



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) e aos arquivos dos trabalhos **que foram entregues anteriormente** pelo sistema SAAT (seu próprio trabalho);
- **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 associadas às questões antes de resolvê-las;
- A utilização de recursos e 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 **awardpace/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 (12 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, data de admissão 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 anexo de nome *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. Implementações que não utilizarem o arquivo *questao1.html* **exatamente** como fornecido serão anuladas (com respectiva nota zero);
- 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;
- No arquivo *sql-tabelas.sql* há um exemplo de código SQL para listar os funcionários;
- Envie o arquivo *questao1.php* pelo sistema SAAT;

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 própria 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 o mesmo deve considerar que **salMin** seja 0 e retornar o array com os dados de todos os funcionários cadastrados.

Outras instruções:

- O arquivo PHP deve referenciar o arquivo *conexaoMysql.php* e a função *mysqlConnect()*;
- A página deve prevenir ataques do tipo SQL *injection*;
- Não crie arquivos adicionais além do script *questao2.php*. Envie apenas esse script;
- Um código SQL de exemplo é apresentado no arquivo *sql-tabelas.sql*

Questão 3 (10 pontos)

Crie uma página HTML contendo um campo textual de formulário e um botão **buscar** para que o usuário possa buscar pelos funcionários cadastrados no banco de dados que ganham mais que o salário informado no campo textual.

a) Crie uma função JavaScript (no próprio arquivo HTML) que faça uma requisição Ajax assíncrona, utilizando o XMLHttpRequest ou a API Fetch (com ou sem `async/await`), para buscar no servidor os dados dos funcionários que possuem salário maior que o salário informado no campo textual. O script PHP criado na questão anterior deve ser utilizado. A busca deve ocorrer quando o usuário clicar no botão **buscar**. O formulário não deve ser submetido da forma convencional.

b) Crie uma função JavaScript que receba os dados dos funcionários retornados pela função anterior e apresente os mesmos na página HTML de forma clara e bem estruturada. A função deve mostrar os dados fazendo uma atualização dinâmica da árvore DOM da página.

Envie o arquivo HTML (contendo todo o código JavaScript) pelo sistema SAAT. Não crie arquivos adicionais.