

## Desafio 6

Prólogo: neste desafio você terá uma introdução sobre services em um contexto de compartilhamento de informações entre componentes, chamadas para o backend juntamente com HTTP Client, executando seus principais métodos e suas diferenças e por fim RxJS a fim de ter um controle maior das requisições abertas e tratamento das informações, de erros, etc.

### Item 1: Services

- a) Qual o propósito de *services* no angular?
- b) Porque no Angular há uma distinção entre *components* e *services*?
- c) Como as *services* são disponibilizadas em um *component*?
- d) Que tipo de tarefas uma *service* pode ter?
  
- e) Verdadeiro ou Falso. Uma *service* precisa obrigatoriamente estar em pelo menos um módulo?  
Justifique sua escolha.
  
- f) Verdadeiro ou Falso. Uma *service* é do tipo de padrão de projeto **Singleton**?  
Justifique sua escolha.
  
- g) Escreva uma service que deverá ter os seguintes métodos e em cada um deverá escrever uma implementação básica (pode ser usado Arrays):
  - getUsers(): <any> {}
  - getUserById(userId: number): <any> {}
  - setUsers(users: any): <any> {}
  - deleteUser(userId: number): <any> {}

### Item 1.1: Injeção de Dependências

- a) O que é Injeção de Dependências?
- b) Verdadeiro ou Falso. Injeção de Dependências pode ser apenas de *services*?  
Justifique sua escolha.
  
- c) Verdadeiro ou Falso. O Angular quando vai instanciar uma classe de componente verifica se as injeções de dependência já não estão sendo usadas?  
Justifique sua escolha.

Dica:

<https://angular.io/guide/architecture-services>

<https://angular.io/guide/http>

<https://rxjs.dev/guide/overview>

<https://www.youtube.com/watch?v=pHFFBVPuapg>

## Item 2: HTTP Client

- a) O que é o protocolo HTTP?
- b) Cite outros tipos de comunicação com o backend e faça um breve resumo de cada.
- c) Que recursos o HTTP nos fornece?
- d) Para usar o HttpClient no Angular, como devemos fazer sua importação e injeção?
- e) Verdadeiro ou Falso. O HttpClient pode ser usado com RxJS?  
Justifique sua escolha.
- f) Cite os principais métodos HTTP e faça um breve resumo de cada.
- g) Verdadeiro ou Falso. Com o protocolo HTTP é possível dizer/setar o tipo de resposta do servidor backend?  
Justifique sua escolha.
- h) Cite os tipos de retorno que uma requisição HTTP pode ter? Faça um breve resumo de cada.
- i) Cite os principais status de uma requisição HTTP e faça um breve resumo de cada.
- j) Faça um exemplo de chamadas do tipo GET, POST, PUT, DELETE.
- k) Para que serve o cabeçalho em uma requisição HTTP?
- l) O que é um Interceptor e quais as suas aplicações?
- m) Quais cenários mais comuns podemos usar Interceptors?

Dica:

<https://angular.io/api/common/http/HttpClient>

<https://angular.io/guide/http#communicating-with-backend-services-using-http>

## Item 3: RxJS

- a) O que é o RxJS?
- b) Qual a diferença de **Promises e Observables**?
- c) O que significa ser **baseado em eventos**?
- d) O que é o padrão de projetos **Observer**?
- e) O que é o padrão de projetos **Iterator**?
- f) O que é **programação funcional** com coleções?
- g) Quais os conceitos básicos do RxJS para eventos assíncronos? Descreva todos e dê um exemplo de como utilizá-lo.
- h) Quais são os **operadores de criação**?
- i) Ainda dentro de **operadores de criação**, explique melhor e dê um exemplo para os seguintes operadores:
  - ajax;
  - from;
  - fromEvent;

- generate;
- of;
- interval;
- throwError;
- timer;

j) Quais os **operadores de criação de associação**?

k) Ainda dentro de **operadores de criação de associação**, explique melhor e dê um exemplo para os seguintes operadores:

- concat;
- forkJoin;
- merge

l) Quais os **operadores de transformação**?

m) Ainda dentro de **operadores de transformação**, explique melhor e dê um exemplo para os seguintes operadores:

- concatMap;
- map;
- mapTo;
- mergeMap;
- mergeMapTo;
- switchMap;
- switchMapTo;

Dica:

<https://rxjs.dev/guide/overview>