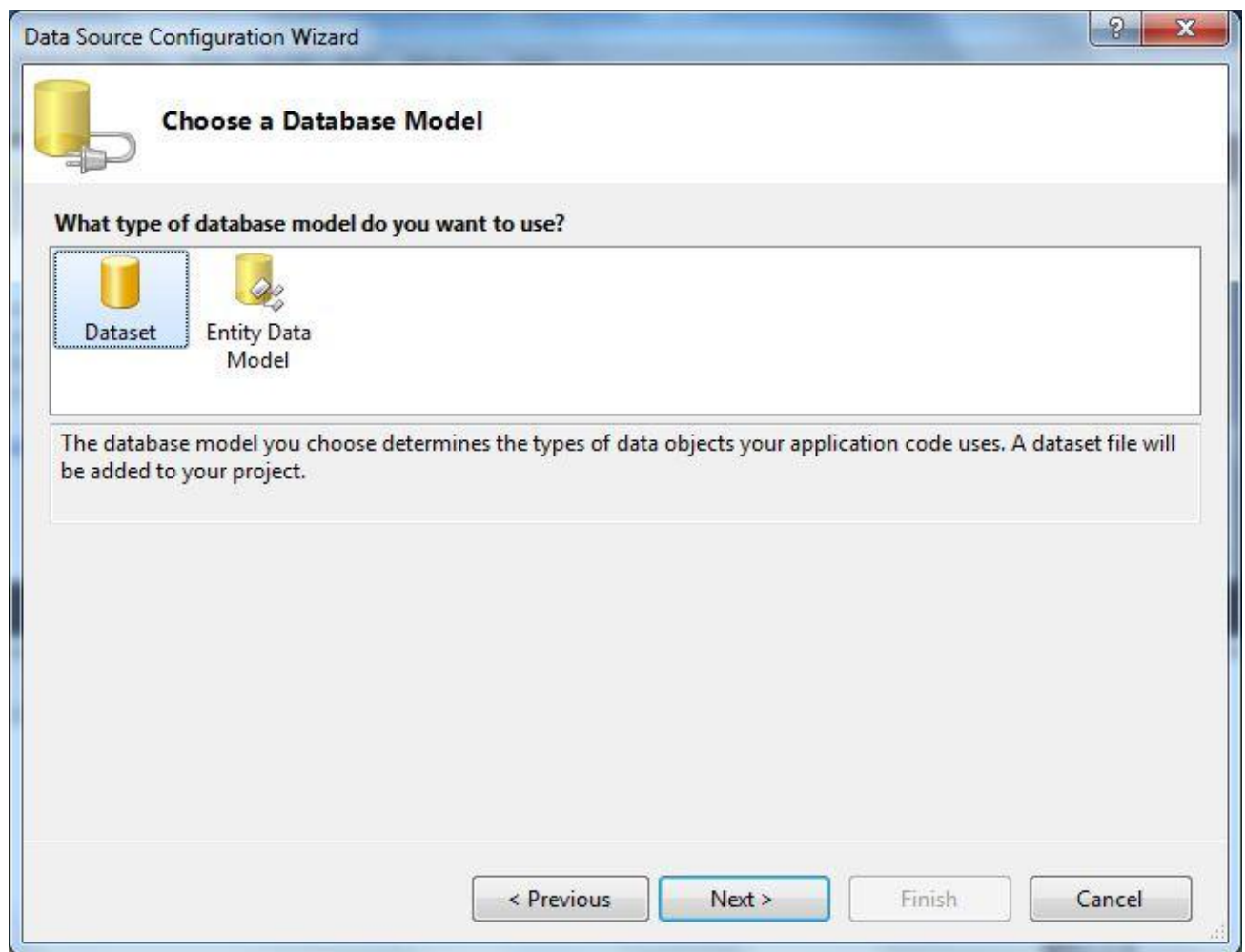


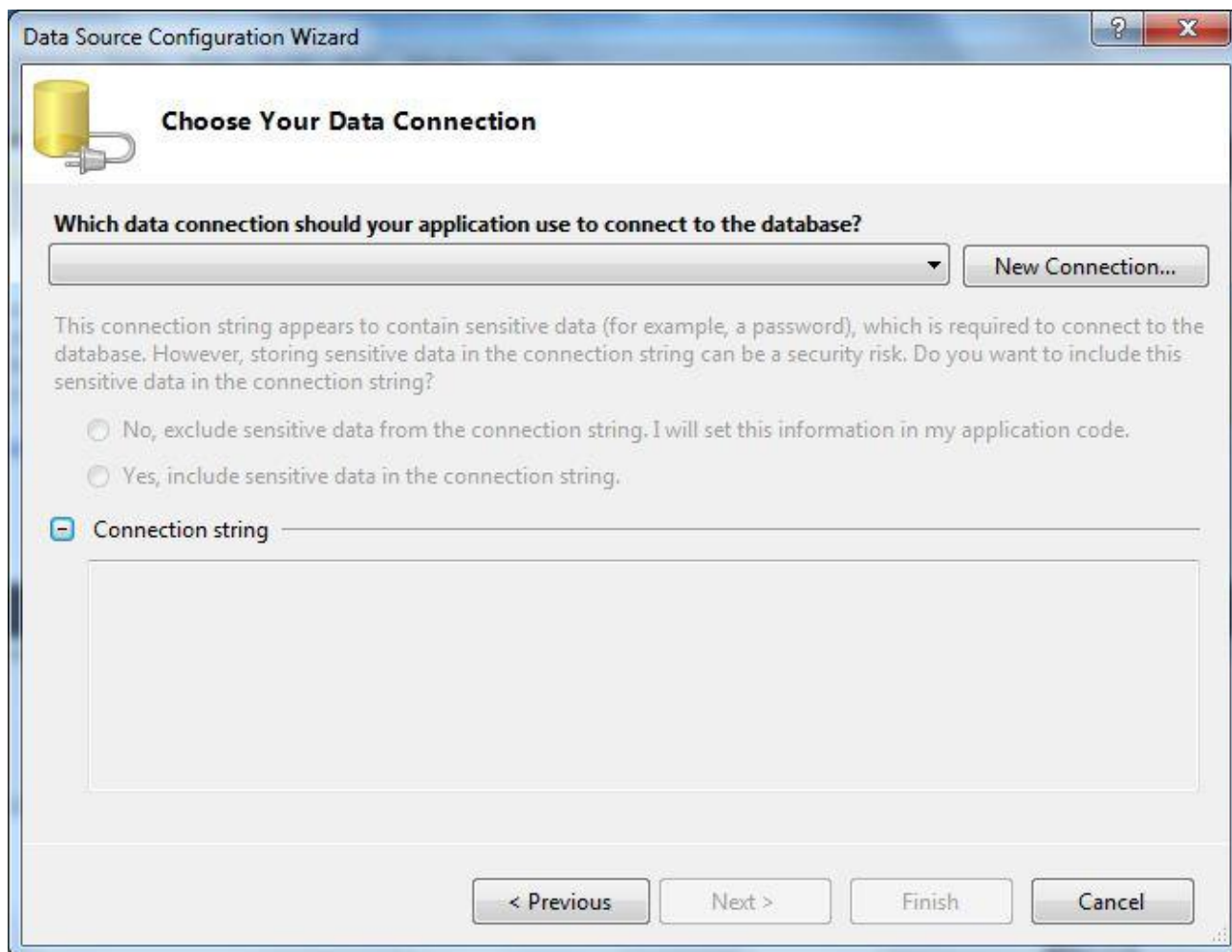
Escolher criar uma nova ligação:



E seguir os passos de escolha de uma origem:







Data Source Configuration Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

New Connection...

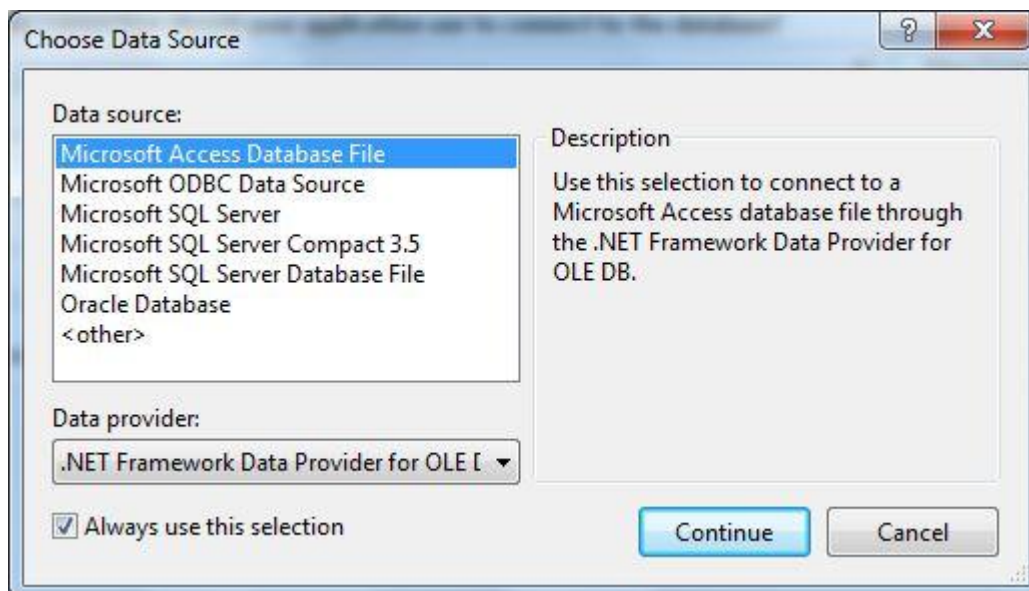
This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set this information in my application code.

☐ Yes, include sensitive data in the connection string.

☒ **Connection string**

< Previous **Next >** **Finish** **Cancel**



Choose Data Source

Data source:

- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Compact 3.5
- Microsoft SQL Server Database File
- Oracle Database
- <other>

Description

Use this selection to connect to a Microsoft Access database file through the .NET Framework Data Provider for OLE DB.

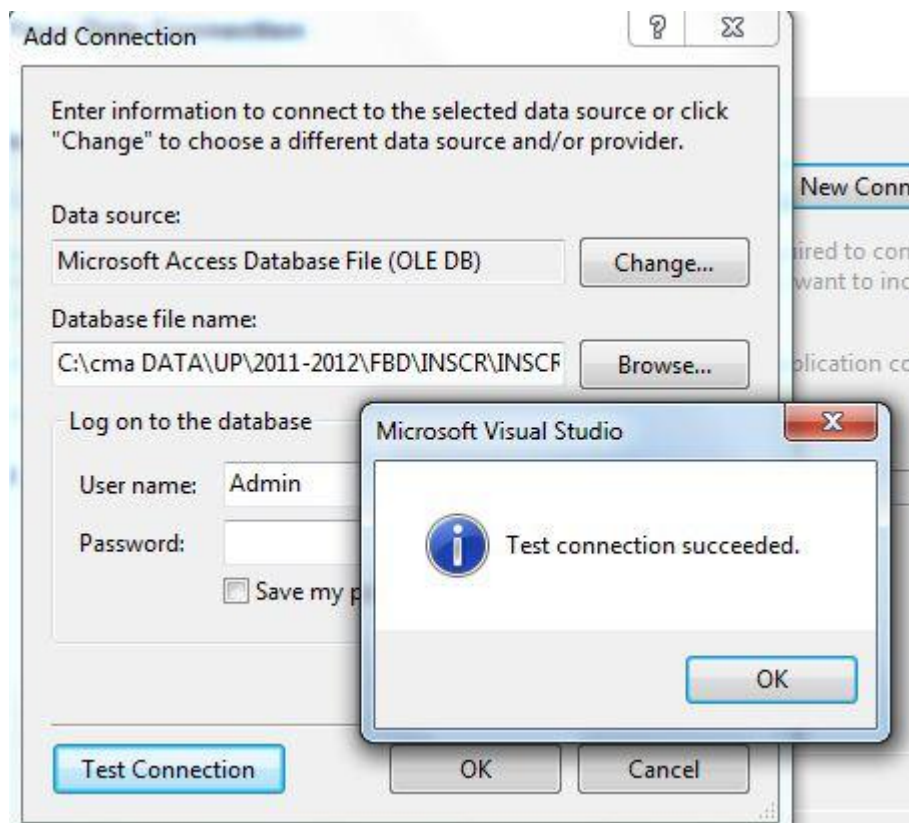
Data provider:

.NET Framework Data Provider for OLE DB

☒ Always use this selection

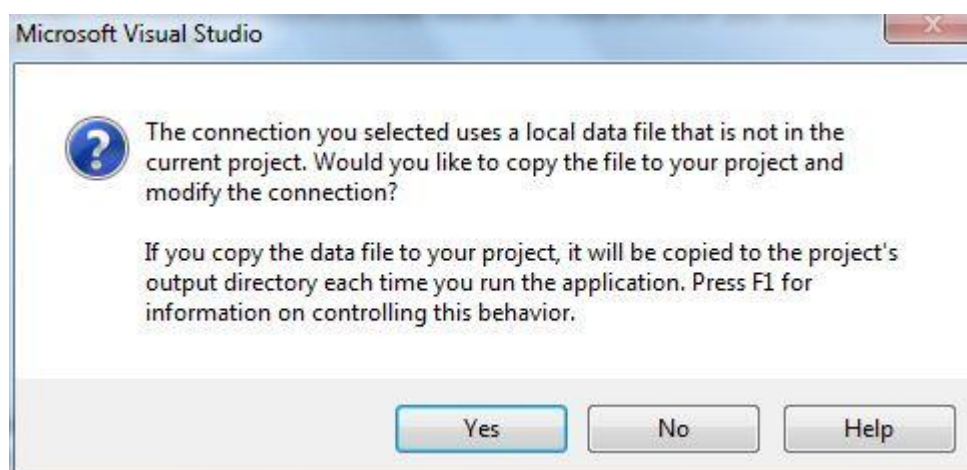
Continue **Cancel**

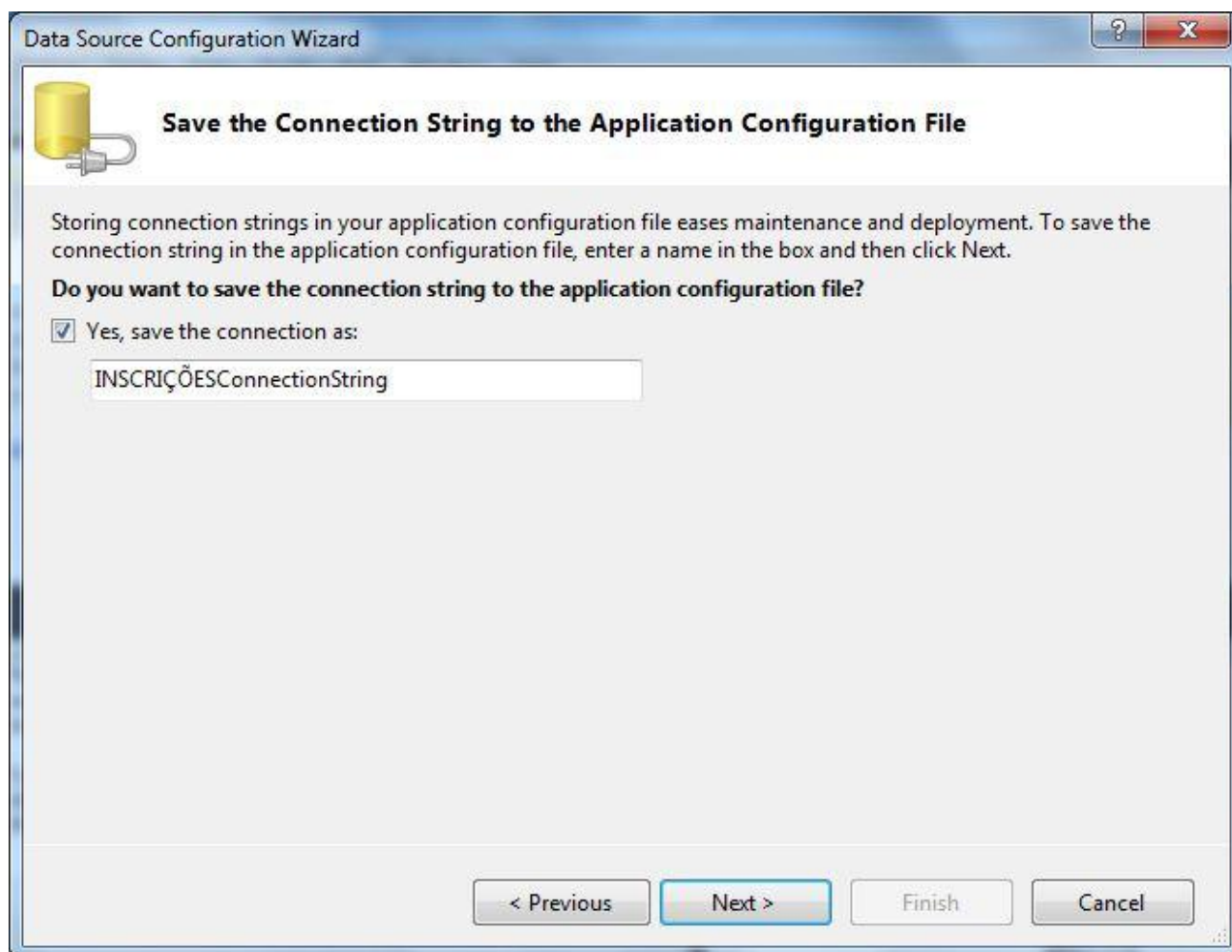
Aparece um quadro onde através do botão [Browse] se pode escolher a base de dados e com o botão [Test Connection] a ligação será testada com o aparecimento de uma janela de confirmação.



Com dois cliques no botão [OK] para fechar a janela de sucesso da ligação e outro na conexão que foi estabelecida volta-se à janela onde se vê a "Connection String" para a ligação pretendida.

No quadro que aparece a seguir será necessário efetuar um clique em [No] para que a base de dados usada pelo programa seja a que foi escolhida e não uma cópia.





No quadro a seguir deverão escolher-se as tabelas que pretendemos e que ficarão armazenadas num DataSet local ao programa:

