



Normalizando dados

Até agora, vimos como o **JavaScript** nos permite acessar informações inseridas por pessoas em um formulário. Mas agora, é hora de darmos um passo adiante, para descobrir como esses dados podem ser manipulados para atender às necessidades de nosso aplicativo.

A criação de um aplicativo é um processo que requer vários tipos de análise, cada uma focada em um problema específico: o design, a interação do usuário com o aplicativo, as tecnologias que serão utilizadas, entre outros.

Nesta oportunidade, vamos nos concentrar em um desses problemas, relevante para construir uma aplicação sólida e escalável: **a manipulação de dados**.

Mas, o que queremos dizer com manipulação de dados? Como já sabemos, todo aplicativo é construído a partir de dados. Alguns deles estão previamente armazenados em nosso aplicativo e/ou em um banco de dados; outros, por outro lado, são coletados à medida que as pessoas interagem com o referido aplicativo.

Vamos pensar em um aplicativo como **Facebook** ou **Twitter**. Se quisermos acessar algum deles pela primeira vez, nosso primeiro passo será concluir o registro correspondente. Nesse momento, devemos preencher um formulário com uma série de dados, que serão coletados por esses aplicativos e armazenados em seu banco de dados.

No entanto, para facilitar as tarefas de coleta, armazenamento e consulta desses dados, é necessário que eles sejam armazenados em um **formato comum**. Em outras palavras, nosso banco de dados deve definir um formato específico com o qual devem ser armazenados os

dados inseridos pelas diferentes pessoas no momento do registro (por exemplo, pode-se estabelecer que todos os campos de texto sejam armazenados em letras minúsculas, sem espaços, pontos ou qualquer tipo de caracteres especiais).

Mas vamos pensar por um segundo em quantas pessoas estão acessando o Facebook e o Twitter a qualquer momento. Como garantir que os dados inseridos por todos eles respeitem os critérios estabelecidos para nosso banco de dados?

Para responder a essa pergunta, precisamos introduzir o conceito de **normalização de dados**.

De que trata esse conceito? Bem, a normalização de dados nada mais é do que uma série de processos, regras ou mecanismos usados para dar um formato comum aos dados coletados em um aplicativo, independentemente de quem é a pessoa que os inseriu ou da forma como isso é feito.

Esse processo pode incluir desde instruções fornecidas ao usuário quando ele é solicitado a preencher um determinado campo (por exemplo: *"Digite seu número de telefone sem o DDD"*), a validações e manipulação dos dados coletados, o que pode ser feito tanto do lado do cliente quanto do lado do servidor.

E então, quais métodos podemos usar para manipular esses dados? Vejamos um exemplo: Vamos imaginar que temos um input solicitando que a pessoa insira seus filmes favoritos, separados por ponto e vírgula.

Filmes favoritos

Insira seus filmes favoritos, separados por ponto e vírgula

Ao capturar o input correspondente, obteremos o seguinte resultado:

```
const filmesFavoritos = document.querySelector('#input-filmes');
console.log(filmesFavoritos);
//"Harry potter; Meu malvado favorito; Avatar"
```

Agora, suponha que para armazenar essa informação em nosso banco de dados, precisamos "**normalizá-la**", transformando-a em um **array** de filmes.

Como podemos realizar essa tarefa? Felizmente, JavaScript nos fornece o método [split\(\)](#). Este método nos permite dividir os caracteres de uma string com base nos critérios que queremos, obtendo como resultado um **array** que contém cada uma das substrings geradas.

Então, voltando ao nosso array de filmes, poderíamos usar **split** para normalizar os dados coletados:

```
const filmesNormalizados = filmesFavoritos.split(';');
console.log(filmesNormalizados );
// [ 'Harry potter', 'Meu malvado favorito', 'Avatar' ]
```

Excelente! Agora nossas informações estão normalizadas e prontas para serem armazenadas.

Talvez agora, estamos nos perguntando se este é o único método que temos para normalizar as informações. Claro que não! Mostraremos alguns outros que serão muito úteis para poder realizar essa tarefa.