

DICAS ASP 2

1.) ENVIO DE E-MAILS

Para este exemplo são utilizados dois ficheiros. Um chamado email.html que vai conter o formulário de envio. O outro chamado email.asp que vai conter o código ASP para o envio.

Para este exemplo utilizamos o componente ASPMail da [ServerObjects](#)

email.html

```
<form action="email.asp" name="formulario" id="formulario" method="GET">
De (e-mail):
<input type="text" name="FromAddress" size="35">
De (nome):
<input type="text" name="FromName" size="35">
Destinatário (email):
<input type="text" name="ToAddress" size="35">
Destinatário (nome):
<input type="text" name="ToName" size="35">
Assunto:
<input type="text" name="Assunto" size="35">
Mensagem:
<textarea name="Mensagem" rows=5 cols=30></textarea>
<input type="submit" value="Enviar">
</form>
```

email.asp

```
<%
Set Mailer = Server.CreateObject("SMTPsvg.Mailer")

Mailer.RemoteHost = "mail.csgnet.org" 'Servidor que vai enviar a
mensagem

ToAddress=Request.form("ToAddress")
ToName=Request.form("ToName")
FromAddress=Request.form("FromAddress")
FromName=Request.form("FromName")
Assunto=Request.form("Assunto")
Mensagem=vbCrLf & vbCrLf & Request.form("mensagem") & vbCrLf
'----- FIM das Definições iniciais -----
--
Mailer.FromName = FromName 'Nome de quem envia o mail
Mailer.FromAddress = FromAddress 'Endereço de quem envia o mail
Mailer.AddRecipient = ToName, ToAddress 'Endereço para quem se envia o e-
mail
Mailer.Subject = Assunto 'Assunto da mensagem
Mailr.BodyText = Mensagem 'Corpo da mensagem (e-mail)
Mailer.Priority = 1 'Prioridade da Mensagem
```

```
'Ainda temos mais possibilidades de configuração das mensagens
'(consultar a página da ServerObjects
'Eis algumas delas:
```

```
'-----
' - Definir a Carbon Copy addresses
'-----
'Mailer.AddCC "Artur", "xyz@abcxyz.net"
```

```
'-----
'- Definir a Blind Carbon Copy addresses
'-----
'Mailer.AddBCC "Arnal Baganha", "ghi@abcxyz.net"
```

```
'-----
'- Define a flag ReturnReceipt
'-     Se for definido como True (Verdadeiro)
'-     e o servidor de mail suportar esta funcionalidade
'-     (e estiver activada) é enviada
'-     uma mensagem a indicar que o mail
'-     foi entregue
'-----
'Mailer.ReturnReceipt = false
```

```
'-----
'- Define a flag ConfirmReading
'-     Se for definido como True (Verdadeiro)
'-     e o servidor de mail suportar esta funcionalidade
'-     (e estiver activada) é enviada
'-     uma mensagem a indicar que o mail
'-     foi lido
'-----
'Mailer.ConfirmRead = false
```

```
if not Mailer.SendMail then
    Response.Write "<b>O envio falhou. O erro foi:</b> </br>"
    Response.Write Mailer.Response
else
    Response.Write "<b>E-mail enviado com sucesso...</b>"
end if
```

```
Set Mailer=nothing
%>
```

2.) ENVIO DE E-MAILS UTILIZANDO O CDONTS

Para este exemplo são utilizados dois ficheiros. Um chamado email.html que vai conter o formulário de envio. O outro chamado email.asp que vai conter o código ASP para o envio.

Para este exemplo utilizamos o objecto CDONTS da Microsoft.

email.html

```
<form action="email.asp" name="formulario" id="formulario" method="GET">
De (e-mail):
<input type="text" name="FromAddress" size="35">
De (nome):
<input type="text" name="FromName" size="35">
Destinatário (email):
<input type="text" name="ToAddress" size="35">
Destinatário (nome):
<input type="text" name="ToName" size="35">
Assunto:
<input type="text" name="Assunto" size="35">
Mensagem:
<textarea name="Mensagem" rows=5 cols=30></textarea>
<input type="submit" value="Enviar">
</form>
```

email.asp

```
<%
Set Mailer = Server.CreateObject("CDONTS.NewMail")

ToAddress=trim(Request.form("ToAddress"))
ToName=Request.form("ToName")
FromAddress=Request.form("FromAddress")
FromName=Request.form("FromName")&"<"& FromAddress &">"
Assunto=Request.form("Assunto")
Mensagem=vbCrLf & vbCrLf & Request.form("mensagem") & vbCrLf
'----- FIM das Definições iniciais -----
--
Mailer.From = FromName
Mailer.To = ToAddress
Mailer.Subject = Assunto
Mailer.Body = Mensagem
Mailer.Importance = 2
Mailer.Send
if err.number = 0 then
    Response.Write "<p align='center' class='Verdana08branco'>Mensagem
Enviada Para " & ToName & "<" & ToAddress & ">!"
else
    Response.Write "<p align='center' class='Verdana08branco'>Foi
detectado um erro: " & Mailer.Response
End if
```

```
Set Mailer=nothing  
%>
```

3.) CONVERSÃO DE TABELA ACCESS EM ARQUIVO XML

Quando as base de dados são estáticas (não são actualizadas constantemente) é muito mais rápido fazer a listagem de um ficheiro XML do que directamente da tabela ACCESS.

Segundo alguns testes por mim efectuados (sem grande rigor diga-se) o tempo de listagem de um ficheiro XML é 4 vezes inferior à leitura e apresentação de uma tabela ACCESS.

Pode fazer o download deste exemplo [Aqui](#)

ex22.asp

```
<%
Set FS=Server.CreateObject("Scripting.FileSystemObject")
Set TStream=FS.CreateTextFile(Server.MapPath("alunos.xml"))
Set rs=Server.CreateObject("ADODB.Recordset")
cn="driver={Microsoft Access Driver (*.mdb)};"
cn=cn & "dbq=" & Server.MapPath("bdAlunos.mdb")
SQL = "SELECT * FROM alunos"
rs.Open sql,cn
TStream.WriteLine "<?xml version='1.0' encoding='ISO8859-1' ?>"
TStream.WriteLine "<LISTA>"
While not rs.EOF
    TStream.WriteLine "<ALUNO>"
    TStream.WriteLine "<ID>" & rs("id") & "</ID>"
    TStream.WriteLine "<NOME>" & rs("nome") & "</NOME>"
    TStream.WriteLine "<LINK>" & rs("link") & "</LINK>"
    TStream.WriteLine "<MENSAGEM>" & rs("msg") & "</MENSAGEM>"
    TStream.WriteLine "<EMAIL>" & rs("email") & "</EMAIL>"
    TStream.WriteLine "</ALUNO>"
    rs.movenext
Wend
TStream.WriteLine "</LISTA>"
Set FS=nothing
Set TStream=nothing
Set rs=nothing
Response.write "Ficheiro Criado com sucesso....."
%>
```

4.) LER XML COM XSL

Podemos ler o conteúdo de um ficheiro XML recorrendo a a um ficheiro de estilos XSL. A leitura pode ser feita através de um pequeno script em javascript ou através de ASP. Aqui são apresentadas as duas hipoteses e ainda o ficheiro de estilos.

De notar que o ficheiro XSL é ele próprio um ficheiro XML

NOTA: O ficheiro *alunos.xml* foi criado com o exemplo anterior.

listaalunos.xsl

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
  <html>
  <body>
    <table border="1" cellspacing="0" cellpadding="0"
bordercolor="Black">
      <tr>
        <th bgcolor="yellow">id</th>
        <th bgcolor="yellow">Nome</th>
        <th bgcolor="yellow">Link</th>
        <th bgcolor="yellow">Mensagem</th>
        <th bgcolor="yellow">E-Mail</th>
      </tr>
      <xsl:for-each select="LISTA/ALUNO">
        <tr>
          <td><xsl:value-of select="ID"/></td>
          <td><xsl:value-of select="NOME"/></td>
          <td><xsl:value-of select="LINK"/></td>
          <td><xsl:value-of select="MENSAGEM"/></td>
          <td><xsl:value-of select="EMAIL"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

lerXML.html

```
<html>
<body>
<script language="javascript">
// Load XML
var xml = new ActiveXObject("Microsoft.XMLDOM")
xml.async = false
xml.load("alunos.xml")
```

```
// Load the XSL
var xsl = new ActiveXObject("Microsoft.XMLDOM")
xsl.async = false
xsl.load("listaalunos.xsl")

// Transform
document.write(xml.transformNode(xsl))
</script>

</body>
</html>
```

lerXML.asp

```
<%
'Carrega o XML
set xml = Server.CreateObject("Microsoft.XMLDOM")
xml.async = false
xml.load(Server.MapPath("alunos.xml"))

'Carrega o XSL
set xsl = Server.CreateObject("Microsoft.XMLDOM")
xsl.async = false
xsl.load(Server.MapPath("listaalunos.xsl"))

Response.Write(xml.transformNode(xsl))
%>
```