

CRONOGRAMA DE PRATICA - REACT + TYPESCRIPT (30 DIAS)

Carga horária: ~1h por dia

Objetivo: Consolidar fundamentos e criar projetos funcionais e tipados.

SEMANA 1 - Fundamentos do React + TypeScript

- 1. Setup e ambiente - Criar um projeto com Vite + React + TypeScript. Renderizar um <h1> com seu nome.
- 2. Componentes funcionais - Criar um componente Header tipado que recebe title: string como prop.
- 3. Props e tipagem - Criar um componente Card que recebe title, description e imageUrl.
- 4. Estado (useState) - Criar um contador (Count.tsx) com tipagem explicita (number).
- 5. Eventos tipados - Criar um input controlado com onChange e tipagem de ChangeEvent<HTMLInputElement>.
- 6. Renderizacao condicional - Mostrar 'Carregando...' ate que um estado loaded seja true.
- 7. Mini-projeto - Pagina de perfil simples com componentes, estados e props.

SEMANA 2 - Listas, Hooks e Tipagem avancada

- 8. Listas e map() - Renderizar lista de tarefas tipadas (id, title, done).
- 9. Manipulacao de listas - Funcoes para adicionar e remover tarefas.
- 10. Tipos personalizados - Criar tipo Task com interface.
- 11. useEffect basico - Exibir mensagem no console ao montar o componente.
- 12. useEffect com dependencias - Criar contador que loga no console a cada atualizacao.
- 13. Comunicacao entre componentes - Passar funcoes como props (onAdd, onRemove).
- 14. Mini-projeto - To-Do List tipada com persistencia no localStorage.

SEMANA 3 - Hooks, Context API e boas práticas

- 15. useReducer - Criar contador com useReducer.
- 16. useContext basico - Criar ThemeContext (tema claro/escuro).
- 17. Tipagem de Context - Tipar provider (ThemeContextProvider).
- 18. Custom Hooks - Criar hook useLocalStorage tipado.
- 19. Union Types - Criar estados 'loading' | 'success' | 'error'.
- 20. Componentes genericos - Criar List<T> generica.
- 21. Mini-projeto - Sistema de login simulado com contexto e estado global.

SEMANA 4 - Integracoes, Routing e Projeto Final

- 22. React Router - Rotas: Home, About, Contact.

CRONOGRAMA DE PRATICA - REACT + TYPESCRIPT (30 DIAS)

- 23. Parametros de rota - Rota dinamica /users/:id.
- 24. Requisicoes HTTP - Buscar dados de API publica.
- 25. Tratamento de erros - Estados de erro e loading.
- 26. Layout - Criar layout com Header, Sidebar e Main.
- 27. ESLint - Corrigir alertas de tipagem.
- 28. Projeto final (parte 1) - Catalogo de usuarios com API, Context e Rotas.
- 29. Projeto final (parte 2) - Implementar busca, filtro e detalhe do usuario.
- 30. Revisao e Deploy - Revisar e publicar no Vercel.

Dicas complementares

- Use TypeScript estrito ('strict': true no tsconfig.json).
- Faça commits descritivos a cada exercício.
- Transforme exercícios em mini projetos pessoais.
- Após 30 dias, você terá base sólida para React profissional.