**Tomadas de decisão e da codificação**

**Resiliência: O que fazer para mitigar possíveis erros e controlar os possíveis erros recebidos da API?**

Procurar salvar o máximo dos logs possíveis, sendo em arquivos ou até mesmo em base de dados.

Procurar colocar o máximo de informações possíveis:

- URL;

- Parâmetros;

- Cabeçalhos;

- Parâmetros de autenticação;

- Texto de resposta;

Já tive experiências em que acabei criando funções para reprocessar requisições paradas com erro, e até mesmo uma dashboard com aberturas de chamado em ferramenta própria.

Outra coisa bem legal, seria criar um disparo de e-mail diário ou a cada x horas, mostrando totais ou um detalhamento dos erros.

**Performance: Quais boas práticas são aplicadas em banco de dados e no código para garantir performance?**

- Procurar usar o máximo de relacionamento possível entre tabelas;

- Caso for realizar selects com muitos jions, é interessante criar views;

- Tentar nunca usar select \* from;

- Evitar utilizar muito IN e NOT IN;

- Em alguns casos é interessante até criar procedures ao invés de deixar as regras no código;

- Só utilizar agrupamentos, order by, distinct se realmente for necessário;

**Segurança: Como garantir segurança para as APIs do sistema?**

Caso vários usuários diferentes forem acessar os dados, procurar criar tokens individuais, e definir permissões de endpoints de acordo com grupos de usuários;

Existem diversas regras para gerenciar os tokens de acesso, portanto devemos estudar a melhor, mais simples e segura possível de acordo com o sistema;

**Simultaneidade: Como trabalhar com simultaneidade se milhares de requisições forem solicitadas simultaneamente?**

Implantar limites de requisições por usuários, serviços, regiões é uma boa prática, porém deve ser bem implantada, pois existem diversas formar de fazer.