

### O início da interatividade

QXD0193 - Projetos de Interfaces Web

Prof. Bruno Góis Mateus (brunomateus@ufc.br)

#### Agenda

- Introdução
- CGI
- Formulários
- JavaScript

## Introdução

#### Introdução

- Em 1994 o navegador Mosaic já dava suporte ao uso de formulários para envio de dados para o servidor
- Estes formulários foram o mecanismo por trás da primeira geração de aplicativos Web
- Neles, a entrada do usuário é obtida em elementos de formulário
  - O envio do formulário remete essa entrada para o servidor
  - O servidor processa a entrada e gera uma nova página HTML (normalmente com novos elementos de formulário) para exibição pelo cliente

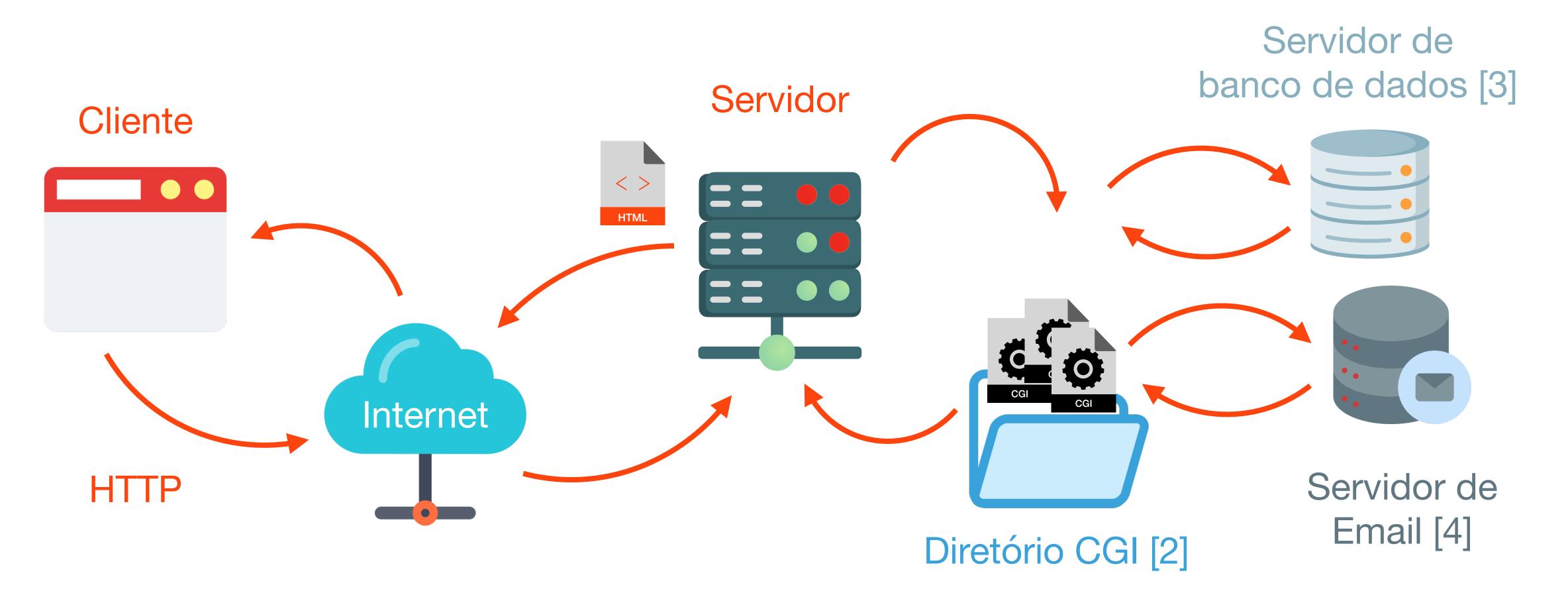
#### Introdução

- Diante do potencial apresentado pela web, o time do National Center for Supercomputing Applications (NCSA) desenvolveu naquela época uma tecnologia que permitiu o uso de páginas dinâmicas
  - Common Gateway Interface (CGI)
- Em 1994, páginas dinâmica poderiam ser servidas via método POST do HTTP
- Além disso o Mosaic, também da NCSA, já era capaz de lidar com formulários HTML
- Um novo mundo se abria naquele momento



- Anos antes da invenção do JavaScript, a especificação do que viríamos a conhecer como CGI começou a sair do papel
- Esta tecnologia foi pioneira na adição de interatividade em páginas web
- Foi inventada em 1993 no NCSA, dona de um dos mais notáveis servidores web da época, o NCSA httpd
  - No futuro o seu código-fonte serviu base para o Apache HTTP Server
- A ideia era possibilitar que um programa pudesse ser iniciado via página web e que seu resultados pudesse ser enviado ao cliente

- Geralmente, o servidor HTTP tem uma pasta, que é designada como uma coleção de arquivos, que podem ser enviados para navegadores da Web
- CGI estende esse sistema, permitindo designar um diretório dentro da coleção de documento contendo scripts executáveis
  - Também conhecido como diretório CGI
  - Por exemplo, /usr/local/apache/htdocs/cgi-bin
- Se um navegador solicitar a URL que aponta para um arquivo dentro do diretório CGI (por exemplo, http://exemplo.com/cgi-bin/printenv.pl), em vez de simplesmente enviar o arquivo para o navegador web, o servidor HTTP executa o script especificado e passa a saída do script para o navegador da web.



- O script CGI podiam ser escritos em qualquer linguagem de programação
  - Inicialmente eram escritos em C
  - Script em Perl foram muitos populares

#### CGI e Formulários HTML

- Remontam aos primórdios da Web e são anteriores à própria JavaScript
- São o mecanismo por trás da primeira geração de aplicativos Web
- Neles, a entrada do usuário é obtida em elementos de formulário
  - O envio do formulário remete essa entrada para o servidor
  - O servidor processa a entrada e gera uma nova página HTML (normalmente com novos elementos de formulário) para exibição pelo cliente

#### Exemplo de uso de CGI

#### Exemplo de uso de CGI

```
#!/usr/bin/perl
local ($buffer, @pairs, $pair, $name, $value, %FORM);
# Read in text
$ENV{'REQUEST METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST METHOD'} eq "POST") {
   read(STDIN, $buffer, $ENV{'CONTENT LENGTH'});
 else {
   $buffer = $ENV{'QUERY STRING'};
# Split information into name/value pairs
@pairs = split(/&/, $buffer);
foreach $pair (@pairs) {
   (\$name, \$value) = split(/=/, \$pair);
   $value =~ tr/+/ /;
   value = ~ s/%(...)/pack("C", hex($1))/eg;
   FORM{name} = value;
```

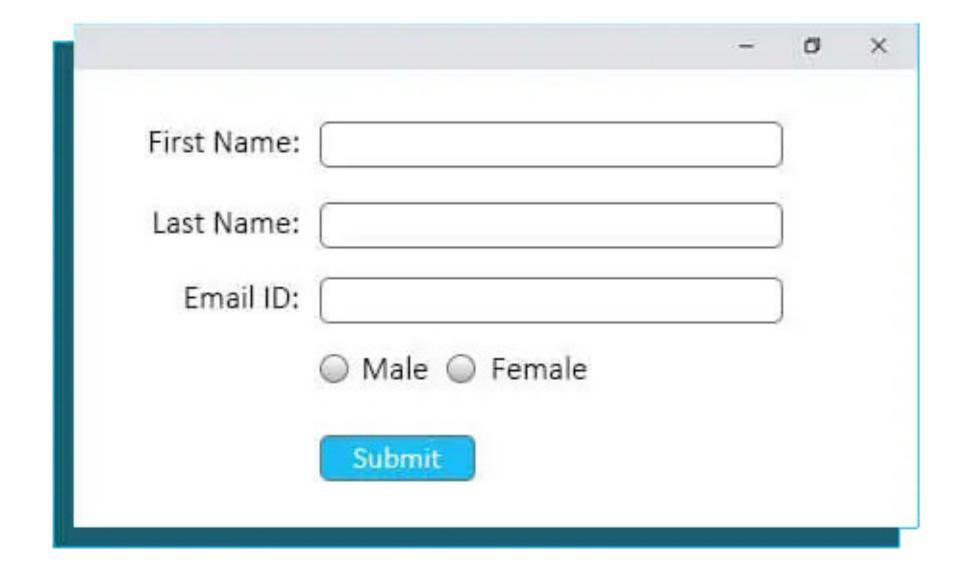
```
$first_name = $FORM{first_name};
$last_name = $FORM{last_name};

print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print "<head>";
print "<head>";
print "<body>";
print "<h2>Hello $first_name $last_name - Second CGI Program</hd>
//diployed CGI Program
CGI Program
/h2>";
print "</body>";
print "</body>";
print "</html>";

1;
```

#### Exemplo de uso de CGI

- Apesar de não ser um componente dinâmico no navegador, como o JavaScript, possibilitou a execução de programas interativos na web dos tempos de 1993-1994
- No entanto, ainda em 1994, PHP, inicialmente chamada de Personal Home Page Tools, se destacou e acabou tomando o espaço do CGI



#### Prática

- Usando NodeJs crie um formulário de cadastro e verifique se as opções enviadas pelo usuário são válidas
  - Caso sejam válidas retorne uma página com a mensagem de sucesso
  - Caso contrário, retorne a página do formulário indicando o local do erro
- Implemente o envio dos dados do formulário via GET e POST

#### Obtendo parâmetros de uma requisição GET

 Precisamos utilizar acessar o objeto a URL dentro do objeto que encapsula a requisição HTTP

#### Obtendo parâmetros de uma requisição POST

 Precisamos utilizar acessar o objeto a URL dentro do objeto que encapsula a requisição HTTP

#### Referências

- Web Design History Timeline
- Common Gateway Interface
- Build Your Own Web Server From Scratch In Node.JS
- How To Create a Web Server in Node.js with the HTTP Module
- 1993: CGI Scripts and Early Server-Side Web Programming
- PERL and CGI Tutorial

# Por hoje é só