



Esta lista de perguntas teóricas tem como objetivo revisar os principais conceitos vistos durante o treinamento em .NET.

### Parte 01 - Conceitos

---

- 1) Defina Classes, Atributos e Métodos
- 2) O que são relacionamentos de Herança e Associação?
- 3) Defina sobrecarga de métodos
- 4) Defina construtores
- 5) O que são tipos genéricos. Exemplo: <T>
- 6) O que são métodos abstratos?
- 7) Qual o objetivo de criarmos interfaces (Contratos) em um projeto?
- 8) O que significa SOLID? Qual a importância do estudo e prática dos 5 princípios do SOLID?
- 9) Explique o que são métodos estáticos
- 10) O que é uma ConnectionString?
- 11) Explique o padrão MVC (Model, View e Controller)
- 12) O que é uma API? Explique o padrão REST
- 13) Explique para que servem os métodos HTTP POST, PUT, DELETE e GET
- 14) Qual o objetivo de uma camada Cross Cutting?
- 15) Explique para que servem classes de Repositório.
- 16) Explique para que servem classes de Entidade.
- 17) O framework .NET permite desenvolver projetos baseados em MVC e em API, explique as principais diferenças entre estes dois tipos de projetos.
- 18) Para uma separação total dos ambientes de FRONTEND e BACKEND qual a melhor solução? MVC ou API? Justifique sua resposta
- 19) O que ORM? Explique o conceito de Mapeamento Objeto Relacional
- 20) Explique a arquitetura DDD (Domain Driven Design) e suas camadas

### Parte 02 – Tecnologias

---

- 1) Explique a finalidade do framework Dapper
- 2) Em um projeto MVC, para que servem os métodos declarados nos controllers como IActionResult?
- 3) O que são DataAnnotations? Qual a sua função dentro do MVC ou API?
- 4) Para que serve a Annotation [Authorize]?
- 5) Para que serve o arquivo appsettings.json?
- 6) O que é @Razor? Como é utilizado em projetos MVC?
- 7) Em projetos API, o que são ENDPOINTS?

- 8) Explique o que é o formato JSON e qual o seu papel em projetos API.
- 9) Para que serve o Swagger?
- 10) O que é o framework JWT? Explique o tipo de autenticação BEARER AUTHENTICATION.
- 11) Explique a diferença entre os métodos POST, PUT, DELETE e GET em uma API.
- 12) Em projetos API, o que é a configuração de CORS?
- 13) Explique o que significa cada código do protocolo HTTP abaixo:
  - a. HTTP 200
  - b. HTTP 400
  - c. HTTP 500
  - d. HTTP 401
- 14) Explique a função da classe Startup.cs em um projeto .NET CORE MVC ou API
- 15) Explique o conceito de injeção de dependência e o uso da Annotation [FromServices]
- 16) Para que serve a função ModelState.IsValid?
- 17) O que são expressões LAMBDA?
- 18) Para que serve a classe SqlConnection do namespace System.Data.SqlClient?
- 19) Sobre Angular, explique a função de:
  - a. HttpClient
  - b. FormsModule
  - c. RouterModule
  - d. ngForm
  - e. ngModel
  - f. \*ngFor
  - g. Subscribe
- 20) Explique a função de uma MasterPage no projeto MVC
- 21) O que são StoredProcedures em banco de dados?
- 22) Sobre o EntityFramework defina o que são Mappings e DataContext
- 23) O que é o recurso de CodeFirst Migrations do EntityFramework

### Parte 03 – Demais assuntos:

---

- 1) O que são Testes Unitários, fale sobre o padrão TDD
- 2) Fale sobre as tecnologias mais “antigas” do framework .NET:
  - a. WindowsForms
  - b. WebForms
  - c. WPF
- 3) Diferencie JSON e XML.
- 4) O que é o Microsoft Azure?
- 5) Para que serve o GitHub