

Construindo Aplicações REST com Slim PHP

Desenvolvimento de Aplicativos Web
João Eduardo Montandon

COLÉGIO TÉCNICO

Introdução

- Slim: Biblioteca desenvolvida em PHP que permite fazer aplicações REST de forma rápida e prática
- Principais Características
 1. Suporte nativo aos principais métodos HTTP (GET, POST, PUT, DELETE)
 2. Tratamento inteligente de rotas (URIs)
 3. Suporte a sessões, *caching*, e criptografia



<http://www.slimframework.com>

Slim: Instalação

Via Composer

```
$ cd my/project/folder/  
$ composer require slim/slim
```

Mais sobre Composer aqui: <https://getcomposer.org/>

Slim: Instalação

Via Download

1. Baixar Slim em <https://github.com/slimphp/Slim/releases>
2. Adicionar código no index.php

```
<?php  
require 'Slim/Slim.php';  
\Slim\Slim::registerAutoloader();
```

Slim: Instalação

Criar arquivo **.htaccess**

```
RewriteEngine On  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteRule ^ index.php [QSA,L]
```

Hello, REST

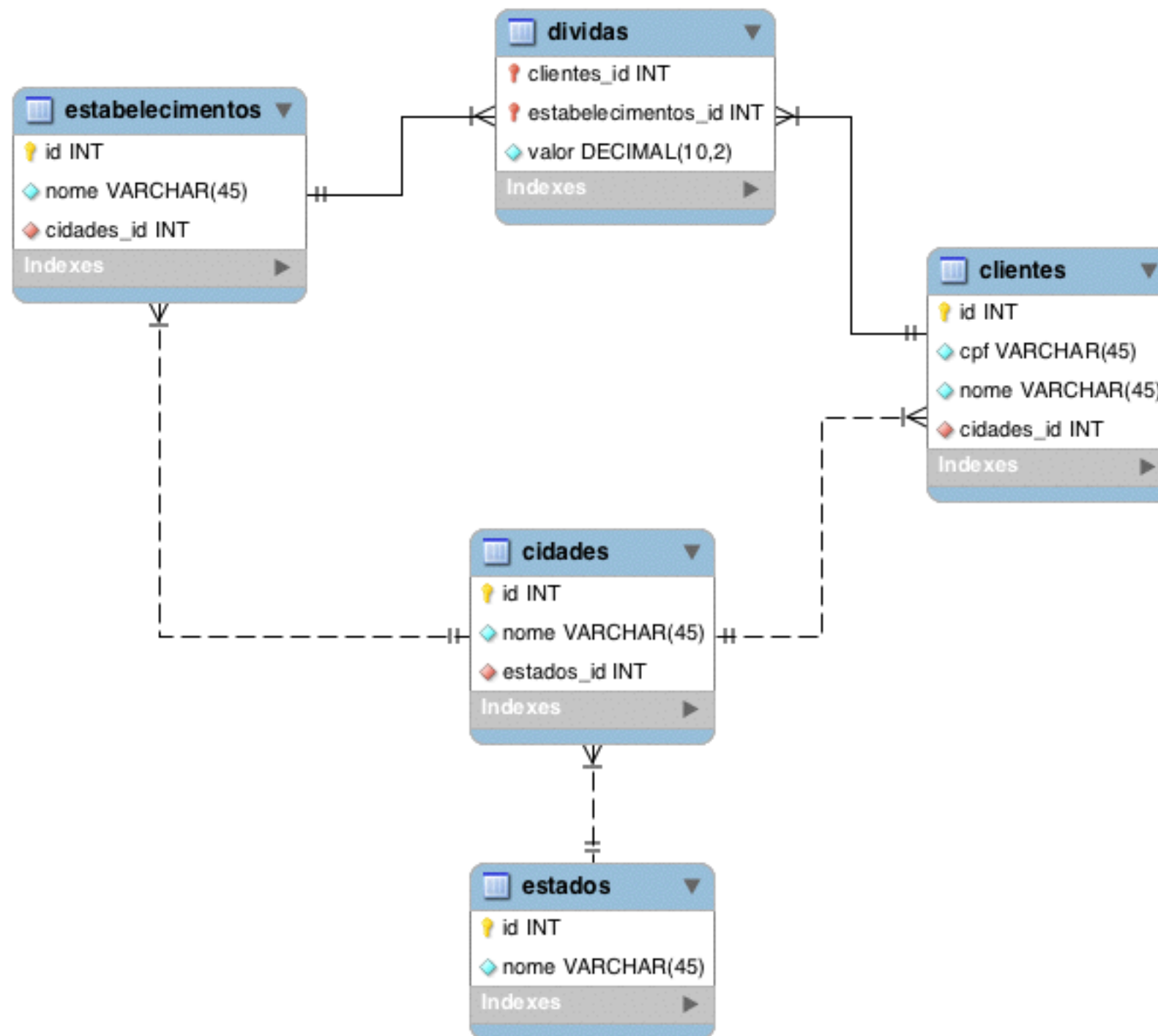
```
$app = new \Slim\Slim();
```

```
$app->get('/hello/:name', function ($name) {  
    echo "Hello, " . $name;  
});
```

```
$app->run();
```

<http://localhost:8888/hello/REST>

Construindo um App. REST



Construindo um App. REST

- E agora, o que fazer?
- Toda vez que quisermos implementar um serviço devemos:
 1. Definir **o que** será **exposto** pelo serviço
 2. Definir **o formato** desse **conteúdo**

Construindo um App. REST

O que será exposto?

- A. Definir quais entidades, e suas respectivas ações, que serão expostas
- B. Criar uma rota para cada uma das ações no sistema

Construindo um App. REST

Entidade	clientes
GET	GET /clientes GET /clientes/:cpf
POST	POST /clientes {cpf, nome, cidades_id}
PUT	PUT /clientes/:id {cpf, nome, cidades_id}
DELETE	DELETE /clientes/:id

Construindo um App. REST

```
$app->get('/clientes/:cpf'), function ($cpf) {  
    echo "GET /clientes/$cpf";  
});
```

```
$app->get('/clientes', function() {  
    echo "GET /clientes";  
});
```

```
$app->post('/clientes', function() {  
    echo "POST /clientes";  
});
```

```
$app->put('/clientes/:id'), function($id) {  
    echo "PUT /clientes/$id";  
});
```

```
$app->delete('/clientes/:id', function($id) {  
    echo "DELETE /clientes/$id";  
});
```

Construindo um App. REST

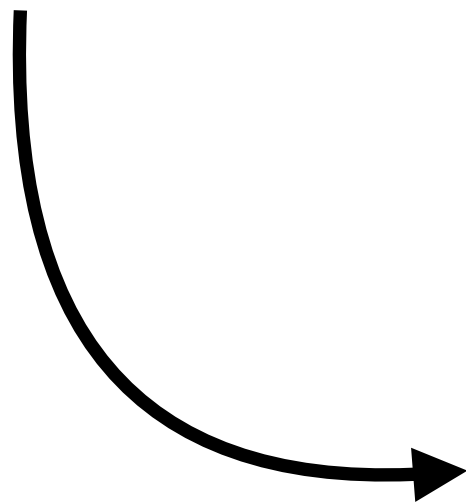
Formato do conteúdo

- Duas opções para envio/recebimento: XML ou JSON
- Vamos utilizar o JSON:
 - ✓ O PHP possui funções que codificam (saída) e decodificam (entrada) um objeto para JSON

Construindo um App. REST

GET /clientes/11122233311

```
$app->get('/clientes/:cpf', function ($cpf) {  
    //recupera o cliente  
    $cliente = ClienteDAO::getClienteByCPF($cpf);  
    echo json_encode($cliente);  
});
```

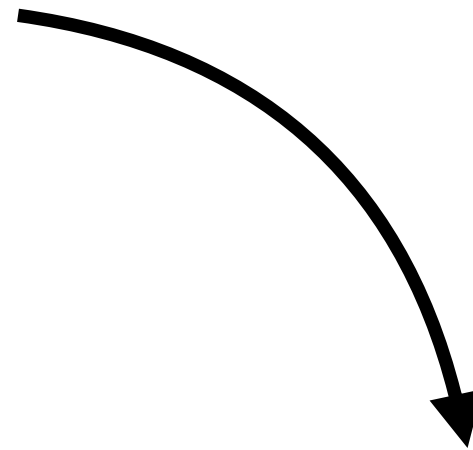


```
{  
    "id": "5",  
    "cpf": "12312312311",  
    "nome": "Marcos",  
    "cidades_id": "0"  
}
```

Construindo um App. REST

POST /clientes

```
{  
  "cpf": "22244466611",  
  "nome": "Juan",  
  "cidades_id": "0"  
}
```



```
$app->post('/clientes/:id', function ($id) {  
    // recupera o request  
    $request = \Slim\Slim::getInstance()->request();  
  
    // cadastra o cliente  
    $cliente = json_decode($request->getBody());  
    $cliente = ClienteDAO::updateCliente($cliente);  
  
    echo json_encode($cliente);  
});
```

Construindo um App. REST

Aparando arestas

- Definir *content-type*

```
$app->response->headers->set('Content-Type', 'application/json');
```

- Alterar status code

```
$app->response->setStatus(400);
```

- Redirecionando requisições

```
$app->response->redirect('/foo', 303);
```

Aplicações REST



<https://github.com/jemaf/restphp>

Construindo Aplicações REST com Slim PHP

Desenvolvimento de Aplicativos Web
João Eduardo Montandon

COLÉGIO TÉCNICO