Nome: Natã Sato Rodrigues

SITUAÇÃO DE APRENDIZAGEM

- **O que é o NodeJS?**

O Node.js pode ser definido como um **ambiente de execução Javascript *server-side***.

Isso significa que com o Node.js é possível criar aplicações Javascript para rodar como uma aplicação *standalone(*programas completamente autossuficientes, sem ter um software auxiliar*)* em uma máquina, não dependendo de um browser para a execução, como estamos acostumados.

Apesar de recente, o Node.js já é utilizado por grandes empresas no mercado de tecnologia, como Netflix, Uber e LinkedIn.

O principal motivo de sua adoção é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de Microsserviços e componentes da arquitetura *Serverless*. Inclusive, os principais fornecedores de produtos e serviços Cloud já têm suporte para desenvolvimento de soluções escaláveis utilizando o Node.js.

- **O que é o NPM?**

O NPM é uma ferramenta do Node.js para o gerenciamento de pacotes. Ele permite instalar, desinstalar e atualizar dependências em uma aplicação por meio de uma simples instrução na linha de comando. Sempre que um projeto é criado por meio do gerenciador, é adicionado um arquivo chamado package.json, que contém a relação dos pacotes instalados no ambiente. Assim, quando for preciso realizar alguma alteração, o NPM verifica esse arquivo e faz as atualizações necessárias de forma simples e rápida. Isso contribui para manter a organização do projeto e de suas dependências, além de evitar erros de configurações ao fazer a instalação de pacotes de forma manual.

- **Qual o comando para iniciar um projeto em Node?**

Após a instalação do Node com o NPM o passo seguinte é justamente **criar o seu projeto utilizando o npm**.

Para isso basta abrir o seu prompt de comandos ou terminal, criar a pasta onde ficará o seu projeto e digitar o comando:

npm init.

Após executar o npm init, o npm vai perguntar algumas informações básicas para montar o package.json. São elas:

- **package name**: Nome do projeto. (Padrão: nome da pasta onde o comando foi executado)
- **version**: Versão do projeto. (Padrão: 1.0.0)
- **description**: Uma descrição para o projeto.
- **entry point**: Arquivo padrão que será utilizado para executar a aplicação. (Padrão: index.js)
- **test command**: Comando para executar os testes da aplicação.
- **git repository**: URL do repositório git onde o código-fonte da aplicação será armazenado.
- **keyword**: Palavras-chave relevantes para ajudar as pessoas a encontrarem o seu projeto.
- **author**: Autor do projeto.
- **license** Tipo de licença do projeto. (Padrão: ISC)

Após informar todos esses dados o programa apresenta no console o conteúdo do arquivo package.json que será criado com as informações que passamos. Para confirmar a criação do arquivo para responder yes.

Agora é só criar um projeto index.json na pasta raiz onde está o packge.json e programar!

- **Qual o comando para instalar uma dependência no projeto?**

O comando para instalar uma dependência no projeto é

npm i --save.

Quando você instala um módulo utilizando a flag --save, o módulo é salvo em dependências, dentro do package.json.

Por exemplo: no nosso projeto, nós iremos utilizar o **express** e o **mongoose**. O **Express** é uma web framework que irá nos ajudar a facilitar nosso trabalho ao trabalhar com aplicações web no **NodeJS**. Já o

mongoose é um módulo que vai nos ajudar a modelar nossa base de dados, criada em **Mongo DB**. Para instalá-lo, você vai utilizar o comando:

```
npm i --save express mongoose
```

E ele fica salvo no package.json, assim:

```
"dependencies" : {
  "express": "^4.12.2",
  "mongoose": "^3.8.24"
}
```

- **Para que serve o ExpressJS?**

ExpressJS é um popular framework web estruturado, escrito em JavaScript que roda sobre o ambiente node. js em tempo de execução. Este módulo explica alguns dos principais benefícios deste framework, como configurar o seu ambiente de desenvolvimento e como executar tarefas comuns de desenvolvimento e implantação da web. Ele cria abstrações de rotas, middlewares e muitas outras funções para facilitar a criação tanto de API's quanto SPA's.

Um exemplo bacana de uso dele é a exposição de uma API simples de get que pode ser feita com poucos cliques em menos de 10 minutos. O Express está voltado para a criação e obtenção dos dados a partir do seu servidor. Independente da linguagem que os irá utilizar.

- **O que é o ReactJS?**

O **React** é a biblioteca mais popular do **JavaScript,** ele é baseado em componentes, o que permite o reaproveitamento de código e facilita a manutenção. No padrão de arquitetura MVC — Model View Control — ou Modelo Visão Controle, em português, é comparado ao desenvolvimento da camada View, que é a interface com o usuário (UI). Ela oferece uma resposta excelente para o usuário adicionar comandos usando um novo método de renderizar sites.

Os componentes dessa ferramenta foram desenvolvidos pelo Facebook. Ela foi lançada em 2013 como uma ferramenta JavaScript de código aberto.

Além de receber atualizações do Facebook, conta com uma grande comunidade ativa, com mais de 1.300 colaboradores no GitHub que ajudam a aprimorar o código e usada por mais de 3,4 milhões de projetos. Atualmente, ela permanece na frente das suas principais competidoras, como a Angular e a Bootstrap, as duas bibliotecas JavaScript mais bem vendidas.

- **O que é uma SPA?**

Um SPA é uma aplicação web que roda em uma única página, se assemelhando a um aplicativo desktop ou um mobile, são leigamente chamadas de “páginas ajax”, um bom exemplo que gosto de usar é o Gmail do Google, ele é um SPA, a navegação na aplicação rola toda em uma única página e todo o conteúdo é carregado de uma vez ou obtido dinamicamente (ou seja, via requisições Ajax).

A aplicação SPA pode ser construída de diversas formas, a mais comum é ser auxiliado por um framework Javascript como Angular ou Vue.js, ambos possuem sistemas de rotas e clientes HTTP para fazer requisições a recursos externos (uma API, por exemplo).

- **O que são componentes e quais as vantagens de componentizar o front-end?**

Um componente é algo que sozinho tem um sentido, ele pode ser único, ou um conjunto de vários outros componentes.

Vamos visualizar o para-brisa, que é um componente único, e temos a exata noção de sua funcionalidade.

Agora vamos analisar a roda do carro, aqui temos um conjunto de outros componentes, dos quais podemos citar: o pneu, o aro e os parafusos. Cada um tem um sentido, porém é junto que podem desempenhar sua funcionalidade.

O que estou querendo mostrar com esses exemplos, é que podemos ter um paralelo entre o carro e nossas páginas. O carro é montado a partir de conjunto de componentes e, quando olhamos para ele, podemos identificar cada um deles. E na consonantização é isso que fazemos com nossas

páginas, desenvolvendo-as a partir de outros componentes, possibilitando com isso, que ao analisarmos o código, identifiquemos cada um deles.

Para se ter um projeto bem componentizado, é necessário treinarmos nossa abstração, para assim conseguirmos identificar um componente perdido no meio da nossa página.

Benefícios da componentização.

- Quando nos utilizamos da componentização, temos bastante ganhos agregados, dos quais podemos citar: **a reutilização de trechos de código; o isolamento de contexto; a legibilidade do código; a redução das tags das páginas e a padronização do projeto; Facilidade na convergência tecnológica através do uso de tecnologia orientada a componentes/serviços; colabora na integração de informações entre vários canais de acesso.**

Fontes:

https://www.opus-software.com.br/node-js/

https://rockcontent.com/br/blog/npm/

https://dicasdejavascript.com.br/como-criar-um-projeto-nodejs-com-npm/

https://blog.da2k.com.br/2015/03/03/gerenciando-corretamente-dependencias-em-nodejs-save-ou-save-dev-1/#:~:text=Pode%20usar%20somente%20npm%20i,dependencies%20e%20devDependencies%20pra%20você.

https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs

https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript

https://blog.schoolofnet.com/o-que-e-uma-spa-single-page-application/

https://inside.contabilizei.com.br/componentização-no-front-end-3fc889a363df