

CGI Realmente Fácil

ou: Escrevendo scripts CGI para processar forms na Web

[Ir para as Notas de Rodapé](#)

[Ir para Outros Tutoriais](#)

[In English](#) (em inglês)

[En español](#) (em espanhol), traduzido por [René Alvarez](#)

[Auf deutsch](#) (em alemão), traduzido por [Friedemann Wachsmuth](#)

[In het Nederlands](#) (em dutch), traduzido por [Simon Amstel](#)

São 16:00 horas, seu chefe precisa de um script CGI pronto até as 16:30, e você não sabe nem o que significa CGI. Você veio ao lugar certo.

Não é difícil, apesar de qualquer papo intimidador que você possa ter ouvido. Se você consegue ler de STDIN e escrever em STDOUT, você pode escrever scripts CGI. Se você já é um programador, esta cartilha pode ensinar-lhe o básico sobre CGI em uns poucos minutos. *Se você não é um programador, esta cartilha não irá ajudá-lo muito -- lamento. Aprenda um pouco de programação, até mesmo em shell, e volte quando estiver preparado. Boa sorte!*

Esta cartilha se concentra em escrever scripts CGI para processar forms em HTML na Web. Ela deixa de fora alguns detalhes, mas pode fazê-lo ganhar velocidade *rapidamente* (literalmente uma hora ou menos), e cobre 90% das situações do mundo real. Quando você sentir necessidade, consulte a [especificação completa da CGI](#). Se precisar de ajuda para escrever forms em HTML, veja [este fragmento de tutorial](#) ou [esta documentação na NCSA](#).

Depois de ler esta página, veja as [notas de rodapé](#) para exemplos de scripts e outros tópicos.

O que é CGI?

CGI não é uma linguagem. É um protocolo simples que pode ser usado para comunicar forms da Web com seu programa. Um script CGI pode ser escrito em qualquer linguagem que possa ler de STDIN, escrever em STDOUT, e ler variáveis de ambiente, ou seja, virtualmente qualquer linguagem de programação, incluindo C, Perl, ou até mesmo shell scripts.

Estrutura de um Script CGI

Esta é a sequência típica de passos para um script CGI:

1. Leia o input do form do usuário.
2. Faça o que desejar com os dados.
3. Escreva a resposta HTML em STDOUT.

O primeiro e o último passos são descritos abaixo.

Lendo a entrada (input) do Form do Usuário

Quando o usuário submete o form, seu script recebe os dados do form como um conjunto de pares nome-valor. Os nomes são o que você tiver definido nas tags INPUT, SELECT ou TEXTAREA, e os valores são o que quer que o usuário tenha digitado ou selecionado. (Os usuários podem também submeter arquivos com forms, mas esta cartilha não aborda esse assunto.)

Este conjunto de pares nome-valor vem na forma de uma longa cadeia de caracteres, a qual você precisa desmembrar. Não é muito complicado, e existe uma infinidade de rotinas para fazer isso para você. Aqui está [uma em Perl](#), [uma mais simples em Perl](#), ou [uma em C](#). Para um cenário CGI mais elaborado, veja o módulo [CGI.pm](#) em Perl. O [diretório CGI em Yahoo](#) inclui várias; rotinas CGI (e scripts prontos), em diversas linguagens.

Se isso for suficiente para você, pule para a próxima seção. Se preferir fazê-lo você mesmo, ou estiver apenas curioso, aqui está o formato da cadeia longa:

"nome1=valor1&nome2=valor2&nome3=valor3"

Portanto, basta quebrá-la nos caracteres "&" e "=". Depois, faça mais duas coisas com cada nome e valor:

1. Converta todos os caracteres "+" em espaços; e
2. Converta todas as seqüências "%xx" para o caractere cujo valor ASCII é "xx" em hexadecimal. Por exemplo, converta "%3d" para "=".

Isso é necessário porque a cadeia longa original é **URL-encoded**, para permitir a presença de símbolos como "=", "&" e outros na entrada do usuário.

Mas onde você obtém a cadeia longa? Isso depende do método HTTP com que o form foi submetido:

- Para submissões GET, ela está na variável de ambiente **QUERY_STRING**.
- Para submissões POST, leia-a de STDIN. O número exato de bytes a ler está na variável de ambiente **CONTENT_LENGTH**.

(Se você esté tentando imaginar a diferença entre GET and POST, veja a [nota de rodapé sobre isso](#). Resposta curta: POST tem propósito mais genérico, mas GET é bom para forms pequenos.)

Enviando a Resposta de Volta para o Usuário

Primeiro, escreva a linha

```
Content-type: text/html
```

mais uma linha vazia, em STDOUT. Depois disso, escreva sua página de resposta (em HTML) em STDOUT, e ela será enviada para o usuário quando seu script estiver pronto. É só isso.

Sim, você está gerando código HTML na hora. Não é difícil; na verdade, é bem simples. HTML foi concebida para ser simples o bastante a ponto de ser gerada assim.

Se você quiser devolver uma imagem ou outra resposta não-HTML, [veja aqui como fazê-lo](#).

É isso aí. Boa Sorte!

Viu como é fácil? Se você ainda não acredita em mim, vá em frente e escreva um script. Certifique-se de colocar o arquivo no lugar certo em seu servidor, e torne-o executável; veja esta [nota de rodapé](#) para mais dicas.

Antes de escrever scripts demais, aprenda sobre [questões de segurança em CGI](#).

Quando você precisar saber mais sobre CGI, veja a [especificação completa da CGI na NCSA](#). O [W3C](#) também mantém uma [página sobre CGI](#) com links para todos os documentos de que você possa necessitar.

Ah sim, **CGI** significa **Common Gateway Interface** -- em outras palavras, uma forma padrão ("common") de comunicação ("interface") entre diferentes processos ("gateway").

Outras Coisas com que Brincar

Veja a [Página de Notas de Rodapé](#) para o seguinte:

1. [Programas-exemplo em CGI](#)
2. [Script CGI para um Mailer](#)
3. [Segurança com Scripts CGI](#)
4. [Colocando seu Script no Servidor](#)
5. [Enviando um Arquivo Existente como Resposta](#)
6. [Outras Variáveis de Ambiente CGI Úteis](#)
7. [Retornando uma Imagem ou Outra Resposta Não-HTML a partir de um Script CGI](#)
8. [Qual é a diferença entre GET e POST?](#)
9. [Adquirindo Maior Controle com Scripts para Header Não-desmembrado](#)

© 1996-1998 [James Marshall](#)

(comentários são bem-vindos; para perguntas, por favor verifique primeiro a [FAQ](#))

Traduzido em 1999 por [Claudio F. Chagas](#)

Última Modificação: 18 Abril 1998

<http://www.jmarshall.com/easy/cgi/portuguese/>