

RSS é o acrônimo para **Really Simple Syndication** - e pode ser considerado uma forma padrão para compartilhar o conteúdo do seu web site.

(Este sistema também é conhecido como Feeds. O termo Feed vem do verbo em inglês "alimentar". Na internet, os "RSS feeds" são listas de atualização de conteúdo de um determinado site.)

No fundo , o RSS é um padrão XML com tags pré-definidas que descreve as informações que você pode compartilhar.

Segundo o Wikipédia - <http://pt.wikipedia.org/wiki/RSS> , temos as seguintes definições e conceitos relacionados com RSS.

O RSS é um descendente do RDF (*Resource Description Framework*) e segue as definições da W3C para RDF, que é descendente do XML (*Extensible Markup Language*) e segue as especificações da W3C para o XML, que por sua vez é descendente do SGML (*Standard Generalized Markup Language*).

No arquivo **RSS** são incluídas informações como título, página (endereço exato de onde há algo novo), descrição da alteração, data, autor, etc... de todas as últimas atualizações do site ao qual ele está agregado. De poucos em poucos minutos o arquivo RSS é atualizado mostrando as atualizações recentes.

O **RSS** é um formato padronizado mundialmente, que funciona com linguagem XML (*Extensible Markup Language*), e é usado para compartilhar conteúdo Web. Ele permite, por exemplo, que o administrador de um site de notícias crie um arquivo XML com as últimas manchetes publicadas, a fim de compartilhá-las mais rapidamente com seus leitores. Este arquivo poderá ser lido através de qualquer ferramenta que seja capaz de entender o formato XML do RSS.

Serve para receber uma lista de atualizações dos sites escolhidos, no momento em que elas ocorram. Muito usado em sites de notícias, sites de previsão do tempo, informações sobre o trânsito, informações econômicas e blogs. O Gmail (*Serviço de Email do Google*) também utiliza **RSS** no seu mecanismo.

Para usar o **RSS**, você precisa de um programa que leia os arquivos XML, criando um índice das atualizações ocorridas em cada site escolhido. Uma das maneiras é baixar gratuitamente um programa chamado agregador de conteúdo, que funciona de forma semelhante à de um leitor de e-mail.

Leitor de Notícias (Rss/XML) desenvolvido pela Technical Informatica (<http://www.nota.com.br>). Pequeno (apenas 705kb), em português e gratuito. Da acesso às manchetes atuais das principais fontes de notícias em português, espanhol e inglês. Novas fontes de notícia são inseridas automaticamente. Permite ao usuário inserir fontes de notícias de sua preferência.
<http://www.nota.com.br/portal/index.php?page=download.php&detalhe=34>

No frigir dos ovos , RSS é apenas um arquivo **XML** com tags definidas que pode ser lido de forma compartilhar informação. Abaixo temos um exemplo de um arquivo RSS padrão:

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>macoratti</title>
    <link>http://www.macoratti.net</link>
    <description>meu primeiro rss</description>
    <language>por-br</language>
    <copyright>Copyright (C) Nacoratti.net 2005</copyright>
    <lastBuildDate>data de hoje</lastBuildDate>
    <webMaster>macoratti@yahoo.com.br</webMaster>
    <ttl>1</ttl>
    <item>
      <title>Criando o RSS com VB.NET</title>
      <description>Este é meu programa para criar um RSS para compartilhar informações de um site.</description>
      <pubDate>novembro , 10 Nov 2005</pubDate>
    </item>
  </channel>
</rss>
```

Dando uma olhada nas **tags** principais temos: *(algumas não foram usadas no exemplo acima.)*

<rss> É a tag raiz e possui um atributo version. A última versão é a 2.0

<channel>	A raiz rss pode conter tags <channel>. Esta tag contém também outras tags como : <title> , <link> e <item>
<title>	representa o título do RSS
<link>	representa a URL do web site fornecedor do rss.
<description>	detalha as informações do rss
<copyright>	informação de copyright
<generator>	Nome do programa ou aplicação que gera o rss.
<rating>	Avaliação do canal
<skipHours>	Tempo em que estará disponível
<author>	E-mail do autor da notícia

Além destas tags pode haver uma ou mais tags **<item>** . Esta tag representa o item atual que você deseja compartilhar.

A esta altura você já deve estar se perguntando : *"Quando esse cara vai falar sobre como gerar um RSS no vb.net"* ?

Na verdade a plataforma .NET possui diversas classes para tratar o formato XML, e , neste artigo eu vou usar a classe **TextWriter** para gerar o nosso **RSS**.

Para poder desenvolver algo que sirva como uma solução genérica e funcione com qualquer web site vamos criar uma classe no VS.NET. A fonte para as nossas tags **<item>** RSS será um DataSet geralmente preenchido a partir de uma tabela de um banco de dados.

Obs: A classe foi adaptada de alguns exemplos que pequisei na web.

Esta classe terá as seguintes propriedades e métodos :

Propriedades :

- **OutputStream** - um objeto stream onde a alimentação será disparada.
- **RssTitulo** - Define o valor da tag <title>
- **editorURL** - representa o a tag <link>
- **Descricao** - representa a tag <description>
- **CopyRight** - define a tag <copyright>
- **Generator** - define a tag <generator>
- **ItemFonte** - define um objeto DataSet que contem os itens.
- **ItemTituloCampo** - nome da DataColumn que representa a tag <title>
- **ItemURLCampo** - DataColumn que representa a tag <link>
- **ItemDescricaoCampo** - DataColumn que representa a tag <description>
- **ItemPublicacaoCampo** - DataColumn que representa a tag <pubDate>

Métodos:

- **publicarRSS** : Método estático que escreve a marcação RSS para o **OutPutStream**.

A seguir temos um código em VB.NET sobre as propriedades e métodos.

Public Class geraRss

'propriedades públicas (o correto seria usar Property)

```
Public OutputStream As Stream
Public RssTitulo As String
Public EditorURL As String
Public Descricao As String
Public Copyright As String
Public Generator As String
Public ItemFonte As DataSet
Public ItemTituloCampo As String
Public ItemURLCampo As String
Public ItemDescricaoCampo As String
Public ItemPublicacaoCampo As String
```

Public Shared Function publicarRss(ByVal r As Rss)

```
Dim writer As New XmlTextWriter(r.OutputStream, System.Text.Encoding.ASCII)
```

```
writer.WriteStartElement("rss")
writer.WriteAttributeString("version", "2.0")
writer.WriteStartElement("channel")
writer.WriteElementString("title", r.RssTitulo)
writer.WriteElementString("link", r.EditorUrl)
writer.WriteElementString("description", r.Descricao)
```

```
writer.WriteString("copyright", r.Copyright)
writer.WriteString("generator", r.Generator)
```

For Each row As DataRow In r.ItemSource.Tables(0).Rows

```
writer.WriteStartElement("item")
writer.WriteString("title", row(r.ItemTituloCampo))
writer.WriteString("link", row(r.ItemUrlCampo))
writer.WriteString("description", row(r.ItemDescricaoCampo))
writer.WriteString("pubDate", CType(row(r.ItemPublicacaoCampo), DateTime).ToString("ddd, dd MMM yyyy"))
writer.WriteEndElement()
```

Next

```
writer.WriteEndElement()
writer.WriteEndElement()
writer.Flush()
```

End Function

End Class

No código acima o método **publicarRss()** é o mais importante.

1. Criamos uma instância da classe [system.XML.XmlTextWriter](#). Esta classe é usada para escrever documentos XML.
2. Passamos o **OutPutStream** e definimos o modo ASCII para o exemplo.

Os métodos da classe **XmlTextWriter** que usamos são :

- **WriteStartElement** - escreve uma tag aberta para a tag definida;
- **WriteAttributeString** - escreve um atributo para a tag atual aberta;
- **WriteElementString** - escreve tag inicial seguida pelo texto e fecha a tag.
- **WriteEndElement** - escreve o fim da tag recentemente aberta.
- **Flush** - libera toda saída do buffer para o destino.

Gerando RSS

A classe foi criada e esta pronta , só falta mostrar como usar , certo ?

Vamos supor que os dados do seu site que você quer compartilhar via **RSS** esteja em uma tabela com a seguinte estrutura:

(Na verdade a estrutura abaixo é a estrutura da tabela artigo do banco de dados jcmsoft.mdb do meu site)

ArtigoID	Int
Artigo	VarChar(255)
Descricao	VarChar(255)
local	VarChar(100)
Data	DateTime

Teremos que criar um código que obtenha o dataset gerado a partir desta tabela. Para fazer isto criamos uma função chamada **geraDataSet()** que acessa a tabela do banco de dados , seleciona todos os registros e preenche um dataset a partir de um objeto DataAdapter.

O código para fazer esta tarefa pode ser o seguinte :

```
Function geraDataSet() as DataSet

Dim cnn as New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\teste\jcmsoft.mdb")

sql = "select * from artigos order by ArtigoID desc"

Dim da As New OleDbDataAdapter(sql,cnn)
Dim ds as New DataSet()

da.Fill(ds,"artigos")

Return ds

End Function
```

Nota: A função acima será agregada a classe Rss.

Juntando todas as partes vamos criar uma aplicação ASP.NET onde na página principal iremos criar uma instância da classe **geraRSS()** e definir as suas propriedades para finalmente chamar o método **geraDataSet()**.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    Dim r As New Rss
    Dim ds As DataSet = geraDataSet()

    r.OutputStream = Response.OutputStream
    r.RssTitulo = "Macoratti.net últimos artigos"
    r.EditorURL = Request.Url.Host
    r.Description = "Macoratti.net - Tudo para VB e VB.NET"
    r.Copyright = "Copyright (C) - www.macoratti.net"
    r.Generator = "Gerador de RSS"
    r.ItemFonte = ds
    r.ItemTituloCampo = "Artigo"
    r.ItemDescricaoCampo = "Descricao"
    r.ItemPublicacaoCampo = "Data"
    r.ItemURLCampo = "local"
```

```
    Response.ContentEncoding = System.Text.Encoding.UTF8
    Response.ContentType = "text/xml"
```

```
    Rss.PublicarRss(r)
```

```
    Response.End()
```

```
End Sub
```

Agora vamos criar um projeto do tipo ASP.NET web Application com o nome de **geraRss1** usando a linguagem VB.NET

A localização do projeto é o diretório virtual padrão **inetpub/wwwroot/geraRss1**. **(para a minha máquina)**

Após o projeto ser iniciado inclua uma classe no menu Project opção **Add Class** e altere o nome do arquivo para **geraRss.vb**

Após isto inclua o código conforme abaixo :

```
Imports System.Xml
```

```
Imports System.io
```

```
Imports System.Data.oledb
```

```
Public Class Rss
```

```
    'propriedades públicas (o correto seria usar Property)
```

```
    Public OutputStream As Stream
```

```
    Public RssTitulo As String
```

```
    Public EditorURL As String
```

```
    Public Descricao As String
```

```
    Public Copyright As String
```

```
    Public Generator As String
```

```
    Public ItemFonte As DataSet
```

```
    Public ItemTituloCampo As String
```

```
    Public ItemURLCampo As String
```

```
    Public ItemDescricaoCampo As String
```

```
    Public ItemPublicacaoCampo As String
```

```
    Public Shared Function publicarRss(ByVal r As Rss)
```

```
        Dim writer As New XmlTextWriter(r.OutputStream, System.Text.Encoding.ASCII)
```

```
        writer.WriteStartElement("rss")
```

```
        writer.WriteAttributeString("version", "2.0")
```

```
        writer.WriteStartElement("channel")
```

```
        writer.WriteElementString("title", r.RssTitulo)
```

```
        writer.WriteElementString("link", r.EditorUrl)
```

```
        writer.WriteElementString("description", r.Descricao)
```

```
        writer.WriteElementString("copyright", r.Copyright)
```

```
        writer.WriteElementString("generator", r.Generator)
```

```
        For Each row As DataRow In r.ItemFonte.Tables(0).Rows
```

```
            writer.WriteStartElement("item")
```

```
            writer.WriteElementString("title", row(r.ItemTituloCampo))
```

```
            writer.WriteElementString("link", row(r.ItemURLCampo))
```

```
            writer.WriteElementString("description", row(r.ItemDescricaoCampo))
```

```
            'writer.WriteElementString("pubDate", CType(row(r.ItemPublicacaoCampo), DateTime).ToString("ddd, dd MMM yyyy")
```

```
            writer.WriteElementString("pubDate", CType(Now, DateTime).ToString("ddd, dd MMM yyyy"))
```

```
            writer.WriteEndElement()
```

```
        Next
```

```
writer.WriteEndElement()
writer.WriteEndElement()
writer.Flush()
```

End Function

Function geraDataSet() As DataSet

```
Dim cnn As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\teste\jcmsoft.mdb")
Dim sql As String = "select * from artigos order by ArtigoID desc"
```

```
Dim da As New OleDbDataAdapter(sql, cnn)
Dim ds As New DataSet
da.Fill(ds, "artigos")
```

```
Return ds
```

End Function

End Class

Nota: Estou gravando a data como sendo a data de hoje.(now) . Temos também um novo método chamado **geraDataSet()**.

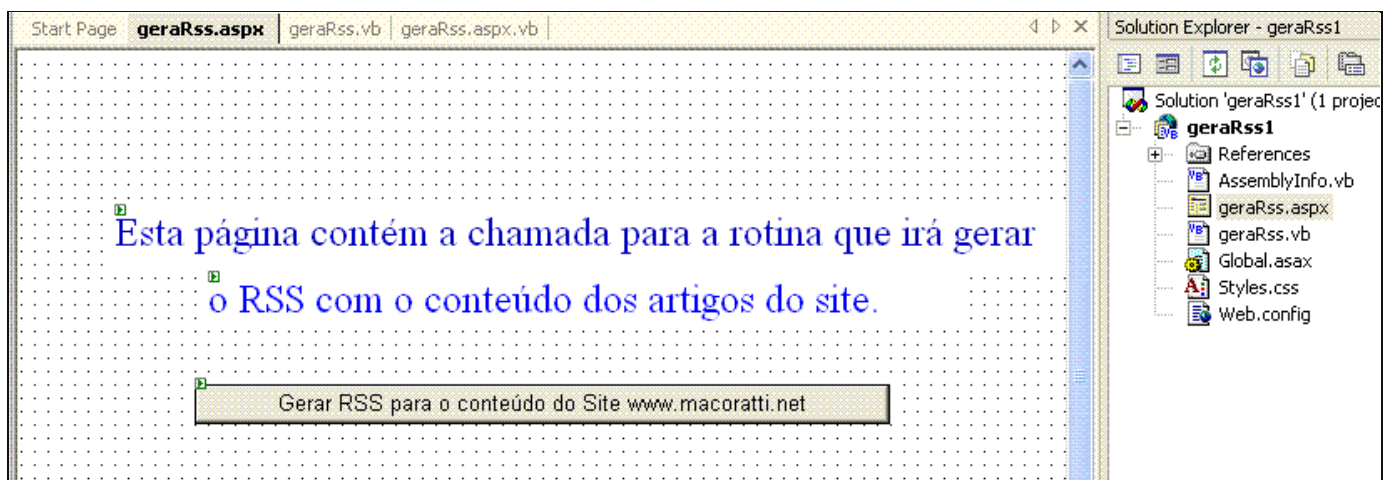
Agora no formulário padrão inclua um botão de comando e inclua o código abaixo no evento **Click**. Altere também o nome do formulário para **geraRss.aspx**

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Dim r As New Rss
Dim ds As DataSet = r.geraDataSet()

r.OutputStream = Response.OutputStream
r.RssTitulo = "Macoratti.net últimos artigos"
r.EditorURL = Request.Url.Host
r.Descricao = "Macoratti.net - Tudo para VB e VB.NET"
r.Copyright = "Copyright (C) - www.macoratti.net"
r.Generator = "Gerador de RSS"
r.ItemFonte = ds
r.ItemTituloCampo = "Artigo"
r.ItemDescricaoCampo = "Descricao"
r.ItemPublicacaoCampo = "Data"
r.ItemURLCampo = "local"
Response.ContentEncoding = System.Text.Encoding.UTF8
Response.ContentType = "text/xml"
Rss.PublicarRss(r)
Response.End()
End Sub
```

Abaixo temos a figura exibindo o layout da pagina os arquivos do projeto:



Executando o projeto em localhost e usando o IIS temos o seguinte resultado:

```
Endereço: http://localhost/geraRss1/geraRss.aspx

- <rss version="2.0">
- <channel>
  <title>Macoratti.net ?ltimos artigos</title>
  <link>localhost</link>
  <description>Macoratti.net - Tudo para VB e VB.NET</description>
  <copyright>Copyright (C) - www.macoratti.net</copyright>
  <generator>Gerador de RSS</generator>
- <item>
  <title>C# - Projetos Open Source e livros gr?tis sobre ASP.NET.</title>
  <link>net_shr1.htm</link>
  <description>Uma rela??o com projetos C# Open Source e dois livros gr?tis sobre ASP.NET e IIS 6.0.</description>
  <pubDate>sex, 11 nov 2005</pubDate>
</item>
- <item>
  <title>VB.NET - Casting : CType ou DirectCast.</title>
  <link>vbn_cast.htm</link>
  <description>Fazendo Casting. CType ou DirectCast ?</description>
  <pubDate>sex, 11 nov 2005</pubDate>
</item>
- <item>
  <title>ASP.NET - Armazenando objetos DataSet na sess?o</title>
  <link>aspn_dss.htm</link>
  <description>Armazenando objetos DataSet na sess?o; recuperando dados da sess?o.</description>
  <pubDate>sex, 11 nov 2005</pubDate>
</item>
```

Pronto ! o seu gerador de **RSS** esta funcionando que é uma beleza. Não esqueça de fazer os ajustes para o seu caso alterando os nomes da tabela e dos campos.

Pegue o projeto completo aqui:  [geraRss1.zip](#)

Eu sei é apenas VB.NET mas eu gosto !!! 💡

José Carlos Macoratti