

Trabalhando com objetos e classes

Ao iniciar sua jornada no mundo da programação orientada a objetos, é importante compreender que, na maioria das vezes, vocês não começarão um projeto do zero. A realidade do mercado de trabalho envolve a manutenção de sistemas já existentes, onde o foco será entender e melhorar códigos criados por outras pessoas. Nesse sentido, trabalhar em um projeto já em andamento nos ajuda a simular essa experiência.

Não é necessário saber todas as linhas de código ou dominar cada detalhe do projeto. O fundamental é que vocês dominem bem os conceitos básicos de orientação a objetos, como classes, objetos, atributos e métodos, pois essas são as ferramentas que permitirão entender, modificar e melhorar qualquer código que encontrarem no futuro.

Swagger

O Swagger é uma ferramenta poderosa que facilita a documentação e o teste de APIs. Ele gera automaticamente uma interface gráfica interativa onde você pode visualizar e testar os endpoints de uma aplicação. Isso é muito útil para desenvolvedores, pois permite entender e interagir com a API sem precisar escrever código adicional.

No nosso projeto, o Swagger ajuda a entender como os diferentes serviços da aplicação funcionam, mostrando os métodos disponíveis (como GET, POST, PUT, DELETE), os parâmetros que eles recebem e as respostas que retornam. Isso torna o trabalho mais simples, permitindo que você teste e explore a API diretamente pelo navegador, além de ser uma excelente ferramenta de comunicação entre desenvolvedores e equipes.

API

As chamadas para os serviços da nossa API podem ser feitas de diversas maneiras, cada uma com suas vantagens. Uma delas é via Swagger, onde vocês podem testar e interagir diretamente com os endpoints através de uma interface gráfica amigável, facilitando o entendimento do funcionamento da API.

No entanto, não se limitam ao Swagger. Vocês também podem realizar essas mesmas chamadas diretamente pelo navegador (em casos de requisições simples, como o método GET) ou utilizando ferramentas como o Postman, que permite testar de forma mais avançada, incluindo envio de parâmetros, cabeçalhos, e autenticação.

Além disso, qualquer outro sistema ou aplicação pode fazer chamadas à nossa API, desde que siga os padrões estabelecidos (como o protocolo HTTP e o formato JSON). Isso é essencial em ambientes de integração entre sistemas, onde diferentes plataformas consomem as mesmas APIs para troca de informações.

Etapas:

- Importar no Eclipse
- Rodar
- Testar
- Debugar
- Melhorar
- Implementar