Roteiro do podcast

- efeito sonoro

Introdução : sejam bem vindos ao nosso podcast star temp. Hoje vamos apresentar a vocês um dos frameworks mais usados quando falamos de java script ... O node js.

- nesse podcast, entraremos de cabeça no mundo dos frameworks, falando da sua definição, importância, suas funções, seus diferentes tipos, e mais especificamente sobre o node js.

-vamos lá? (efeito sonoro)

- afinal o que é um framework? Um framework é um conjunto de componentes que auxiliam o usuário a desenvolver um software de uma forma mais simples e eficaz. Eles são importantes para ajudar o desenvolvedor a definir a estrutura do projeto fornecendo as ferramentas para esse processo de criação.

- já sobre suas funções, podemos dizer que um simplificador em questão do desenvolvimento do programa, o que aumenta a produtividade e melhora a qualidade do código, facilitando assim a manutenção a longo prazo e garantindo segurança ao código gerado.

- Existem diferentes tipos de frameworks, e vamos abordar sobre isso agora (efeito sonoro).

- Vamos começar com o Angular que é muito famoso no front-end, ele é uma opção excelente se o que procura é um padrão mais usado que consta com uma comunidade enorme.

- Temos também o ASP.NET core, o padrão da Microsoft que é focado em modernizar o desenvolvimento web.

- Já o flask é um framework pensado para o python que atua no back-end das aplicações web, conhecido pela sua simplicidade e velocidade de funcionamento.

- Em penúltimo temos o Django que é alternativa para lidar com python no back-end, permitindo a manipulação de bancos de dados , a autenticação do usuário, feed RSS.

- Agora vamos embarcar no framework escolhido do podcast de hoje [...] o Node.js com Express.js

O JavaScrip é uma linguagem de programação com foco no front-end, ou seja, ele é executado no navegador, no “client side”, mas com o passar do tempo se tornou possível rodar o JavaScrit no back-end, no “server side”, e é nesse momento que surge o node.js.

O node é uma plataforma de desenvolvimento que permite a execução de código JavaScript no servidor. é um ambiente de execução que na prática se reflete na possibilidade de criar aplicações em uma máquina servidora, sem a necessidade do navegador, e trabalhar de forma a não bloquear a aplicação no momento de sua execução, colocando os processos mais demorados para um segundo plano.

Ele funciona da seguinte forma, quando executamos um comando escrito em JavaScript, o Node.js interpreta esse comando e faz a sua conversão para a linguagem de máquina a ser executada pelo computador. Por esse motivo, o Node.js também pode ser referido como um JavaScript Runtime, ou um programa de execução do JavaScript.

Uma característica marcante desse framework o conceito de módulo, que visa organizar o código em partes pequenas e especializadas, além de aplicar o princípio de encapsulamento ao “esconder” o código e expor somente o necessário a outras partes da aplicação.

(efeito sonoro)

- Entre os diversos frameworks populares que utilizam Node.js, vamos citar aqui o Express.js, que é muito utilizado para construir APIs e aplicações web. Ele fornece uma variedade de recursos, incluindo roteamento, onde é possível mapear URLs para funções de manipulação de solicitações, permitindo que você defina como o aplicativo responde a solicitações HTTP específicas, middleware, que são funções usadas para executar tarefas comuns, como registro de solicitações, análise de corpo de solicitação e manipulação de solicitações e respostas HTTP. Com o Express, os desenvolvedores podem criar APIs RESTful, sites dinâmicos e muito mais, de forma rápida e eficiente.

- O Node.js é uma ferramenta de código aberto e gratuita. Ela também é multiplataforma, o que nos permite entregar uma solução para rodar em ambientes Windows, Linux ou MacOS, e seu interpretador é baseado no V8 da Google (um interpretador JavaScript desenvolvido pela Google e utilizado pelo Chrome, com o objetivo de acelerar a execução de aplicações desenvolvidas em JavaScript).

Para essa instalação, nós vamos usar a versão LTS (Long-term support, ou suporte de longo prazo, em português) porque é a mais estável. A versão Current é a mais atualizada, porém pode conter alguns bugs.

No processo de instalação do Node.js nos é entregue também o NPM (Node Package Manager), que vai nos permitir compartilhar os módulos que criamos com a comunidade além de possibilitar a reutilização de módulos criados por outras pessoas desenvolvedoras. NPM é uma tecnologia responsável pelo gerenciamento e execução desses pacotes e dependências. Através do NPM podemos gerenciar todas dependências dos projetos JavaScript que estamos desenvolvendo, com ele temos acesso a uma gama enorme de bibliotecas e frameworks JS.

- (efeito sonoro) É necessário passar por um processo simples para a instalação do Node.js, mostraremos como isso deve ser feito em Windows e MacOS.

- Vamos para a instalação no Windows, começamos acessando a página de download do Node.js, ao clicar na opção Windows Installer da versão LTS, será iniciado o download automático do pacote instalador. Quando o download estiver concluído, basta clicar no instalador que foi baixado para que ele seja executado. Uma janela será exibida e iremos prosseguir clicando em Next, chegando à tela de licenças. Aqui basta preencher a checkbox para concordar com os termos de uso e avançar com o clique em Next novamente. Na tela seguinte, vai aparecer a opção de alterar o local em que o Node será instalado. Optamos por manter o caminho padrão, clique em Next. A seguir, serão escolhidas quais as funcionalidades do Node queremos que sejam instaladas. A recomendação é que não altere nada e faça a instalação de todas, mas caso queira alterar essa configuração, é importante garantir que ao menos as opções Node.js runtime, npm package manager e Add to PATH estejam selecionadas. Em seguida, vai aparecer a opção de instalar as ferramentas para módulos nativos. Em algumas situações específicas, alguns pacotes do npm precisam ser compilados em sua máquina para rodarem no Windows sem impedimentos. Para isso, é necessário que essas ferramentas para módulos nativos estejam instaladas, pois elas irão cuidar da compatibilidade com o Windows. Portanto, para evitar problemas futuros, selecione a checkbox para que seja feita a instalação automática dessas ferramentas. Feito isso, basta selecionar a opção install, que vai aparecer na tela seguinte, e aguardar até que a instalação seja finalizada. Nesse momento, pode ser que algumas janelas do terminal se abram, que é justamente a responsável pela instalação das ferramentas para módulos nativos. Basta clicar em qualquer tecla para continuar e esperar até que seja finalizada. Assim que a instalação terminar, é interessante que você reinicie a máquina, para garantir que Node.js foi escrito corretamente no caminho do PATH do Windows. Mas antes de reiniciar tenha certeza de que a instalação terminou. Por fim, para verificar se a instalação ocorreu corretamente, basta abrir o Prompt de Comando e rodar os comandos node --version. Para verificar a versão do gerenciador de pacotes para Node (o NPM) rode os comandos npm –version. Agora basta digitar node, e você já pode executar JavaScript através do terminal. E a instalação do Node.js foi concluída, além da verificação que ele foi instalado corretamente no Windows.

- (efeito sonoro) A instalação no macOS é bastante similar com a instalação do Windows, pois também usa um pacote de instalação. Inicialmente, vamos até o site de download do Node. As versões LTS já estão disponíveis por padrão. Então, selecionamos a opção macOS Installer (.pkg). Ao selecionarmos essa opção, o pacote de instalação será baixado. Assim que o download terminar, clicamos no arquivo baixado e executamos. Depois, podemos prosseguir ao clicar em \*Continue\*, sem nos esquecermos de aceitar os termos de uso. Clique em \*Close\* e agora vamos conferir se o Node e o npm foram realmente instalados, comece digitando node --version e npm –version, se o prompt responder com suas respectivas versões, mostra que conseguimos concluir a instalação do Node e do npm. E pronto! Já instalamos.

- (efeito sonoro) E agora? Como posso utiliza-lo? O Node.js pode ser usado em APIs, aplicações web em tempo real como chats ou ferramentas de múltiplos usuários, jogos multiplayer, aplicações que demandam alta escalabilidade e servidores de streaming de dados. Empresas como LinkedIn, Walmart, PayPal, Uber e Netflix, usam o Node.js em suas aplicações, destacando benefícios como escalabilidade, velocidade de entrega e performance excepcional. Além disso, esse framework é multiplataforma, open source e multi-paradigma, ou seja, podemos criar desde soluções web a aplicações desktop, o Node.js tem seu código aberto, o que nos possibilita contribuir com o projeto e até mesmo realizar customizações e com o JavaScript podemos adotar várias formas de codificar usando paradigmas como a orientação a objetos, o funcional e o orientado a eventos.

- Outras grandes vantagens são a facilidade no desenvolvimento rápido de aplicações backend, oferta de recursos pré-construídos, como roteamento, manipulação de solicitações HTTP e autenticação, há comunidades ativas de desenvolvedores, o que pode facilitar o suporte e a resolução de problemas, a escalabilidade e o gerenciamento eficiente de recursos.

- (efeito sonoro) Olá, ouvintes! Chegamos ao final do nosso episódio sobre o Express.js, e vamos recapitular os pontos mais importantes.

Express.js é um framework minimalista e flexível para Node.js, que tem ganhado destaque no desenvolvimento de aplicações web e APIs. Ele permite que desenvolvedores criem servidores robustos com facilidade, gerenciem rotas e middlewares de forma eficiente, e integrem diversos módulos para estender suas funcionalidades.

Então, se você ainda não experimentou o Node.js, vale a pena dar uma chance e ver como ele pode transformar seu desenvolvimento web. Espero que tenham gostado do episódio de hoje! Até a próxima!.