



Questão 01

Qual a diferença entre os métodos de ciclos de vida `ngOnInit` e `ngAfterViewInit`?

- a) O `ngOnInit` é acionado antes do componente ser destruído e o `ngAfterViewInit` é acionado assim que o componente está pronto para uso.
- b) Ambos são acionados quando o componente está pronto para uso e a página está carregada na view.
- c) O `ngOnInit` é acionado assim que os elementos injetados no componente como o `@ViewChild` estão prontos e o `ngAfterViewInit` é acionado assim que o binding dos dados está pronto para ser usado.
- ☒ d) O `ngOnInit` é acionado assim que o binding dos dados está pronto para ser usado, enquanto que o `ngAfterViewInit` é acionado assim que os elementos injetados no componente como `@ViewChild` estão prontos.
- e) O `ngOnInit` é acionado sempre que o angular modificar dados do databind de propriedades e o `ngAfterViewInit` é acionado antes do angular destruir os componentes.

Questão 02

O que é o location strategy e qual a diferença entre `HashLocationStrategy` e `PathLocationStrategy`?

- a) O location strategy define como o angular irá fazer a localização da pessoa e a diferença é que o `HashLocationStrategy` usa GPS enquanto que o `PathLocationStrategy` utiliza os roteadores em que a pessoa está conectada.
- ☒ b) O location strategy define como o angular irá tratar as urls nas rotas e a diferença é que o `HashLocationStrategy` usa o `#` (âncora) para manter o usuário sempre na mesma página, enquanto que o `PathLocationStrategy` usa urls normais.
- c) O location strategy define como o angular irá tratar as urls nas rotas e a diferença é que o `HashLocationStrategy` usa urls normais enquanto que o `PathLocationStrategy` usa o `#` (âncora) para manter o usuário sempre na mesma página.
- d) O location strategy posiciona elementos na página com base em suas coordenadas, sendo que o `HashLocationStrategy` utiliza um hash da coordenada e o `PathLocationStrategy` utiliza coordenadas x,y.

- e) O location strategy define como o angular irá tratar as urls nas rotas e não há diferença entre o HashLocationStrategy e o PathLocationStrategy.

Questão 03

Os itens abaixo definem passos à se seguir para criar uma rota em angular com parâmetros:

- I) Adicionar um placeholder no path da rota para que seja substituído pela informação que deseja passar como parâmetro.
- II) Salvar o parâmetro no localStorage e ler quando a rota for ativada.
- III) Criar um link para página usando o routerLink com os parâmetros.
- IV) Criar um link para página usando método navigate da classe Router.
- V) Receber os parâmetros no componente de destino usando a propriedade snapshot.params da classe ActivatedRoute.
- VI) Receber os parâmetros no componente de destino usando a propriedade params da classe Router.

Dos passos listados, quais são corretos para receber parâmetros na navegação de rotas?

- a) I, II e VI.
- ☒ b) I, III, IV e V.
- c) III, IV, V e VI.
- d) III, IV, V.
- e) I, III, IV e VI.

Questão 04

Qual das alternativas abaixo não representa um passo necessário para criação de subrotas em angular?

- a) Criar os componentes pai e filhos.
- b) Adicionar no componente pai ou <router-outlet> no seu template.
- c) Criar links com routerLink ou o navigate do router.
- d) Adicionar rotas filhas no módulo de rotas usando a propriedade children.
- ☒ e) Adicionar um parâmetro na rota para representar o nome dos componentes filhos.

Questão 05

Avalie o código abaixo e escolha a alternativa que melhor condiz com o que ele faz:

```
providedIn: 'root'
})
export class UserExitGuard implements
CanDeactivate<PessoaAddPageComponent> {
  canDeactivate(
    component: PessoaAddPageComponent,
    currentRoute: ActivatedRouteSnapshot,
    currentState: RouterStateSnapshot,
    nextState?: RouterStateSnapshot): Observable<boolean |
UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
    return component.canExit();
  }
}
```

- a) Define uma classe de serviço com o nome "UserExitGuard".
- b) Define um componente "PessoaAddPageComponent" que pode ser desativado.
- c) Define um componente "UserExitGuard" que pode ser desativado.
- d) Define um guarda de rota que impede o usuário de sair da página.
- ☒ e) Define um guarda de rota que verifica se é possível desativar o componente "PessoaAddPageComponent".

Questão 06

Leia o texto a seguir:

Em aplicações angular, é possível utilizar guards para proteger as rotas e impedir que o usuário acesse determinadas páginas sem estar autenticado ou autorizado. Além disso, é possível também configurar rotas com parâmetros dinâmicos que podem ser passados através da URL.

Com base na aula e no texto acima, qual opção combina melhor os recursos de guardas com os parâmetros de URL?

- ☒ a) Verificar se o usuário é dono do recurso de id passado na url e com isso proibir ou não o acesso ao detalhe do item baseado em uma guarda que verifique o id passado pela url.
- b) Definir uma guarda que verifica se o usuário está logado, retornando verdadeiro caso a sessão do mesmo esteja aberta com o login válido.

- c) Definir uma guarda que verifica se o sistema pode sair da página caso um formulário esteja sendo preenchido.
- d) Utilizar um guard canActivate para garantir que o usuário tenha as permissões necessárias para acessar a rota, e passar os parâmetros de URL como parte do corpo da requisição.
- e) Usar um guard canLoad para impedir que a rota seja carregada se o usuário não estiver autenticado, e passar os parâmetros de URL no header da requisição.