

Nome: _____

A estrutura de dados dos arquivos segue em anexo, utilizar servidor de preferência, ex. *WAMP, LAMP, etc.*

Resolver as questões e enviar via e-mail um arquivo *.zip* com todos os arquivos *.php* na estrutura proposta.

1) Desenvolva um formulário com 5 campos em “formulario.php”

Fazer esta página inicial em PHP, e estilizar o HTML usando CSS básico, este formulário deve conter os campos abaixo:

- Nome
- Data de nascimento
- Sexo
- Idade (Somente leitura)
- E-mail

2) Validar os campos utilizando Javascript

É permitida a utilização de bibliotecas de terceiros como: JQuery, Angular, Prototype, etc... As validações devem seguir a relação abaixo:

- Nome => mínimo 3 dígitos e máximo 20 dígitos
- Data de nascimento => apenas uma máscara dd/mm/aaaa
- Sexo => 'M' para masculino e 'F' para feminino
- Idade => deverá ser calculada de forma automática, baseando-se na data de nascimento informada
- E-mail => Validar o formato se contem "@", ".com", tudo em minuscúlo

3) Enviar os dados do formulário para outra página “tratamento.php”

Para esta questão, deve-se utilizar apenas o método **POST**, validar se todos os campos do formulário foram recebidos com sucesso.

- Caso contrário, deverá exibir a mensagem de erro: “Campo inválido: XXXXXXXX”, onde “XXXXXXX” deve ser o nome do campo.
- Se o erro acontecer, a execução deve parar por aqui e não deve prosseguir.

4) Refazer as mesmas validações, mas agora com as variáveis no PHP

Para realizar esta questão, utilizar as mesmas regras citadas no item 2 (Validações Javascript) utilizando as variáveis no PHP.

- Caso ocorra algum erro, exibir a seguinte mensagem: “Campo informado incorretamente: XXXXXXXX”, onde “XXXXXXX” deve ser o nome do campo.
- Se o erro acontecer, a execução deve parar por aqui e não deve prosseguir.

5) Criar um COOKIE com os dados do formulário

Criar um COOKIE com todos os dados recebidos do formulário.

6) Salvar as informações no Banco de Dados

Para prosseguir fique atento: haverá um arquivo chamado “conexao.php” já com os dados para conectar no MySQL local.

- Criar um banco de dados, chamado “**prova_php**”.
- Criar uma tabela chamada “**clientes**”, utilizando a seguinte estrutura de dados:

| Coluna | Tipo | Null |
|-----------|--------------|------|
| id | int(15) | Não |
| nome | varchar(255) | Não |
| data_nasc | varchar(255) | Não |
| sexo | varchar(255) | Não |
| email | varchar(255) | Não |

- Você deverá utilizar o arquivo “conexao.php” para realizar as queries SQL.

Validar se o e-mail informado já está cadastrado

a) Se o e-mail já estiver cadastrado:

- Exibir apenas uma mensagem: “**Cliente já cadastrado**”.
- Limpar o COOKIE existente
- A execução para por aqui, e não deve prosseguir.

b) Se o e-mail não estiver cadastrado:

- Salvar os dados no Banco de Dados (utilizar as informações do **COOKIE**, e **NÃO** das variáveis recebidas)
- Exibir a mensagem: “**Cliente cadastrado com sucesso**”.
- Limpar o COOKIE existente.
- Redirecionar o cliente para o formulário inicial.

7) Listagem de clientes cadastrados

Na tela inicial (item 1), abaixo do formulário, deve-se exibir uma listagem com todos os clientes já cadastrados (id, nome, email)

8) Localizar o cliente

Na listagem (item 7), deve-se exibir um campo de busca para localizar o cliente desejado.

- Este filtro não deve ser feito no 'server', apenas no 'client' usando Javascript.
- O filtro pode utilizar bibliotecas de terceiros (jquery, angular, prototype, etc)

9) Remover um cliente

Na listagem (item 7), deve-se exibir um botão para remover o cliente desejado.

- Após remover o cliente, atualizar a listagem/tela.

10) Editar registro de clientes

Na listagem (item 7), deve-se exibir um botão para editar o cliente desejado.

- A edição do registro pode ser feita como desejar (em outra tela ou no mesmo formulário)

- Após editar o cliente, atualizar a listagem/tela.