



Universidade Federal de Uberlândia
Faculdade de Computação – Prof. Daniel A. Furtado
10º Trabalho de Programação para Internet
Programação Server-Side com Acesso à Banco de Dados

Instruções Gerais

- Esta atividade deve ser realizada **individualmente**;
- Utilize apenas as tecnologias HTML5, CSS, JavaScript, Bootstrap 5, PHP e MySQL;
- Sintaxe da XHTML como ou
 não é permitida (anulará o trabalho);
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Ao construir o website, utilize dados fictícios. **Jamais utilize** dados pessoais como seu nome, CPF, endereço, e-mail etc.;
- 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 validado utilizando as ferramentas disponíveis nos endereços **validator.w3.org** e **jigsaw.w3.org/css-validator** (não deve conter nenhum erro ou *warning*);
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Material de Apoio e Dicas Gerais

<https://furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo7-Banco-de-Dados.pdf>

Para visualizar eventuais erros do MySQL/PHP, acesse sua conta do infinityfree e configure:

Accounts → if0_xxx → Control Painel → Software → Alter PHP Config → Alter PHP Directives → Display Errors → ON

Display Errors	<input type="radio"/> Off <input checked="" type="radio"/> On
MB String Input	<input type="text" value="auto"/>
PHP Timezone	<input type="text" value="America/New_York"/>
<input type="button" value="Alter PHP directives"/>	

Exercício 1

Descompacte o arquivo <http://www.furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Mysql.zip> e coloque os exemplos online seguindo os passos a seguir:

1. Crie um novo banco de dados no **infinityfree** conforme instruções do **slide 5**;
2. Siga os passos do **slide 7** para executar o código SQL disponibilizado no arquivo **sql-tabelas.sql** (veja pasta raiz dos exemplos);
3. Abra o arquivo **conexaoMysql.php** (na pasta raiz) e altere os dados de conexão inserindo os dados do seu banco de dados (veja figura do **slide 6**);
4. Copie a pasta raiz contendo todos os exemplos para o servidor. O exemplo deve ficar disponível no endereço **seudominio.com/trabalho10**. Neste trabalho não é necessário criar subpastas para cada exercício;
5. Ao acessar a pasta raiz dos exemplos no navegador deverá abrir um **menu de opções**. Abra o **Exemplo 1 – Hello World** e verifique se os dados dos alunos estão sendo listados adequadamente;
6. Abra o arquivo **index.php** do **Exemplo 1** e acrescente comentários no código PHP explicando as operações.

Exercício 2

1. Acesse o **Exemplo 2 – Cadastrar aluno** e cadastre dois novos alunos. Liste os dados;
2. O código PHP do exemplo está vulnerável a ataques do tipo **SQL Injection**. Simule um ataque cadastrando um novo aluno e inserindo a string a seguir no campo **telefone** do formulário:

tol'); DELETE FROM aluno; -- comment

3. Acrescente comentários no código PHP do arquivo **cadastro-vulneravel.php** explicando o motivo da vulnerabilidade. Não utilize explicitamente na explicação as palavras Injeção, Injection, Ataques, XSS etc.;
4. Coloque o código vulnerável dentro de um **comentário de bloco** e acrescente o código adequado utilizando **prepared statements** para corrigir a vulnerabilidade. Em seguida, cadastre novos alunos e repita o ataque de injeção para verificar se o problema foi resolvido.

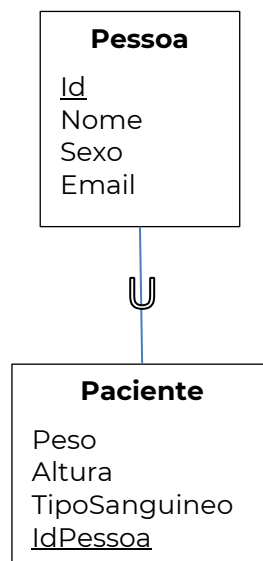
Exercício 3

1. Acesse o **Exemplo 3 – Cadastrar Cliente (MVC simplificado)** no navegador e cadastre dois novos clientes. Verifique os dados cadastrados escolhendo a opção de listagem;
2. Na tela de listagem, clique no link **Excluir** para remover um dos clientes;
3. Abra o exemplo no **VS Code** e analise o código dos arquivos:
 - a. cliente.php
 - b. index.html
 - c. controlador.php
 - d. clientes.html (analisar superficialmente, pois o conteúdo referente às requisições Ajax – GetJSON – ainda será apresentado nas próximas aulas)
4. Faça uma cópia da pasta **Ex3-cadastro-mvc** e renomeie a cópia para **Ex3-cad-produto-mvc**. Faça as adaptações necessárias para permitir o cadastro e listagem de produtos (ao invés de clientes). Devem ser cadastrados os dados: **nome do produto, marca e descrição**. Uma nova tabela deve ser criada no banco de dados;
5. Acrescente links no **menu de opções** para incorporar a nova funcionalidade de cadastro de produtos.

Exercício 4

1. Acesse o **Exemplo 4 – Cadastro em duas tabelas** no navegador e cadastre dois novos clientes. Observe que os dados são inseridos nas tabelas correlacionadas **Cliente** e **enderecoCliente** utilizando o conceito de transação. Liste os dados escolhendo a opção **Exemplo 4 - Mostrar clientes e endereços**;
2. Abra o exemplo no **VS Code** e analise o código dos arquivos:
 - a. index.html
 - b. cliente.php -> método Create
 - c. controlador.php
 - d. clientes.html (analisar apenas superficialmente)
3. Acrescente uma página HTML e um script PHP para permitir o cadastro de **Pacientes** de acordo com o diagrama a seguir. O formulário de cadastro deve ser único, com campos para o preenchimento do Nome, Sexo, Email, Peso, Altura e Tipo Sanguíneo do paciente. O código PHP de cadastro deve ser inserido utilizando um método de nome **Create** dentro de uma classe de nome **Paciente** em um arquivo **paciente.php**. O método deve receber todos os dados por parâmetro e inserí-los, utilizando uma **transação**, nas duas tabelas correlacionadas por **chave estrangeira** (tabela **Pessoa** e tabela **Paciente**). Acrescente a funcionalidade no arquivo **controlador.php**.

As tabelas devem ser criadas conforme diagrama a seguir. O campo **Id** em **Pessoa** deve ser **chave primária** do tipo **auto_increment**. O campo **IdPessoa** em **Paciente** deve ser **chave primária** (sem auto_increment) e também **chave estrangeira** conectando com **Pessoa**. Não acrescente atributos além daqueles mostrados no diagrama.



4. Acrescente um link no **menu de opções** para incorporar a nova funcionalidade de cadastro de pacientes com transações.

Exercício 5

Acesse o **Exemplo 5 – Testar login** no navegador e preencha o formulário de login utilizando dados de clientes cadastrados anteriormente. Experimente uma senha incorreta e observe o resultado. Em seguida, abra o exemplo no **VS Code** e analise o código dos arquivos.

Disponibilização Online

O trabalho deve ser entregue pelo sistema SAAT e disponibilizado online utilizando o subdomínio gratuito registrado em site de hospedagem. Como este trabalho consiste de modificações dos arquivos de exemplo, não é necessário criar subpastas para cada exercício. Ao acessar o endereço a seguir, deverá abrir a página contendo o **menu de opções** modificado:

seusubdominio.com/**trabalho10**

Entrega

Além da disponibilização online, a pasta raiz deve ser compactada no formato **zip** e enviada pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data limite indicada pelo professor em sala de aula.

Adicione também um arquivo de nome **link.txt**, na pasta raiz, contendo a URL do trabalho online (para a pasta raiz do trabalho).

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**.