Exercícios em PHP

Thiago H. P. Silva

Instalação

- LAMP: Linux + Apache + MySQL + PHP
 - 1. ctrl+alt+t
 - 2. sudo apt-get install tasksel
 - 3. sudo tasksel install lamp-server
 - 4. sudo chmod 777 /var/www/html/
 - 5. echo "<?php print "oi"; ?>" > /var/www/html/teste.php
 - 6. Abra o navegador e insira o seguinte endereço:
 - localhost/teste.php

Instalação

- MySQL
 - 1. ctrl+alt+t
 - 2. sudo mysql -u root
 - 3. CREATE USER 'aluno'@'%' IDENTIFIED BY 'teste\$123';
 - 4. FLUSH PRIVILEGES;
 - 5. EXIT;

Instalação

- MySQL Workbench
 - https://dev.mysql.com/downloads/workbench/
 - 2. Selecione a versão do seu Linux (ctrl+alt+t →uname -a)
 - 3. Faça o download
 - 4. ctrl+alt+t
 - 5. sudo dpkg -i <arquivo.deb>
 - 6. Teve dependências?
 - sudo apt-get install <dependência>
 - 7. Abra o MySQL Workbench
 - Crie uma nova conexão MySQL Connections ⊕ ⊗
 - Username: aluno
 - Senha: teste\$123



Successfully made the MySQL connection

Information related to this connection:

OSI Model

APPLICATION HTTP, SNMP, FTP Interface for end point service Examples are web browsing and email

PRESENTATION WMV, JPEG, PNG

Formats application data for delivery Examples are compression and encryption

SESSION Connection Management

Manages sessions between application process

TRANSPORT TCP, UDP Host to host communications Segments and Diagrams

NETWORK IP Source and destination IP addresses www.google.com = IP address Packets

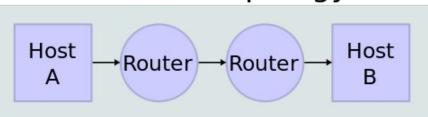
DATA LINK MAC, FCS Source and destination MAC addresses Ethernet Frames

PHYSICAL Data Encoding

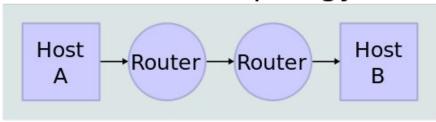
Physical media Layer 1



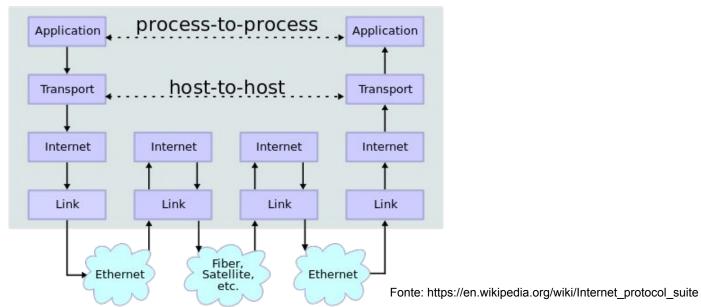
Network Topology



Network Topology



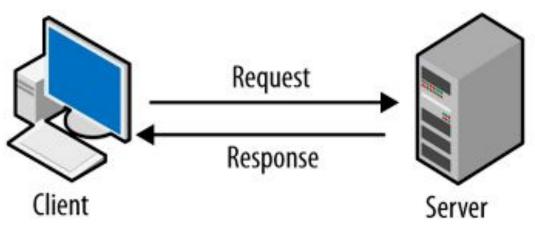
Data Flow



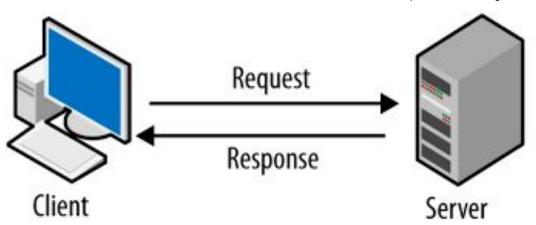
- Hypertext Transfer Protocol
 - Comunicação → Camada de Aplicação (modelo OSI)
 - Base na comunicação de dados da World Wide Web

- Hypertext Transfer Protocol
 - Comunicação → Camada de Aplicação (modelo OSI)
 - Base na comunicação de dados da World Wide Web
- Hipertexto
 - Texto estruturado que utiliza ligações lógicas entre entidades
 - HTTP → Transferencia de Hipertextos

Fonte: https://darvishdarab.github.io/cs421_f20/docs/readings/client_server/

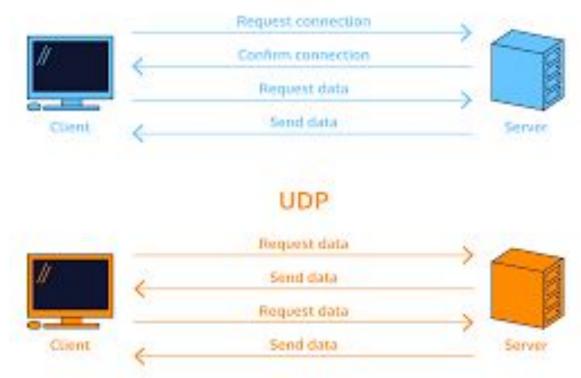


Fonte: https://darvishdarab.github.io/cs421_f20/docs/readings/client_server/

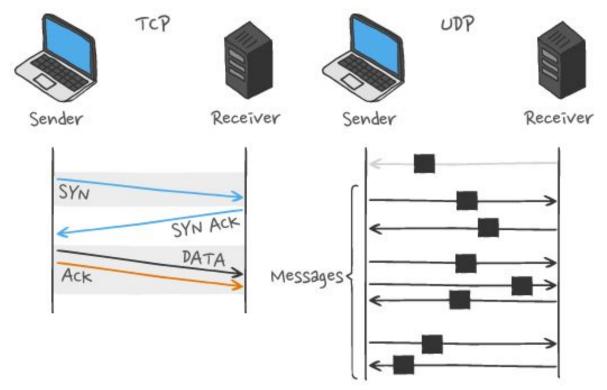


- → Implementação via **sockets** TCP/IP (interfaces de intercomunicação entre aplicações):
 - Transmission Control Protocol (TCP)
 - User Datagram Protocol (UDP)

TCP



Fonte: https://www.codecademy.com/



Fonte: https://www.codecademy.com/

- Como são as mensagens trocadas?
 - Cabeçalhos
 - 1. general-header
 - 2. request-header
 - 3. response-header
 - 4. entity-header

- Como são as mensagens trocadas?
 - Corpo
 - Dados enviados ao servidor
 - Recursos que foram requisitados

Como são as mensagens trocadas?

o Corpo

Exemplo	Descrição
text/plain	Arquivo no formato texto (ASCII)
text/html	Arquivo no formato HTML, utilizado como padrão para documentos Web
Image/gif	Imagem com o formato GIF
Image/jpeg	Imagem com o formato JPEG
application/zip	Arquivo compactado
application/json	Arquivo no formato JSON
application/xml (ou text/xml)	Arquivo no formato XML

Fonte: https://en.wikipedia.org/wiki/Internet_protocol_suite

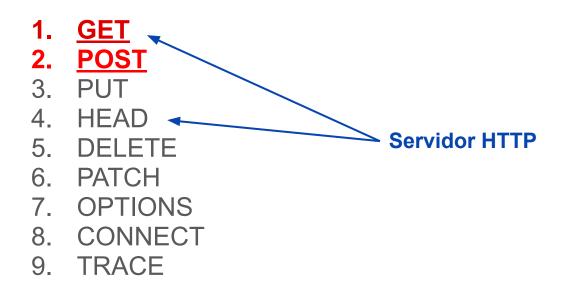
• Quais métodos de solicitação existem?

- Quais métodos de solicitação existem?
 - 1. <u>GET</u>
 - 2. POST
 - 3. PUT
 - 4. HEAD
 - 5. DELETE
 - 6. PATCH
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. <u>GET</u>
 - 2. POST
 - 3. PUT
 - 4. HEAD
 - 5. DELETE
 - 6. PATCH
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

Servidor HTTP

Quais métodos de solicitação existem?



- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD
 - 5. DELETE
 - 6. PATCH
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD \rightarrow GET without *body*
 - 5. DELETE
 - 6. PATCH
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. $HEAD \rightarrow GET$ without *body*
 - 5. DELETE → delete
 - 6. PATCH
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD \rightarrow GET without *body*
 - 5. DELETE → delete
 - 6. PATCH → partial modifications
 - 7. OPTIONS
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD \rightarrow GET without *body*
 - 5. DELETE → delete
 - 6. PATCH → partial modifications
 - 7. OPTIONS → communication options
 - 8. CONNECT
 - 9. TRACE

- Quais métodos de solicitação existem?
 - 1. GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD \rightarrow GET without *body*
 - 5. DELETE \rightarrow delete
 - 6. PATCH → partial modifications
 - 7. OPTIONS → communication options
 - 8. CONNECT → two-way communications (a tunnel)
 - 9. TRACE

- Quais métodos de solicitação existem?
 - GET
 - 2. POST
 - 3. PUT → create/update
 - 4. HEAD \rightarrow GET without *body*
 - 5. DELETE \rightarrow delete
 - 6. PATCH → partial modifications
 - 7. OPTIONS → communication options
 - 8. CONNECT → two-way communications (a tunnel)
 - 9. TRACE → loop-back tests (debugging)

GET

- GET é usado para solicitar dados de um recurso específico
- Forma de consulta com nome/valor
- Query é enviada na URL de uma solicitação GET

/teste.php?name1=value1&name2=value2

<u>GET</u>

Como obtemos os nomes e valores em PHP?

/teste.php?name1=value1&name2=value2

<u>GET</u>

Como obtemos os nomes e valores em PHP?

/teste.php?name1=value1&name2=value2

Variável especial: \$_GET

GET

Como obtemos os nomes e valores em PHP?

/teste.php?name1=value1&name2=value2

Variável especial: \$_GET

```
<?php echo var_dump($_GET); ?>
```

POST

- Usado para enviar dados a um servidor para criar/atualizar um recurso
- Os dados enviados ao servidor com POST são armazenados no corpo da solicitação HTTP

POST teste.php HTTP/1.1

Host: localhost

name1=value1&name2=value2

POST

- Como obtemos os nomes e valores em PHP?
- Variável especial: \$_POST

```
<?php echo var_dump($_POST); ?>
```

- Usando o método POST
- Crie o seguinte formulário

Nome:	Enviar
-------	--------

- Como action, informar a própria página
- Se o método de requisição for POST, então imprima o nome informado se ele não for vazio e, caso contrário indique "Nome não informado."
- Caso o método não seja POST, não imprimir nada.
 Obs: use a variável global \$_SERVER

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Nome: <input type="text" name="nome">
        <input type="submit" value="Enviar">
        </form>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['nome'];
    if (empty($name)) {
       echo "Nome nāo informado.";
    } else {
       echo $name;
    }
}</pre>
```

- Faça um programa para exibir a tabuada de um número informado pelo usuário
 - Gerar as opções do SELECT HTML
 - Enviar a seleção direto para o servidor quando um novo item for selecionado

- 2. Faça um programa para exibir a tabuada de um número informado pelo usuário
 - Gerar as opções do SELECT HTML
 - Enviar a seleção direto para o servidor quando um novo item for selecionado
 - JavaScript: onchange="this.form.submit()"
 - JQuery: \$('#selectId').on('change', function() { this.form.submit(); });

Projeto 1. Crie uma calculadora virtual com processamento no lado do servidor

