

## 1. JavaScript:

- Conceitos fundamentais (variáveis, funções, objetos, etc.)
- DOM Manipulation
- Eventos
- AJAX e Fetch API
- ES6 e posteriores
- Frameworks e bibliotecas (por exemplo, React, Vue.js)

**Projeto Prático:** Criar um aplicativo web simples usando JavaScript puro ou um framework como React ou Vue.js.

## 2. TypeScript:

- Introdução ao TypeScript
- Tipos e interfaces
- Classes e herança
- Decorators

**Projeto Prático:** Converter um projeto JavaScript em TypeScript e aproveitar a tipagem estática para identificar e corrigir erros.

## 3. Dart:

- Conceitos básicos da linguagem
- Tipos e variáveis
- Funções
- Classes e herança

**Projeto Prático:** Desenvolver um aplicativo móvel simples usando o Flutter, o framework que utiliza Dart.

## 4. Java:

- Conceitos básicos da linguagem (sintaxe, estruturas de controle, etc.)
- Programação orientada a objetos
- Exceções
- Coleções
- Threads

**Projeto Prático:** Criar um aplicativo de console ou uma aplicação web básica usando Java.

## 5. Angular 2+:

- Componentes e templates
- Diretivas e Pipes
- Roteamento
- Formulários reativos
- Serviços e injeção de dependência

**Projeto Prático:** Desenvolver um aplicativo web completo usando o Angular.

## 6. Spring Framework:

- Spring MVC
- Spring Boot
- Spring Data e JPA
- Configuração e injeção de dependência

**Projeto Prático:** Construir uma aplicação web usando o Spring Framework, que utilize o Spring Boot para simplificar a configuração.

## 7. Bootstrap:

- Layout responsivo
- Componentes pré-estilizados

**Projeto Prático:** Criar uma página web responsiva usando o Bootstrap para facilitar o layout e o design.

**Projeto Prático:** Implementar um projeto com o NGZorro e criar um formulário avançado com validações e elementos de UI personalizados.

## 9. Material UI:

- Componentes do Material UI

**Projeto Prático:** Construir uma aplicação web com o Material UI, aproveitando a estética e interações oferecidas pelo framework.

## 10. NGRX:

- Conceitos de gerenciamento de estado
- Store, Actions e Reducers

**Projeto Prático:** Implementar um aplicativo usando o NGRX para gerenciamento de estado.

## 11. Git:

- Controle de versão
- Branching e merging

**Projeto Prático:** Criar um repositório Git e colaborar em um projeto com outros colegas, usando pull requests e revisões de código.

## 12. Docker:

- Conceitos básicos de containerização
- Docker Compose

**Projeto Prático:** Dockerizar uma aplicação e configurar o ambiente usando o Docker Compose.

## 13. HTML e CSS:

- Fundamentos do HTML
- Estilização com CSS (seletores, layout, flexbox, grid)

**Projeto Prático:** Criar uma página web com layout responsivo e estilização adequada usando HTML e CSS.

## 14. JSON:

- Estrutura de dados em JSON

**Projeto Prático:** Consumir uma API que retorna dados em JSON e exibi-los em uma aplicação web.

## 15. Cookie e Local Storage:

- Armazenamento de dados no cliente

**Projeto Prático:** Criar um aplicativo que utilize cookies ou o local storage para armazenar preferências do usuário.

## 16. NPM e Node.js:

- Uso do NPM para gerenciamento de pacotes
- Conceitos básicos do Node.js

**Projeto Prático:** Desenvolver um pequeno utilitário ou aplicativo usando pacotes do NPM e funcionalidades do Node.js.

**Projeto Prático:** Integrar uma funcionalidade de mensageria e recursos de liberação com o Togglz em uma aplicação.

## 18. Autenticação, Autorização e Criptografia:

- Diferença entre autenticação e autorização
- Conceitos básicos de criptografia

**Projeto Prático:** Implementar um sistema de autenticação e autorização em um aplicativo web e usar técnicas de criptografia para proteger senhas.

## 19. MVC (Model-View-Controller):

- Conceitos e padrão de arquitetura

**Projeto Prático:** Desenvolver uma aplicação seguindo o padrão MVC para organizar a estrutura do projeto.

## 20. Arquitetura Limpa (Clean Architecture):

- Conceitos e benefícios
- Separação de camadas

**Projeto Prático:** Aplicar os princípios da Arquitetura Limpa em um projeto real, garantindo a separação adequada de camadas.

## 21. Programação Orientada a Objetos (POO):

- Encapsulamento, herança, polimorfismo
- Princípios SOLID

**Projeto Prático:** Implementar um conjunto de classes e objetos que utilizem conceitos de POO e aplicar os princípios SOLID quando apropriado.

## 22. Design Patterns (Padrões de Projeto GOF):

- Conhecer alguns padrões de projeto comuns

**Projeto Prático:** Identificar e aplicar padrões de projeto relevantes em um projeto real.

## 23. Testes Unitários, End to End e de Integração:

- Diferença entre os tipos de teste
- JUnit e ferramentas de teste

**Projeto Prático:** Escrever testes unitários e de integração para um projeto Java.

## 24. Teste de Mutação:

- Conceito de teste de mutação

**Projeto Prático:** Realizar testes de mutação em um projeto Java.

## 25. Banco de Dados:

- Postgres, MongoDB e Redis (conceitos e diferenças)

**Projeto Prático:** Utilizar cada banco de dados em um projeto específico, adaptando o modelo de dados e realizando operações básicas.

## **26. CI/CD (Integração Contínua e Entrega Contínua):**

- Conceitos e ferramentas (por exemplo, Jenkins)

**Projeto Prático:** Configurar

um pipeline de CI/CD para um projeto usando Jenkins ou outra ferramenta de automação.

## **27. Cloud Computing e Azure:**

- Conceitos básicos de Cloud Computing
- Conhecer os serviços do Azure

**Projeto Prático:** Hospedar uma aplicação na plataforma Azure ou utilizar serviços específicos da nuvem.

## **28. Sistemas Operacionais (Linux e Windows):**

- Comandos básicos do terminal
- Gerenciamento de processos

**Projeto Prático:** Configurar um ambiente de desenvolvimento em um sistema operacional Linux ou Windows.

## **29. Protocolo TCP/IP:**

- Compreensão dos conceitos do protocolo TCP/IP

**Projeto Prático:** Implementar uma aplicação de rede simples usando o protocolo TCP/IP.

**Projeto Prático:** Implantar uma aplicação em um servidor Payara ou Glassfish.

## **31. Wireframes, Grids, Cores, Tipografia e Componentização:**

- Design de interface e elementos de UI/UX
- Conceitos de componentização

**Projeto Prático:** Criar um conjunto de wireframes e implementar uma biblioteca de componentes reutilizáveis para um projeto de design de interface.

## 32. Acessibilidade, Usabilidade e Responsividade:

- Práticas para tornar o projeto acessível, usável e responsivo

**Projeto Prático:** Implementar melhorias de acessibilidade e usabilidade em um projeto web existente, tornando-o responsivo em diferentes dispositivos.