

# Lista 11 – HTML / CSS / JavaScript

Elaboração de um sistema organizado em *front-end* e *back-end*. O objetivo será simular um sistema de autenticação de usuário sem fazer acesso a um banco de dados. Os dados do usuário (*e-mail* e senha) serão inseridos por meio de um formulário HTML (*front-end*) e enviados ao *back-end* ao se clicar no botão de submissão do formulário. O *back-end* irá ler os dados e processá-los, devolvendo uma resposta adequada ao navegador.

Para o desenvolvimento use como base a apresentação atualizada '[aula 12 - DOM com JS \(19.10.2022\).pdf](#)', a partir do *slide* 52.

1. O cliente *front-end* será um formulário em HTML/CSS contendo somente 2 campos de entrada de dados de texto (*e-mail* do usuário e senha) e um botão para submissão do formulário.
  - a) A página deverá ser feita de forma responsiva utilizando o Bootstrap.
  - b) O campo nome será do tipo `email` e o de senha do tipo `password`.
  - c) O botão deverá ser do tipo `submit` e seu texto 'Entrar'.
  - d) Deverá haver uma validação simples no formulário. Ele não poderá ser submetido se algum dos campos estiver vazio.
  - e) O formulário será enviado primeiramente através do método `GET`.
2. O *back-end* será feito em `node.js` utilizando a biblioteca `Express.js` para criar o servidor, tratar as requisições feitas pelo cliente e gerenciar as rotas (*end-points*).

- a) Os dados serão tratados de várias formas diferentes pelo *back-end*, mas basicamente ele irá ler os dados enviados pelo *front-end* e irá compará-los com dados fixos no código-fonte. Isso será feito de várias formas diferentes, especificadas abaixo. Se os dados entrados no formulário forem iguais aos do código-fonte então deverá ser mostrada uma nova página ao usuário com a mensagem que ele entrou no *site*. Caso contrário será mostrada uma mensagem de acesso negado. Em ambas as situações será exibido também o botão para retornar à página de *login*.
- b) Na primeira versão o *back-end* deverá simplesmente declarar duas variáveis em JS e usá-las para realizar o procedimento do exercício acima.
- c) Para a segunda versão, crie uma nova rota com o método `app.get()`. Essa versão irá utilizar duas listas (*arrays*) com 5 elementos cada. Uma delas terá os *e-mails* e a outra terá as senhas. O código deverá procurar o *e-mail* informado no formulário na lista de *e-mails*. Se encontrar deverá verificar na lista de senhas, na posição correspondente, se ela é a mesma da informada no formulário. A seguir proceder com o procedimento descrito na letra (a). Não esqueça de alterar o atributo `action` no formulário HTML para corresponder a este novo endereço.
- d) Para a terceira versão, crie uma nova rota com o método `app.get()`. Essa versão irá utilizar um objeto literal do JS com os dados de 5 usuários no seguinte formato:

```
const dados = {
  usuarios: [
    { nome: 'Fulano', senha: '1234' },
    { nome: 'Beltrano', senha: '4321' },
    .
    .
    .
  ]
};
```

O código deverá percorrer a lista `usuarios` do objeto `dados` acima para fazer a autenticação procedendo então novamente com o processo da letra (a). Não esqueça de alterar o atributo `action` no formulário HTML para corresponder a este novo endereço.

- e) Para a quarta versão, crie uma nova rota com o método `app.get()`. Essa versão irá declarar uma classe chamada `Usuario` com os atributos `email` e `senha`.

Para isso use a nova notação de classes do ES6. Reveja na apresentação '[aula 18 - js \(04.11.2022\).pdf](#)', *slide* 10, da profa. Fran, para fazer isso. Faça uso dos construtores e também de *getters* e *setters*. Crie um método `autentica()` que receba um `email` e `senha` e retorne `true` se eles forem iguais aos do objeto em questão e `false` caso contrário.

O código deverá montar um *array* com 5 objetos desta classe `Usuario`. O código deverá percorrer este *array* para fazer a autenticação usando o método `autentica()` acima, procedendo então novamente com o processo da letra (a).

Melhorias:

- Altere o método de envio do formulário para `POST` de forma que a senha não apareça na URL. Para isso veja o exemplo do *slide* 56 da apresentação '[aula 12 - DOM com JS \(19.10.2022\)](#)'.
- Implementar o processo de comunicação entre *front-end* e *back-end* usando *AJAX* através do objeto `XMLHttpRequest`.
- Em vez de devolver uma nova página, utilize *AJAX* + *DOM* + *JSON* de forma a utilizar somente a página inicial de *login* e alterar elementos da página para informar se o processo de autenticação ocorreu com sucesso ou não.