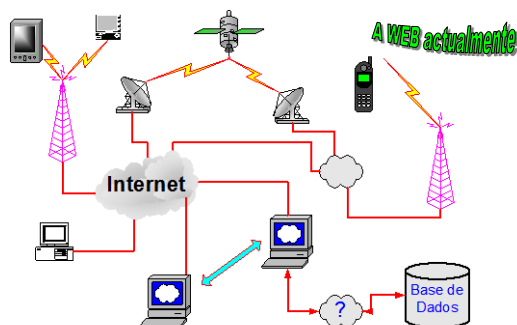




- Introdução
- Necessidade de Gerar Páginas Dinamicamente
- O que é um CGI?
- Passagem de dados (servidor web ↔ CGI)
- Exemplos



Introdução



Introdução

- A WEB oferece um mecanismo simples para publicação de informação estática;
 - Através da criação de páginas HTML que residem no servidor WEB
- As páginas estáticas não permitem interacção contextualizada com o utilizador;
- Existe a necessidade de poder interagir com o servidor WEB de modo a ser possível realizar operações interactivas;
 - Ex: Consulta de email, Banca online, e-commerce, etc...



Necessidade de Gerar Páginas Dinamicamente

- O protocolo HTTP oferece de raiz mecanismos para transferir para o servidor os dados que o utilizador gera;
 - Métodos GET, POST, ...;
- Os servidores WEB têm de processar, dinamicamente, os pedidos efectuados pelos clientes;
- Que fazer com esses dados?



Necessidade de Gerar Páginas Dinamicamente

- Os servidores WEB não sabem, de um modo geral, o que fazer com os dados dinâmicos que recebem;
- É “impossível” ter um servidor que saiba processar todos os dados gerados dinamicamente;



Necessidade de Gerar Páginas Dinamicamente

- Em alternativa os servidores disponibilizam ferramentas para processar os dados;
- O método tradicional mais utilizado é usar o mecanismo *gateway programs*.

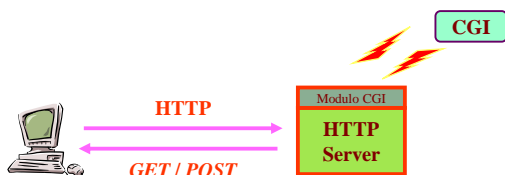


CGI – Common Gateway Interface

- Foi o primeiro mecanismo a surgir que possibilitava a geração dinâmica de informação na WWW;
- As páginas podem assim ser construídas em tempo real em vez de terem que ser previamente guardadas em ficheiros no servidor;
- Consiste na execução de um programa no servidor de WWW desencadeado pelo acesso, de um utilizador, a uma determinada página;



O Conceito de *gateway programs*

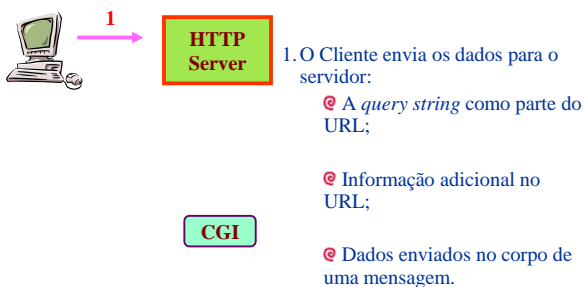


CGI – Modo de funcionamento

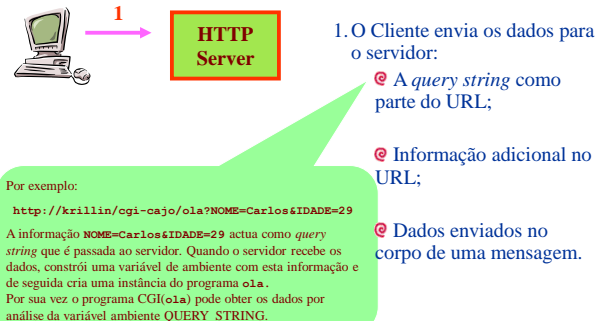
- O servidor WEB lança uma aplicação CGI se for feito um pedido HTTP que a refira;
- Antes de o servidor WEB lançar o CGI define um conjunto de variáveis de ambiente;
 - Ex: `QUERY_STRING`, `REQUEST_METHOD`, etc...
- Redirecciona o `<stdin>` e `<stdout>` de modo ao CGI interagir por omissão com o servidor web;
- O programa CGI lê os dados à sua disposição, interpreta esses dados e envia como resposta uma página HTML;
- O servidor web recebe a página de resposta do CGI e devolve-a na mensagem HTTP de resposta ao pedido feito originalmente pelo browser;



O Conceito de *gateway programs*



O Conceito de *gateway programs*





O Conceito de gateway programs



1

HTTP Server

1. O Cliente envia os dados para o servidor:

• A *query string* como parte do URL;

• Informação adicional no URL;

CGI

Esta informação adicional é referente a *path*. Mais uma vez é passada ao programa CGI através de variáveis ambientais. Um exemplo disso pode ser:

`http://krillin/cgi-cajo/ola/dir/file?NOME=Carlos+Jorge&IDADE=29`

ISEL-DEIETC - Secção de Engenharia de Sistemas

13



O Conceito de gateway programs



1

HTTP Server

1. O Cliente envia os dados para o servidor:

• A *query string* como parte do URL;

• Informação adicional no URL;

• Dados enviados no corpo de uma mensagem.

Em alternativa a passar a informação através do URL, os dados podem ser passados ao programa CGI no corpo de uma mensagem. Este método é usado tipicamente no tratamento de *forms* HTML. Uma das vantagens face à passagem dos dados usando o URL, é que, a dimensão dos dados não fica limitada à dimensão das variáveis ambientais.

ISEL-DEIETC - Secção de Engenharia de Sistemas

14



O Conceito de gateway programs



HTTP Server

2

CGI

2. O servidor envia os dados para o programa CGI:

• Argumentos na linha de comando;

• Variáveis de ambiente;

• Através do *standard input*.

ISEL-DEIETC - Secção de Engenharia de Sistemas

15



O Conceito de gateway programs



HTTP Server

2

CGI

2. O servidor envia os dados para o programa CGI:

• Argumentos na linha de comando;

• Variáveis de ambiente;

• Através do *standard input*.

Os dados são passados ao programa CGI como argumentos na linha de comando. Este mecanismo só é usado quando o pedido teve origem com o método GET usando uma *query* do tipo ISINDEX.

ISEL-DEIETC - Secção de Engenharia de Sistemas

16



O Conceito de gateway programs



HTTP Server

2

CGI

2. O servidor envia os dados para o programa CGI:

• Argumentos na linha de comando;

• Variáveis de ambiente;

• Através do *standard input*.

Os dados a passar ao programa CGI são colocados em variáveis ambientais antes de o programa CGI se executar. É esta a forma utilizada caso seja recebido um comando HTTP GET. Além dos dados que foram enviados pelo browser, são também criadas variáveis de ambiente com informação adicional, tais como:

- directoria raiz do servidor WEB;
- método utilizado no pedido (*REQUEST_METHOD*);
- dados concatenados no URL (*QUERY_STRING*);
- tipo de servidor (*SERVER_SOFTWARE*);
- nome do domínio;
- etc.

ISEL-DEIETC - Secção de Engenharia de Sistemas

17



O Conceito de gateway programs



HTTP Server

2

CGI

2. O servidor envia os dados para o programa CGI:

• Argumentos na linha de comando;

• Variáveis de ambiente;

• Através do *standard input*.

O programa CGI pode ler os dados do *standard input*. É deste modo que os programas CGI conseguem obter os dados que foram enviados pelos clientes usando o método POST. Os servidores criam sempre uma variável de ambiente que indica se os dados foram passados usando o método GET ou POST (*REQUEST_METHOD*).

ISEL-DEIETC - Secção de Engenharia de Sistemas

18



O Conceito de gateway programs



HTTP Server

3

CGI

3. O CGI envia os dados para o servidor através do seu *standard output*:

- Com processamento posterior por parte do servidor Web;
- Sem processamento posterior por parte do servidor Web;



O Conceito de gateway programs



HTTP Server

3. O CGI envia os dados para o servidor através do seu *standard output*:

- Com processamento posterior por parte do servidor Web;
- Sem processamento posterior por parte do servidor Web;

O programa CGI devolve o resultado ao servidor por escrita dos dados no *standard output*. De um modo geral a informação produzida por um CGI é composta por duas partes:

1. Conjunto de directivas para o servidor. Estas directivas ajudam o servidor a compor a informação que vai ser enviada ao cliente (*Cabeçalhos HTTP*).
2. Os dados que representam o pedido que o utilizador efectuou.

Estas duas partes são separadas por uma linha composta por CRLF (*carriage return linefeed*).



O Conceito de gateway programs



HTTP Server

3

3. O CGI envia os dados para o servidor através do seu *standard output*:

- Com processamento posterior por parte do servidor Web;
- Sem processamento posterior por parte do servidor Web;

Existe uma excepção ao método descrito em (1). Se o nome do programa CGI começa por *nph* (*non-parsed header*) o servidor não efectua nenhum pré-processamento sobre os dados e envia-os directamente para o *browser*. É da responsabilidade do programa CGI providenciar a totalidade do cabeçalho da resposta HTTP.



O Conceito de gateway programs



HTTP Server

4

4. O servidor devolve os dados ao cliente que vê o seu pedido satisfeito.

CGI



CGI - Exemplo

Exemplo de um CGI em BASH

```
#!/bin/bash
echo "Content-type: text/html"
echo
echo "<html> <head> <title> Ola - Exemplo de um CGI </title>
</head>"
echo "<body> <h1> Exemplo de um CGI</h1>"
echo "<p>"
date
echo "</p>"
echo "</body> </html>"
```

Data.cgi



Tecnologias para Geração de Páginas Dinamicamente

• CGI (*Common Gateway Interface*);

• FastCGI;

• Server Extensions APIs;

• ISAPI (*Internet Server API*);

• SSI (*Server Side Includes*);

• SSJS (*Server Side JavaScript*);

• ASP (*Active Server Pages*);

• JSP (*Java Server Pages*);

• Servlets;



Geração de Páginas Dinamicamente

Tecnologia	Suporte em várias linguagens	Portável	Interag com o servidor em tempo de execução
CGI	✓	✓ / X	X
FastCGI	✓	✓ / X	X
ASP	✓	X	
JSP	X (Java)	JVM	✓
ISAPI	✓	X	✓
SSJS	X (JavaScript)	✓	
Servlets	X (Java)	JVM	✓



Formulários HTML

(forma de recolher dados para enviar ao servidor)



Formulários HTML

- É uma das opções disponíveis no HTML para fazer a recolha de dados de modo a serem enviados para o servidor;
- Permite especificar se queremos enviar os dados através do método HTTP **GET** ou **POST**;
 - GET** – Os dados são concatenados no fim do URL no pedido **HTTP**.
 - POST** – Os dados são inseridos no corpo da mensagem do pedido **HTTP**.



HTML – Formulários

- Elemento que permite a introdução de dados, pelo utilizador, no browser e envia para o servidor HTTP (ex: inserir palavras num motor de procura, responder a um inquérito)
- <form> ... </form>** – Delimita um formulário
 - Action** – Atributo que indica o URL para onde enviar os dados (CGI, Servlet, jsp, asp, ...)
 - Method** – Atributo que especifica o método de envio dos dados
 - Method= "post" – Indica que é para ser enviada uma mensagem HTTP do tipo **POST** com os dados no corpo da mensagem
 - Method= "get" Indica que é para ser enviada uma mensagem HTTP do tipo **GET** com os dados concatenados no URL
 - Target** – Indica a frame onde será visualizada a resposta
 - Pode ser uma frame específica referindo o seu nome (ex: target="myframe")
 - Pode ser uma frame relativa
 - Target= "_blank" – Nova janela
 - Target= "_self" – Na própria frame
 - Target= "_parent" – No frame pai
 - Target= "_top" – No corpo da própria janela

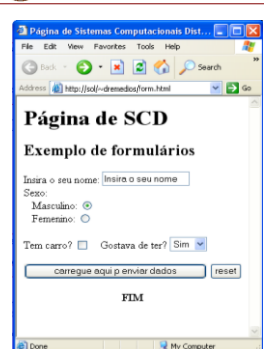


HTML – Formulários (cont)

- Dentro de um formulário podemos ter os elementos:
 - <input>** – Cria uma entrada no formulário.
 - Atributos – accept, align, alt, checked, disabled, maxlength, name, size, readonly, src, type, value
 - Não tem etiqueta de fim
 - Tipos possíveis para o atributo type:
 - button
 - checkbox
 - file
 - hidden
 - image
 - password
 - radio
 - reset
 - submit
 - text
 - <select> ... </select>** – Permite especificar uma drop down list
 - Os seus atributos são:
 - disabled
 - multiple
 - name
 - size
 - <option>** – introduz uma opção na lista, e os seus atributos são:
 - disabled
 - label
 - selected
 - value
 - <textarea> ... </textarea>** – Área em que pode ser introduzido texto (Ex: comentários)
 - name
 - cols
 - rows



HTML – Formulários - Exemplo



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html">
<title>Página de SCD</title>
</head>
<body>
<h1>Página de SCD</h1>
<h2>Exemplo de formulários</h2>
<form method="GET" action="http://phoenix-dremedios/cgi-bin/inspertsia.cgi">
  Insira o seu nome: <input type="text" size="17" name="Nome" value="Insira o seu nome">
  <br>
  Sexo:
  <br>
  <input type="radio" name="sexo" value="masculino" checked="" /> Masculino
  <br>
  <input type="radio" name="sexo" value="feminino" /> Feminino
  <br>
  Tem carro? <input type="checkbox" name="carro" value="carro">
  <br>
  <input type="checkbox" name="carro" value="carro"> Gostava de ter?
  <br>
  <input type="radio" name="sim" value="sim" /> Sim
  <br>
  <input type="radio" name="nao" value="nao" /> Não
  <br>
  <input type="submit" value="carregar aqui p enviar dados">
  <input type="reset" value="reset">
</form>
<p style="text-align: center;"></p>
</body>
</html>
```



```
#!/bin/bash

Content-Type: text/html"
echo "Content-Type: text/html"

#exatar qual o metodo de envio dos dados
if [ $REQUEST_METHOD = "GET" ]
then
    # GET - Lev dados da QUERY_STRING
    DADOS=QUERY_STRING
else
    #post - Lev dados do stdin
    read input from stdin
    read DADOS

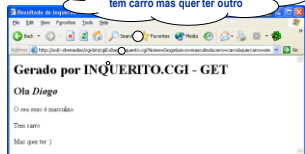
    echo "HEAD"
    echo "Content-Type: text/html"
    echo "Content-Type"

# Vamos parar o resto da pagina
if [ $REQUEST_METHOD = "GET" ]
then
    # GET
    echo "TITLE=Resultado do Imperto - GET/TITLE"
    echo "BODY"
    echo "cli=Gerado por INQUERTO.CGI - GET/CLI"
else
    #post
    echo "TITLE=Resultado do Imperto - POST/TITLE"
    echo "HEAD"
    echo "BODY"
    echo "cli=Gerado por INQUERTO.CGI - POST/CLI"
fi
fi
```

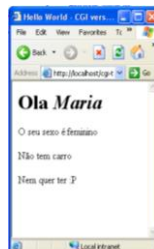
```
# fazer o parsing dos DADOS

# iterar os varios parametros
IFS=" ";
for param in `DADOS`; do
  # echo "param: $param"
  case $param in
    Nome=*)      echo "CH201a <I> echo $param | cut -c6-20 </I> <I>";;
    sexo=*)      echo "cp0 seu sexo seacute; 'echo $param | cut -c6-20 </I>";;
    CARRO=carro) echo "cp0 qual carro/p?";;
    quercar=carro) echo "cp0Ma qual tar 1/p?";;
    quercar=mao) echo "cp0Men qual tar 1P/p?";;
  esac
done
```

Escolhendo no formulário do slide anterior o Nome: Diogo, tem carro mas quer ter outro



Escolhendo no formulário o
Nome: Maria, não tem carro mas
também não quer ter



**Código fonte,
ficheiro**

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include "../Include/html-lib.h"
#include "../Include/cgi-lib.h"
```

O CGI
inquerito.cgi é o
ficheiro binário
que resulta de
compilar
inquerito.c

```
int main() {
    char sname[64];
    llst entries;

    read_cgi_input(&entries);
    html_header("Hello World - CGI version");

    char * username = cgi_valentries, "Name";
    char * sexo = cgi_valentries, "sex";
    char * correo = cgi_valentries, "correo";
    char * carrera = cgi_valentries, "carrera";

    printf("condicao, \"Ola %s<br/>\", username);
    h1(sname);
    printf("cipo seu nome e %s<br/>", sexo);
    if(correio[0] != '\0')
        printf("Nao tem correo: %s", correo);
    if(strlen(carrera, "sim"))
        printf("cipo-Mas quer ter : %s<br/>");
    else
        printf("cipo-Nao quer ter : %s<br/>");
    jelse
        printf("cipo-Tem correo: %s<br/>");
    html_end();
}

return 1;
}
```

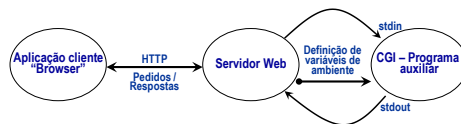


- Os programas CGI podem ser realizados recorrendo a qualquer tipo de linguagem de programação ou de scripting (*BASH, C shell, C, Perl, Tcl*).
- O resultado destes programas pode ser virtualmente em qualquer formato (texto, HTML, POSTSCRIPT, PDF, imagens, etc)
- Desde que o cliente tenha capacidade para o tratar.
- O programa CGI pode saber quais os tipos que pode retornar consultando o cabeçalho HTTP "Accept:..."














-
- Common Gateway Interface (CGI)

- Define uma interface entre o servidor Web e programas que se executem na mesma máquina;
- Caso o pedido ao servidor Web referencie um programa e este tenha permissões para ser executado, este dá início à sua execução;
- A comunicação entre o servidor Web e o programa auxiliar é feita através dos mecanismos oferecidos pelo sistema operativo (stdin, stdout e as variáveis de ambiente);
- O output gerado pelo programa auxiliar é devolvido pelo servidor web para o browser, normalmente corresponde a uma página HTML;



- RFC's:
 - RFC 1945 – HTTP 1.0
 - RFC 2616 – HTTP 1.1
- HTML 4.0 Sourcebook
 - Graham, Ian S., – John Wiley & Sons; 4th edition
- CGI Programming Unleashed;
 - *Eugene Eric Kim*
- Folhas da cadeia
 - Criação e Utilização de CGI's
 - *Nuno Oliveira, Carlos Gonçalves*
- Especificação CGI
 - <http://hoohoo.ncsa.uiuc.edu/cgi/overview.htm>
 - <http://hoohoo.ncsa.uiuc.edu/cgi/examples.htm>



-  WWW – World-Wide Web;
-  HTML – HyperText Markup Language;
-  HTTP – HyperText Transfer Protocol;
-  URL – Uniform Resource Locator;
-  URI – Uniform Resource Identifier;
-  URN – Uniform Resource Name;
-  RFC – Request For Comments;
-  CGI – Common Gateway Interface;
-  API – Application Programming Interface;
-  ASP – Active Server Pages;
-  JSP – JavaServer Pages;