

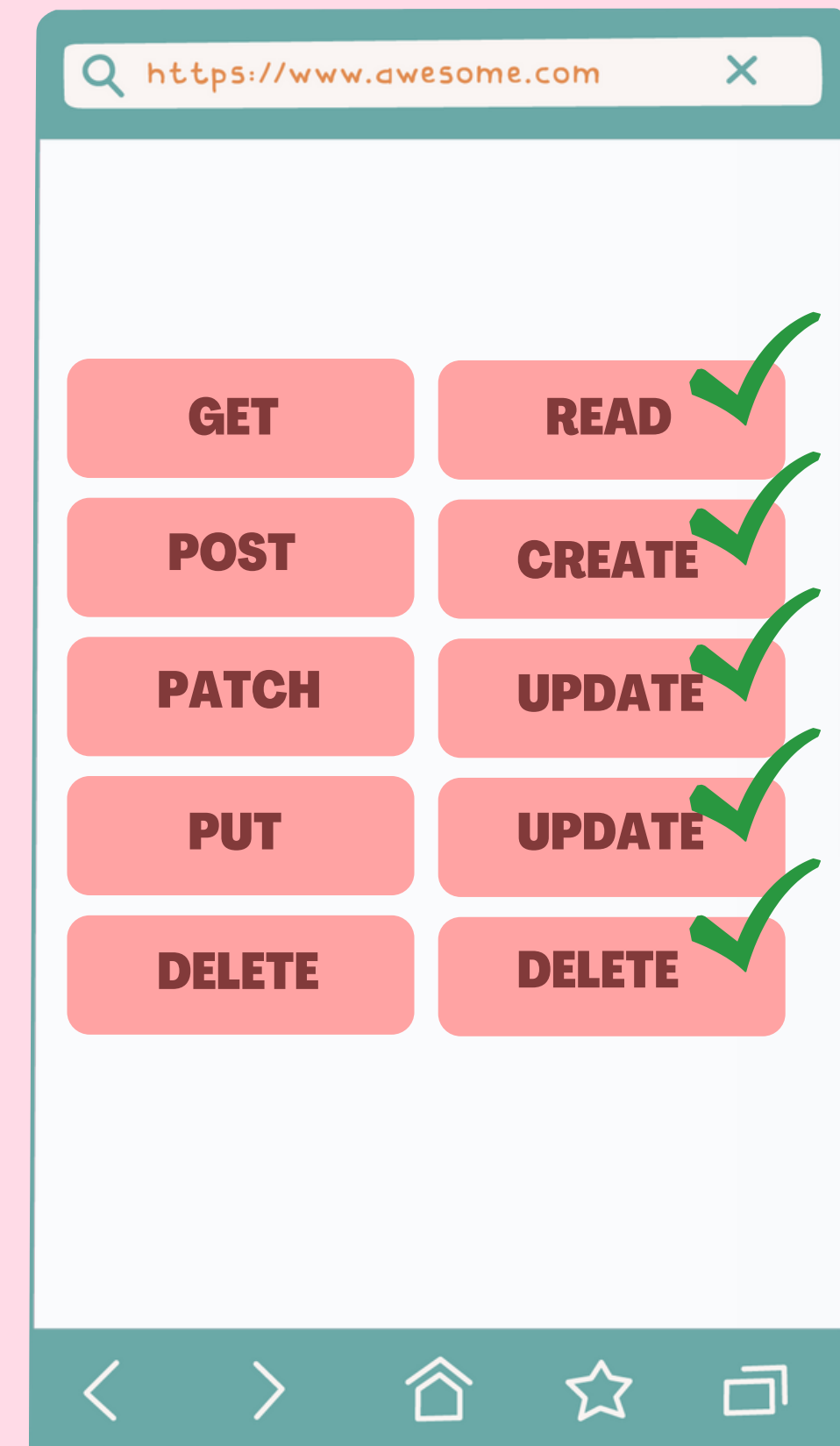
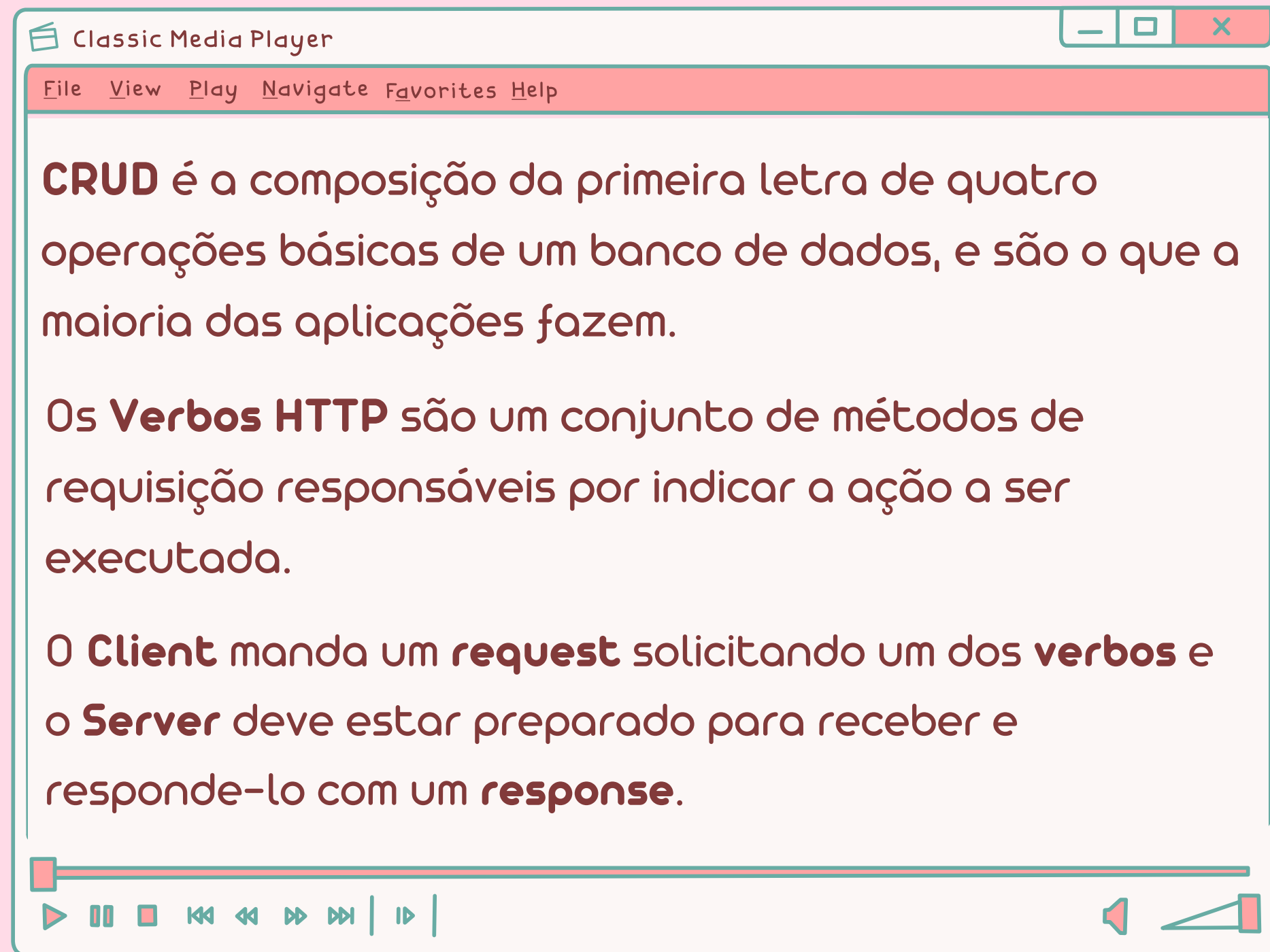


Sim, ele voltou! aquele que é um dos mais temidos de todas as edições da {reprograma}.

Todas as sisters estão convocadas para o  
***PROJETO GUIADO***

Semana 10 - Backend - Prof May

# Bora lembrar?



# Tipos de Parâmetros nas requisições

## ***request.query***

Awesome Module (Running)

NÃO faz parte do url e é passado no formato key=value. Esses parâmetros devem ser definidos pela desenvolvedora da API.

Quando queremos criar filtros para fazer consultas na nossa aplicação, o ideal é sempre usar o req.query. Quero filtrar por ano? Quero filtrar por cor? Por tipo? Por diretor? Vamos usar a Query.

**EX.: GET /livros/findByYear?**<sup>[key]</sup><sup>[value]</sup>**year=2000**

End Program

## ***request.params***

Awesome Module (Running)

São partes variáveis de um caminho de URI. Eles são tipicamente usados para apontar para um recurso específico dentro de uma coleção. Um URL pode ter vários parâmetros de caminho, cada um denotado com chaves { } **OU** dois pontos . Quando quero filtrar/deletar/atualizar usando um identificador único (username, cpf, ID) usamos o req.params;

**EX.: GET /livros/:id**

End Program

## ***request.body***

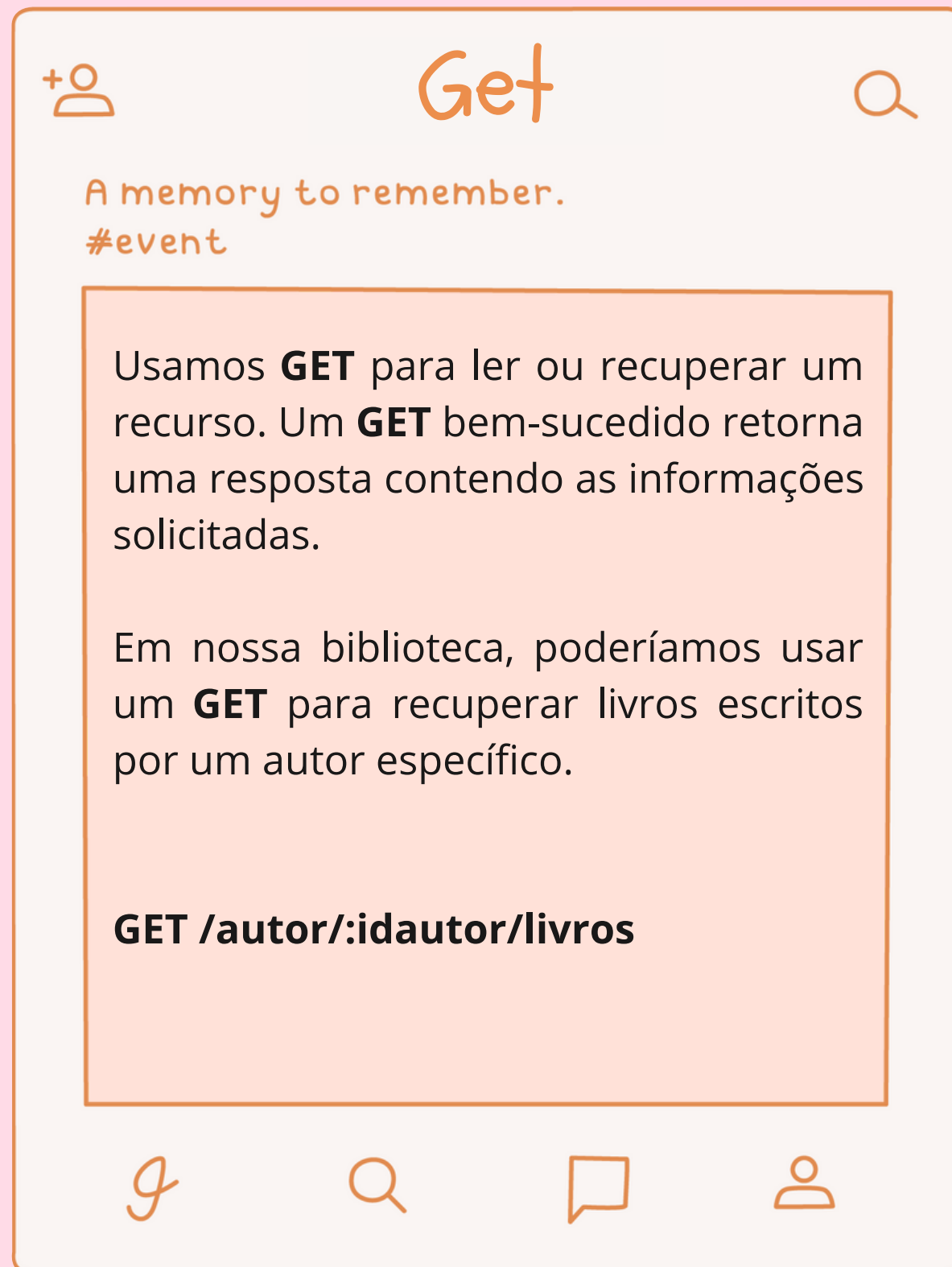
Awesome Module (Running)

É usado para enviar dados que serão cadastrados no banco, podem ser combinados com query ou path params.

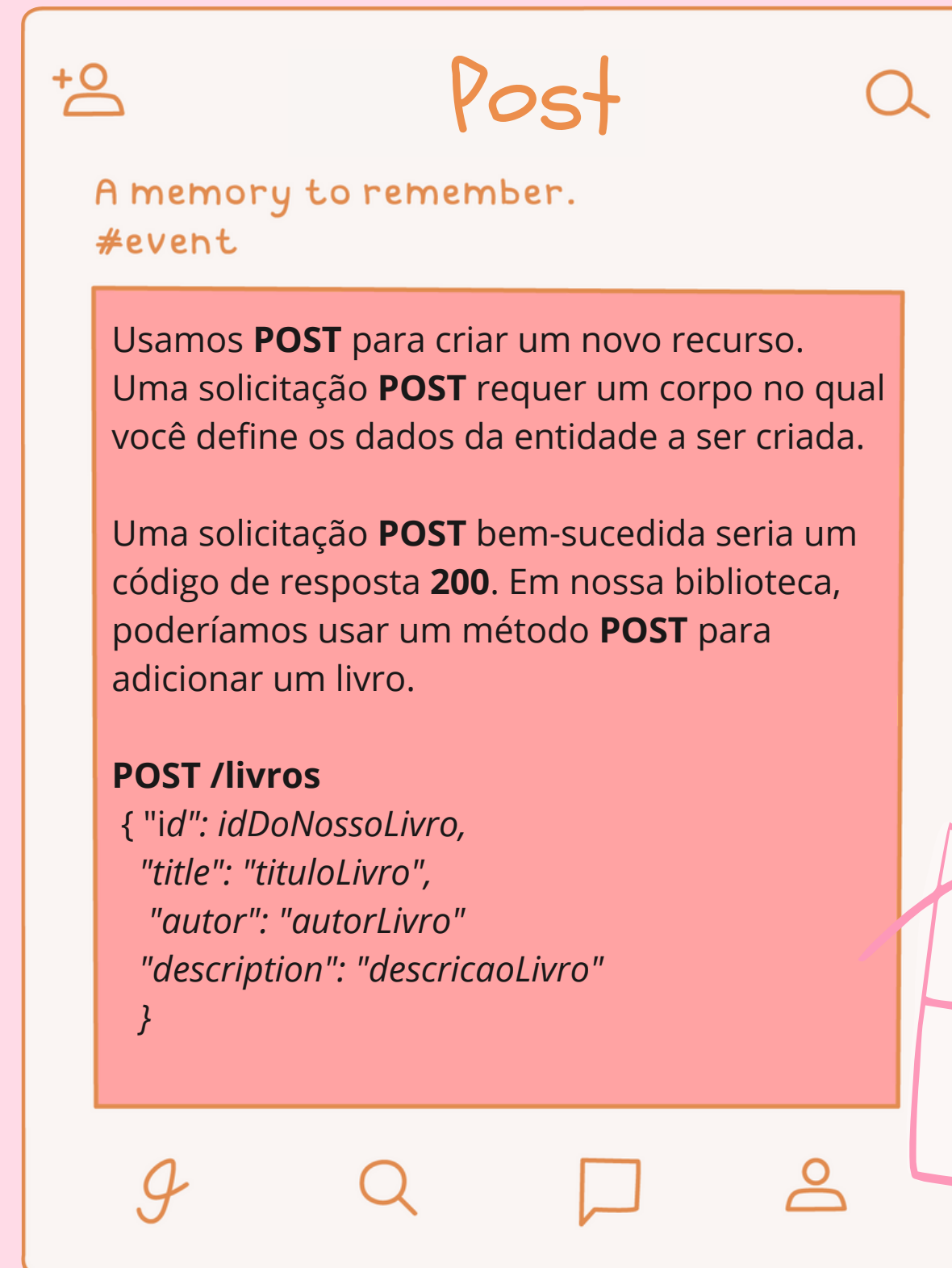
**EX.: {"filme": "Midsommar"}**

End Program

# HTTP - GET



# HTTP - POST



## HTTP - PUT

## HTTP - PATCH

## HTTP - DELETE



Put



A memory to remember.  
#event

O **PUT** substitui **todos** os atuais dados do recurso de destino pelos dados passados na **requisição**. Perceba, que estamos falando de uma atualização integral. Existe, então, a possibilidade de atualizar todo o recurso em apenas uma requisição.

**PUT /autor/:idautor**

```
{ "id": idDoNossoLivro,  
  "title": "tituloLivro",  
  "autor": "autorLivro"  
  "description": "descricaoLivro" }
```



Patch



A memory to remember.  
#event

O **Patch** aplica modificações parciais em um recurso. Logo, é possível modificar apenas uma parte do recurso. Perceba que **NÃO** estamos falando de um subconjunto de uma atualização completa (PUT), estamos falando de uma atualização parcial, o que torna as coisas mais flexíveis.

**PATCH /autor/:idautor**

```
{ "title": "novoTitulo" }
```



Delete




A memory to remember.  
#event

Usamos o método **DELETE** para remover um recurso ou uma coleção de recursos.

Quando em um formulário você clica no botão de **"Excluir"**, o evento que está sendo disparado passa pelos recursos do método **DELETE**.

**DELETE /autor/:idautor**



 **Doug Out**  
15 mins



Wonderful day!

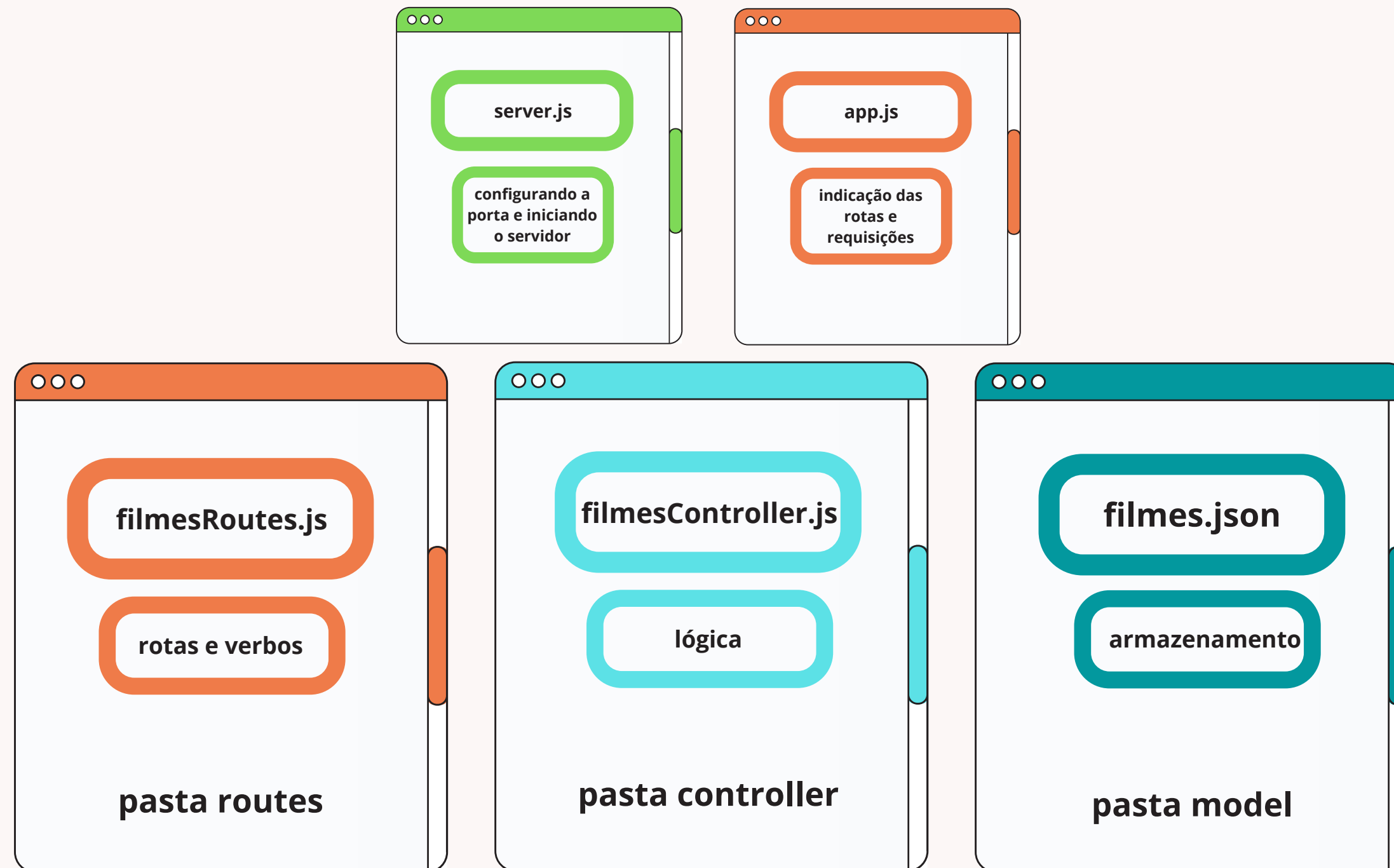
# Arquitetura MVC

MVC é um padrão de arquitetura de software, separando sua aplicação em 3 camadas. A camada de interação do usuário(view), a camada de manipulação dos dados(model) e a camada de controle(controller)

Já que estamos lidando com um projeto que tem somente back-end, não lidaremos com as views, porém lidamos com as rotas(routes).

O MVC nada mais é que uma forma de organizar o nosso código

 Like  Comment  Share



acessando JSON ●  
criando rotas ●  
criando lógica ●  
configurando a porta e iniciando o server ●



***Eai, gatinhas!***  
***Dúvidas?***





{reprograma}fy

⏏ ⏮ 🔊 4:50/7:05



## Projeto Guiado

46 views - May 27, 2022

👍 114K 💬 1K ➦ SHARE ⌵ SAVE ⋮



# {reprograma}fy

Que tal montarmos nossa própria playlist de músicas? Na nossa playlist poderemos favoritar nossas músicas e ver quais artistas tocam a música. Além disso, poderemos inserir novas músicas ou remover da nossa playlist as que não queremos mais.



## Contratos que deverão ser entregues

- |                 |   |
|-----------------|---|
| <b>[GET]</b>    | <i>/musics</i><br>- Retorna todas as músicas                          |
| <b>[GET]</b>    | <i>/musics/:id</i><br>- Retornar apenas uma música específica         |
| <b>[GET]</b>    | <i>/musics/artists</i><br>- Retornar músicas de um artista específico |
| <b>[POST]</b>   | <i>/musics</i><br>- Cadastrar nova música                             |
| <b>[PUT]</b>    | <i>/musics/:id</i><br>- Atualizar uma música específica (exceto o ID) |
| <b>[DELETE]</b> | <i>/musics/:id</i><br>- Deletar uma música específica                 |
| <b>[PATCH]</b>  | <i>/musics/:id/favorited</i><br>- Favoritar/desfavoritar música       |

# ***{reprograma}fy***

E se eu quiser ouvir um podcast?  
No nosso **{reprograma}fly** você  
poderá listar todos os podcasts e  
também filtra-los por tópico. Além  
disso, pode inserir novos podcasts ou  
remover algum já existente, além de  
conseguir avaliá-lo.

## ***Contratos que deverão ser entregues***

- [GET]** */podcasts*  
- Retorna todos os podcasts
- [GET]** */podcasts/topic*  
- Retornar apenas os podcasts de um tópico específico
- [POST]** */podcasts*  
- Cadastrar novo podcast
- [DELETE]** */podcasts/:id*  
- Deletar um podcast específico
- [PATCH]** */podcasts/:id/stars*  
- Alterar a nota de um podcast