



# Aula 05

**Como funciona uma API?**



# Nas aulas passadas...

- 👉 HTML, CSS e JavaScript
- 👉 Conceitos react (props, componentes, states, sintaxe)
- 👉 Como iniciar um projeto com vite
- 👉 O que é o Nodejs? 🤔

# Aula 05 - API

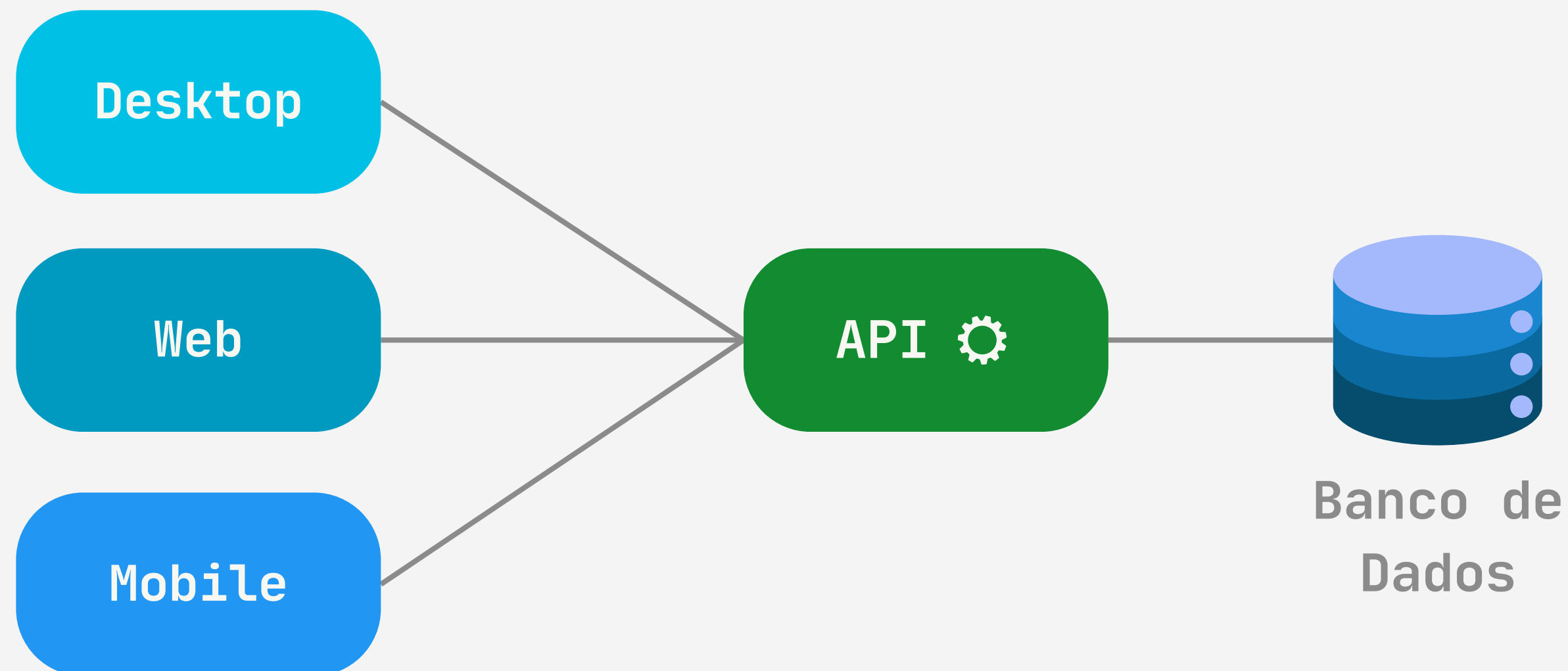
O que vamos ver hoje?

- 👉 O que é uma API?
- 👉 O que é uma Express e MongoDB?
- 👉 O que são métodos HTTP, dados da requisição, status code e Rest API.

Ao final da aula teremos todos os conceitos necessários para darmos inicio a construção das rotas da nossa API.

# O que é uma API?

Uma API (Application Programming Interface) trata-se de um conjunto de definições e protocolos que permite que diferentes sistemas de software se comuniquem entre si.





# Express

Framework web minimalista e flexível para Node.js, projetado para simplificar o desenvolvimento de aplicativos web e APIs.

Ele fornece um conjunto robusto de funcionalidades para criar servidores web e manipular requisições HTTP com facilidade. Os mais utilizados são:

**GET**

**POST**

**PUT**

**DELETE**

**PATCH**

# Métodos HTTP

São verbos responsáveis por indicar uma ação a ser executada para um dado recurso.

- 👉 **GET** - obter dados
- 👉 **POST** - criar dados
- 👉 **PUT** - atualizar dados
- 👉 **DELETE** - deletar dados
- 👉 **PATCH** - atualizar dados\* (Atualização parcial)

# Dados da requisição

👉 Request body

```
async store(req, res) {  
  const { email } =  
    req.body  
  ...  
}
```

👉 Route params

```
async show(req, res) {  
  const { userId } =  
    req.params  
  ...  
}
```

👉 Query params

```
async index(req, res) {  
  const { subject } =  
    req.query  
  ...  
}
```



# Status Code

São respostas padronizadas emitidas pelos servidores em resposta às requisições.

👉 **100 - 199** - Respostas de informações

👉 **200 - 299** - Respostas de sucesso

👉 **300 - 399** - Redirecionamentos

👉 **400 - 499** - Erros de cliente

👉 **500 - 599** - Erros de servidor

## Mais comuns:

👉 **200** - Ok

👉 **201** - Created

👉 **204** - No Content

👉 **400** - Bad Request

👉 **401/403** - Unauthorized

👉 **404** - Not Found

👉 **405** - Method Not Allowed

👉 **409** - Conflit

👉 **500** - Internal Server Error

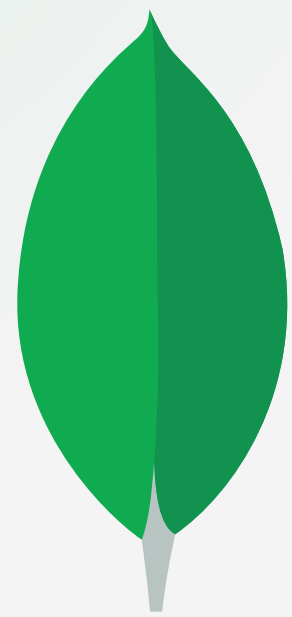
# Padrão REST

Padrão para construção de **API's** que utiliza o protocolo HTTP e retorna uma representação dos dados (**JSON**).

```
{
  "users": [
    {
      "id": 1,
      "name": "Marcos",
    },
    {
      "id": 2,
      "name": "Natália",
    }
  ]
}
```

```
{
  "cars": [
    {
      "id": 1,
      "name": "Hilux",
    },
    {
      "id": 2,
      "name": "Corolla",
    }
  ]
}
```

```
{
  "tasks": [
    {
      "id": 1,
      "title": "Task-1",
    },
    {
      "id": 2,
      "title": "Task-2",
    }
  ]
}
```



mongoDB<sup>®</sup>

# O que é o mongo?

É um banco de dados NoSQL orientado a documentos, que armazena os dados em formato JSON.

```
_id: ObjectId('6630354c97f678a51c783f01')
name : "Marcos Renê"
email : "marcosrenedev@gmail.com"
password : "123456"
confirm_password : "123456"
role : "STUDENT"
__v : 0
avatar : "1714435612988-wallpaper-astronaut-4k.jpg"
```

**Praticando...**