



### Questão 01

O que é um formulário reativo em Angular?

- a) É um formulário criado diretamente no template HTML usando a diretiva `[(ngModel)]`.
- b) É um formulário criado diretamente no template HTML usando a diretiva `[(ngModel)]`.
- ☒ c) É um formulário criado programaticamente no código TypeScript usando `FormGroup` e `FormControl`.
- d) É um formulário que depende diretamente do template HTML para ser criado e validado.
- e) É um formulário criado pela utilização e tags específicas no html do componente.

### Questão 02

Em um formulário reativo, como é feito o binding entre o formulário e o componente ou input?

- a) Através da diretiva `[(ngModel)]`.
- ☒ b) Através da diretiva `formControlName`.
- c) Através da diretiva `formGroupName`.
- d) Não é possível fazer binding em um formulário reativo.
- e) Através da diretiva `name`.

### Questão 03

Quais são as principais vantagens de se utilizar o formulário reativo em uma aplicação Angular?

- ☒ a) Maior controle e poder para as validações e lógicas de interação com o usuário.
- b) Maior facilidade para construir formulários simples e diretos.
- c) Maior facilidade para trabalhar em equipe.
- d) Maior facilidade para fazer testes unitários.
- e) Maior controle do que pode ou não ser visto no formulário.

### Questão 04

O que são validadores no Angular?

- a) Classes que permitem criar formulários dinâmicos e mais complexos.

- b) Funções que realizam validações no template HTML.
- ☒ c) Classes que implementam uma interface específica para validar os inputs em formulários reativos.
- d) Diretivas utilizadas para fazer o binding de dados em formulários reativos.
- e) Diretivas usadas para validar informações em formulários template driven.

### Questão 05

Qual método é utilizado para aplicar mais de um validador em um campo de formulário reativo?

- a) Validator.required
- b) Validator.email
- c) Validator.minLength
- d) Validator.pattern
- ☒ e) Validator.compose

### Questão 06

Leia o texto a seguir:

Ao editar um item em um aplicativo web ou mobile híbrido, temos a opção de armazenar dados localmente usando o localStorage. Isso permite que o dado seja persistido após o sistema deixar estar rodando em memória RAM, porém o localStorage tem algumas limitações, como o limite máximo, geralmente de 5Mb, e o fato de poder armazenar apenas strings.

Como base no texto acima e no código estudado nessa aula para editar, listar e salvar um item no localStorage, podemos dizer que são desvantagens do localStorage:

- a) Armazenamento local de dados: O LocalStorage permite armazenar dados localmente no dispositivo do usuário, tornando possível que os aplicativos possam funcionar offline, sem depender de uma conexão constante com a internet.
- b) A utilização do LocalStorage para armazenamento de dados pode dificultar a experiência do usuário, tornando o carregamento de informações mais lento e menos fluido.
- c) Como os dados ficam armazenados localmente no dispositivo, o tráfego de rede pode ser reduzido, diminuindo assim o consumo de dados do usuário.
- ☒ d) Lidar com a conversão dos dados para JSON e a manipulação desses em memória antes de salvar os itens
- e) Ao armazenar dados localmente, os desenvolvedores têm mais controle sobre como esses dados são armazenados e como são acessados, o que pode levar a uma pior qualidade e segurança dos dados.