## Enunciado

##### Lembra da nova marca fintech que nos contatou para aconselhamento técnico sobre a arquitetura dos microsserviços que planejavam implementar? Bem, eles voltaram "em forma de chips" e bateram na porta do departamento de arquitetura da DH Soluções.

##### Como esperado, eles decidiram lançá-lo com pouco conhecimento e os problemas não demoraram a aparecer. Entre eles, podemos citar:

##### A complexidade de testar, implementar e entender o que cada componente do sistema distribuído resolve (para o negócio). Ao mesmo tempo, a dificuldade de emular ambientes ao tentar rodar dezenas ou centenas de microsserviços em um mesmo ambiente de desenvolvimento local.

R: É orientado ao cliente utilizar o padrão **Rective Microservices**, pois esse padrão auxilia na flexibilidade e escalabilidade dos microsserviços, deixando também as resoluções dos problemas mais pragmáticas.

##### Problemas para ter uma configuração única (centralizada) do universo de microsserviços da empresa, e permitir atualizá-la para impactar os microsserviços afetados.

R: Nesse caso, é orientado o cliente utilizar o padrão **Central Configuration** para ter as configurações dos microsserviços agrupados em 1 único local.

##### Os obstáculos que eles encontram na orquestração do grande número de microsserviços. Eles nos disseram que têm cerca de 15, mas não têm certeza de quantos!

R: É orientado ao cliente utilizar os padrões **Service registry e Service Discovery.**

##### A visibilidade que alguns serviços exigem para o exterior (domínio público) e aqueles que precisam ser ocultados por serem de uso puramente interno, como os que acessam dados "sensíveis" de clientes (dados de identificação privada, movimentos financeiros, investimentos em andamento, etc.)

R: É orientado ao cliente utilizar o padrão **Edge Server**.

##### Perda de rastreabilidade de erros de serviço. Eles nos explicam que algo recorrente que acontece é que o sistema —visto que tenta fazer transferências para outra conta— falha ao se autenticar novamente com o usuário. E tentou-se o mesmo com um código de confirmação.

##### R: É orientado ao cliente utilizar os padrões **Log Aggregation e Circuit Breaker**, para conseguirem rastrear o erro e também dar um feedback mais eficaz para o usuario.

##### Finalmente, eles descobrem que precisam passar por milhares de logs, de diferentes fontes, e tentar fazer uma "referência cruzada" deles para ver por que há erros ou comportamento anormal no sistema.

R: É orientado ao cliente utilizar o padrão **Log Aggregation**.

##### Alinhado ao nosso trabalho anterior, vamos ajudá-lo novamente, mas contando com os padrões de design que conhecemos, associando-os aos diferentes problemas mencionados acima.

##### Boa sorte!