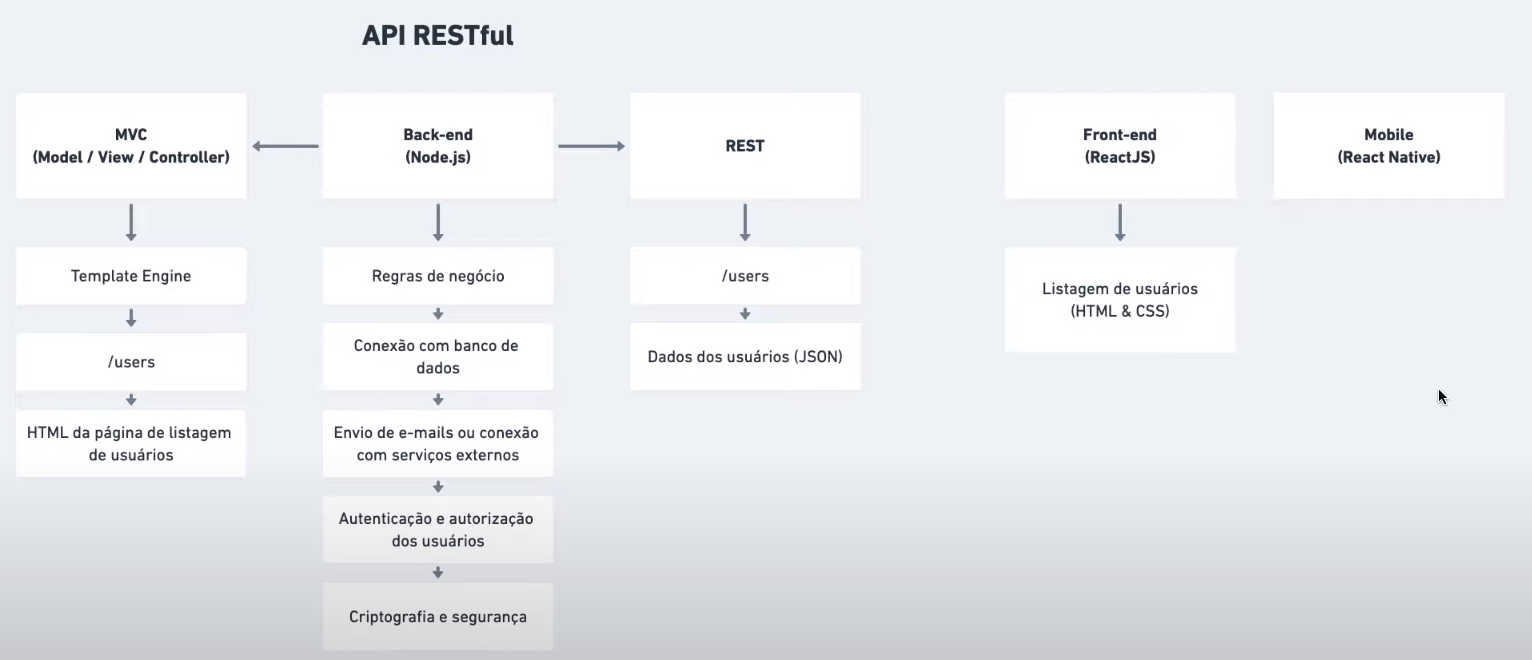
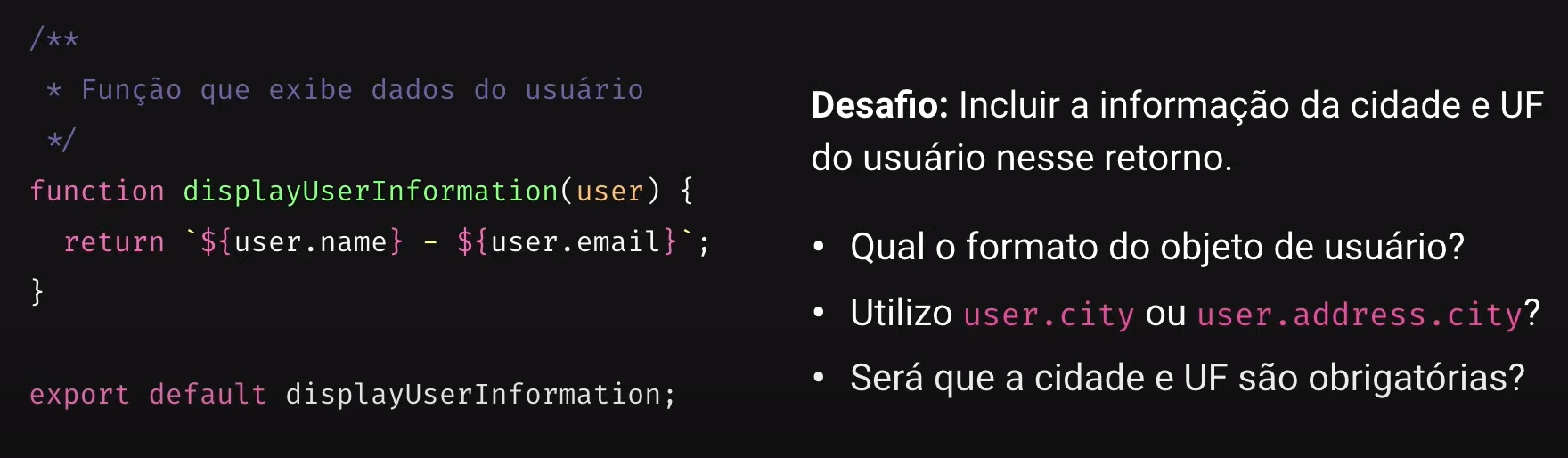
NLW a nova semana omnistack



Porque type script?



Permite o IntelliSense

Iniciando o prjeto :

**λ npm init -y**

Instalar o express :

**λ npm install express**

Instalar o type express,apenas como dependência de desenvolvimento :

**λ npm install @types/express -D**

Instalar o typescript:

**λ npm install typescript -D**

Instalar o transpilador do TS no node, em dev

**λ npm install ts-node -D**

Criar o arquivo de parametos o TS

**λ npx tsc –- init**

Iniciando a aplicação

**λ npx ts-node src/server.ts**

Tipo nodemon

**λ npm install ts-node-dev -D**

iniciando a aplicação com o

**λ npx ts-node-dev src/server.ts**

Incluir script no package.json

"scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

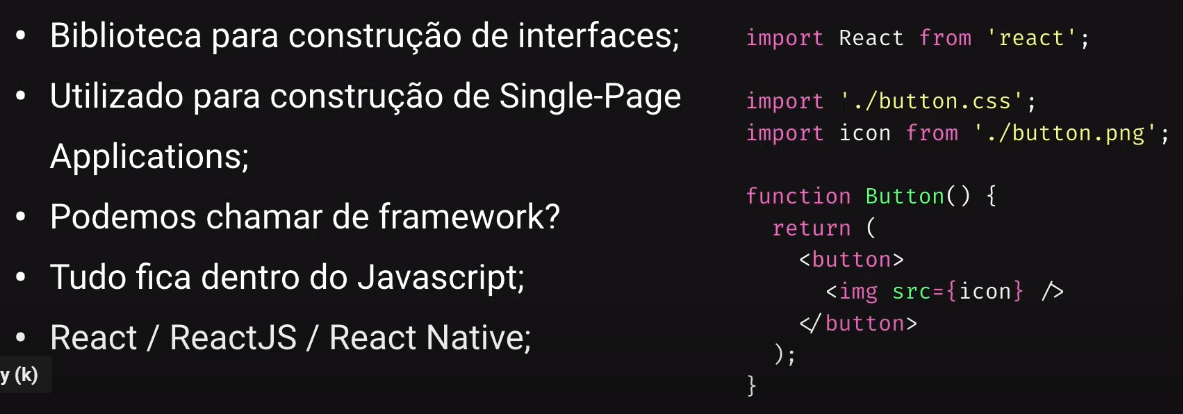
**"dev": "ts-node-dev  src/server.ts"**

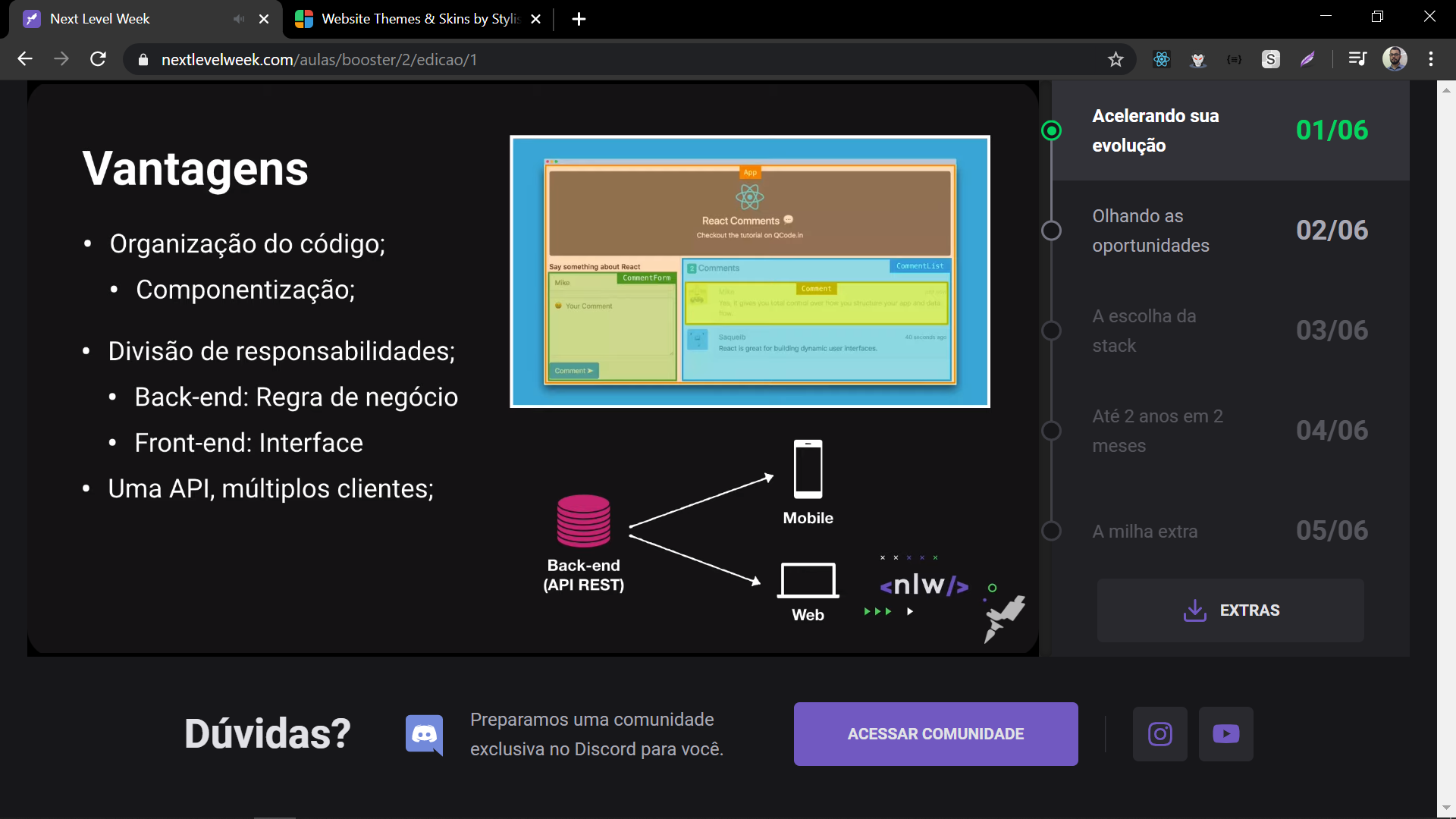
**},**

Para rodar a aplicação:

**λ npm run dev**

**REACT**





Criar o projeto react

**λ npx create-react-app web --template=typescript**

**REACT NATIVE**

|  |  |
| --- | --- |
|  |  |

**Expo**

**Pq usar :**

****



ROTA : localhost:3333/users

RECURSO: users

**BANCO DE DADOS**

Orm KNEX

**λ npm install knex**

Banco de dados SQLite

**λ npm install sqlite3**

Arquivo de conexão

import knex from 'knex'

import path from 'path'

const connection = knex({

    client: 'sqlite3',

    connection:{

        filename:path.resolve(\_\_dirname,'database.sqlite')

    } ,

    useNullAsDefault: true,

})

export default connection;

arquivo de migration 00\_create\_points.ts

import Knex from 'knex'

export async function up(knex:Knex){

    return knex.schema.createTable('points',table => {

        table.increments('id').primary();

        table.string('image').notNullable();

        table.string('name').notNullable();

        table.string('email').notNullable();

        table.string('whatsapp').notNullable();

        table.decimal('latitude').notNullable();

        table.decimal('longitude').notNullable();

        table.string('city').notNullable();

        table.string('uf').notNullable();

    });

}

export async function down(knex:Knex){

    return knex.schema.dropTable('points');

}

Arquivo de configurações do Knex

import import path from 'path'

module.exports = {

    client: 'sqlite3',

    connection:{

        filename:path.resolve(\_\_dirname,'src', 'database', 'database.sqlite')

    },

    migrations:{

        directory: path.resolve(\_\_dirname,'src', 'database', 'migrations')

    },

    seeds:{

        directory: path.resolve(\_\_dirname,'src', 'database', 'seeds')

    },

    useNullAsDefault: true,

};

Executando as migrations

**λ npx knex migrate:latest --knexfile knexfile.ts migrate:latest**

inserido o script no package.json:

"knex:migrate":"knex migrate:latest --knexfile knexfile.ts migrate:latest"

Para executar:

**λ npm run knex:migrate**

**seeds**

**criar arquivo com inserts**

import Knex from 'knex'

export async function seed(knex:Knex){

    await knex('items').insert([

        {title:'Lâmpadas',image:'lampadas.svg'},

        {title:'Pilhas e Baterias',image:'baterias.svg'},

        {title:'Papéis e Papelão',image:'papeis-papelao.svg'},

        {title:'Resíduos Eletrônicos',image:'eletronicos.svg'},

        {title:'Resíduos Orgânicos',image:'organicos.svg'},

        {title:'Óleo de Cozinha',image:'oleo.svg'},

    ]);

}

**Inserir o caminho da pasta seeds no knexfile**

**Adicionar script no package.json**

"knex:seed":"knex --knexfile knexfile.ts seed:run"

**Para executar :**

**λ npm run knex:seed**

**CORS**

Para instalar + types:

**λ npm install cors**

**λ npm install @types/cors -D**

**importar no server.ts**

**multer**

**λ npm install multer**

**λ npm install @types/multer**

**Celebrate**

**λ npm install celebrate**

**λ npm install @types/hapi\_\_joi -D**

**Aplicação WEB**

**Pesquisar : imagem extensão svg**

**Conceitos React:**

**- JSX**

**- Componente**

**- Propriedade**

**- Estado e imutabilidade**

**Fontes do google**

**Instalar ícones :**

**λ npm install react-icons**

**para transitar entre as paginas, é preciso criar as rotas.**

**Instalar:**

**λ npm install react-router-dom**

**λ npm install @types/react-router-dom -D**

**para usar o mapa:**

**instalar o leaflet**

**λ npm install leaflet react-leaflet**

**λ npm install @types/react-leaflet -D**

**importar o css em**

<https://leafletjs.com/examples/quick-start/>

carregar os itens da api

instalr axios

**λ npm install axios**

**API do IBGE**

https://servicodados.ibge.gov.br/api/v1/localidades/estados

https://servicodados.ibge.gov.br/api/v1/localidades/estados/{UF}/municipios

**instalar o filearea: react-dropzone**

**λ npm install react-dropzone**

**App Mobile**

**Instalar o expo**

**λ npm install -g expo-cli**

**iniciar a plicação:**

**λ expo init mobile**

**? Choose a template: expo-template-blank-typescript**

**Instalando as fontes ubuntu e roboto**

**λ expo install expo-font @expo-google-fonts/ubuntu @expo-google-fonts/roboto**

**instalar o React navigation**

**λ** **npm install @react-navigation/native**

**λ** **expo install react-native-gesture-handler react-native-reanimated react-native-screens react-native-safe-area-context @react-native-community/masked-view**

**controle de navegação em pilha**

λ npm install @react-navigation/stack

**tem que parar a app(expo) para fazer as instalações.**

**mapas**

λ expo install react-native-maps

λ expo install expo-constants

**pacote para o react trabalhar com imagens svg**

λ expo install react-native-svg

**instalr o axios**

λ npm install axios

**ter acesso a localização:**

λ expo install expo-location

**Mail composer:**

λ expo install expo-mail-composer

**GIT**