

REST

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

“

REST é um tipo de arquitetura de serviços que **proporciona normas** entre sistemas informáticos para estabelecer a forma **como** vão se **comunicar**.



”

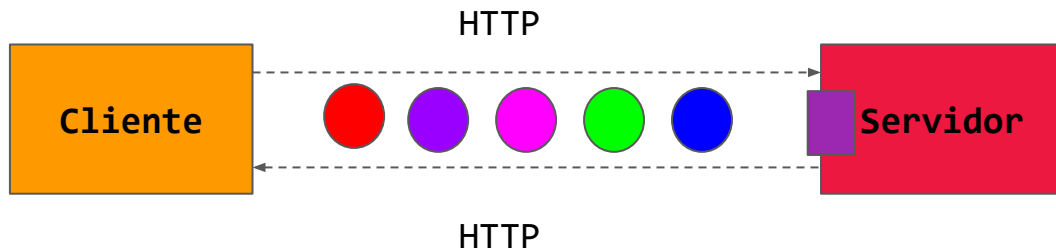
Índice

1. [Conceitos-chave](#)
2. [Formatos de envio de dados](#)

1 | Conceitos-chave

Arquitetura cliente-servidor

REST é uma arquitetura cliente-servidor, porque deve permitir que tanto a aplicação cliente como a aplicação servidor se desenvolvam ou escalem sem interferirem uma com a outra. Ou seja, permite a integração com qualquer outra plataforma e tecnologia, tanto cliente como servidor.



Recursos uniformes

Do lado do servidor, uma **arquitetura REST** expõe os clientes a uma interface uniforme.

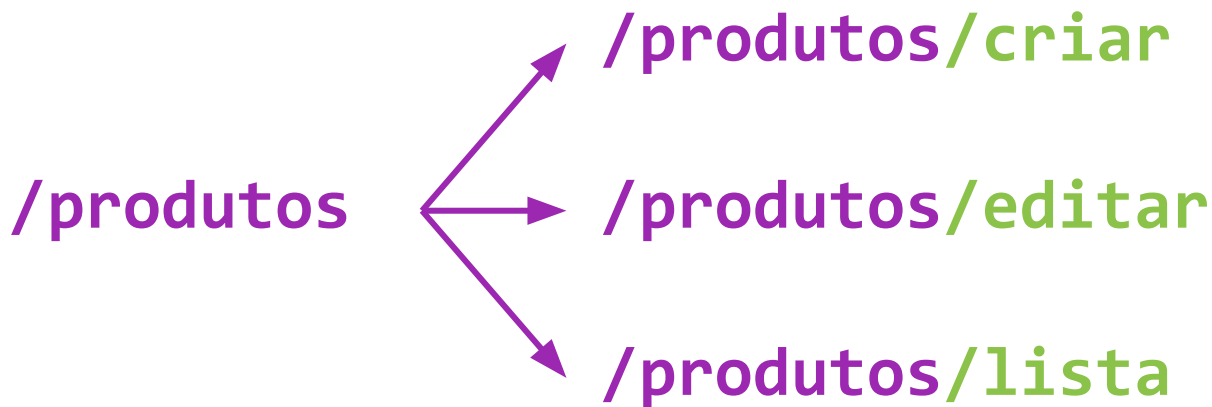
- Todos os recursos no servidor têm um nome sob a forma de URL ou hyperlink.
- Todas as informações são trocadas através do protocolo HTTP.

Chamamos estas URLs de **endpoints**, ou seja, o servidor expõe um conjunto de **endpoints** aos clientes para que possam ter acesso. A esta interface uniforme, ou seja, ao conjunto de **endpoints**, chamamos **API**.

Um **endpoint** está ligado ao recurso que solicitamos, tal recurso deve ter apenas um identificador lógico, e este deve permitir o acesso a toda a informação relacionada. Vejamos um exemplo a seguir.

Recursos uniformes

O servidor expõe a URL `/produtos/lista`. Este endpoint será ligado ao recurso que nos devolve a lista de produtos solicitados.



Sem estado (*stateless*)

REST propõe que todas as **interações** entre o cliente e o servidor devem ser tratadas como **novas** e de forma absolutamente **independente e sem guardar estado**.

Portanto, se quiséssemos por exemplo, que o servidor faça a distinção entre usuários logados ou convidados, devemos enviar todas as informações de autenticação necessárias **em cada pedido que fizermos ao servidor**.



Isto permite o desenvolvimento de aplicações mais **confiáveis, performantes e escaláveis**.

Cacheable

Em **REST**, o cache de dados é uma ferramenta muito importante, implementada do lado do cliente para melhorar o desempenho e reduzir a procura no servidor.



Ao colocar caches no cliente, evitamos fazer pedidos ao servidor.

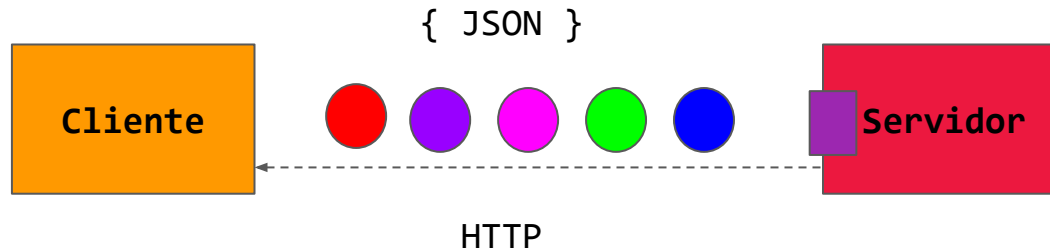
2

Formatos de envio de datos

Formatos

Quando o servidor envia um pedido, transfere uma representação do estado do recurso solicitado para o requisitante. Tal informação é entregue através de HTTP em um destes formatos: JSON (JavaScript Object Notation), RAW, XLT ou texto sem formato, URL-encoded.

JSON é o mais popular.



JSON

Quando queremos enviar dados no formato JSON, devemos adicionar um cabeçalho nos headers que diz:

```
{  
  "Content-Type": "application/json"  
}
```

```
JSON  
{  
  "id": 1,  
  "title": "..."  
}
```

RAW

É utilizado para enviar dados de texto sem qualquer formatação particular.

TEXT

É utilizado para enviar dados que não estão no formato JSON, como arquivos HTML e CSS.

URL-encoded

Indica que os dados codificados de URL serão enviados para nós.
Portanto, envia-nos algo muito semelhante a um query string.

Um dado enviado por este método se pareceria o seguinte:

```
{ } email%3Dcosme%40fulanito.fox%26password%3Dverysecret
```

Resumo

Uma arquitetura REST se caracteriza por seguir os seguintes princípios:

- Deve ser uma arquitetura **cliente-servidor**.
- Tem que ser **sem estado**, ou seja, não há necessidade de serviços para armazenar sessões de usuários (cada pedido ao servidor tem de ser independente dos outros).
- Deve apoiar um sistema de **caches**.
- Deve fornecer uma interface uniforme, para que a informação seja transferida de forma normalizada.
- Tem de ser um sistema em camadas invisíveis para o cliente.

DigitalHouse>
Coding School