



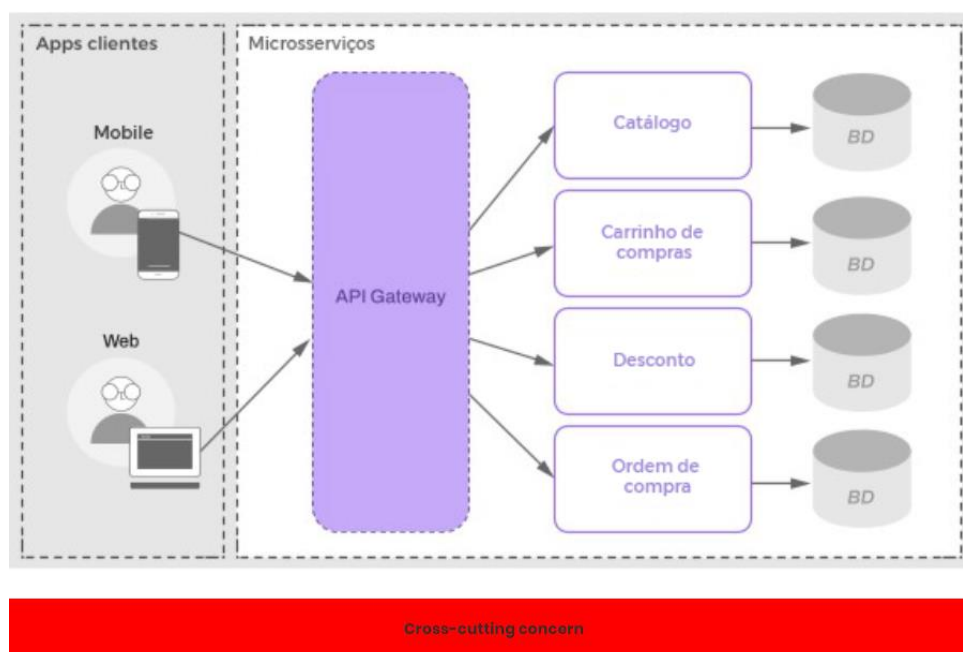
## Vamos praticar!

O gerenciamento da arquitetura da DH Fintech precisa começar a acrescentar tarefas relacionadas com cross-cutting concerns em todos os seus microsserviços. Para isso, solicita que adicionemos o Spring Cloud Gateway aos microsserviços do produto e do usuário encontrados no ZIP anexado. Devemos encapsular os endpoints desses dois microsserviços na porta 8090.

### O que devemos fazer?

1. Identificar os serviços a serem acrescentados, analisando o código.
2. Identificar as rotas apropriadas para estes serviços. Estes devem fazer sentido semântico — por exemplo, em serviços “Produtos”, o caminho não deve ser “/cursos”. Lembre-se da boa prática de que as rotas devem ser plurais (com algumas exceções), respeitar os parâmetros das rotas — quando vai na rota e quando vai na querystring, ou seja, “/produtos/12” ou “/produto?id=12”.
3. Uma vez identificadas as rotas, analisar a maneira mais eficiente e clara de aplicar os filtros.
4. Impactar as etapas anteriores nos diversos arquivos de código e configuração.

Abaixo, compartilhamos um diagrama de exemplo de uma aplicação. Podemos ver que os diferentes clientes entram através do API Gateway, e por sua vez, o API Gateway é redirecionado para cada microsserviço. Podemos fazer um gráfico semelhante para o exercício e depois fazer as rotas.





Em uma programação orientada a um aspecto, os cross-cutting concerns são funcionalidades gerais de um programa que afetam outras funcionalidades centrais do programa, sem que elas próprias façam parte dessa funcionalidade central. Por exemplo, em uma aplicação de gerenciamento de registros médicos, a indexação desses registros é uma função “núcleo” (ou core) desse sistema; enquanto rastrear mudanças nesses registros, autenticar usuários, ou os dados criptografados seriam uma cross-cutting concerns. Eles compõem a funcionalidade, mas não fazem parte do core do sistema.