

# Introdução ao **Strapi**

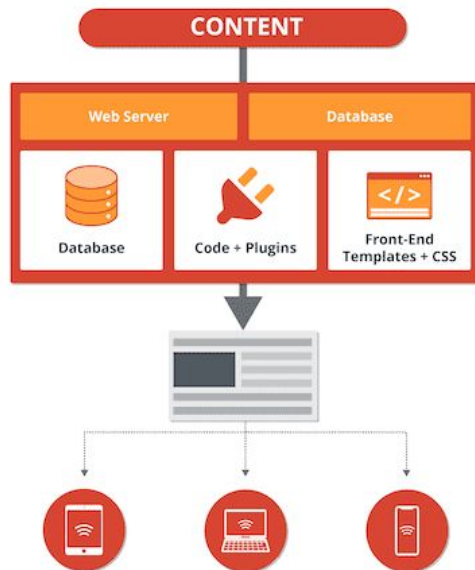
O que é, como funciona, quais vantagens e mais.

“O **Strapi** é um **headless CMS** open-source feito 100% em JavaScript, totalmente customizável e orientado para desenvolvedores preocupados em agilidade.

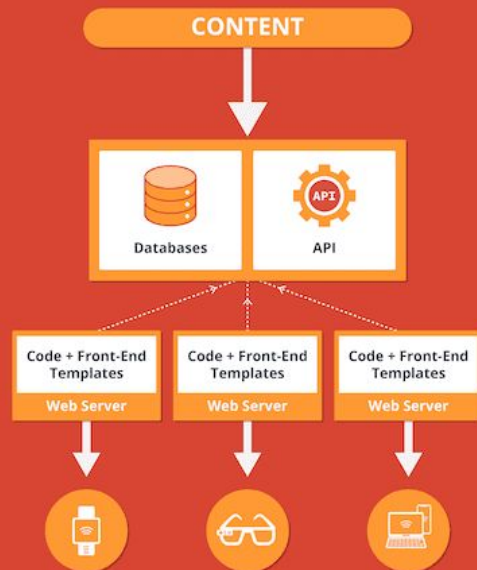
// Um **CMS** (*Content Management System* – Sistema de Gerenciamento de conteúdo) é um software responsável por gerenciar o conteúdo, ou seja, permitir **criação**, **edição** e **organização** de um determinado conteúdo.

# CMS Tradicional x Headless CMS

## TRADITIONAL CMS



## HEADLESS CMS



# Vantagens do **Headless CMS**

- Mesma entrega para diferentes tipos de dispositivos
- Agnóstico de frameworks (mesma base funciona em React, Vue, Angular, etc)
- Mais fácil na manutenção
- Mais ágil para criação de protótipos

# Vantagens do **Strapi**

- 100% open-source
- Altamente customizável (API e painel)
- Funciona com Rest API ou GraphQL ao apertar de um botão
- Self Hosted (você controla seus dados e onde colocá-los)
- Roadmap bem definido e grande investimento

# Algumas features nativas do **Strapi**

- Suporte a múltiplos bancos de dados (SQLite, MongoDB, Postgres, MariaDB)
- Documentação automática com plugin de one-click
- Integração a Webhooks
- Autenticações e Permissões por padrão
- Integração com diferentes serviços (Cloudinary, SendGrid, Algolia, Redis, etc)
- Sistema de emails
- Sistema de assets que otimiza imagens e cria em diferentes tamanhos!



# O que podemos fazer com o **Strapi**

- Sites estáticos
- Sites Institucionais para empresas
- Sites de notícias
- Ecommerce
- O que você quiser!

# Por que usar o **Strapi** ao invés de criar algo próprio ou Wordpress, Contentful, DatoCMS?

- Mais agilidade
- Codebase toda em JavaScript (mais fácil em manutenção)
- Sem mensalidades e controle do próprio dado (Contentful, DatoCMS)

// E como funciona o **Strapi**?

# Possui 3 tipos de estrutura de dados

- Collection Types (conjunto de dados padrão – Ex.: usuários, produtos, categorias)
- Single Types (dado único – Ex.: homepage, footer, menu)
- Component Types (estrutura de dados reutilizadas – Ex.: galerias, hero)

# E vários tipos para fields



**Texto** Small or long text like title or description



**Rich Text** A rich text editor with formatting options



**Number** Numbers (integer, float, decimal)



**Data** A date picker with hours, minutes and seconds



**Booleano** Yes or no, 1 or 0, true or false



**Relação** Refers to a Collection Type



**Component** Group of fields that you can repeat or reuse



**Email** Email field with validations format



**Senha** Password field with encryption



**Enumeração** List of values, then pick one



**Mídia** Files like images, videos, etc



**JSON** Data in JSON format



**UID** Unique identifier



**Dynamic Zone** Dynamically pick component when editing ...

## Cria a Estrutura de Dados





### Category

Nenhuma descrição para este Tipo De Conte...

2 fields

Configure the view

Add another field

<div>Ab</div>	name	Texto	 
<div>Ab</div>	slug	Texto	 

+ ADD ANOTHER FIELD TO THIS COLLECTION TYPE

## Cria o conteúdo

### Category

11 registros encontrados

Filtros

<input type="checkbox"/>	Id	Name ▾	Slug
<input type="checkbox"/>	1	Action	action
<input type="checkbox"/>	2	Adventure	adventure
<input type="checkbox"/>	11	Horror	horror
<input type="checkbox"/>	7	Indie	indie

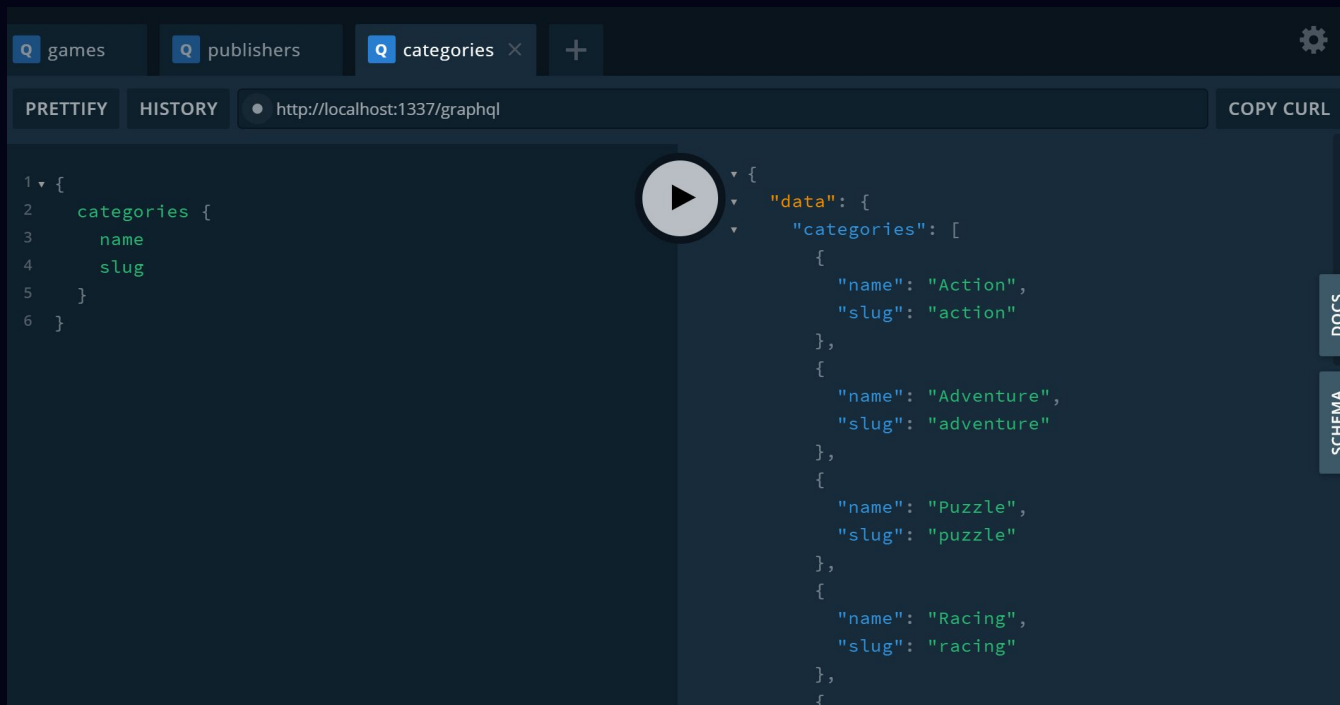
## API criada!

```
localhost:1337/categories
// 20200622125228
// http://localhost:1337/categories

[
  {
    "id": 1,
    "name": "Action",
    "slug": "action",
    "created_at": "2020-04-11T01:49:08.993Z",
    "updated_at": "2020-04-11T01:49:08.993Z"
  },
  {
    "id": 2,
    "name": "Adventure",
    "slug": "adventure",
    "created_at": "2020-04-11T01:49:16.954Z",
    "updated_at": "2020-04-11T01:49:16.954Z"
  },
  {

```

# Suporte a GraphQL!



**E muito mais...**