



Pós-graduação em Desenvolvimento Mobile
APIs e Web Services
(Prof. Luiz Vitor Ramos Silva)
Tarefa 2

Nome: Ismália Dulce Gonçalves Santiago

De acordo com o que foi entregue da tarefa 1, ao qual foi executado a etapa de Brainstorm do API-First.

Continue as etapas do API-First:

1) Estabeleça as partes interessadas da API

- O Product Team ou o Product Owner, que realiza decisões e planejamentos acerca do design da API;
- Os desenvolvedores da API (back-end e front-end), aqueles que irão implementar, colocar a API em produção e monitorá-la;
- Os usuários finais (prestador de serviço e usuário cliente), que farão uso do aplicativo mobile e do site.

2) Crie um guia de estilo "convenção"

Ideias inspiradas na página "Convenções de nomeação" do "Guia de projeto da API" das APIs do Google Cloud (https://cloud.google.com/apis/design/naming_convention).

- Caso a Ideia Original seja vendida para uma multinacional bilionária, seria um bom padrão manter os nomes usados nas APIs em inglês americano;
- Os nomes de serviços devem ser nomes de DNS sintaticamente válidos que possam ser resolvidos para um ou mais endereços de rede. Os nomes de serviços das APIs públicas da empresa Ideia Original podem ser: **xxx.ideiaoriginal.com**;
- Os nomes de pacotes para APIs com controle de versões precisam terminar com a versão. Por exemplo: **package ideiaoriginal.application.v3**;
- Os IDs de coleções precisam usar a forma plural e lowerCamelCase, além da ortografia e semântica do inglês americano. Por exemplo: **users**, **categories** ou **availableServices**;
- Se for necessário representar uma data como uma string, o formato deverá ser DD-MM-YYYY para melhor entendimento dos usuários brasileiros. Por exemplo: **30-07-2014**.

3) Projete o contrato de APIs

Cheque o arquivo **swagger.yaml**, anexo junto a esse documento.

4) Propor uma governança de API

No caso da governança de uma API, precisaremos tomar uma variedade de decisões. Aqui temos algumas ideias de questionamentos baseados no site <https://balipodo.com.br/melhor-tomada-de-decisao-melhor-governanca-da-api/>. A partir desses questionamentos, podemos debater algumas decisões de governança.

- Qual API devemos construir a seguir?

A decisão de qual API é mais importante de construir e a priorização no backlog de quais funcionalidades devem ser desenvolvidas primeiro devem ser discutidas com o Product Team ou com o Product Owner, considerando obrigatoriamente o que o cliente considera que vai entregar mais valor em primeiro lugar.

- Quais são os parâmetros de design para essas APIs?

Alguns parâmetros foram definidos no item 2.

Definir um design system para as interfaces das APIs usando uma ferramenta como o Figma seria adequado para deixar todas as APIs padronizadas e com a cara da empresa.

- Quais são os detalhes técnicos de implementação de cada API?

Seguindo a tendência do mercado, podemos seguir com frameworks Javascript na versão web. Por exemplo, back-end em Node.js, e o front-end em React ou Angular.

O aplicativo mobile pode utilizar Flutter ou React Native, já que os requisitos não exigem algo além da capacidade desses frameworks híbridos, e podemos ter um aplicativo funcionando tanto em iOS quanto em Android ao final do desenvolvimento.

- Uma vez implantada no ecossistema de TI, quando e como a API deve ser mantida?

Primeiramente, a API deve ser implantada apenas quando tiver passado pelas mãos do time de QA, ou seja, passado por testes automatizados e manuais, testes funcionais, testes de carga etc.

Posteriormente, a API deve ser mantida por um time de operações que consiga manter a disponibilidade dos servidores, estejam eles na nuvem ou não.

Pode-se monetizar a API a fim de conseguir manter os servidores funcionando com alta disponibilidade.

- Por quanto tempo você deve manter essa API antes de substituí-la?

Até não haver mais formas de a empresa sustentar financeiramente a API.