

Meses 1-4: Fundamentos de JavaScript

Semanas 1-12 (16 de Outubro - 8 de Janeiro): Fundamentos de JavaScript

- **16 a 20 de Outubro: Variáveis, Tipos de Dados e Operadores**
 - Conceitos: `let`, `const`, `var`, tipos primitivos, operadores aritméticos, lógicos e relacionais.
 - Prática: Criação de scripts que manipulem valores de diferentes tipos e utilizem operadores para cálculos simples.
- **21 a 25 de Outubro: Condicionais e Loops (for, while)**
 - Conceitos: `if`, `else if`, `else`, `switch`, `for`, `while`, `do while`.
 - Prática: Construção de scripts que tomem decisões baseadas em condições e repitam ações com loops.
- **26 a 30 de Outubro: Funções (declaração, expressões, funções de seta)**
 - Conceitos: Declaração de funções, parâmetros, retorno, funções anônimas e arrow functions.
 - Prática: Criar funções reutilizáveis que façam operações matemáticas e manipulem strings.
- **31 de Outubro a 4 de Novembro: Objetos e Arrays**
 - Conceitos: Arrays e objetos em JavaScript, manipulação de estruturas de dados complexas.
 - Prática: Construção de arrays e objetos e aplicação de métodos como `map`, `filter`, `reduce`.

- **5 a 9 de Novembro: Seleção de Elementos e Manipulação de DOM**

- Conceitos: Manipulação de elementos HTML usando JavaScript (ex.: `document.querySelector`, `getElementById`).
- Prática: Criação de interações básicas com páginas web, como alterar o conteúdo de elementos ou estilos.

- **10 a 14 de Novembro: Eventos e Manipulação de Eventos**

- Conceitos: Eventos do DOM (ex.: `click`, `keydown`, `submit`).
- Prática: Manipular eventos em formulários e botões, criando interações dinâmicas com a página.

- **15 a 19 de Novembro: Callbacks e Promises**

- Conceitos: Funções de callback e Promises para controle de fluxo assíncrono.
- Prática: Utilizar funções assíncronas simulando chamadas a APIs com `setTimeout` e `fetch`.

- **20 a 24 de Novembro: `async/await`**

- Conceitos: Introdução a `async` e `await` para simplificar Promises.
- Prática: Refatorar códigos de Promises para `async/await`, simulando requisições HTTP.

- **25 a 29 de Novembro: Estruturas de Dados (Pilhas, Filas, Listas Ligadas)**

- Conceitos: Estruturas de dados básicas e sua implementação em JavaScript.
- Prática: Criar pilhas, filas e listas ligadas para operações de inserção e remoção.

- **30 de Novembro a 4 de Dezembro: Algoritmos Básicos (Busca, Ordenação)**

- Conceitos: Algoritmos de busca (`linear`, `binária`) e ordenação (`bubble sort`, `merge sort`).
- Prática: Implementar algoritmos de ordenação e busca para arrays.

- **5 a 9 de Dezembro: Revisão e Prática**

- Revisar os principais conceitos e aplicar em projetos práticos de JavaScript.
-

Meses 5-8: Avançando no JavaScript e

Node.js

Semanas 13-24 (10 de Dezembro - 9 de Março): Conceitos Básicos de Node.js

- **10 a 14 de Dezembro: Configuração do Ambiente Node.js**
 - Instalação do Node.js, configuração de ambiente de desenvolvimento e uso básico de `npm`.
- **15 a 19 de Dezembro: Módulos e npm**
 - Uso de módulos internos (`fs`, `path`), criação de módulos personalizados e gerenciamento de pacotes com `npm`.
- **20 a 24 de Dezembro: Criando Servidores Básicos com HTTP**
 - Configurar um servidor HTTP simples com o módulo `http` do Node.js, respondendo a requisições básicas.
- **25 a 29 de Dezembro: Trabalhando com Arquivos e Streams**
 - Manipulação de arquivos (leitura/escrita) e uso de Streams para grandes volumes de dados.
- **30 de Dezembro a 3 de Janeiro: Planejamento do Projeto Node.js**
 - Definição dos requisitos para um projeto simples em Node.js.
- **4 a 8 de Janeiro: Desenvolvimento e Revisão do Projeto**
 - Implementação do projeto definido, revisão de código e correção de bugs.

- **9 a 13 de Janeiro: Instalação e Configuração do Express.js**
 - Configuração de rotas e middlewares básicos usando o framework Express.js.
- **14 a 18 de Janeiro: Criação de Rotas e Middleware**
 - Criação de rotas dinâmicas e implementação de middleware para manipulação de requisições/respostas.
- **19 a 23 de Janeiro: Manipulação de Requisições e Respostas**
 - Manipulação de dados nas requisições e envio de respostas com diferentes status e formatos.
- **24 a 28 de Janeiro: Criação de APIs RESTful com CRUD**
 - Implementação de uma API completa com operações CRUD (Create, Read, Update, Delete).
- **29 de Janeiro a 9 de Março: Revisão e Prática com APIs RESTful**
 - Revisão geral e aperfeiçoamento das APIs desenvolvidas, integrando com banco de dados e melhorias de performance.
- **29 de Janeiro a 2 de Fevereiro: Introdução à Autenticação Simples (JWT ou Token Simples)**
 - Aprenda como proteger rotas da sua API usando autenticação simples com JSON Web Tokens (JWT) ou tokens gerados manualmente.
 - Objetivo: Criar uma rota protegida que só pode ser acessada por usuários autenticados.

- **3 a 7 de Fevereiro: Validação de Dados nas Requisições**

- Explore como validar dados recebidos em suas APIs para garantir que tudo chegue corretamente. Você pode usar bibliotecas como **Joi** ou validar manualmente no Node.js.
- Objetivo: Criar validações para diferentes rotas (ex: cadastro de usuário).

- **8 a 12 de Fevereiro: Upload de Arquivos (Imagens e Documentos)**

- Aprenda a configurar o upload de arquivos em suas APIs, por exemplo, para salvar imagens de perfil ou documentos.
- Objetivo: Criar uma API que permita o upload e o armazenamento de arquivos no servidor.

- **13 a 17 de Fevereiro: Paginação e Filtros em APIs**

- Aprenda a implementar funcionalidades de paginação e filtros nas suas rotas para lidar com grandes volumes de dados.
- Objetivo: Criar uma API que retorne resultados paginados ou filtrados por critérios como data, nome, etc.

- **1 a 9 de Março: Revisão Geral e Melhoria de Projetos**

- Use este tempo para revisar tudo o que foi feito até agora, melhorar as funcionalidades que você criou e corrigir possíveis bugs.
 - Objetivo: Consolidar o aprendizado e melhorar seus projetos anteriores.
-

Meses 9-12: Fundamentos do React e React Native

Semanas 25-32 (10 de Março - 9 de Maio): React e React Native

- **10 a 14 de Março: Configuração do Ambiente React**
 - Instalação de ferramentas (Node.js, npm, Create React App) e configuração inicial do ambiente.
- **15 a 19 de Março: Componentes e JSX**
 - Criação de componentes funcionais e uso de JSX para renderização.
- **20 a 24 de Março: Estado e Propriedades**
 - Gerenciamento de estado e passagem de propriedades entre componentes.
- **25 a 29 de Março: Eventos e Manipulação de Eventos**
 - Captura e tratamento de eventos de usuários em componentes React.
- **30 de Março a 3 de Abril: Hooks Básicos (useState, useEffect)**
 - Introdução aos hooks para manipular estado e efeitos colaterais.
- **4 a 8 de Abril: Projeto Prático (Lista de Tarefas)**
 - Desenvolvimento de um projeto prático para consolidar conceitos, como uma lista de tarefas.
- **9 a 13 de Abril: Configuração do Ambiente React Native**
 - Instalação do ambiente para desenvolvimento de aplicativos móveis com React Native.

- **14 a 18 de Abril: Componentes e Estilização com React Native**
 - Criação de componentes e estilização para aplicativos móveis.
 - **19 a 23 de Abril: Navegação com React Navigation**
 - Implementação de navegação entre telas em um aplicativo React Native.
 - **24 a 28 de Abril: Integração com APIs e Armazenamento Local**
 - Integração com APIs e armazenamento local (ex.: AsyncStorage).
 - **29 de Abril a 9 de Maio: Revisão e Projeto Prático (Aplicativo de Lista de Compras)**
 - Criação de um aplicativo completo utilizando React Native, focado em interações com APIs e gerenciamento de estado.
-

Meses 13-16: Next.js

Semanas 33-40 (10 de Maio - 13 de Junho): Fundamentos e Desenvolvimento com Next.js

- **10 a 14 de Maio: Configuração do Ambiente Next.js**
 - Instalação e configuração de um projeto Next.js com SSR (Server-Side Rendering).
- **15 a 19 de Maio: Páginas e Rotas**
 - Criação de páginas dinâmicas e configuração de rotas estáticas e dinâmicas no Next.js.
- **20 a 24 de Maio: Renderização do Lado do Servidor**
 - Entendimento e implementação da renderização no lado do servidor com Next.js.
- **25 a 29 de Maio: Projeto Prático (Blog com Next.js)**
 - Desenvolvimento de um blog utilizando Next.js com SSR e rotas dinâmicas.
- **30 de Maio a 3 de Junho: Integração com APIs**
 - Consumo de APIs e integração com backend em um projeto Next.js.
- **4 a 8 de Junho: Otimização de Performance**
 - Técnicas de otimização de performance, como carregamento sob demanda e otimização de imagens.
- **9 a 13 de Junho: Segurança e Autenticação**
 - Implementação de autenticação com JWT, OAuth e boas práticas de segurança no Next.js.