



www.coldfusionbrasil.com.br

© 2000

VARIÁVEIS CGI	3
CGI.NOME_DA_VARIÁVEL	3
VARIÁVEIS CGI SERVIDOR	3
VARIÁVEIS CGI CLIENTE	3
USANDO AS VARIÁVEIS CGI	4
 REDIRECIONANDO APLICAÇÕES	 4
 INTERAGINDO COM ARQUIVOS	 5
ENVIANDO ARQUIVOS ATRAVÉS DE UM FORMULÁRIO	5
SOLUCIONANDO CONFLITO DE NOMES DE ARQUIVO	6
DEFININDO OS TIPOS DE ARQUIVOS A SEREM ENVIADOS	7
PEGANDO OS RESULTADOS DO ARQUIVO ENVIADO	7
MOVENDO, RENOMEANDO, COPIANDO E DELETANDO ARQUIVOS	7
MOVENDO UM ARQUIVO	7
RENOMEANDO UM ARQUIVO	8
COPIANDO UM ARQUIVO	8
DELETANDO UM ARQUIVO	8
LENDO, ESCRREVENDO E ATUALIZANDO ARQUIVOS TEXTO	8
LER ARQUIVO TEXTO	8
ESCREVENDO UM ARQUIVO TEXTO	9
ATUALIZANDO UM ARQUIVO TEXTO	9
 INTERAGINDO COM DIRETÓRIOS	 9
LISTAR DIRETÓRIO	9
CRIAR DIRETÓRIO	9
DELETAR DIRETÓRIO	10
RENOMEAR DIRETÓRIO	10

Variáveis CGI

Em nosso último módulo vimos como tratar variáveis, vamos iniciar este novo módulo conhecendo mais um tipo de variável tratada pelo Cold Fusion, são as variáveis de ambiente CGI.

Como já sabemos, cada vez que um browser faz uma requisição ao servidor, uma série de variáveis de ambiente são criadas. Algumas pelo servidor, outras pelo próprio browser. Estas variáveis contêm dados sobre a transação entre o browser e o servidor, desde endereço IP, tipo de browser, e autenticação do usuário.

Para criar uma referência a este tipo de variável, iremos fazer da mesma maneira como visto no módulo anterior:

CGI.nome_da_variável

Variáveis CGI servidor

Neste quadro estão as variáveis mais comuns criadas no servidor.

Variável	Descrição
SERVER_SOFTWARE	Apresenta o nome e a versão do software do servidor
SERVER_NAME	Mostra o nome host, apelido DNS, ou endereço IP do servidor
GATEWAY_INTERFACE	A revisão da especificação CGI que o servidor obedece
SERVER_PROTOCOL	O nome e a revisão de informação do protocolo que a requisição chegou
SERVER_PORT	O número da porta que a requisição foi enviada
REQUEST_METHOD	O método o qual a requisição foi criada. Como "GET", "POST", etc
PATH_INFO	O caminho extra que foi dado pelo cliente
SCRIPT_NAME	Um caminho virtual onde o script está sendo executado
QUERY_STRING	Informação de query que segue o ? na URL que referencia este script
REMOTE_HOST	O nome de host que criou a requisição
REMOTE_ADDR	O endereço IP do host que criou a requisição
CONTENT_TYPE	Mostra o tipo de conteúdo de dados
CONTENT_LENGTH	Tamanho do conteúdo que foi passado pelo cliente

Variáveis CGI cliente

Neste quadro estão as variáveis criadas pelo browser e passadas juntas com a requisição

Variável	Descrição
HTTP_REFERER	Mostra o documento referenciado. O documento que ligou para ou enviou dados através de form
HTTP_USER_AGENT	O browser que está sendo usado para enviar a requisição.

Usando as variáveis CGI

Podemos querer identificar os visitantes de nossas aplicações, para isso podemos utilizar as variáveis CGI. Vejamos como:

```
Index.cfm
<cfset IP=#cgi.remote_addr#>
<cfset DeOnde=#cgi.http_referer#>
<cfset Browser=#CGI.HTTP_USER_AGENT#>
<cfoutput>
IP: #IP#<br>
De onde veio: #DeOnde#<br>
Browser: #Browser#
</cfoutput>
```

Redirecionando Aplicações

Podemos redirecionar uma aplicação para outra aplicação ou uma URL qualquer usamos a tag CFLOCATION. Esta tag é muito útil quando queremos passar para outra aplicação automaticamente ou enviar um usuário para outra URL através de parâmetros dinâmicos.

Sintaxe:

```
<CFLOCATION URL="endereço">
```

Onde:

Endereço – É o nome da aplicação ou URL para onde queremos redirecionar o usuário.

Como vimos acima que podemos identificar o usuário de nossa aplicação através de seu IP, endereço de onde veio e o browser que utiliza, montaremos agora algumas aplicações que de acordo com esses dados irão redireciona-lo para outras seções do site.

```
Redireciona1.cfm
<CFSET IP=#cgi.remote_addr#>

<CFIF #Browser# IS '200.210.177.34'>
  <CFLOCATION URL="IP20021017734.htm">
<CFELSEIF #Browser# IS '127.0.0.1'>
  <CFLOCATION URL="IP127001.htm">
</CFIF>
```

Redireciona2.cfm

```
<CFSET DeOnde=#cgi.http_referer#>
```

```
<CFIF #DeOnde# IS 'http://www.coldfusionbrasil.com.br/'>
```

```
  <CFLOCATION URL="usuariosCFBrasil.htm">
```

```
<CFELSEIF #DeOnde# IS 'http://www.uol.com.br/'>
```

```
  <CFLOCATION URL="usuariosUol.htm">
```

```
</CFIF>
```

Redireciona3.cfm

```
<CFSET Browser=#CGI.HTTP_USER_AGENT#>
```

```
<CFIF #Browser# IS 'Mozilla/4.08 [en] (Win98; I)'>
```

```
  <CFLOCATION URL="usuariosNetscape.htm">
```

```
<CFELSEIF #Browser# IS 'Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)'>
```

```
  <CFLOCATION URL="usuariosExplorer.htm">
```

```
</CFIF>
```

Interagindo com arquivos

Já vimos que o Cold Fusion nos permite interagir de muitas formas com o usuário, mas como trabalhar com arquivos? Estaremos vendo agora exatamente como o Cold Fusion trata arquivos.

A tag responsável por este tipo de interação é a <CFFILE>, com ela você poderá enviar arquivos (upload) para o servidor, mover, renomear, copiar e deletar arquivos, e ler, escrever e atualizar arquivos texto.

Enviando arquivos através de um formulário

Como montar um simples formulário HTML já foi mostrado em um de nossos módulos, agora iremos desenvolver um formulário mais avançado, com a possibilidade de enviar arquivos da máquina cliente para o servidor. Para fazer isto, usaremos a tag *INPUT* da seguinte forma:

```
<INPUT TYPE="file" NAME="arquivo">
```

EnviaArquivo.cfm

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Enviando arquivos</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM ACTION="RecebeArquivo.cfm" ENCTYPE="multipart/form-data"
METHOD="post">
```

```
  <INPUT TYPE="file" NAME="arquivo">
```

```
  <INPUT TYPE="submit" VALUE="Enviar arquivo">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

* Note que o atributo *ENCTYPE="multipart/form-data"* é necessário para enviar arquivos através do formulário.

Criado o formulário, veremos agora como tratar isto do lado da aplicação cfm. Quando o arquivo é enviado através do formulário, ele ainda não foi salvo no servidor, foi apenas codificado e enviado como dado de formulário. É a tag **CFFILE** que é responsável pela decodificação deste dado e então salvar no servidor.

Para isso é necessário alguns parâmetros, vejamos quais são:

ACTION	que identifica o tipo de ação que a tag deve fazer. Neste caso deve ser UPLOAD
FILEFIELD	nome do campo dado no formulário HTML
DESTINATION	local onde o arquivo será salvo no servidor

Portanto, a aplicação ficará assim:

```
RecebeArquivo.cfm
<CFFILE ACTION="UPLOAD"
        FILEFIELD="arquivo"
        DESTINATION="C:\Uploads\">

<CFOUTPUT>Arquivo enviado com sucesso!</CFOUTPUT>
```

* Não use escopas (#) no parâmetro **FILEFIELD**

Solucionando conflito de nomes de arquivo

Muitas vezes ao enviarmos algum arquivo, podemos nos deparar com a existência de um outro com o mesmo nome. O Cold Fusion está preparado para tratar estes tipos de ocorrências, para isto você deve usar o parâmetro **NAMECONFLICT**.

O parâmetro **NAMECONFLICT** pode ter os seguinte valores:

ERROR	Valor padrão. O arquivo não será salvo, e o Cold Fusion irá retornar uma tela de erro.
SKIP	Não salva o arquivo nem mostra erro. Apenas ignora o arquivo.
OVERWRITE	Substitui o arquivo existente pelo que está sendo enviado.
MAKEUNIQUE	Salva o arquivo com um nome diferente, que é armazenado na variável "ServerFile".

Vejamos agora como fica nossa aplicação:

```
RecebeArquivo.cfm
<CFFILE ACTION="UPLOAD"
        FILEFIELD="arquivo"
        DESTINATION="C:\Uploads\"
        NAMECONFLICT="makeunique">

<CFOUTPUT>Arquivo enviado com sucesso!</CFOUTPUT>
```

Definindo os tipos de arquivos a serem enviados

Quando enviamos arquivos, poderemos querer restringir seus tipos em determinadas aplicações. Por exemplo, podemos querer evitar o envio de arquivos executáveis para não ocorrer nenhum tipo de invasão.

O parâmetro que nos permite fazer este tipo de restrição é o ACCEPT, só será permitido o envio de arquivos com o tipo especificado. Quando utilizado, o tipo de conteúdo MIME do arquivo deve estar compatível com o critério de restrição especificado, caso contrário será gerado um erro. Os tipos MIME estão registrados em seu browser.

Vamos colocar agora em nossa aplicação o parâmetro ACCEPT para aceitar apenas o envio de arquivos html:

```
RecebeArquivo.cfm
<CFFILE ACTION="UPLOAD"
        FILEFIELD="arquivo"
        DESTINATION="C:\Uploads\"
        NAMECONFLICT="makeunique"
        ACCEPT="text/html">

<CFOUTPUT>Arquivo enviado com sucesso!</CFOUTPUT>
```

Pegando os resultados do arquivo enviado

Logo após o envio de um arquivo, podemos querer saber algumas informações sobre este arquivo enviado para uso posterior. As informações geradas podem ser o nome do arquivo, o diretório onde foi salvo, entre outros.

Veremos agora alguns dos mais importantes:

ContentType	Tipo do conteúdo MIME do arquivo enviado
FileSize	Tamanho do arquivo enviado
ServerDirectory	Diretório no servidor onde o arquivo enviado foi salvo
ServerFile	Nome do arquivo salvo no servidor
ServerFileExt	Extensão do arquivo enviado, exemplo txt e não .txt
ServerFileName	Nome do arquivo sem a extensão
TimeLastModified	Data e hora da última modificação do arquivo

Para fazer referência a estas informações utilize "File" antes, exemplo "File.ServerFile".

Movendo, Renomeando, Copiando e Deletando arquivos

Agora que já sabemos como enviar um arquivo para o servidor, iremos ver como utilizar a tag CFFILE para gerenciar arquivos dentro do servidor.

Movendo um arquivo

Iremos mover o arquivo *exemplo.html* de c:\uploads\ para c:\registro\

```
MoveArquivo.cfm
<CFFILE ACTION="MOVE"
        SOURCE="c:\uploads\exemplo.html"
        DESTINATION="c:\registro\">
```

Renomeando um arquivo

Agora vamos renomear o arquivo exemplo.html para *final.html*.

```
RenomeiaArquivo.cfm
<CFFILE ACTION="RENAME"
        SOURCE="c:\registro\exemplo.html"
        DESTINATION="c:\registro\final.html">
```

Copiando um arquivo

Vamos copiar o arquivo final.html para um outro diretório no servidor.

```
CopiaArquivo.cfm
<CFFILE ACTION="COPY"
        SOURCE="c:\registro\final.html"
        DESTINATION="c:\backup\">
```

Deletando um arquivo

Iremos agora apagar o arquivo exemplo.html do diretório uploads. Note que agora a tag CFFILE utiliza o parâmetro FILE.

```
DeletaArquivo.cfm
<CFFILE ACTION="DELETE"
        FILE="c:\uploads\exemplo.html">
```

Lendo, Escrevendo e Atualizando arquivos texto

Ainda gerenciando arquivos no servidor, podemos utilizar a tag CFFILE para trabalhar com arquivos texto, isto pode ser muito útil para criar arquivos de log, ou até mesmo arquivos html.

Ler arquivo texto

Para lermos um arquivo é necessário que este exista no diretório. O arquivo lido tem seu conteúdo armazenado em uma variável, que pode ser utilizada para mostrar o conteúdo do arquivo na tela ou inserir em um banco de dados. Vejamos como utilizar:

```
<CFFILE ACTION="READ"
        FILE="c:\guestbook\mensagens.htm"
        VARIABLE="mensagens">
```

O conteúdo do arquivo *mensagens.htm* do diretório *guestbook* foi armazenado na variável *exemplo*. Para mostrar seu resultado na tela basta usar a tag CFOUTPUT.

```
<CFOUTPUT>#mensagens#</CFOUTPUT>
```


Escrevendo um arquivo texto

Podemos utilizar a CFFILE para escrever um texto padrão, ou um arquivo com conteúdo dinâmico com dados enviados através de um formulário.

```
<CFFILE ACTION="WRITE"
        FILE="c:\guestbook\mensagens.htm"
        OUTPUT="Nome: #Form.Nome#
        Mensagem: #Form.Mensagem#">
```

O arquivo *mensagens.htm* é criado com os dados enviados pelo usuário, que foram passados através do parâmetro OUTPUT.

Atualizando um arquivo texto

Com a CFFILE podemos ainda incluir mais texto ao final do arquivo.

```
<CFFILE ACTION="APPEND"
        FILE="c:\guestbook\mensagens.htm"
        OUTPUT="Nome: #Form.Nome#
        Mensagem: #Form.Mensagem#">
```

Interagindo com diretórios

O Cold Fusion nos permite além de interagir com arquivos, interagir com diretórios também. Você vai poder listar, criar, deletar e renomear diretórios.

Listar diretório

Você pode listar um diretório com ordem, ou somente com os tipos de arquivos desejados.

No exemplo abaixo, iremos listar o diretório *updates*, e somente serão mostrados arquivos com extensão *.htm* em ordem alfabética

MostraDiretorio.cfm

```
<CFDIRECTORY ACTION="LIST"
        DIRECTORY="c:\uploads"
        NAME="uploads"
        FILTER="*.htm"
        SORT="name ASC">
```

```
<cfloop query=uploads>
<CFOUTPUT>#name#<br></CFOUTPUT>
</cfloop>
```

Criar diretório

Para criar um diretório a partir da tag CFDIRECTORY usaremos:

```
<CFDIRECTORY ACTION="CREATE"
        DIRECTORY="c:\uploads">
```

Deletar diretório

Para deletar um diretório a partir da tag CFDIRECTORY usaremos:

```
<CFDIRECTORY ACTION="DELETE"  
              DIRECTORY="c:\uploads">
```

** Para que você possa deletar um diretório, é necessário que o mesmo esteja vazio*

Renomear diretório

Para renomear um diretório a partir da tag CFDIRECTORY usaremos:

```
<CFDIRECTORY ACTION="RENAME"  
              DIRECTORY="c:\uploads"  
              NEWDIRECTORY="c:\updates">
```

Note que renomeamos o diretório de uploads para *updates*.