### **1. Planejamento e Estruturação do Projeto:**

* Primeiro, vamos planejar a estrutura do projeto, definindo as pastas e arquivos necessários.
* Decida se você deseja utilizar o NestJS para o backend e o Angular para o frontend, como sugerido nos desafios opcionais, ou se prefere usar outras tecnologias.
* Defina a estrutura das rotas da API, os modelos de dados necessários e as validações a serem implementadas.

### **2. Desenvolvimento do Backend:**

* Crie um projeto Node.js usando o NestJS, caso opte por isso. Caso contrário, você pode usar o Express.js ou outro framework de sua preferência.
* Implemente as rotas CRUD para as pessoas e endereços, utilizando os métodos HTTP apropriados (GET, POST, PUT, DELETE).
* Utilize um banco de dados (como MongoDB, MySQL, PostgreSQL) para armazenar as informações das pessoas e endereços.
* Implemente a integração com a API de consulta de CEPs para preencher automaticamente os detalhes do endereço.

### **3. Desenvolvimento do Frontend:**

* Se optou por usar o Angular, crie um projeto Angular e configure as rotas e serviços necessários.
* Desenvolva componentes para listar, cadastrar, editar e excluir pessoas e seus endereços.
* Integre a aplicação frontend com a API backend que você criou, consumindo as rotas CRUD.
* Implemente as funcionalidades adicionais, como exibição da idade da pessoa e dias restantes para o próximo aniversário.

### **4. Validações e Tratamento de Erros:**

* Implemente validações nos campos de entrada tanto no frontend quanto no backend, utilizando bibliotecas como Joi, Yup ou outras.
* Adicione tratamento de erros em todas as camadas da aplicação, fornecendo mensagens de erro claras e informativas.

### **5. Desafios Opcionais:**

* Se desejar, utilize o NestJS para o backend e o Angular para o frontend, conforme sugerido nos desafios opcionais.
* Adicione suporte à paginação na listagem de pessoas e implemente um mecanismo de busca para filtrar pessoas com base em critérios específicos.
* Implemente um sistema de autenticação de usuários para acessar a API e o frontend, utilizando JWT (JSON Web Tokens) ou outra abordagem de sua preferência.

### **6. Testes e Deploy:**

* Realize testes unitários e de integração para garantir o funcionamento correto da aplicação.
* Faça o deploy da aplicação em um ambiente de produção, utilizando serviços como Heroku, AWS, Google Cloud Platform, etc.

Lembre-se de manter um bom padrão de código, documentação e seguir as melhores práticas de desenvolvimento. Se tiver alguma dúvida específica durante o desenvolvimento, estou aqui para te ajudar. Boa sorte!