

Serviços e Interoperabilidade de Sistemas

RESTful Web Services (PHP/Yii2)

Enquadramento - Modelo SOA

Service Oriented Architecture

Ideia principal (é um conceito!):

Componentes separados que suportam o processo de negócio

Lembrar competitividade e mudanças no mercado!

Disponível
na rede

Acesso
Standard

Fácil
alteração

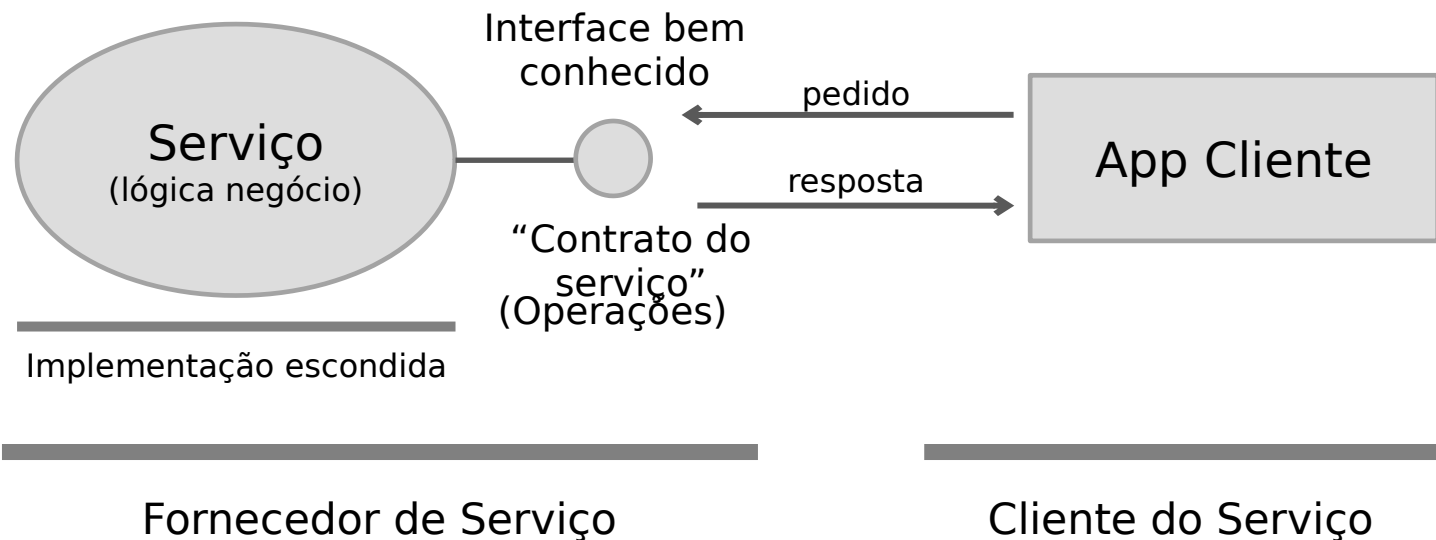
Fácil
reutilização

**Redução de tempo para responder a mudanças no
negócio!**

Enquadramento - Modelo SOA

SOA: mas, o que é um serviço?

Uma **unidade** básica da lógica de negócio



O cliente “não quer saber” como a função de negócio está implementada. Pretende apenas usar...

Enquadramento - Modelo SOA

O que é um Web Service

Materialização do modelo SOA

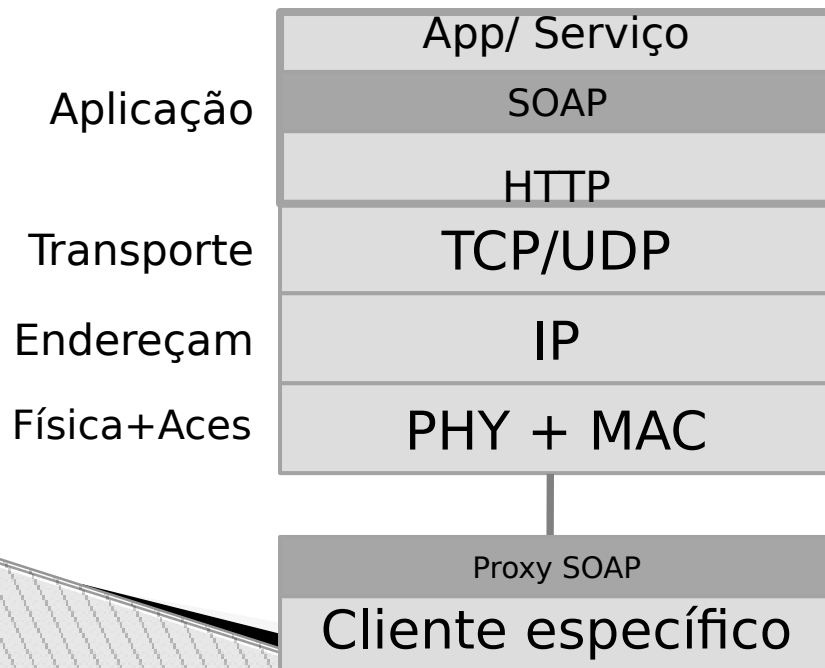
Web Service = Web + Service

É um serviço que 'vive' na Web

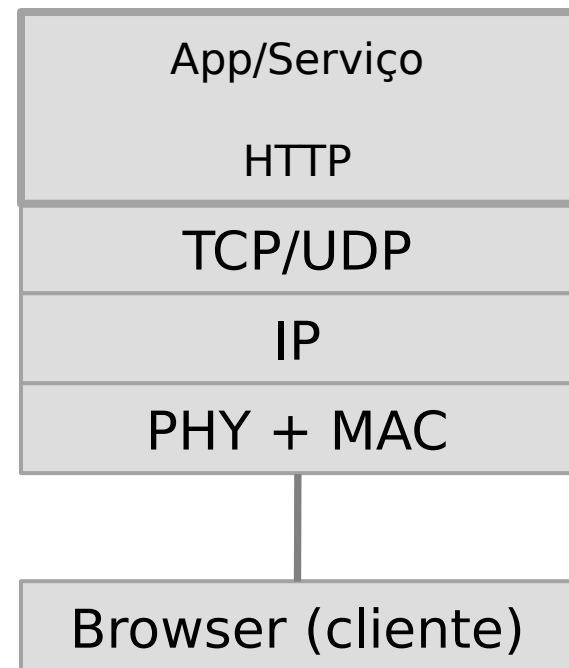
Web Services

Dois grandes tipos de Serviços Web:

Serviços Web SOAP



Serviços Web RESTful



100% Web

Web Services

As tecnologias Web

O que é Web?

Web Services

As tecnologias Web

A web é um **espaço global** de partilha de informação/serviços

Objetivo: troca de páginas web entre servidores HTTP e os clientes (browsers)

A Web é simplesmente:

HTTP + URIs + HTML

Serviços Web RESTful

Os serviços Web RESTful são 100% Web

Olham para a Web como um sistema de partilha de informação extremamente escalável

Os serviços Web RESTful dependem apenas de:

URIs – endereço/identificação do recurso

Estado do recurso se o recurso é alterado, o próximo cliente vê as alterações.

Interface uniforme – Verbos HTTP e nada mais

Serviços Web RESTful

Os serviços Web RESTful são 100% Web!

As operações/ações são inspiradas nas bases de dados:

CRUD - *create, read, update, delete*

POST - Cria novo recurso

GET - Obtém recurso(s)

PUT - Atualiza recurso

DELETE - Apaga recurso

90% do tráfego Web é GET

Serviços Web RESTful

Lembrar formato de mensagem HTTP

```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request
Message
Header

A blank line separates header & body

Request Message Body

```
HTTP/1.1 200 OK
```

```
Date: Sun, 08 Feb xxxx 01:11:12 GMT
```

```
Server: Apache/1.3.29 (Win32)
```

```
Last-Modified: Sat, 07 Feb xxxx
```

```
ETag: "0-23-4024c3a5"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 35
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<h1>My Home page</h1>
```

Status Line

Response Headers

Response
Message
Header

A blank line separates header & body

Response Message Body

Serviços Web RESTful

Os serviços Web RESTful são 100% Web!

Exemplos:

POST <http://www.abc.com/api/clientes>

No *request body* passamos a informação do novo cliente

Serviços Web RESTful

Os serviços Web RESTful são 100% Web!

Exemplos:

GET <http://www.abc.com/api/clientes>

Retorna todos os clientes

GET <http://www.abc.com/api/clientes/1>

Retorna o cliente de ID 1

Serviços Web RESTful

Os serviços Web RESTful são 100% Web!

Exemplos:

PUT <http://www.abc.com/api/clientes/2>

No *request body* passamos a nova informação do cliente de ID 2

Serviços Web RESTful

Os serviços Web RESTful são 100% Web!

Exemplos:

DELETE <http://www.abc.com/api/clientes/5>

Remove cliente de ID 5

Serviços Web RESTful

Os serviços Web RESTful na prática

Cada recurso (serviço) é identificado por um URI

www.pecas-loja.org/pecas

ou

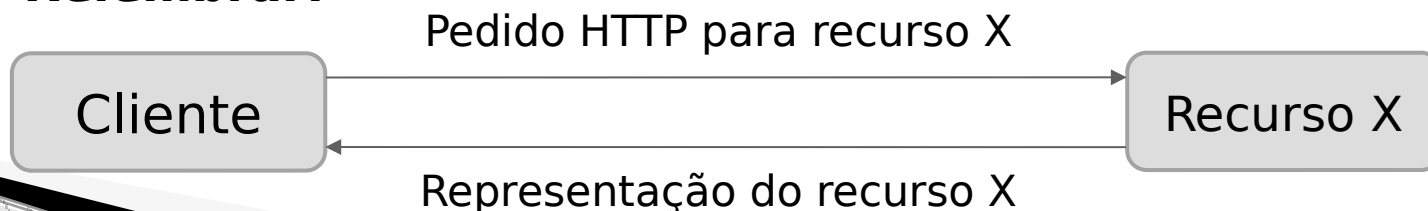
www.pecas-loja.org/pecas/099

nomes(recursos) e
não verbos!

E nunca:

~~www.pecas-loja.org/pecas/getpeca?id=099~~

Relembrar:



Serviços Web RESTful

Definição de URIs para os recursos (serviços)

Não é fácil!

Idealmente, cada conceito dentro da organização que tenha representação é candidato a ser exposto como recurso (serviço)

É como aprender Programação Orientada a Objeto depois de aprender Programação Procedimental

Exemplos:

<http://empresa.com/api/fatura>

<http://empresa.com/api/cliente>

<http://empresa.com/api/encomenda>

<http://empresa.com/api/fornecedor>

Serviços Web RESTful

Definição de URIs para os recursos (serviços)

Ainda assim, podemos usar apenas um URI?



Serviços Web RESTful

Definição de URIs para os recursos (serviços)

Ainda assim, podemos usar apenas um URI?

Sim, porque existem as “query strings”

GET servicotemp.org/salaestar?addtemp=22

Localização do recurso Chave Valor

Mas vai **contra** as recomendações RESTful! Neste caso não usa o verbo **POST** para registar a temperatura!

Serviços Web RESTful

E o estado, é guardado no recurso (servidor) ou no cliente?

No servidor é impossível (escala Web!)

Clientes devem guardar o estado!

Exemplo:

Se o servidor envia um token/password ao cliente, este **deve** enviar o mesmo em todos os pedidos seguintes

Não é possível ao servidor memorizar o IP do cliente (ou algo do género) para saber que este já foi autenticado antes!

Serviços Web RESTful

Erros?

Certos programadores retornam sempre **Status Code**=“**200 OK**” e detalham o erro, a existir, no response body

Não é uma boa prática já que força a criação de clientes especializados e não interoperáveis

200 – Ok

404 – Not Found

500 – Internal Server Error

Etc.

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

1. XAMP/WAMP
2. *Composer e o plugin "fxp/composer-asset-plugin:1.2.0"*
3. Instalar Yii2 para aplicação baseada na **template** Basic:
 - `cd c:\xampp\htdocs`
 - `composer create-project --prefer-dist yiisoft/yii2-app-basic yii2Basic`

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

Se necessário base de dados:

<http://127.0.0.1/phpmyadmin/>

Exemplo:

basic database

tabela **clientes**

id, nome, morada

Configurar no ficheiro **config\db.php**

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

Teste da aplicação criada:

<http://127.0.0.1/yii2basic/web/>

Usando o Webserver
(Apache) do XAMP

Usando Webserver embebido do php:

`c:\xampp\htdocs\yii2Basic>yii serve localhost:8888`

Depois: <http://localhost:8888/>

Serviços Web RESTful

Template Advanced:
Front end + backend

Criação de serviços Web RESTful em PHP/Yii2

Aplicação criada...



Congratulations!

You have successfully created your Yii-powered application.

Get started with Yii

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

Serviços Web RESTful

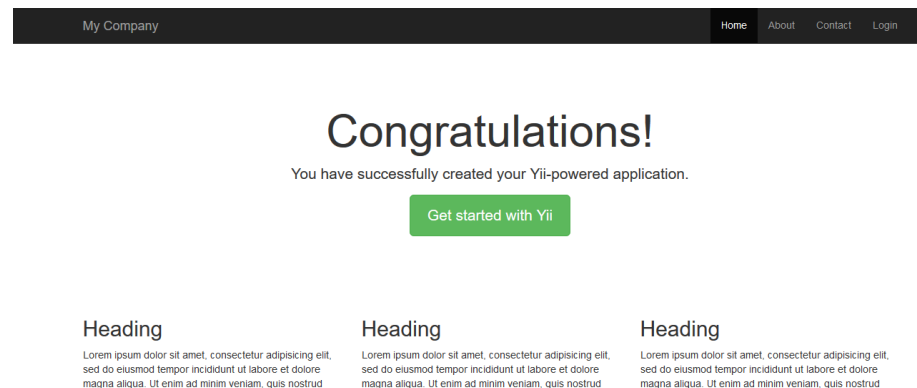
Criação de serviços Web RESTful em PHP/Yii2

- **Nota:**
 - Um serviço RESTful **não tem** interface gráfica
 - É usado para M2M (machine-to-machine)
- Vamos instalar, no Firefox, o plugin **RestClient**
 - Para testarmos os serviços RESTful...
 - (Chrome => Postman)

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **URLs ‘bonitas’:**
 - Quando clicamos no menu **Contact** vemos como URL: `http://localhost:8888/index.php?r=site%2Fcontact`



Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **URLs ‘bonitas’:**
 - No **config/web.php** descomentar bloco **urlManager**

```
'urlManager' => [  
    'enablePrettyUrl' => true,  
    'showScriptName' => false,  
    'rules' => [  
    ]  
]
```

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **URLs 'bonitas':**
 - No **config/web.php** descomentar bloco **urlManager**
 - Agora obtemos a URL:
http://localhost:8888/site/contact



Controller

View

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- Para criar um **formulário** de clientes para CRUD iríamos criar um **model** e um **CRUD**...
- **Nesta UC apenas nos preocupamos com a API REST**
 - Formulários não fazem qualquer sentido aqui

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **RESTful webservice para clientes:**

Objetivo: os clientes poderem fazer:

GET `http://localhost:8888/clientes`

GET `http://localhost:8888/clientes/9`

POST `http://localhost:8888/clientes`

PUT `http://localhost:8888/clientes/2`

DELETE `http://localhost:8888/clientes/3`

Clientes, faturas,
fornecedores,
produtos,
encomendas, posts,
etc.

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

RESTful webservice para clientes:

1. Tabela para guardar os dados?

Criar **model** (localhost:8888/**gii**)

- Table : clientes
- Model: Clientes

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

RESTful webservice para clientes:

2. Quem recebe o pedido dos clientes?

Criar **controller** (localhost:8888/**gii**) em:

app\controllers**ClientesController**

(Não é necessária View. Desligar.)

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

RESTful webservice para clientes:

- Derivar **controller** de **ActiveController** e fornecer model:

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

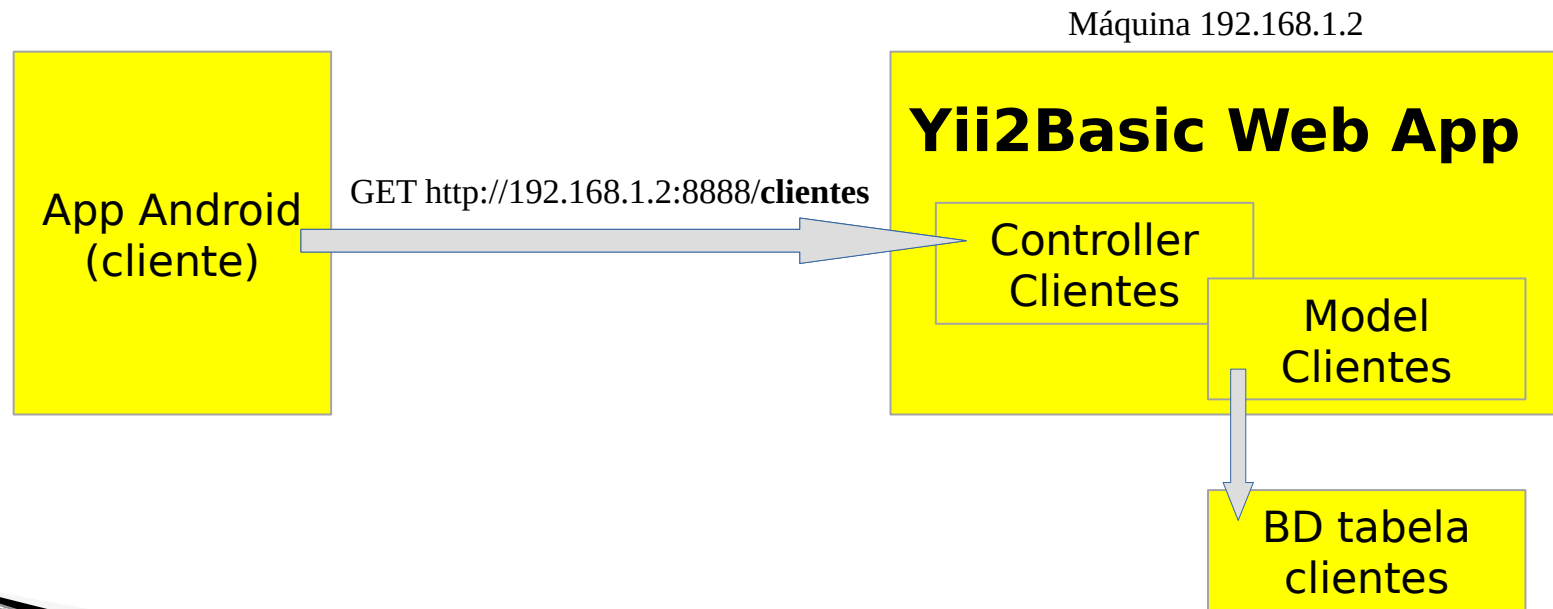
RESTful webservice para clientes:

```
<?php
namespace app\controllers;
use yii\rest\ActiveController;
class ClientesController extends ActionController
{
    public $modelClass = 'app\models\Clientes';
}
```

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

RESTful webservice para clientes:



Yii2Basic Web App à escuta no porto 8888

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **RESTful CRUD para tabela clientes:**
 - Teste do GET:

Method URL

Headers

Header name	Header value
Accept: application/json	
Content-Type: application/json	
content-type	application/json

Body

Request Body

[+] Response

```
{ "id": 4, "nome": "ccc", "morada": "ccc" }
```

Serviços Web RESTful

► Criação de serviços Web RESTful em PHP/Yii2

- **RESTful CRUD para tabela clientes:**

- Teste do POST:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8888/clientes
- Headers:**
 - Accept: application/json
 - Content-Type: application/json
- Body:** {"nome": "zzz", "morada": "kkk"}
- Response:** {"nome": "zzz", "morada": "zzz", "id": 7}

A yellow highlight box contains the text: **Post, Put, Delete e GET {id} não funcionam...**

Serviços Web RESTful

► Criação de serviços Web RESTful em PHP/Yii2

- **RESTful CRUD para tabela clientes:**

- Teste do POST:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8888/clientes
- Headers:**
 - Accept: application/json
 - Content-Type: application/json
- Body:**

```
{ "nome": "zzz", "morada": "kkk" }
```

Below the configuration, a response is shown:

```
{ "nome": "zzz", "morada": "zzz", "id": 7 }
```

Overlaid on the bottom of the screenshot is a yellow text box with the following text:

Post, Put, Delete e GET {id} não funcionam.
Teremos que registar a nova **rota** e suporte **JSON**.

Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

config/web.php

Registo de nova rota (controlador)

```
'urlManager' => [  
    'enablePrettyUrl' => true,  
    'showScriptName' => false,  
    'rules' => [  
        ['class' => 'yii\rest\UrlRule', 'controller' => 'clientes',  
        'pluralize' => false],  
    ]  
]
```

Minúsculas

Para evitar problemas...



Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

Ligar suporte para JSON: (config/web.php)

```
'components' => [  
    'request' => [  
        'cookieValidationKey' => '1VuIjiNm2kN78qsNDdJ7bEK2WDOFPTZS',  
  
        'parsers' => [  
            'application/json' => 'yii\web\JsonParser',  
        ],  
    ],
```



Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **RESTful CRUD para tabela clientes:**
 - Teste do POST:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8888/clientes
- Headers:**
 - Accept: application/json
 - Content-Type: application/json
 - Header name: content-type, Header value: application/json
- Body:** {"nome": "zzz", "morada": "kkk"}
- Response:** {"nome": "zzz", "morada": "zzz", "id": 7}



Serviços Web RESTful

Criação de serviços Web RESTful em PHP/Yii2

- **Convenção de nomes para controllers (REST) / chamada**
 - **ClientesController** → /http://localhost:8888/**clientes**
 - **ClientesApiController** → /http://localhost:8888/**clientes-api**

