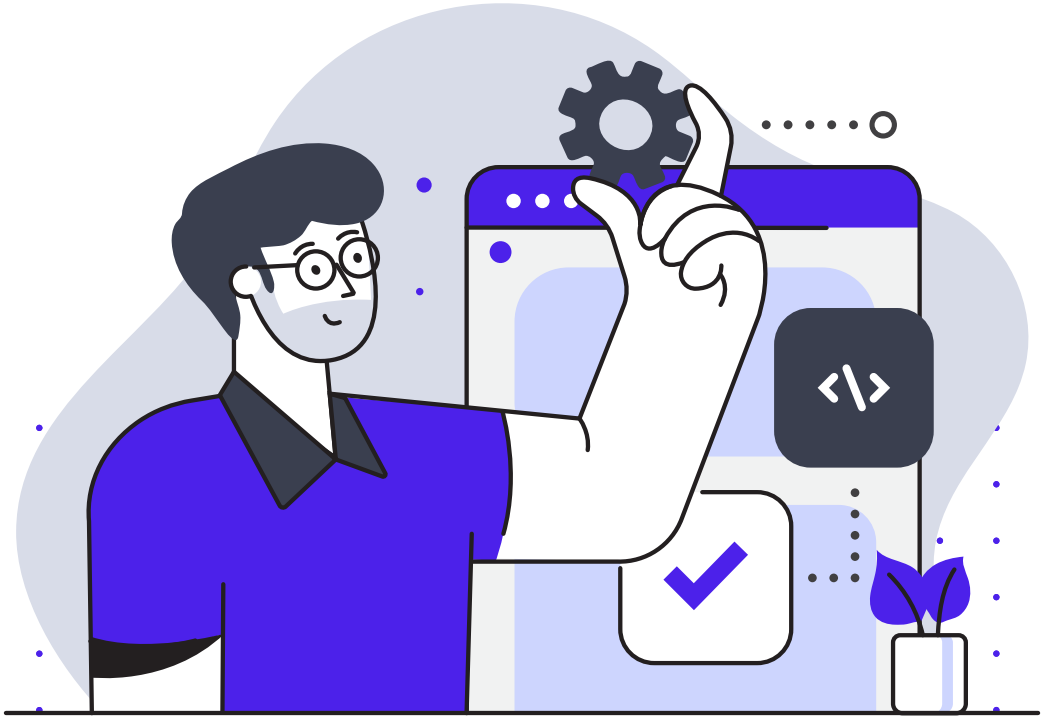
GR1D

Conheça o modelo de maturidade estratégica de APIs

Saiba, neste *ebook*, tudo o que você precisa sobre o modelo de maturidade estratégica de APIs e descubra como analisar a maturidade da sua.

Índice



Introdução

Com a evolução da programação e a ramificação das linguagens disponíveis, é cada vez mais necessário fazer o uso de **APIs**. Sendo assim, é essencial entender em qual nível de maturidade uma API se encontra, para que seja feito o melhor uso dessa API e das integrações que se deseja manter.

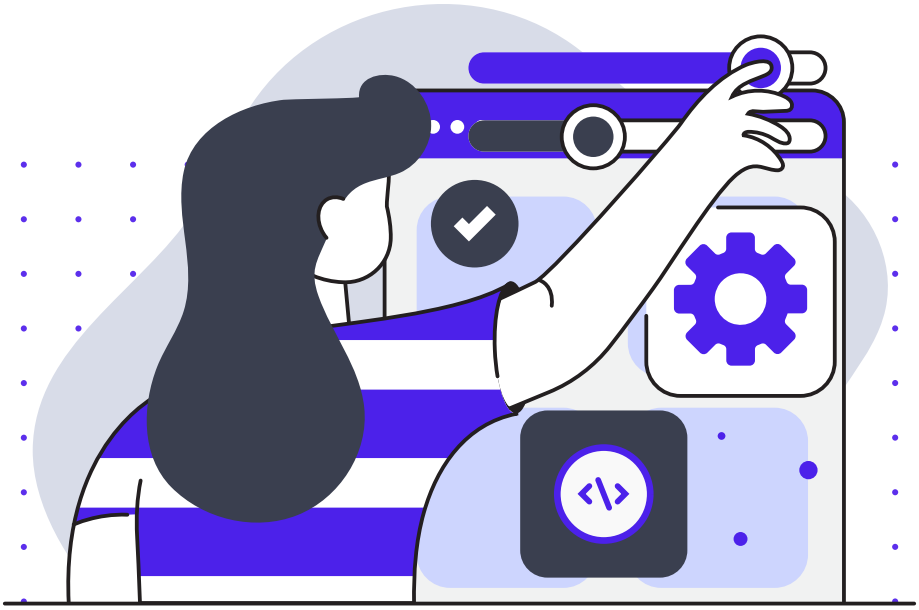
De acordo com Gartner nesse ano de 2021, 90% das aplicações *WEB* terão mais chances de ataque por APIs do que pela UI (*User Interface*), ante 40% em 2019. Isso gera um enorme risco para empresas que fazem uso de APIs mal estruturadas e com baixo nível de maturidade.

Outro fator importante que se perde com uma baixa maturidade de API é a quantidade de negócios realizados, já que implica em pouca segurança de dados e da própria aplicação. Afinal, em 2019 foi criada a LGPD, que obriga as empresas a zelar pela segurança de qualquer dado "sensível" de pessoas físicas ou jurídicas.

Para contornar esses problemas, saber a maturidade da **API** que será utilizada é fundamental e ter uma estratégia de maturidade de API se faz preciso.

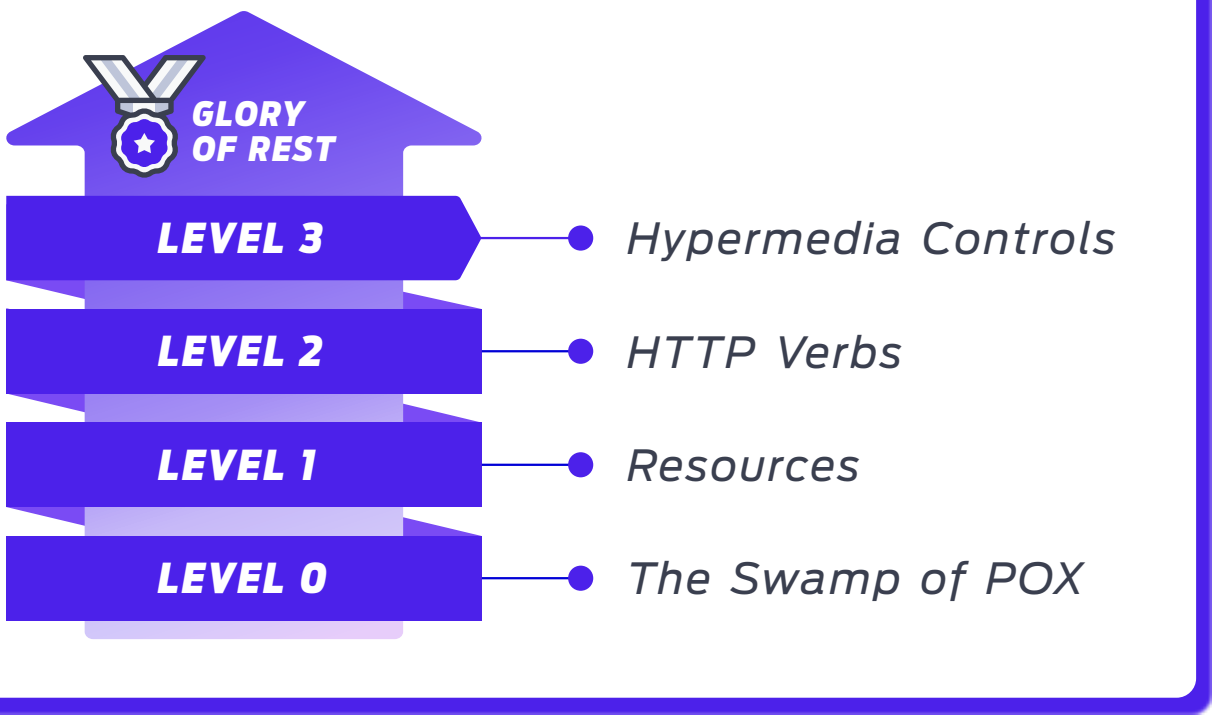
Nesse material, você irá conhecer os 5 pontos principais para se ter uma boa "estratégia de maturidade de APIs", extraindo o máximo potencial de cada API usada em seu negócio.

Quais os níveis de maturidade de uma API?



Uma **API** possui 4 níveis de maturidade, conhecidos como “Modelo de Maturidade *REST* de Richardson”. Eles indicam desde sua facilidade de uso pelos desenvolvedores até a capacidade de ser interpretada pelo *CLIENT* (normalmente navegadores *web*).

Os quatro níveis de maturidade são:



O primeiro nível de maturidade é o Nível 0 e o mais alto é nível 3. Quanto maior o nível de maturidade, maior a segurança e a facilidade de uso da API.

Nível 0: *The Swamp of POX*

O Nível 0 – *The Swamp of POX (Plain of XML)* – de maturidade é o mais baixo dos níveis de maturidade. Os serviços que estão no Nível 0 só possuem uma URI e utilizam apenas um protocolo *HTTP*, fazendo uso de *XML* – método mais primitivo de comunicação entre 2 aplicações e, muitas vezes, inseguro para aplicações recentes.

Contudo, mesmo o Nível 0 possui regras a serem seguidas, que permitem a padronização das APIs e facilitam o uso pelos desenvolvedores.

Nível 1: *Resources*

Nesse nível de maturidade, é possível acessar os dados da API por diversas URIs, sendo que cada URI identifica um *Resource* diferente. Apesar disso, no Nível 1 ainda é usado apenas um protocolo *HTTP* (normalmente *POST*).

Como é possível utilizar uma URI dedicada a cada *Resource*, o nível de usabilidade pelos desenvolvedores aumenta.

Nível 2: *HTTP Verbs*

As **APIs** que estão no Nível 2 fazem uso de URIs e de Protocolos *HTTP*, além de códigos de *Status*, melhorando a comunicação com outras aplicações. De outro modo, facilitam a interpretação por outros desenvolvedores que desejam criar uma integração com uma API que esteja no Nível 2.

Com a maturidade atingindo o nível 2, é possível fazer todo o *CRUD* dentro de uma única URI por *Resource* e saber se a ação que foi solicitada para API foi concluída ou não. Afinal, agora a API sempre irá responder com um código de *Status*, na faixa de 2xx quando a operação deu certo e na faixa de 4xx quando a operação não pode ser concluída por algum motivo.

Nível 3: *Hypermedia Controls (HATEOAS)*

Este é o maior nível de maturidade que uma API pode atingir, de acordo com o modelo de Richardson, que é costumeiramente chamado de *HATEOAS* (acrônimo para *Hypermedia As The Engine Of Application State*).

Com esse nível de maturidade, os retornos que a API gera são autoexplicativos, facilitando seu "consumo" e agilizando processos de integração. Ao chegar a esse nível a API pode ser considerada como "*RESTful* API completa".

Desenvolvendo APIs com uma estratégia voltada para a maturidade

Agora que você já conhece os 4 níveis de maturidade de uma API, é de suma importância pensar em como obter o nível ideal de **maturidade estratégica.**

Para alcançar o máximo potencial de uma API e gerar lucro para um negócio baseado em uso de API, é necessário ter uma boa estratégia, que garanta o máximo potencial de mercado da API a ser criada/usada.

Podemos citar aqui uma estratégia orientada por 5 pontos:

- Alinhamento com negócio
- Capacitação dos desenvolvedores
- Medindo valor o comercial
- Gerenciamento do ciclo de vida da API
- Comunicação

Alinhamento com negócio

Um alinhamento da API com os valores do negócio é importante, pois pode gerar novos canais de oportunidades e parcerias, aumentando o retorno financeiro de uma API.

Para melhor entendimento, podemos mensurar o Alinhamento com negócio em 6 níveis. Veja cada um deles a seguir.

Nível 0: Nesse nível, a maior parte da empresa não entende a API, sendo que os líderes estão focados em redução de custo e reusabilidade.

Nível 1: O valor da API aqui é meramente uma extensão de um Serviço *SOA* ou apenas uma fonte de acesso de dados, tendo seu valor atrelado a facilidade de integrar com sistemas *back-end*.

Nível 2: Neste ponto, grande parte da equipe fica focada em habilitar funcionalidade para integração mobile e inovações dentro da API. Entretanto, ainda existe muito esforço descoordenado pela falta de um gerenciamento da API como um produto.

Nível 3: Nesse nível, já existe um processo definido para entradas, seleção e priorização da API, baseado no uso e valor do negócio.

Nível 4: Tirar valor da API é um objetivo de toda a empresa e uma equipe já é dedicada à multifuncionalidade da API e seu gerenciamento. Nesse nível, desenvolvedores externos se engajam e visam o lucro com extensão das funcionalidades da API do provedor.

Nível 5: Colaboração vira o foco na plataforma da API, que é continuamente revisada e otimizada em cada característica para atingir as metas de negócio.

Capacitar desenvolvedores

Capacitar desenvolvedores é a chave para o sucesso de uma API como um produto, pois eles são os primeiros a consumirem a **API**. Logo, ela deve ser o mais fácil quanto possível para que os desenvolvedores façam uso dela. Caso contrário, eles irão procurar funcionalidades similares em outro fornecedor.

A capacitação de desenvolvedores também pode ser dividida em 6 níveis diferentes:

Nível 0: A empresa não tem consciência de oportunidades e valores que podem fazer a comunidade de desenvolvedores se engajar e capacitar para dar suporte aos objetivos de negócio.

Nível 1: Os desenvolvedores estão pouco engajados, tendo resultado em pequenas customizações ou soluções pontuais. As documentações não são autoexplicativas e contêm informações rasas sobre as funcionalidades. Aqueles que querem contribuir devem se inscrever de forma manual, sendo que ainda não existe o conceito de catálogo de API ou plataforma de API.

Nível 2: Criar recursos para dar suporte a desenvolvedores com testes consistentes que ajudam a entender as funcionalidades da API. O versionamento, processo de atribuir um nome único ou uma numeração única para indicar o estado de um *software*, já é utilizado.

Nível 3: Neste nível, os desenvolvedores já podem se autorregistrar. Contudo, a aprovação do cadastro ainda é feita manualmente, podendo os desenvolvedores terem acesso a uma maior customização da API. Além disso, eles têm um maior suporte com códigos de "amostra" e *SDK's* (*Kit* para desenvolvimento de *software*).

Nível 4: O autorregistro pode ser feito de maneira fácil, sem necessidade de o time responsável pela API liberar acesso. Os desenvolvedores têm acesso total a *SDK's*, informações completas da API e treinamentos interativos.

Nível 5: Neste nível, além do autorregistro, desenvolvedores têm acesso ao autogerenciamento. Isto inclui o uso de Marketplaces.

Para a melhor capacitação de desenvolvedores é indicado que seja feita a correta implantação de versionamento e padronizações de design e documentação, tornando o uso da API mais fácil.

Medindo valor comercial

Medir o valor comercial é o propósito final em uma estratégia de API. Uma API existe para promover um objetivo comercial da empresa. Assim, é necessário dispor de ferramentas para medir e melhorar o valor da API, pensando em como ela é usada, por quem é utilizada, que tipo de valor ela entregou ou criou e como esse valor pode ser melhorado.

Medir o valor comercial de uma API tem 6 níveis de evolução:

Nível 0: Não existe uma organização ou mesmo um sistema para coletar dados para analisar ou melhorar o valor comercial.

Nível 1: Os provedores da API compreendem a necessidade de ter meios de mensurar o valor comercial da API para investidores internos e externos.

Nível 2: Os times de desenvolvimento do provedor da API são orientados para coleta de dados, mas ainda de forma isolada de outras partes.

Nível 3: As métricas estão bem definidas e são coletadas para entender a performance da API. A empresa tem várias ferramentas e começa a combinar os dados para obter insights sobre a API.

Nível 4: Neste nível, a empresa começa a coletar dados avançados e em tempo real, permitindo a análise de performance da API em cada funcionalidade/módulo.

Nível 5: A API contribui para os objetivos comerciais da empresa. Quando a companhia atinge esse nível de maturidade de mensuração do valor comercial, ela revisa os dados coletados com seu plano de negócio, melhorando os resultados comerciais.

Gerenciamento do ciclo de vida da API

Gerenciar o tempo de vida de uma API consiste em práticas que governam a arquitetura, design e implementação. O gerenciamento do ciclo de vida da API entrega benefícios, como a interoperabilidade, assim como uma melhor comunicação com os "clientes" (consumidores da API), com geração de *roadmaps* claros.

Os níveis que o gerenciamento do ciclo de vida da API pode ter são:

Nível 0: Não existe nenhum mecanismo para gerenciar o ciclo de vida da API.

Nível 1: A organização usa faz uso de mecanismo de versionamento e documentações simples. Cada time responsável pela API determina seu próprio método de gerenciar a vida da API, gerando problemas de integração entre os times.

Nível 2: API inclui ambiente de pré-produção para dar suporte aos desenvolvedores e testes.

Nível 3: O processo para o gerenciamento do ciclo de vida da API é bem definido, porém o entendimento sobre a gestão ainda é raso entre os times de desenvolvimento.

Nível 4: Com processo de gerenciamento do ciclo de vida da API bem definido, o processo de interação entre os times de desenvolvimento também é estruturado de forma consistente. Alguns sistemas de automação simples já são utilizados, permitindo retornar a versões anteriores.

Nível 5: O ciclo de vida da API é amplamente automatizado em toda a organização, com um processo fortemente estabelecido. O acesso ao histórico completo de versões e documentações ajuda desenvolvedores, permitindo que eles resolvam problemas mais rapidamente.

Comunicação

Uma comunicação segura, confiável e flexível gera um fluxo adaptável e protegido de informações entre aplicações e componentes da API. A segurança da API permanece com um ponto importante, que faz uma comunicação ser clara e confiável a ponto de estar no topo da estratégia de maturidade de API.

Os níveis da comunicação efetiva são:

Nível 0: Nenhum meio de comunicação entre provedor e desenvolvedores existe, nem mesmo termos de privacidade ou políticas de uso.

Nível 1: O time de desenvolvimento entende a importância do gerenciamento de tráfego, autenticação, autorização, qualidade do serviço e segurança. Algumas partes do projeto já possuem mecanismos de segurança.

Nível 2: Os times implementam sistema de autenticação, autorização e mecanismos de mensagens baseados em práticas da internet, mas de forma isolada. Isso acaba gerando diferentes termos de uso e políticas de privacidade.

Nível 3: Princípios e padrões básicos são estabelecidos e implementados usando toda estrutura da API, como segurança, gerenciamento de tráfego, qualidade de serviço, dados privados e diversos aspectos da rotina de funcionamento da API.

Nível 4: Segurança, privacidade, qualidade e padrões de comunicação são adotados proativamente dentro de toda a organização.

Nível 5: A organização inicia mudança no gerenciamento de tráfego e segurança de proativo para preditivo. O gerenciamento de ciclo de vida da API e o sistema de *firewall* funcionam em conjunto, para detectar qualquer ameaça e/ou antecipar mudanças no tráfego e nível de serviço.

Como a GR1D auxilia sua empresa a alcançar a maturidade de APIs

Para que a maturidade ideal de sua API seja atingida é necessário levar em conta cada nível da estratégia multidimensional, elevando os resultados comerciais ao máximo possível.

Por isso, a GR1D mantém em sua base **APIs** de alto nível de maturidade com estratégias para entregar o melhor resultado para todos os envolvidos no processo de uso do marketplace.

Conheça o Marketplace de APIs da GR1D

GR1D