**Aula 1**

**Ambiente de desenvolvimento**

Node.js (opcional: instalar pelo chocolately)

Visual Studio Code (extensões opcionais: Dracula Offcial, Material Icon Theme)

**BACKEND**

Iniciar Node: Entrar pelo terminal na pasta do projeto usar o comando npm init -y para criar o arquivo de configuração package.json

Instalar o framework espress (adiciona funcionalidades para o projeto como rotas, parametros e etc) através do comando npm install express

Criar o arquivo index.js. Para rodar o arquivo: node index.js

**FRONTENT**

ReactJS: Utilizar o NPX (parecido com npm mas executa um pacote sem necessidade de instalação) em uma pasta chamada frontend. Executar o comando npx create-react-app frontend. Testar no terminal com o comando npm start

**MOBILE**

App Expo pode ser baixado na play store para poder utilizar apis nativas como mapa, camera, etc..

**AULA 2**

**BACKEND**

Insomnia ou postman: Aplicativo para testar as rotas do backend

Tipo de parâmetros:

Query Params - Parâmetros nomeados enviados na rota após o simbolo “?” (users?nome=Jonathan)

Route Params - Parâmetros utilizados para identificar recursos (users/:id)

Request Body - Corpo da requisição, utilizado para criar ou alterar recursos (para a requisição consiga entender o JSON deve-se adicionar o código “app.use(express.json());”

Para não precisar cancelar e levantar o server toda hora instalar o nodemon - npm install nodemon -D (“-D“ instala como dependência apenas de desenvolvimento). Utilizar nodemon ao invés de node.

Banco de dados: SQLite.

Formas de conexão:

- Driver: SELECT \* FROM users

- **Query Builder**: table(‘users’).select(‘\*’).where()

KNEX - > npm install knex

SQLite -> npm install sqlite3

npx knex init

Criar migrate: npx knex migrate:make create\_ongs

Executar migrate: npx knex migrate:latest

Desfazer último: npx knex migrate:rollback

Criar pasta para os controllers

Autenticação fica no request.headers

reponse.status(401).json({error: ‘não permitido’}); - envia erro

Cors: Segurança, indica qual frontend pode acessar o backend

npm install cors

**AULA 3**

**FRONTEND**

**Limpando estrutura**

Deletar readme.md, app.css, app.test.js, index.css, logo.svg, serviceWorker.js, setupTests.js, robots.txt, manifest.json, logo.png,

Limpar o index.js e App.js pois não importará os arquivos deletados

**Conceitos do React**

Componente é uma função em JS que retorna um HTML. JSX (java script xml) é quando o HTML está integrado no javascript

Criar um componente Header.js, importar o react (import React from ‘react’;)

Propriedades: entre chaves {props.title}

Propriedade automática .children {props.children}

Estado: import {useState}

Criar pasta assets dentro da pasta src para guardar imagens

Crair pasta pages dentro da pasta src para os componentes

Icones: npm install react-icons

Configurar rotas: npm install react-router-dom

Criar routes.js na pasta raiz

Comunicação entre o frontend e backend: npm install axios

Criar o arquivo api.js dentro da pasta services

useHistory() voltar ou ir para rota específica

Classe javascript de internalização Intl

Intl.NumberFormat

**AULA 4**

**MOBILE**

Reactive Native.

Instalar o Expo: npm install -g expo-cli;

Rodar no terminal expo -h e iniciar o projeto: expo init mobile, selecionar template blank;

Rodar yarn start e vai abrir um qr code que deve ser aberto no celular pelo app Expo (baixar no celular), outra opção é rodar pelo simulador android;

Criar pasta src na raiz e pasta pages dentro da pasta src, cada componente tem sua pasta dentro da pasta page com os arquivos index.js e styles.css;

Rotas

Criar routes.js dentro da pasta src;

Npm install @react-navigation/native

Instalar as dependencias do expo

Expo install ... (procurar na web, são muitas. Reactnavigation.org - getting started)

Criar pasta assets dentro de src

Instalar axios para comunicação com backend

Instalar Intl para internalização moeda

**AULA 5**

**Funcionalidades avançadas**

Validação: npm install celebrate

Testes

TDD: Desenvolvimento dirigido a testes;

Npm install jest;

Npx jest --init;

Criar na raiz a pasta tests;

Criar na pasta tests as pastas units e integration;

Criar um arquivo .spec.js na pasta unit;

Npm install supertest -D ( D dependência apenas em desenvolvimento)

Deploy

Backend

Heroku - plano grátis para testes;

Digital ocean - aplicação comercial pequena;

(Azure, AWS amazon, Google Cloud) - aplicação grande

Frontend

Netlify