



Hora de Início: 20:40

Hora de Término: 22:30

Instruções Gerais – Leia atentamente!

- O arquivo correspondente à solução de cada questão deve ser enviado pelo sistema SAAT (www.furtado.prof.ufu.br) até o horário de término indicado acima. **Apenas um arquivo deve ser enviado por questão, sem compactar. Arquivos compactados serão desconsiderados e a respectiva questão será anulada.**
- Esta prova é **individual** e pode ser realizada com consulta apenas aos materiais disponibilizados pelo professor no endereço www.daniel.prof.ufu.br (incluindo slides, códigos de exemplo e videoaulas). Arquivos de trabalhos **não são** permitidos.
- **Questões não enviadas até a hora limite serão automaticamente perdidas pelo aluno** (com resp. nota 0). Resolver as questões no tempo apropriado faz parte desta avaliação, dados os materiais disponibilizados;
- **NÃO É PERMITIDA** a consulta a outros materiais e websites, arquivos de semestres anteriores, arquivos de colegas etc. O uso de qualquer material desse tipo será interpretado como plágio e resultará na anulação das provas de **todos os alunos envolvidos**, com respectiva nota zero;
- Leia atentamente as especificações das questões antes de resolvê-las;
- A utilização de recursos ou tecnologias não apresentadas em aula resultará na **anulação da questão**;
- As questões devem ser resolvidas utilizando os recursos de maneira adequada e da forma mais eficiente possível;
- Questões resolvidas de formas diferentes daquelas solicitadas não serão consideradas;
- Para acessar o banco de dados, criar tabelas e testar os códigos, recomenda-se a utilização da conta do **infinityfree** criada anteriormente para realização dos trabalhos;
- A comunicação com o banco de dados deve ser feita utilizando o PHP Data Objects (PDO);
- Todo código deverá estar 100% indentando. Código sem a devida indentação será considerado ilegível, de difícil manutenção, e resultará em penalização de 30% sobre o valor da questão;
- **Alunos envolvidos em qualquer tipo de plágio, total ou parcial, terão suas provas anuladas e receberão nota zero.**

Questão 1 (10 pontos)

Abra o arquivo anexo de nome *questao1.html* e observe o formulário de cadastro de funcionário, sem alterar o arquivo. **Crie** um script PHP de nome **questao1.php** para (1) receber os dados desse formulário, (2) inseri-los adequadamente nas tabelas **pessoa** e **funcionário** do MySQL e (3) gerar como resposta uma página HTML dinâmica listando os dados de todos os funcionários cadastrados até o momento (nome, cpf, email, admissão, cargo e salário). Veja as instruções adicionais a seguir:

- Apenas um único arquivo PHP deve ser criado. **Não** crie um script para cadastrar os dados e outro para listar os dados;
- Crie as tabelas no MySQL utilizando o código SQL disponibilizado (arquivo *sql-tabelas.sql*);
- Utilize o arquivo fornecido *conexaoMysql.php* e faça os devidos ajustes para se conectar ao seu banco de dados MySQL;
- É obrigatório o uso do arquivo fornecido *questao1.html* sem qualquer alteração (caso contrário a questão será anulada);
- O arquivo *questao1.php* deve fazer referência ao arquivo de conexão *conexaoMysql.php* e à função *mysqlConnect()* desse arquivo;
- A página deve prevenir ataques do tipo SQL *injection* e XSS. O cadastro deve ser feito utilizando os recursos adequados para evitar eventuais inconsistências no banco de dados;
- Envie **apenas** o script **questao1.php**, sem compactar.

Questão 2 (8 pontos)

Crie um script PHP de nome **questao2.php** que retorne como resposta, no formato JSON, um *array* de objetos correspondentes aos funcionários cadastrados no banco de dados com salário superior a um parâmetro de nome **salMin** informado pela URL. Cada objeto desse *array* deve ter os campos **nome**, **e-mail** e **salário** do funcionário. O script **não deve retornar** mensagens ou conteúdos em HTML. Exemplo de chamada do script: `meudominio.com/questao2.php?salMin=2000`.

Caso nenhum parâmetro seja passado ao script, então deve-se considerar `salMin=0`, ou seja, o script deve retornar um *array* de objetos com os dados de todos os funcionários cadastrados.

O arquivo PHP deve referenciar o arquivo *conexaoMysql.php* e a função *mysqlConnect()*. Aspectos de segurança devem ser considerados. Envie **apenas** o script **questao2.php**.

Questão 3 (12 pontos)

Crie uma página HTML (`questao3.html`) contendo um campo textual de formulário e um botão “**buscar com fetch**” para que o usuário possa buscar pelos funcionários cadastrados no banco de dados que ganham mais que o salário informado no campo textual. Em seguida:

a) Crie uma função JavaScript (no próprio arquivo HTML) que faça uma requisição Ajax assíncrona, utilizando a API Fetch com `async/await`, para buscar no servidor os dados dos funcionários com salário maior que aquele informado no campo textual. Observações:

- A requisição deve ser enviada ao script **questao2.php**;
- A busca deve ocorrer apenas quando o usuário clicar no botão **buscar**;
- O formulário **não deve** ser submetido da forma convencional;
- A função não deve exibir diretamente os dados, mas deve chamar uma outra função esse fim (a função a ser criada no item b) a seguir);

b) Crie uma função JavaScript que receba os dados dos funcionários buscados pela função anterior e apresente-os na página HTML de forma clara e bem estruturada. A função deve mostrar os dados fazendo uma atualização dinâmica da árvore DOM. Acrescente o código JavaScript “principal” para viabilizar a busca com o botão.

c) Crie uma terceira função JavaScript para fazer a mesma busca solicitada no item a), porém utilizando o XMLHttpRequest. Não altere a função criada para o item a). Um segundo botão deve ser criado para testar a busca com o XHR sem alterar as funcionalidades anteriores.

Envie apenas o arquivo HTML contendo o código JavaScript pelo sistema SAAT. Não crie arquivos adicionais. Não compacte o arquivo.