

Utilizando Recursos de Paginação

O QUE É PAGINAÇÃO?

Quando se faz uma consulta a uma ou mais tabelas para mostrar seu conteúdo na tela, pode-se obter um número de registros muito grande, de forma que o usuário não se sentirá muito confortável ao navegar por estes registros uma vez que ele terá que usar a barra de rolagem para "subir" ou "descer" a página já que não consegue visualizá-los em uma única tela.

Para solucionar este pequeno problema, utiliza-se o recurso de **Paginação**, onde os registros são colocados em várias páginas, e estas páginas mostrarão sempre um número limitado de registros.

Se uma página tem tamanho 20, isto significa que serão mostrados 20 registros em cada página. Assim, se uma consulta retornar 100 registros, existirão 5 páginas com 20 registros cada. A primeira página mostrará os registros de número 1 até 20, a segunda página mostrará os registros de número 21 até 40 e assim por diante. Cada página terá um botão ou um link para a próxima página e a página anterior.

O QUE É NECESSÁRIO PARA UTILIZAR O RECURSO DE PAGINAÇÃO?

A maior parte dos recursos utilizados para implementar o sistema de paginação serão **propriedades do RecordSet**.

A primeira consideração importante é que **o cursor do objeto Connection deve residir no cliente**. Isto equivale a dizer que a propriedade **CursorLocation** do objeto connection é igual a adUseClient, ou se não quiser utilizar as constantes do arquivo **adovbs.inc**, use o número 3 ao invés de adUseClient. Na verdade, a constante adUseClient tem o valor 3 no arquivo adovbs.inc. Logo, tanto faz usar adUseClient ou 3, pois adUseClient = 3.

CONCLUSÃO: CASO QUEIRA USAR A CONSTANTE adUseClient, VOCÊ DEVERÁ COLOCAR A DIRETIVA <!-- # INCLUDE FILE = "ADOVBS.INC" --> NO INÍCIO DA SUA PÁGINA. PARA MAIORES INFORMAÇÕES SOBRE O INCLUDE, LEIA O ARTIGO [Utilizando a diretiva Include \(e poupando trabalho \)](http://www.aspbrasil.com.br/ver_tutorial.asp?tutorial=42) DO ASPBRASIL.

Se você não compreendeu o porquê dos pré-requisitos do objeto Connection para a utilização dos recursos de paginação, não se preocupe. Por hora, apenas aceite que deverá ser feito desta maneira e siga em frente procurando entender o resto do tutorial. Está sendo preparada uma série de tutoriais que abordam os fundamentos dos objetos RecordSet e Connection que breve estarão disponíveis no ASPBrasil. Esta parte teórica é indispensável para o aprendizado da linguagem e para a formação de um bom programador.

Para criar o objeto Connection, definir a Propriedade CursorLocation e abrí-lo, use o seguinte código:

```
<%
ConnectString = "dsn=nome_odbc;uid=usuario;pwd=senha"
Set Conn = server.CreateObject("ADODB.Connection")
Conn.CursorLocation = 3 ' adUseClient
Conn.Open ConnectString
%>
```

O próximo passo é criar o objeto RecordSet e definir a propriedade **PageSize**, que é a propriedade que diz **quantos registros conterà cada página** e informar que deve ser mostrada a primeira página de registros através da propriedade **AbsolutePage**. É aconselhável setar o valor da propriedade **CacheSize** para o mesmo tamanho da página para otimizar a performance do processo.

```
<%
Set rstTemp = server.CreateObject("ADODB.Recordset")
rstTemp.PageSize = 20 ' define o tamanho da "página"
rstTemp.CacheSize = 20 ' define o tamanho do cache com o mesmo tamanho da
página
rstTemp.AbsolutePage = 1 ' informa que a primeira página de registros será
mostrada
%>
```

PRÓXIMOS PASSOS

A primeira etapa está pronta. Já foi definido o tipo de cursor do objeto Connection e o tamanho da página do RecordSet. Agora será preciso um pouco de código para controlar qual página deverá ser exibida na tela.

Todavia, antes da implementação do código, deve-se entender como funciona todo o mecanismo ASP quando se utiliza paginação.

A primeira coisa a se levar em conta é que a cada vez que o usuário requisitar uma nova página de registros (navegar para a próxima página ou para a página anterior através de um link ou um botão), a mesma página .ASP será recarregada (atualizada) , apenas tendo o valor do **AbsolutePage** alterado para *AbsolutePage + 1* (se o usuário clicar em "Próxima") ou *AbsolutePage - 1* (se o usuário clicar em "Anterior"). Isto faz com que a mesma página .ASP seja recarregada no browser, mas com diferentes **páginas de registros**.

Como a página será recarregada várias vezes conforme o usuário navega entre os registros, é conveniente criar uma variável de sessão para guardar qual a página de registros deve ser exibida (**session("Pagina")**) e uma variável para saber se é a primeira vez ou não que se está carregando a página (**session("PrimeiraVez")**). A cada vez que a página for recarregada, o valor de session("Pagina") será lido e o ponteiro do RecordSet irá para a página que tiver o mesmo número que este valor. Se for a primeira vez que a página está sendo carregada, o ponteiro irá para a primeira página de registros (setando a propriedade AbsolutePage com o valor 1).

Agora que já foi dada uma base teórica de como as coisas funcionam, é interessante ler primeiro o código inteiro para depois analisá-lo em partes. Esta é a forma mais fácil para se entender todo o procedimento.

```

<%
ConnectString= "dsn=banco;uid=usuário;pwd=senha"
If Session("PrimeiraVez") <> "Nao" then ' Se é a primeira vez que a página é
carregada
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.CursorLocation = 3 ' é o mesmo que adUseClient
Conn.Open ConnectString
Set rstTemp = Server.CreateObject("ADODB.Recordset")
rstTemp.CacheSize = 20 ' tamanho do cache
rstTemp.PageSize = 20 ' tamanho da página de registros
sql = "Select * from tabela"
rstTemp.Open Sql, Conn
session("Pagina") = 1
MostraDados
Session("PrimeiraVez") = "Nao"
Else ' Se a página já foi carregada, verifica se deve incrementar ou decrementar o
contador de páginas

```

```

if Request("Navegacao") = "Proxima" then
Session("Pagina") = Session("Pagina") + 1
Else
If Request("Navegacao") = "Anterior" then
Session("Pagina") = Session("Pagina") - 1
End If
End If
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.CursorLocation = 3 ' adUseClient
Conn.Open ConnectString
Set rstTemp = Server.CreateObject("ADODB.Recordset")
rstTemp.CacheSize = 20
rstTemp.PageSize = 20
sql = "Select * from tabela"
rstTemp.Open Sql, Conn
MostraDados
End If

```

```

Sub MostraDados()
Dim Contador
rstTemp.AbsolutePage = Session("Pagina") ' vai para o número da página que está
armazenado em session("pagina")
Response.Write "Existem " & rstTemp.RecordCount & " registros na tabela - Mostrando
página " & Session("Pagina") & " de " & rstTemp.PageCount
If Session("Pagina") <> 1 then
response.write "<a href=""Paginacao.asp?Navegacao=Anterior""> [ Anterior ] "
End If
If Session("Pagina") <> rstTemp.PageCount then
response.write "<a href=""Paginacao.asp?Navegacao=Proxima""> [ Proxima ] "
End If
response.write "
"
For contador = 1 to 20

```

```
Response.Write rstTemp("campo") & "  
"  
rstTemp.MoveNext  
If rstTemp.Eof then Exit For  
Next  
End Sub  
%>
```

A primeira linha importante do código é

```
If Session("PrimeiraVez") <> "Nao" then
```

Esta linha verifica se Session("PrimeiraVez") possui o conteúdo "Não" para saber se é a primeira vez ou não que a página está sendo carregada. Quando a página for carregada pela primeira vez, Session("PrimeiraVez") ainda não existirá, então a condição **Session("PrimeiraVez") <> "Nao"** obrigatoriamente será verdadeira, o que fará com que o ASP processe as linhas imediatamente abaixo ao If até encontrar o Else.

As próximas linhas definem os objetos RecordSet e Connection. **Note que as propriedades CacheSize e PageSize foram setadas com o mesmo valor.** A única propriedade que realmente precisa ser setada é PageSize, que vai dizer quantos registros aparecerão na página, mas por razões de desempenho, recomendo atribuir o mesmo valor de PageSize à CacheSize. PARA MAIORES INFORMAÇÕES LEIA O ARTIGO [Otimizando o desempenho de suas consultas \(propriedade CacheSize \)](http://www.aspbrasil.com.br/ver_tutorial.asp?tutorial=41) <http://www.aspbrasil.com.br/ver_tutorial.asp?tutorial=41> .

Depois disso, Session("PrimeiraVez") tem seu valor setado para "Nao". Esta é uma das chaves do funcionamento deste código, pois quando o usuário clicar no link que vai para a próxima página, o browser recarregará a página e verificará que ela já foi carregada uma vez e saberá que deverá desviar o processamento para a porção Else do If.

Na porção Else, o procedimento é semelhante. A diferença é que precisamos verificar o valor da variável "navegação" que tem seu valor preenchido quando o usuário clicar em um dos links "Anterior" ou "Próxima", que nos permitirá saber se devemos incrementar ou diminuir o valor da propriedade AbsolutePage.