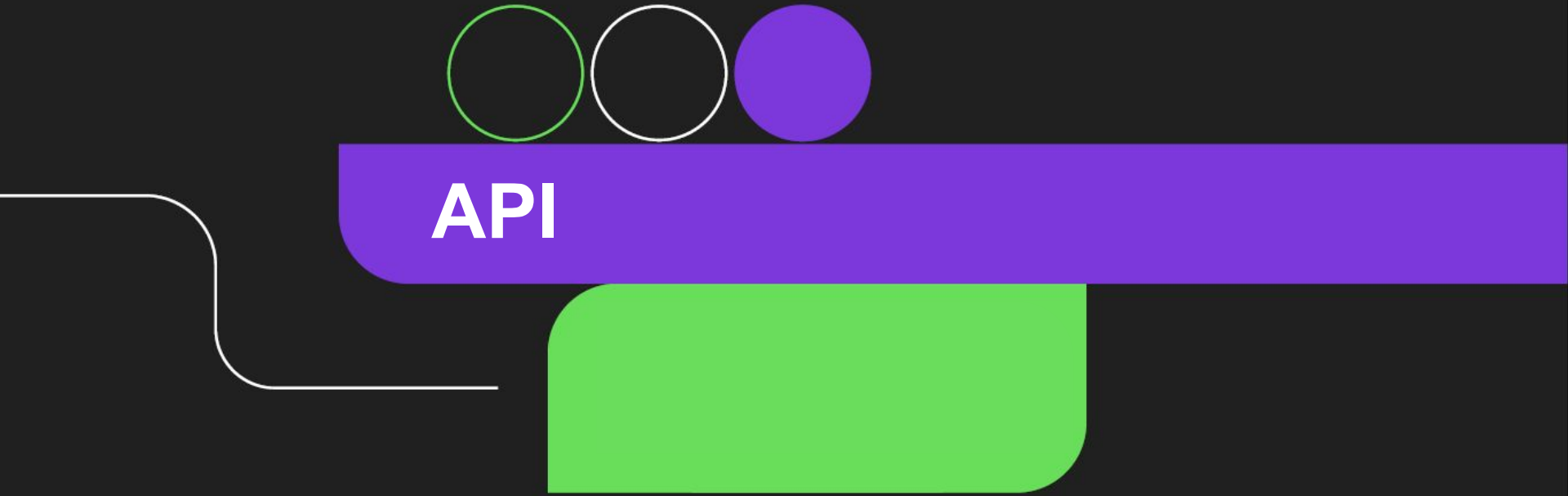
GAMA ACADEMY

API e MVC

01 API

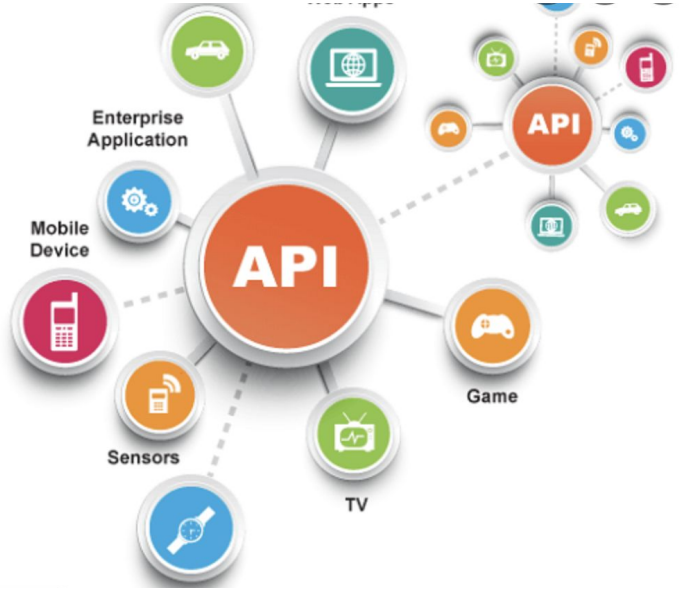
02 MVC


API

Interface de Programação de Aplicativos

O que é API

A sigla API deriva da expressão inglesa *Application Programming Interface* que, traduzida para o português, pode ser compreendida como uma interface de programação de aplicação. Ou seja, API é um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos.



Qual a função de uma API

A função de uma API é, basicamente, facilitar e simplificar o trabalho de desenvolvedores, além de oferecer um padrão para a criação de novas plataformas. Com o uso das APIs, não é necessário criar códigos personalizados para cada função que um programa for executar, o que simplifica a criação de novos aplicativos, softwares e plataformas em geral.

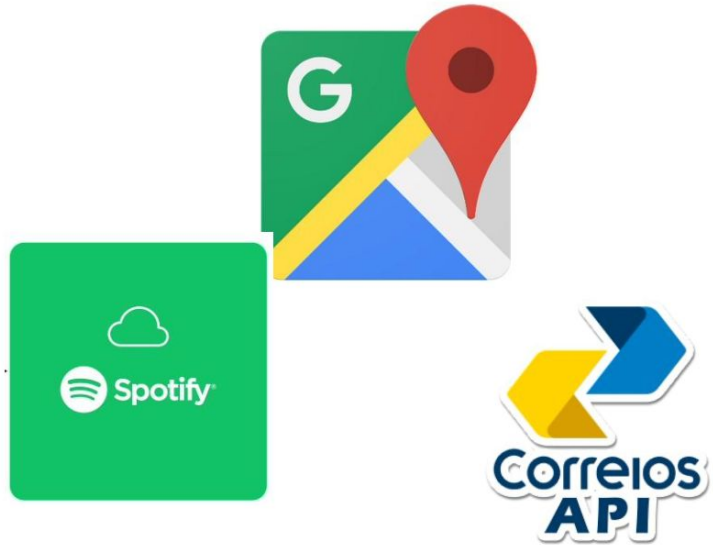
Além disso, as APIs também possuem papel fundamental quando o assunto é segurança, já que também são capazes de bloquear acesso e permissões a dados de software e hardware que algumas aplicações não podem usar.

Exemplos de API

As APIs estão presentes na maioria dos aplicativos que utilizamos no nosso dia a dia. No **WhatsApp**, por exemplo, podemos perceber a integração da lista de contatos salva no dispositivo com os contatos do aplicativo. No **Facebook**, temos a integração com o Instagram, que **permite que fotos postadas no aplicativo também sejam postadas automaticamente no Facebook**.

Web APIs

São um conjunto de instruções e padrões de programação para acesso a um aplicativo de software. Uma empresa de software lança sua API para o público de modo que outros criadores de software possam desenvolver produtos acionados por esse serviço.



APIs Publicas

São aquelas que são disponibilizadas gratuitamente para desenvolvedoras e usuários com restrição mínima. Podem precisar de cadastro, o uso de API Key ou ser completamente abertas.
Elas estão relacionadas com uso externo de dados ou serviços.

APIs Privadas

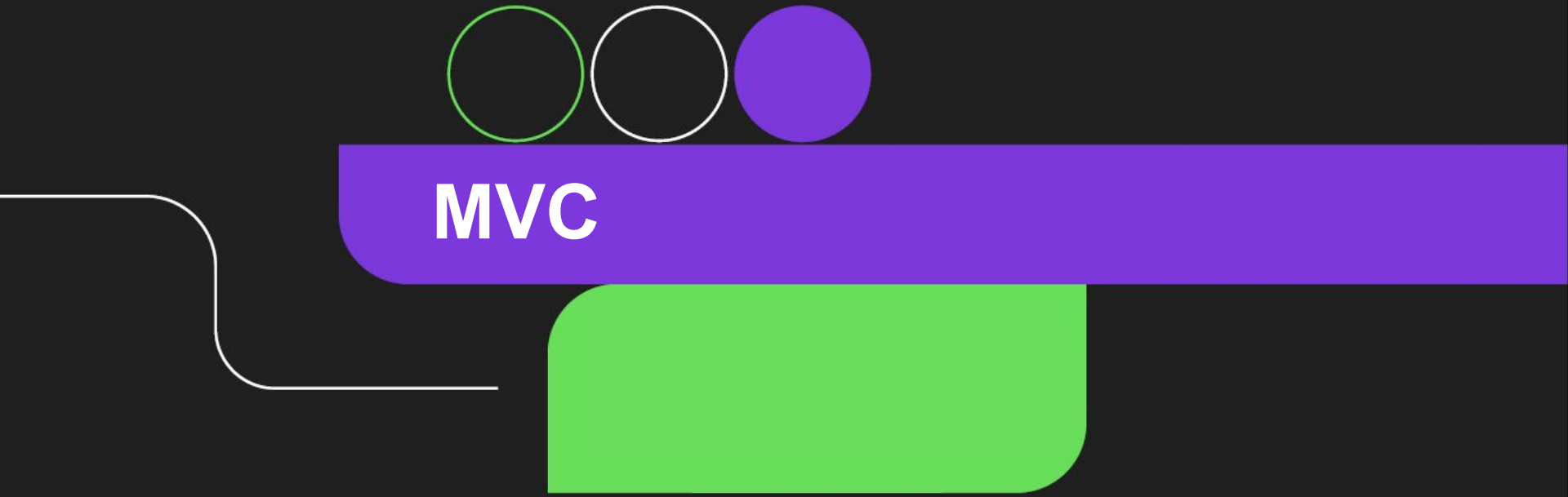
São oposto das APIs públicas. Elas estão ligadas à serviços sigilosos, dados sensíveis, transações de empresas privadas, comunicação e ferramentas interna da empresa, etc

APIs REST e RESTfull

Trata-se de um conjunto de princípios e definições necessários para a criação de um projeto com interfaces bem definidas, **Rest**, que é a abreviatura de **Representational State Transfer**, é um conjunto de restrições utilizadas para que as requisições HTTP atendam as diretrizes definidas na arquitetura.

RESTful

No RESTful, são conhecidos como verbos os métodos do protocolo HTTP:
POST (cria um novo recurso);
GET (lê um recurso);
PUT(atualiza um recurso);
DELETE (deleta um recurso);
HEAD (recupera o header da URI passada);
OPTIONS (recupera os métodos que são possíveis);
TRACE (recupera informações de debug).


MVC

MVC

A principal ideia do padrão arquitetural MVC é a separação dos conceitos - e do código. O MVC é como a clássica programação orientada a objetos, ou seja, criar objetos que escondem as suas informações e como elas são manipuladas e então apresentadas em uma interface simples para o mundo.

A utilização do padrão MVC traz como benefício o isolamento das regras de negócios da lógica de apresentação, que é a interface com o usuário. Isto possibilita a existência de várias interfaces com o usuário que podem ser modificadas sem a necessidade de alterar as regras de negócios, proporcionando muito mais flexibilidade e oportunidades de reuso das classes.

MVC

Apesar de muitas pessoas considerarem essa sigla como um padrão de design de interface, na verdade ele é um padrão de arquitetura de software responsável por contribuir na otimização da velocidade entre as requisições feitas pelo comando dos usuários.

Model ou Modelo

Essa classe também é conhecida como Business Object Model (objeto modelo de negócio). Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e regras de negócios estabelecidas.

MVC

Controller ou Controlador

A camada de controle é responsável por intermediar as requisições enviadas pelo View com as respostas fornecidas pelo Model, processando os dados que o usuário informou e repassando para outras camadas.

Numa analogia bem simplista, o controller operaria como o ‘’maestro de uma orquestra’’ que permite a comunicação entre o detentor dos dados e a pessoa com vários questionamentos no MVC.

MVC

View ou Visão

Essa camada é responsável por apresentar as informações de forma visual ao usuário. Em seu desenvolvimento devem ser aplicados apenas recursos ligados a aparência como mensagens, botões ou telas.

Seguindo nosso processo de comparação o View está na linha de frente da comunicação com usuário e é responsável transmitir questionamentos ao controller e entregar as respostas obtidas ao usuário. É a parte da interface que se comunica, disponibilizando e capturando todas as informação do usuário.

MVC

O MVC funciona como um padrão de arquitetura de software que melhora a conexão entre as camadas de dados, lógica de negócio e interação com usuário. Através da sua divisão em três componentes, o processo de programação se torna algo mais simples e dinâmico.