

Resumo das Aulas - Programacao para Dispositivos Moveis

Aula 01 Introducao a Programacao para Dispositivos Moveis

- Plataformas moveis: iOS, Android e Windows Mobile.
- UI/UX Design: Importancia do design centrado no usuario.
- Linguagens: Java, Swift, JavaScript e C#.
- Ambientes de desenvolvimento: Xcode, Android Studio, Visual Studio.
- Design: Material Design (Android) e Human Interface Guidelines (iOS).
- Tipos de desenvolvimento: Nativo, Multiplataforma (como React Native), e publicacao nas lojas de apps.

Aula 02 React Native, Ambiente de Desenvolvimento e Expo

- React Native: Framework para criar apps nativos usando JavaScript e React.
- Requisitos: Node.js, VS Code.
- Expo: Plataforma que facilita o desenvolvimento e teste de apps React Native.
 - Vantagens: Sem necessidade de codigo nativo, SDK rico, builds e publicacao facilitadas.
- Tipos de workflow: Gerenciado (sem codigo nativo) e Bare (com codigo nativo).
- JSX: Sintaxe parecida com HTML, usada para construir interfaces no React Native.

Aula 03 Conceitos Basicos do React e Componentes

- Props: Parametros passados entre componentes.
- State: Estado interno de um componente.
- Componentes basicos do React Native: Text, View, Button, etc.
- Exercicio pratico: Criar uma tela de login com base em um wireframe do Figma.

Aula 04 Navegacao em Apps React Native

- Tipos de navegacao:

- Stack Navigation: Pilha de telas (navegacao para frente e para tras).
- Tabs Navigation: Abas na parte inferior da tela.
- Drawer Navigation: Menu lateral.
- Uso da biblioteca React Navigation.

Aula 05 Consumo de APIs

- APIs: Interfaces que permitem comunicacao entre sistemas.
- HTTP Requests:
 - Fetch API: Nativa, precisa tratar manualmente JSON e erros.
 - Axios: Mais simples, melhor tratamento de erros.
- API para testes: JSONPlaceholder.
- Exercicio: Criar app que mostra o tempo de 5 cidades usando a API do OpenWeatherMap.

Aula 06 Armazenamento Local e Bibliotecas de Componentes

- AsyncStorage: Biblioteca para salvar dados localmente (ex: login, preferencias).
 - Boas praticas: Usar try/catch, nao guardar dados sensiveis, usar JSON.stringify() para objetos.
- Alternativas: MMKV (mais rapido), SQLite (estruturado), SecureStore (seguro).
- Bibliotecas de UI:
 - Gluestack
 - React Native Elements
 - Tamagui
- Exercicios:
 1. App de clima com adicao/remocao de cidades.
 2. App de tarefas com login e persistencia usando AsyncStorage.