



Centro Universitário de Excelência

SISTEMAS DE INFORMAÇÃO

DAVID MARTINS DOS SANTOS - 240902022

PORTFÓLIO
ANGULAR FRAMEWORK

DAVID MARTINS DOS SANTOS

**PORTFÓLIO
ANGULAR FRAMEWORK**

Trabalho apresentado ao Curso Sistemas de informação
do Centro Universitário ENIAC para a disciplina
Framework angular.

Prof. Lucio Luzetti Criado

Guarulhos

2024

ESTUDO DE CASO

Nos últimos anos, o desenvolvimento de aplicativos web tem sido impulsionado por uma ampla variedade de ferramentas e frameworks, proporcionando experiências interativas e dinâmicas aos usuários. Entre essas tecnologias, o Angular e o Node.js emergiram como pilares fundamentais, revolucionando a maneira como os projetos front-end são concebidos e implementados.

O Angular, um poderoso framework de código aberto desenvolvido pelo Google, tem ganhado destaque pela sua robusta arquitetura e recursos abrangentes. Com sua capacidade de criar aplicativos web escaláveis, o Angular tornou-se a escolha ideal para desenvolver projetos complexos com facilidade e eficiência.



ATIVIDADE

1. Estrutura do Projeto Angular para Gerenciamento de Tarefas

Para atender à missão de criar um aplicativo web para gerenciamento de tarefas,

organizaremos a estrutura do projeto Angular de forma a garantir sua escalabilidade e eficiência.

Diretórios e Arquivos:

src/: Diretório principal do projeto.

- app/: Diretório que contém os componentes, módulos e serviços principais do aplicativo.
 - components/: Diretório para componentes reutilizáveis.
 - task-list/: Componente responsável por exibir a lista de tarefas.
 - task-form/: Componente para adicionar e editar tarefas.
 - services/: Diretório para serviços do aplicativo.
 - task.service.ts: Serviço para lidar com operações de CRUD de tarefas.
 - models/: Diretório para definição de modelos de dados.
 - task.model.ts: Modelo de dados para uma tarefa.
 - app.component.html: Template principal do aplicativo.
 - app.component.ts: Lógica do componente principal.
 - app.module.ts: Módulo principal do aplicativo.
- assets/: Diretório para recursos estáticos como imagens e arquivos de estilo.
- index.html: Página HTML principal do aplicativo.
- styles.css: Arquivo CSS global para estilos gerais.

Relacionamento entre Diretórios e Componentes:

- O componente principal (app.component) é responsável por carregar os outros componentes conforme necessário e coordenar suas interações.
- Os componentes de task-list e task-form são utilizados para exibir a lista de tarefas e adicionar/editar tarefas, respectivamente.
- O serviço task.service é injetado nos componentes para realizar operações de CRUD no banco de dados.

2. Relação entre Node.js e Angular

O Node.js é utilizado no contexto do desenvolvimento front-end com o Angular principalmente para configuração e execução do projeto. O Angular é um framework front-end, e o Node.js, com seu ambiente de tempo de execução JavaScript, fornece uma maneira eficiente de servir arquivos estáticos, gerenciar dependências e executar tarefas de compilação.

Funcionalidades Específicas do Node.js Vantajosas para o Angular:

- **Servidor HTTP:** O Node.js pode ser usado para criar um servidor HTTP que serve os arquivos estáticos do aplicativo Angular durante o desenvolvimento.
- **Gerenciador de Pacotes NPM:** O Node.js vem com o NPM, um gerenciador de pacotes poderoso que facilita a instalação de dependências do projeto Angular.
- **Ferramentas de Compilação e Minificação:** O Node.js pode ser usado em conjunto com ferramentas como Webpack ou Gulp para compilar, transpilar e minificar o código Angular.

Importância na Configuração e Execução do Projeto:

- O Node.js é essencial na configuração do ambiente de desenvolvimento, fornecendo as ferramentas necessárias para instalar dependências, configurar scripts de compilação e iniciar um servidor de desenvolvimento.
- Durante a execução do projeto, o Node.js é usado para servir os arquivos estáticos do Angular e fornecer uma infraestrutura robusta para o desenvolvimento e testes.

Conclusão

Durante o desenvolvimento deste projeto, enfrentei diversos desafios, desde a concepção da arquitetura até a implementação das funcionalidades. A utilização do Angular como framework front-end proporcionou uma estrutura sólida e organizada para o desenvolvimento do aplicativo de gerenciamento de tarefas, enquanto o Node.js desempenhou um papel fundamental na configuração e execução do projeto. A integração entre essas tecnologias permitiu criar uma aplicação web escalável, responsiva e de fácil manutenção. Este projeto demonstrou a importância de escolher as ferramentas certas para alcançar os objetivos do desenvolvimento web moderno.