

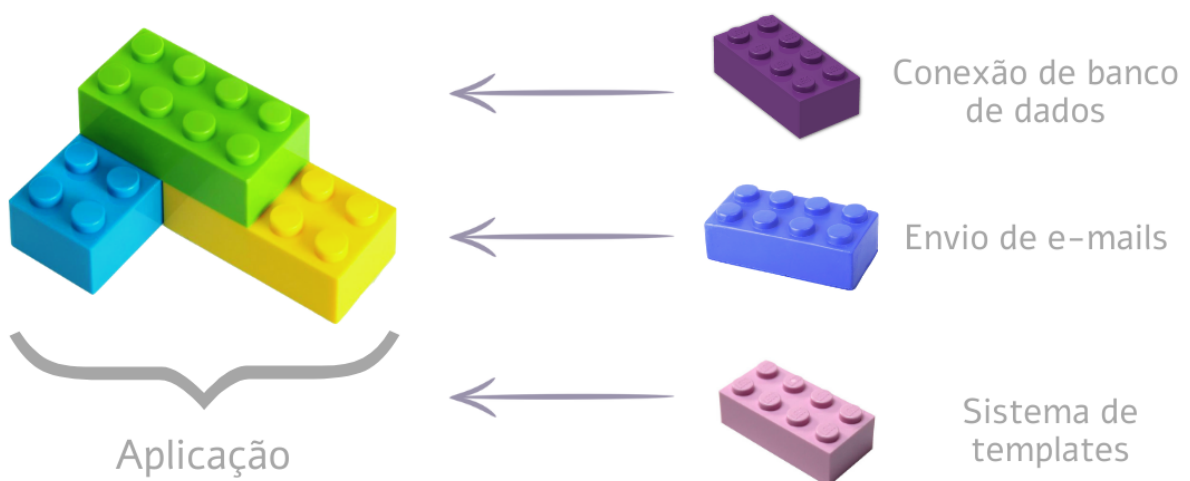
Introdução ao Slim Framework

*O Slim Framework é um **micro-framework**, escrito em PHP para criação de APIs. Por sua baixa complexidade de uso e por ser essencialmente modular, tem se destacado como uma boa opção para o desenvolvimento de APIs, pequenas aplicações ou websites.*

*Como um de seus principais recursos, o Slim implementa um **sistema de gerenciamento de rotas**, permitindo que funções e classes PHP possam ser chamadas por meio de requisições HTTP, tais como chamadas get, post, put e delete.*

Como principais características, podemos citar:

- *Baixa necessidade de configuração;*
- *Por não possuir um padrão de desenvolvimento definido, há liberdade para definir uma estrutura personalizada para o projeto;*
- *Facilidade na integração de aplicações externas ao projeto;*
- *Possui suporte nativo a rotas HTTP;*
- *Possui suporte a injeção de dependências;*
- *Possui uma comunidade ativa, entre outros.*



Micro-Frameworks são Frameworks modularizados que possuem uma estrutura inicial muito mais simples quando comparado a um Framework convencional.

Configuração e Instalação do Slim

A instalação do Slim deve ser realizada por meio do gerenciador de dependências do PHP, o Composer. Com o Composer devidamente instalado em seu computador, acesse a pasta do projeto e digite o seguinte comando:

```
composer require slim/slim:3.*
```

Após a execução do comando, a pasta do projeto deve se parecer com o esquema abaixo:

```
--ProjetoIFPR/  
---Vendor/  
-----autoload.php  
-----Slim/  
-----PSR/  
-----Composer/  
-----Pimple/  
-----Nikic/  
---composer.json  
---composer.lock
```

O Composer é um gerenciador de dependências focado no PHP, e funciona de forma semelhante ao NPM com NodeJS. No Composer ao invés do package.json temos o composer.json, e ao invés de node_modules temos a pasta vendor.

Saiba mais sobre a instalação do composer no Windows [clikando aqui](#).

Gerenciamento de Rotas

Assim como mencionado anteriormente, uma das grandes razões para se usar o Slim é devido ao seu sistema de gerenciamento de rotas. Vamos configurar a primeira rota da API, que corresponderá à raiz da API.

Assegurado que estamos dentro da pasta do projeto, **criaremos um arquivo chamado `index.php`**. O conteúdo desse arquivo de seguir o código abaixo:

```
<?php
require __DIR__ . '/../vendor/autoload.php';

use Slim\App;
use Slim\Http\Request as Request;
use Slim\Http\Response as Response;

$app = new App;

$app->get('/hello/{name}', function (Request $request, Response
$response, array $args) {

    $name = $args['name'];
    $response->getBody()->write("Hello, {$name}");

    return $response;

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

Disponibilizando o website no localhost

No **Terminal** acesse a pasta do projeto e execute o comando:

```
php -S localhost:8080
```

Agora, no seu browser, acesse `localhost:8080` e você verá uma mensagem de **Page Not Found**, agora acesse `localhost:8080/hello/World` e a sua página deve estar funcionando e exibindo o texto "Hello World".

Pronto, instalamos o Slim e o Composer! Estamos prontos para desenvolver a rota da nossa rota index.

Para isto cole o seguinte código no seu arquivo index.php:

```
$app->get('/', function (Request $request, Response $response,  
array $args) {  
    $response->getBody()->write("Our index page! (Home page)");  
    return $response;  
});
```

Acesse localhost:8080 e veja a página funcionando!