



Universidade Federal de Uberlândia

Faculdade de Computação

6º Trabalho de Programação para Internet – Prof. Daniel A. Furtado

Trabalho Individual – Programação *Server-Side*

### INTRUÇÕES GERAIS

- Esta atividade deve ser realizada individualmente;
- Utilize apenas as tecnologias HTML5, CSS, JavaScript, Bootstrap 5 e PHP. **Não é permitido o uso** de tecnologias adicionais como jQuery, React, etc. Trabalhos utilizando tais tecnologias serão anulados;
- Os recursos adequados das linguagens devem ser utilizados;
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As páginas web não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato, etc.;
- O website deve ser bem estruturado utilizando uma pasta raiz e subpastas de acordo com as instruções das videoaulas sobre HTML;
- O website deve ser validado utilizando a ferramenta disponível no endereço [validator.w3.org](http://validator.w3.org). Não deve conter nenhum erro ou *warning*;
- O trabalho deve ser entregue até a data definida pelo professor em aula síncrona;
- Trabalhos enviados por e-mail **não serão considerados** (veja instruções no final);
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues.

Estude a videoaula sobre web dinâmica com PHP disponível no YouTube:

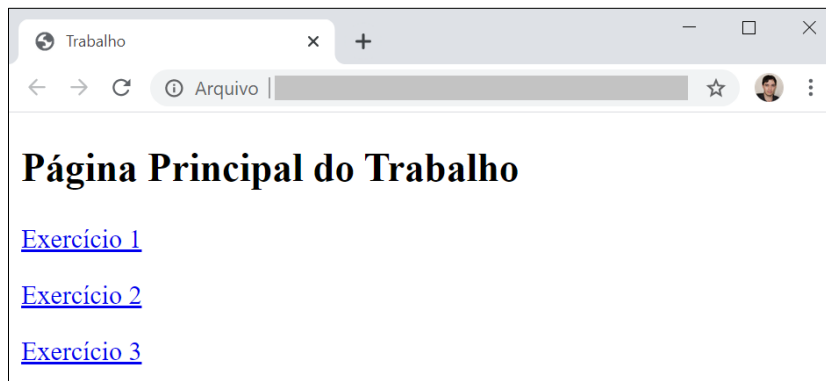
<https://youtu.be/tasLntGDF8c>

Em seguida, faça os exercícios descritos a seguir. As páginas dos exercícios devem ser disponibilizadas em endereços no formato:

[www.seusubdominio.dominio.com/trabalho6/ex1](http://www.seusubdominio.dominio.com/trabalho6/ex1),

[www.seusubdominio.dominio.com/trabalho6/ex2](http://www.seusubdominio.dominio.com/trabalho6/ex2), etc.

A página principal desta atividade deverá conter links para as páginas dos exercícios, conforme exemplo a seguir:



## Exercício 1

---

Crie uma página web utilizando o sistema de grade do Bootstrap 5 que contenha um formulário responsivo com os seguintes campos para cadastro de endereço: **CEP**, **Logradouro**, **Bairro**, **Cidade** e **Estado**. CEP deve ser o primeiro campo do formulário. O campo **Estado** deve ser do tipo *select* e deve aparecer na mesma linha que o campo cidade, exceto quando acessado de smartphones (neste caso, todos os campos devem ser empilhados na vertical).

Em seguida, crie um script PHP que receba o formulário anterior pelo método POST. O script PHP deve gerar uma página dinâmica apresentando os dados em uma grade Bootstrap. As informações de cada endereço devem aparecer sempre na horizontal, exceto quando o arquivo for acessado de um smartphone (neste caso, devem aparecer todos empilhados na vertical).

## Exercício 2

---

Crie um script PHP contendo um vetor com 10 *strings* de nomes de produtos. Crie também um segundo vetor nesse script contendo 10 *strings* representando as respectivas descrições dos produtos (com pelo menos 10 caracteres cada descrição). O script PHP deve receber um parâmetro de nome **qde** pela URL e produzir uma página HTML5 dinâmica contendo uma tabela HTML (utilizando classes do Bootstrap 5) com informações de produtos retirados desses vetores de forma aleatória (utilize a função `rand(0, 9)` do PHP). A tabela deve ter 3 colunas: um número sequencial, o nome do produto e a respectiva descrição do produto. O script PHP deve gerar a tabela dinamicamente contendo uma quantidade de linhas igual ao parâmetro fornecido pela URL (**qde**).

**OBS:** não utilize campos de formulário. O parâmetro deve ser passado diretamente pela URL.

## Exercício 3

---

Crie uma página HTML contendo um formulário com os campos **e-mail** e **senha** e um botão **cadastrar** (adaptar o formulário de login do trabalho de CSS). Em seguida, crie um script PHP para receber os dados do formulário e inseri-los em dois arquivos com nomes, respectivamente, **email.txt** e **senhaHash.txt** (veja código a seguir).

Utilize a função **password\_hash** do PHP, conforme apresentado na videoaula, para gerar uma *string* de *hash* a partir da senha (o *hash* deve ser gravado no arquivo - e não a senha).

O script PHP deve gerar uma página HTML dinâmica confirmando a inserção dos dados. Para salvar as *strings* nos arquivos utilize a função a seguir (dados novos devem sobrescrever os dados anteriores):

```
function salvaString($string, $arquivo)
{
    $arq = fopen($arquivo, "w");
    fwrite($arq, $string);
    fclose($arq);
}
```

## Exercício 4

---

Crie um script PHP para produzir uma página HTML dinâmica contendo os dados armazenados nos arquivos **email.txt** e **senhaHash.txt**. A página deve prevenir ataques XSS (veja detalhes na videoaula sobre PHP). **OBS:** os dados não devem ser armazenados “modificados” nos arquivos (não utilize a função **htmlspecialchars** antes de armazenar os dados). Utilize a função a seguir para carregar as *strings* dos arquivos:

```
function carregaString($arquivo)
{
    $arq = fopen($arquivo, "r");
    $string = fgets($arq);
    fclose($arq);
    return $string;
}
```

## Exercício 5

---

Crie uma página HTML contendo um formulário de login com os campos **e-mail** e **senha** e um botão **Entrar** (adaptar o formulário anterior). Em seguida, crie um script PHP para receber os dados do formulário e fazer a validação dos dados utilizando as *strings* armazenadas previamente nos arquivos **email.txt** e **senhaHash.txt**. A validação deve ser feita de forma adequada utilizando a função **password\_verify** do PHP conforme apresentado na videoaula. Caso os dados estejam corretos, o script PHP deve redirecionar o usuário para uma página de sucesso (criar uma página HTML simples) utilizando as funções **header** / **exit** (veja detalhes na videoaula). Caso contrário, o usuário deve ser redirecionado novamente para a página de login.

## Disponibilização Online

---

O website deve ser colocado online utilizando um serviço de hospedagem gratuito, conforme apresentado no final da primeira videoaula sobre HTML. Recomenda-se a utilização do awardspace.com, conforme detalhado na videoaula.

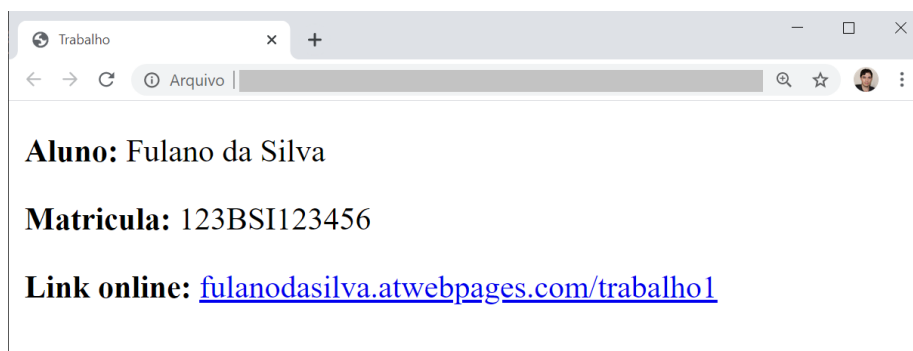
A página principal do trabalho deve ser disponibilizada por meio de um endereço no formato: [www.subdominiocadastrado.dominioescolhido.com/trabalho6](http://www.subdominiocadastrado.dominioescolhido.com/trabalho6), **sem a indicação de um nome de arquivo em particular** como principal.html (veja a parte da videoaula de HTML sobre o arquivo index.html).

O trabalho deve ser disponibilizado online de forma independente dos trabalhos anteriores. Não deve substituir ou referenciar os arquivos de trabalhos anteriores.

## Entrega

---

Primeiramente o trabalho deve ser colocado online até a data limite de entrega indicada pelo professor. Em seguida, envie pelo Sistema de Aplicação de Testes (SAAT), até a data limite, apenas um arquivo HTML simples (arquivo de texto não será aceito) contendo seus dados e um **link que dê acesso imediato ao trabalho online**. Não envie arquivos compactados, arquivos de texto ou arquivos do trabalho pelo sistema SAAT ou por e-mail. Veja a seguir um exemplo de arquivo HTML a ser enviado pelo SAAT:



## Sobre Eventuais Plágios

---

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.