**Fatec franca**

**Avaliação Tópicos especiais**

Typescript

1) por padrão os navegadores não conseguem entender a sintaxe do typescript, apenas o javascript, por isso a necessidade da transpilação (conversão) do typescript para o javascript.

2)concurrently é uma dependência que possibilita executar uma série de scripts criados no package.json simultaneamente. No caso do nosso projeto em específico, foi feito a transpilação e em seguida a execução desse código transpilado.

3)primeiro foi configurado o typescript (tsconfog.json), em seguida criou três diretórios modelos, controles e interfaces e um arquivo principal onde contém as configurações dos middlewares (interceptores de requisição) e o servidor que sobe a API. Os modelos serve como um espelho para o banco de dados, ou seja, os mesmos atributos do modelo deve ser respeitado conforme os do banco. O controle ou controlador (no caso desse projeto) foi separado como o que contém a regra de negócio, conecta com as tabelas ou documentos do banco (depende do banco escolhido) e entrega a resposta obtida para as rotas. Por fim as interfaces que cujo objetivo é padronizar os métodos e atributos, ou seja caso apreca uma variável a mais ou a menos, a interface não deixa passar.

React e Node

4) existem dois diretórios (models e controllers) e dois arquivos, um de servidor e um de rotas. O arquivo de servidor tem a finalidade de importar todos outros arquivos e injetar dentro do servidor (ou seja o express que foi a biblioteca escolhida), o arquivo de rotas serve como desacoplamento de tarfeas ou seja, para o arquivo de servidor não ficar tão grande o arquivo de rotas serve para separar toda essa funcionalidade de roteamento em outro arquivo que por sua vez, será exportado para uso no arquivo principal (o de servidor). Odiretório de modelo é responsavel pelos arquivos de criação de entidades ou documentos refletidos no banco de dados (igual a explicação feita na questão anterior) e por fim o controlador que obtém a regra de negócio (CRUD).

5) SPA é um método de desenvolvimento de aplicações do lado do cliente cujo objetivo é trazer a experiência de estar na mesma página, pois existe apenas um arquivo HTML, e todo resto é construido em javascript, dando a sensação de ter varias arquivos HTML sendo renderizado na mesma tela.

6) tendo um arquivo central, é possível desacoplar arquivos para executar tarefas específicas igual no back-end. Primeiro deve-se criar um roteamneto cujo objetivo é envolver todos componentes e apresenta-los apenas em recursos específicos (em rotas específicas), tendo isso cria um componente de renderização de HTML e alguns métodos para controlar o comportamento do mesmo. No caso do componente de Notas (especificamente de criação) foi feito uma estrutura tipo HTML (chamada de JSX – javascript extends) possibilitando criar caixas de entrada e botões. Em seguida foi capturado os dados e armazenaddos em varáveis e mandado para a API, possibilitando a criação de notas.

7) a consulta de notas (da mesma forma que o cadastro) foi requisitado o acesso a API (node ou typescript) com a lista de cadastros efetuados. Tendo esse *array* de arquivos basta armazenar em uma variável e começar a percorre-lo. Nesse percurso no retorno do arquivo jsx é possível acessar variáveis, e com isso basta criar uma tabela ou itens (depende da necessidade) e imprimir no “HTML”.