

Curso



Criando aplicações PHP com Zend Framework 3, ops, Zend Expressive

Flávio Gomes da Silva Lisboa
www.fgsl.eti.br



Middleware



Templating



Dependency Injection



Routing



Error Handling

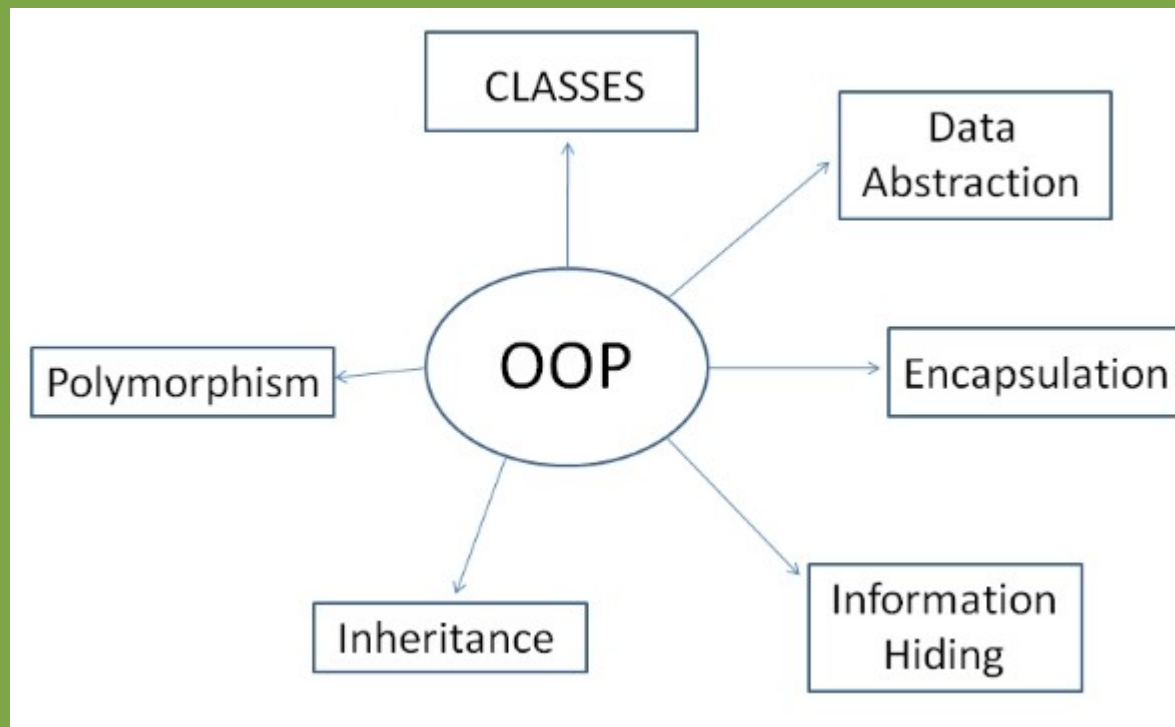
O curso

Descrição: O curso será totalmente **prático**. Os alunos irão compreender os conceitos de criação de uma aplicação Web com Zend Expressive a medida que se defrontarem com os problemas lançados durante o curso e forem apresentados aos componentes gerenciados pelo *microframework*. O principal conceito envolvido, de *middleware* ficará bem claro quando os alunos perceberem a diferença entre a proposta do Zend Expressive e dos *frameworks full-stack*.

O curso

Pré-Requisitos:

Conhecimentos intermediários de PHP (linguagem de programação + programação orientada a objetos)



O curso

Tópicos do Curso:

- Instalação e configuração do Zend Expressive
- Criação do banco de dados e da tabela para o exercício
- Acesso a banco de dados com injeção de dependências
- Roteamento para URLs amigáveis
- Interface com usuário com padrões MVC + Two Step View
- Manipulação de erros na aplicação
- Reuso com sua própria biblioteca

Ingredientes do curso

- PHP 7.0.8
- Apache 2.4
- MySQL 5.7
- PhpMyAdmin 4



O que é Zend Expressive?

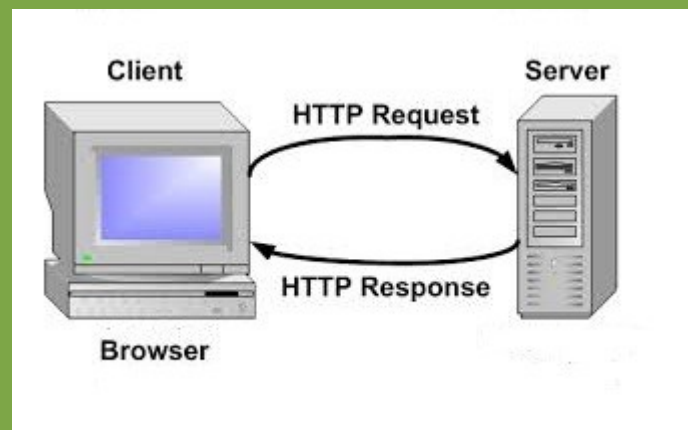
É uma implementação de *middleware* da Zend baseado na PSR-7 (*HTTP message interfaces*).



E o que é um *middleware*?

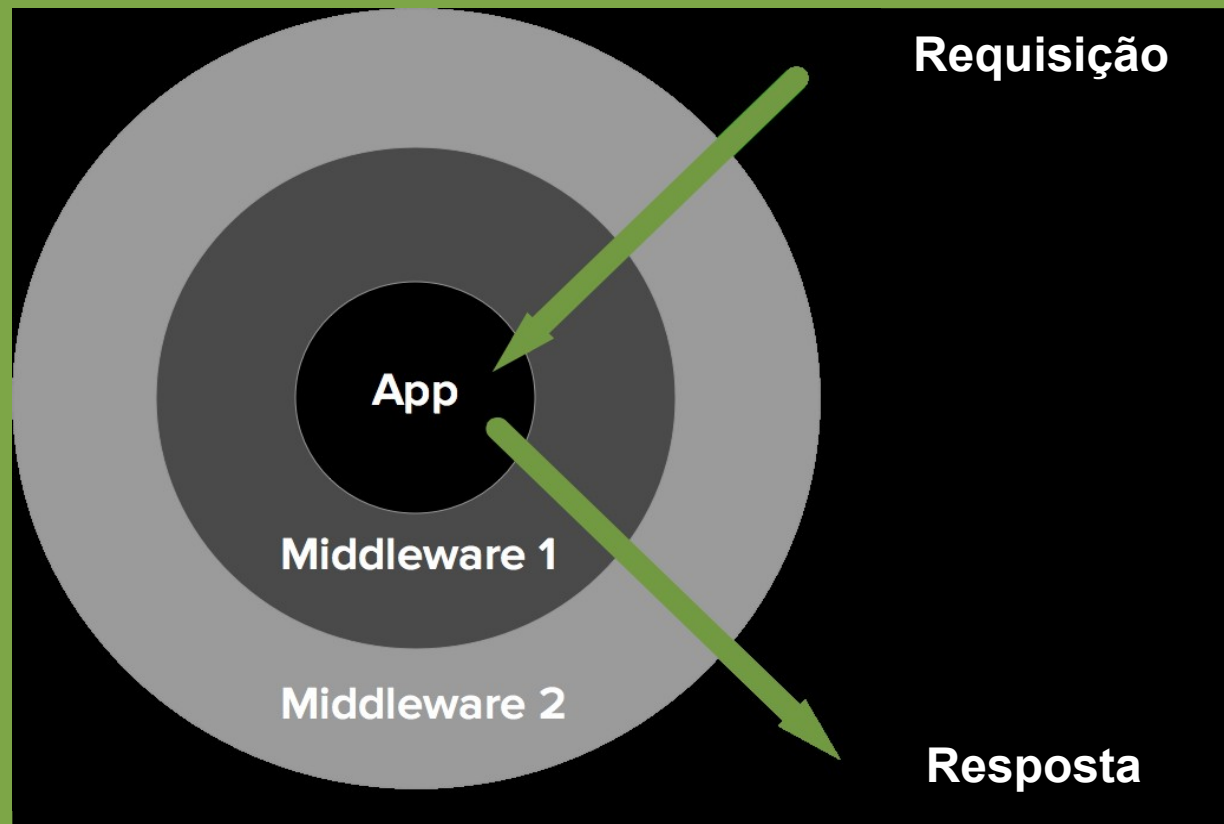
O protocolo HTTP (RFC 2616) define como padrão uma **comunicação síncrona**, na qual um dispositivo emite uma solicitação de um arquivo de texto e recebe uma resposta que pode conter o arquivo solicitado ou não.

De qualquer modo, o funcionamento básico do protocolo é **enviar texto e receber texto**.



E o que é um *middleware*?

É um software que realiza operações entre a requisição e a resposta HTTP que não fazem parte do protocolo.



Certo, então
middleware é só
outro nome pra
aplicação Web...



Não segundo a PSR-15

“Um componente HTTP *middleware* é um **componente individual** que participa junto com outros componentes *middleware* do processamento de uma requisição HTTP e da criação de uma resposta HTTP, como definido pela PSR-7.”

<https://github.com/php-fig/fig-standards/blob/master/proposed/http-middleware/middleware.md>

Middleware X Framework

Pela definição da PSR-15 já percebemos que um *middleware* não é um *framework*, mas pode constituir parte de um.

No caso do *Zend Expressive*, ele é um componente do *Zend Framework 3*.

zend-escaper

Securely and safely escape HTML, HTML attributes, JavaScript, CSS, and URLs.

[!\[\]\(c694a3ff3b077d76910920a6a1593ab4_img.jpg\) Github](#)[!\[\]\(ec9132f1d27c8919987d92907322654d_img.jpg\) Documentation](#)

zend-eventmanager

Implement events, signal slots, aspects, and observers!

[!\[\]\(dd161862f9164df98f62b726e9846241_img.jpg\) Github](#)[!\[\]\(758ebdf4629c903da74c2e079717ae32_img.jpg\) Documentation](#)

zend-expressive

PSR-7 middleware in minutes.

[!\[\]\(a8f9309f944226d1420f5fed22e2b6e6_img.jpg\) Github](#)[!\[\]\(248b91fcdac4810ffd15cf33fb6aec6f_img.jpg\) Documentation](#)

zend-feed

Consume and generate Atom and RSS feeds, and interact with Pubsubhubbub.

[!\[\]\(cbd8541a32dfc32f356f5c6c994b0a21_img.jpg\) Github](#)[!\[\]\(d3e32d099174a7c248ec1f564ee4f69c_img.jpg\) Documentation](#)

zend-file

Locate PHP classfiles.

zend-filter

Programmatically filter and normalize data and files.

Framework



Middleware



Já ouviu alguém falar isso?



Eu não gosto de
usar framework
em PHP, parece
que você está
programando em
Java!

www.fgsl.eti.br

Crie um diretório de projeto

```
$ mkdir expressive  
$ cd expressive
```

Instale o Zend Expressive

\$ composer require zendframework/zend-expressive zendframework/zend-expressive-fastroute zendframework/zend-servicemanager

```
/opt/lampp/htdocs/expressive$ composer require zendframework/zend-expressive zendframework/zend-exp
cemanager
Search for a package:
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Generating autoload files
/opt/lampp/htdocs/expressive$ ls
composer.json  vendor
```

Use o Composer

composer.json ✕

```
{
    "require": {
        "zendframework/zend-expressive": "*",
        "zendframework/zend-expressive-fastroute": "*",
        "zendframework/zend-servicemanager": "3.1.*"
    }
}
```

Use o Composer

```
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing zendframework/zend-escaper (2.5.2)
  Loading from cache

- Installing psr/http-message (1.0.1)
  Downloading: 100%

- Installing zendframework/zend-stratigility (1.2.1)
  Downloading: 100%

- Installing zendframework/zend-expressive-template (1.0.3)
  Downloading: 100%

- Installing zendframework/zend-expressive-router (1.2.0)
  Downloading: 100%

- Installing zendframework/zend-diactoros (1.3.5)
  Downloading: 100%

- Installing container-interop/container-interop (1.1.0)
  Loading from cache

- Installing zendframework/zend-expressive (1.0.0)
  Downloading: 100%

- Installing nikic/fast-route (v1.0.1)
  Downloading: 100%

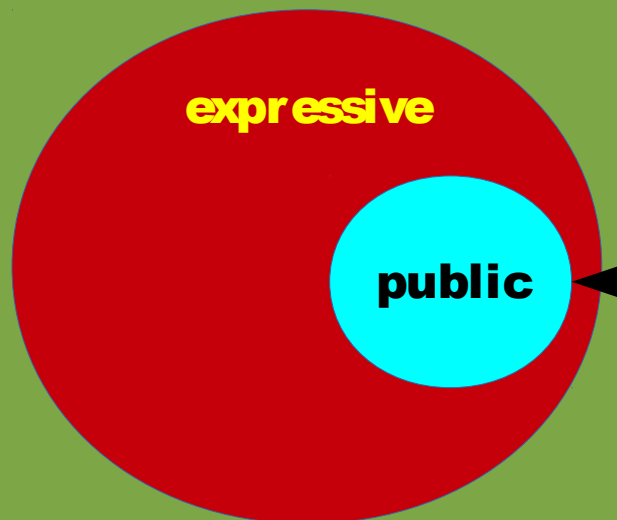
- Installing zendframework/zend-expressive-fastroute (1.2.0)
  Downloading: 100%

- Installing zendframework/zend-servicemanager (3.1.1)
  Loading from cache
```

Crie um diretório público

Para servir sua aplicação com segurança.

```
$ mkdir public
```



Crie um index.php no diretório público

(Este é o *script de bootstrap*)

```
<?php
use Zend\Expressive\AppFactory;

chdir(dirname(__DIR__));
require 'vendor/autoload.php';

$app = AppFactory::create();

$app->get('/', function ($request, $response, $next) {
    $response->getBody()->write('Hello, world!');
    return $response;
});

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

Inicie um servidor web

Em desenvolvimento, você pode usar o próprio PHP para isso (≥ 5.4).

```
$ php -S 0.0.0.0:8080 -t public/
```

```
/opt/lampp/htdocs/expressive$ /opt/lampp/bin/php -S 0.0.0.0:8080 -t public
PHP 7.0.6 Development Server started at Fri Sep  2 09:16:56 2016
Listening on http://0.0.0.0:8080
Document root is /opt/lampp/htdocs/expressive/public
Press Ctrl-C to quit.
```

Simples



Sugestão de ambiente

https://www.apachefriends.org/pt_br/index.html

XAMPP Apache + MariaDB + PHP + Perl

O que é o XAMPP?

XAMPP é o ambiente de desenvolvimento PHP mais popular

XAMPP é completamente gratuito e fácil de instalar a distribuição Apache contendo MariaDB, PHP e Perl. O pacote de código aberto do XAMPP foi criada para ser extremamente fácil de instalar e de usar.



Download

Clique aqui para outras versões



XAMPP para Windows
v5.6.24 (PHP 5.6.24)

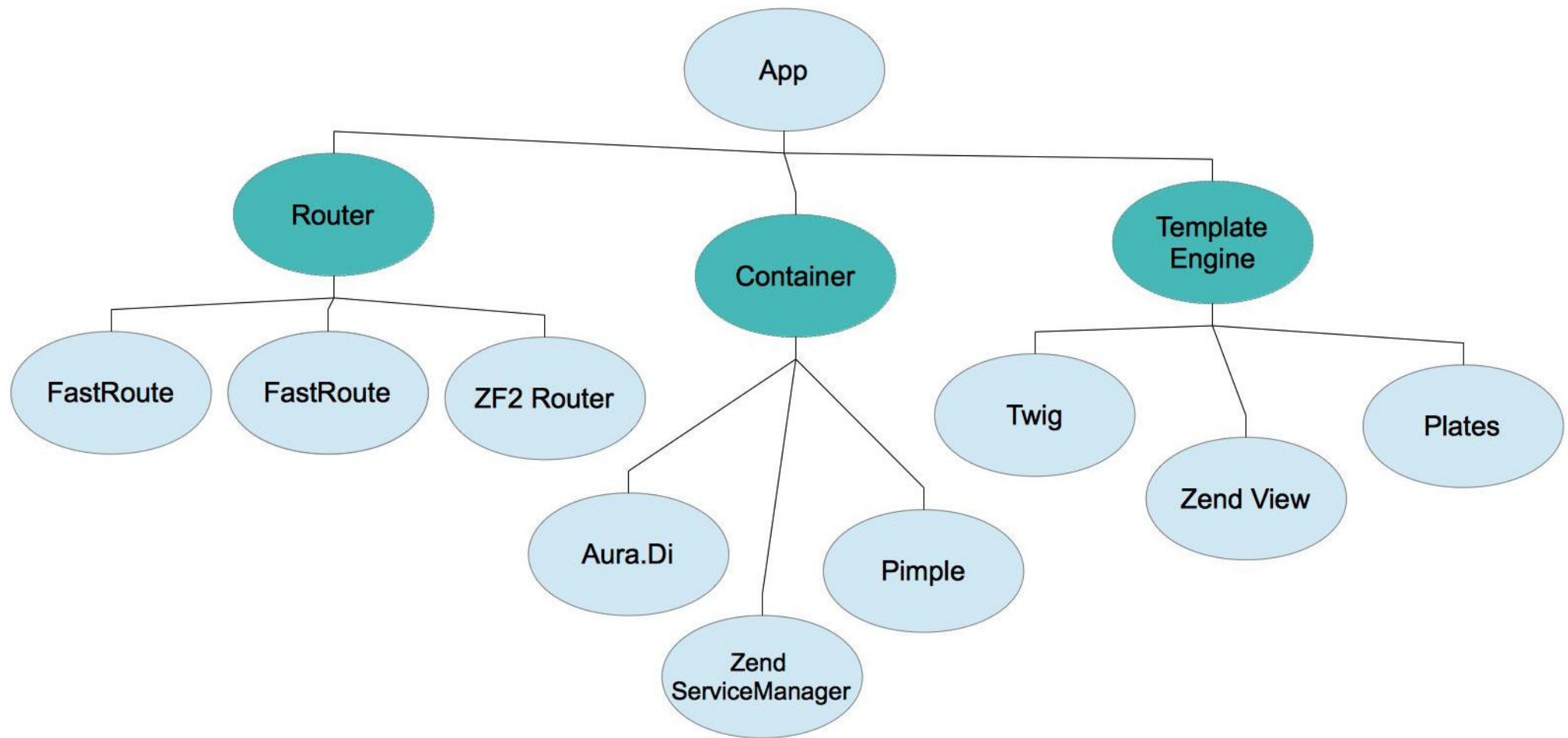


XAMPP para Linux
v5.6.24 (PHP 5.6.24)



XAMPP para OS X
v5.6.24 (PHP 5.6.24)

Aplicação Zend Expressive



Instalação e configuração do Zend Expressive

Precisamos do Composer:

<https://getcomposer.org/download/>

Criamos um esqueleto de projeto no diretório web root:

```
composer create-project zendframework/zend-expressive-skeleton escola
```

Instalação e configuração do Zend Expressive

NOTA

O Composer faz cache dos pacotes baixados, assim se outro projeto na mesma máquina precisar dos mesmo pacotes, eles não serão necessariamente baixados pela rede.

Instalação e configuração do Zend Expressive

Instale o esqueleto completo:

```
Installing zendframework/zend-expressive-skeleton (1.0.3)
- Installing zendframework/zend-expressive-skeleton (1.0.3)
  Downloading: 100%

Created project in escola
> ExpressiveInstaller\OptionalPackages::install
Setup data and cache dir
Setting up optional packages

Minimal skeleton? (no default middleware, templates, or assets; configuration only)
[y] Yes (minimal)
[n] No (full; recommended)
Make your selection (No):
```

Instalação e configuração do Zend Expressive

Instale as dependências recomendadas:

```
Which router do you want to use?
```

- [1] Aura.Router
- [2] FastRoute
- [3] Zend Router

Digite ENTER em todas as opções

```
Make your selection or type a composer package name and version (FastRoute):
```

- Adding package zendframework/zend-expressive-fastroute (^1.0)
- Copying /config/autoload/routes.global.php

```
Which container do you want to use for dependency injection?
```

- [1] Aura.Di
- [2] Pimple
- [3] Zend ServiceManager

```
Make your selection or type a composer package name and version (Zend ServiceManager):
```

- Adding package zendframework/zend-servicemanager (^2.7.3 || ^3.0)
- Copying /config/container.php

```
Which template engine do you want to use?
```

- [1] Plates
- [2] Twig
- [3] Zend View installs Zend ServiceManager
- [n] None of the above

```
Make your selection or type a composer package name and version (n):
```

Instalação e configuração do Zend Expressive

MAS NÃO INSTALAMOS
NENHUM *TEMPLATE*
ENGINE!
COMO VAMOS FAZER AS
PÁGINAS?



Instalação e configuração do Zend Expressive

CALMA AÍ CARA



VOCE TÁ MUITO EXALTADO

GERADORMEMES.COM

Podemos instalar qualquer dependência a qualquer momento adicionando o nome do pacote no arquivo composer.json e executar o comando `composer update`.

Instalação e configuração do Zend Expressive

Mantenha a calma e selecione o manipulador de erros recomendado:

```
Which error handler do you want to use during development?
```

```
[1] Whoops
```

```
[n] None of the above
```

```
Make your selection or type a composer package name and version (Whoops):
```

```
- Adding package filp/whoops (^1.1 || ^2.0)
```

```
- Copying /config/autoload/errorhandler.local.php
```

Instalação e configuração do Zend Expressive

As dependências serão baixadas do packagist.org se não estiverem atualizadas no cache.

```
Installing dependencies (including require-dev)
- Installing zendframework/zend-escaper (2.5.2)
  Downloading: 100%

- Installing psr/http-message (1.0.1)
  Downloading: 100%

- Installing zendframework/zend-stratigility (1.2.1)
  Downloading: 100%

- Installing zendframework/zend-expressive-template (1.0.3)
  Downloading: 100%

- Installing zendframework/zend-stdlib (3.1.0)
  Loading from cache

- Installing zendframework/zend-expressive-router (1.2.0)
  Downloading: 100%

- Installing zendframework/zend-diactoros (1.3.7)
  Downloading: 100%

- Installing container-interop/container-interop (1.1.0)
  Downloading: 100%

- Installing zendframework/zend-expressive (1.0.0)
  Downloading: 100%
```

Instalação e configuração do Zend Expressive

Ao final do processo, o Composer criará o arquivo autoload.php na pasta vendor.

```
zendframework/zend-expressive-template suggests installing zendframework/zend-expressive-platesrenderer (^0.1 to use the Plates template render
er)
zendframework/zend-expressive-template suggests installing zendframework/zend-expressive-twigrenderer (^0.1 to use the Twig template renderer)
zendframework/zend-expressive-template suggests installing zendframework/zend-expressive-zendviewrenderer (^0.1 to use the zend-view PhpRender
r template renderer)
zendframework/zend-expressive-router suggests installing zendframework/zend-expressive-aurarouter (^0.1 to use the Aura.Router routing adapter)
zendframework/zend-expressive-router suggests installing zendframework/zend-expressive-zendrouter (^0.1 to use the zend-mvc routing adapter)
zendframework/zend-expressive suggests installing aura/di (3.0.*@beta to make use of Aura.Di dependency injection container)
zendframework/zend-expressive suggests installing xstreamwayz/pimple-container-interop (^1.0 to use Pimple for dependency injection)
zendframework/zend-expressive suggests installing zendframework/zend-expressive-aurarouter (^1.0 to use the Aura.Router routing adapter)
zendframework/zend-expressive suggests installing zendframework/zend-expressive-platesrenderer (^1.0 to use the Plates template renderer)
zendframework/zend-expressive suggests installing zendframework/zend-expressive-twigrenderer (^1.0 to use the Twig template renderer)
zendframework/zend-expressive suggests installing zendframework/zend-expressive-zendrouter (^1.0 to use the zend-mvc routing adapter)
zendframework/zend-expressive suggests installing zendframework/zend-expressive-zendviewrenderer (^1.0 to use the zend-view PhpRenderer templat
e renderer)
zendframework/zend-expressive-helpers suggests installing mouf/pimple-interop (^1.0 to use Pimple for dependency injection)
zendframework/zend-expressive-helpers suggests installing aura/di (3.0.*@beta to make use of Aura.Di dependency injection container)
zendframework/zend-servicemanager suggests installing ocramus/proxy-manager (ProxyManager 1.* to handle lazy initialization of services)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (>=2.2.1)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
filp/whoops suggests installing symfony/var-dumper (Pretty print complex values better with var-dumper available)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
Writing lock file
Generating autoload files
```

Abertura do projeto no Zend Eclipse PDT

Baixe o Zend Eclipse PDT:

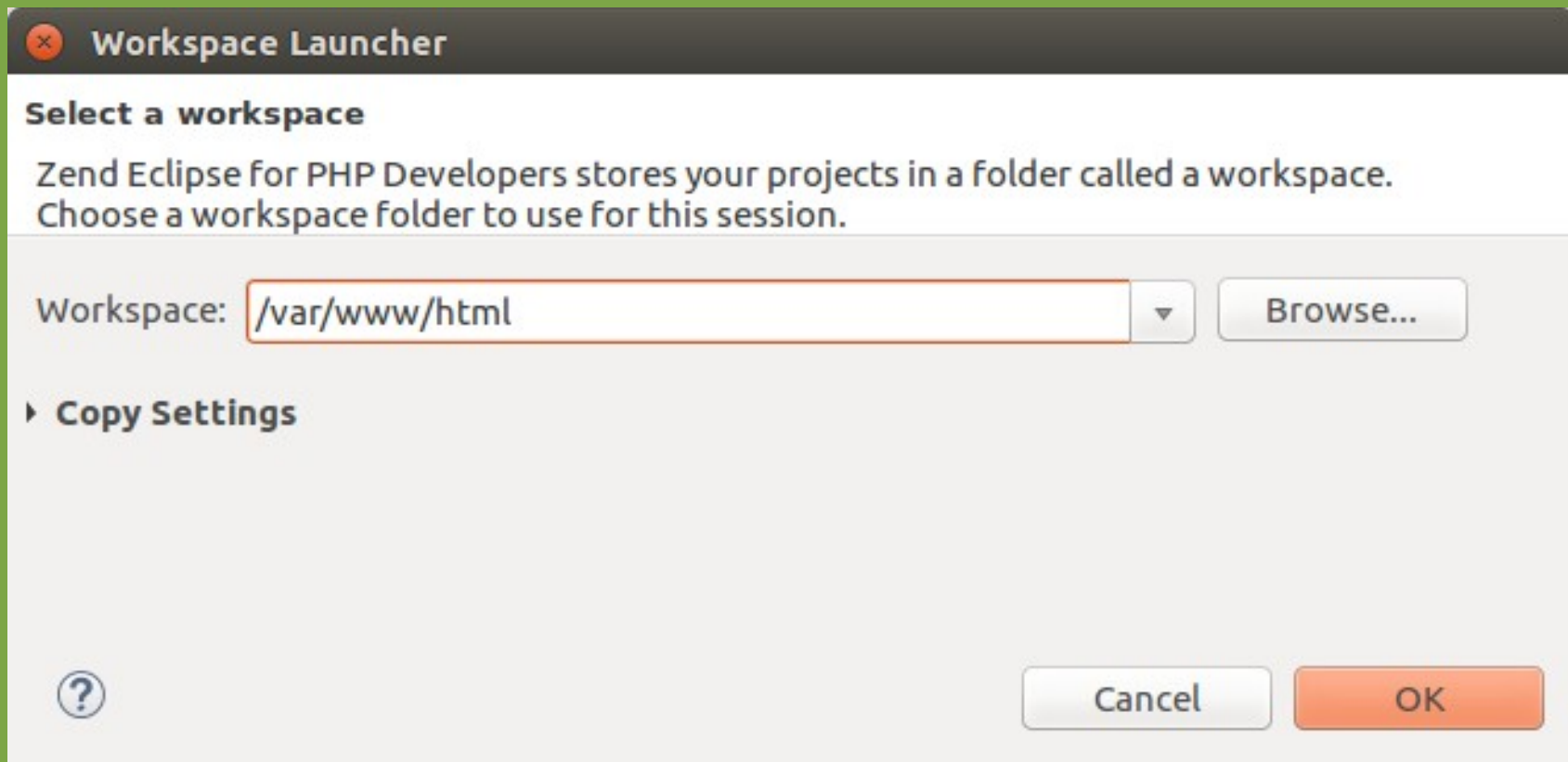
<http://www.zend.com/en/company/community/pdt/downloads>

Você vai precisar de:



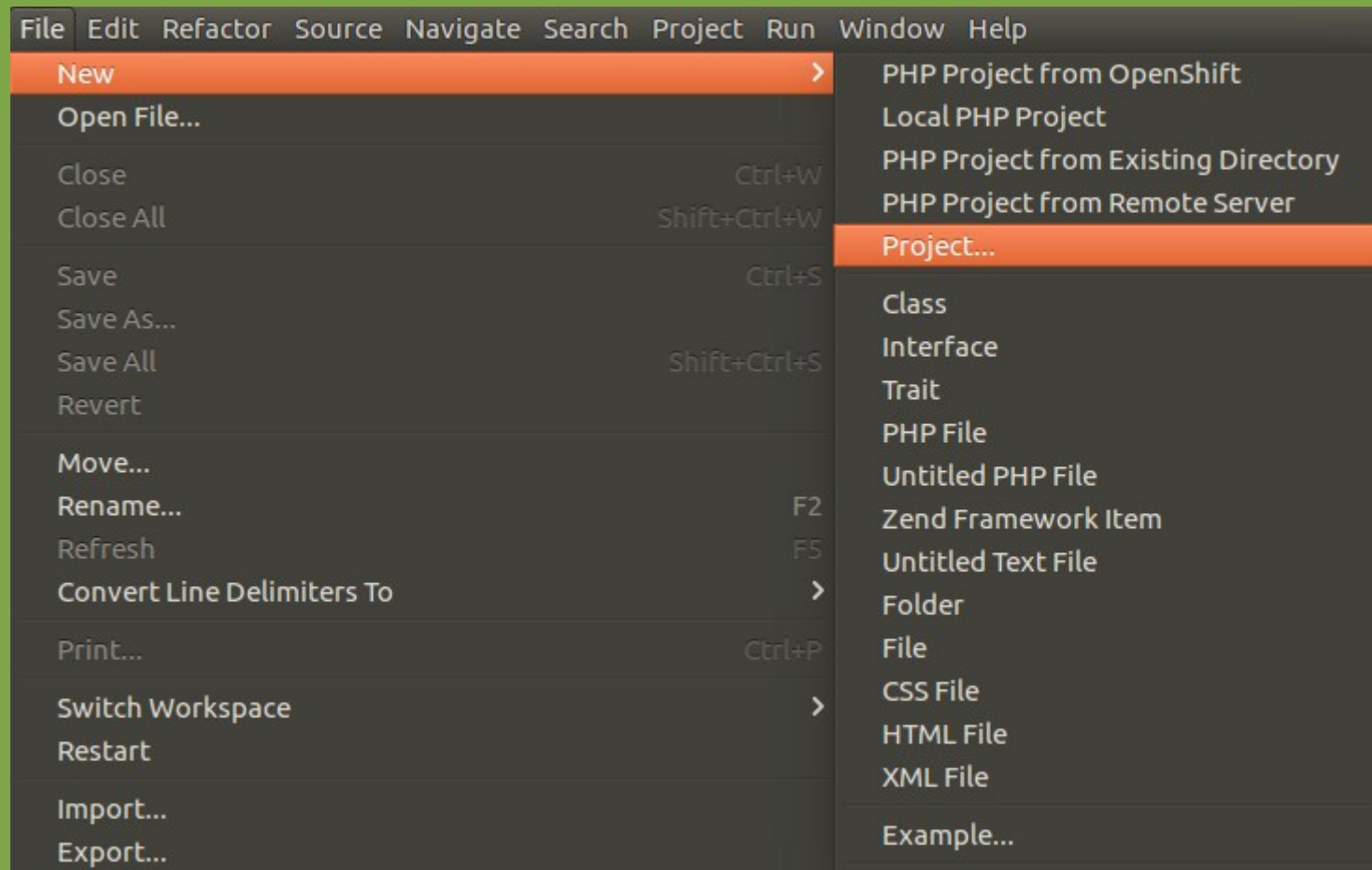
Abertura do projeto no Zend Eclipse PDT

Abra o Zend Eclipse e configure o diretório *web root* como *workspace*:



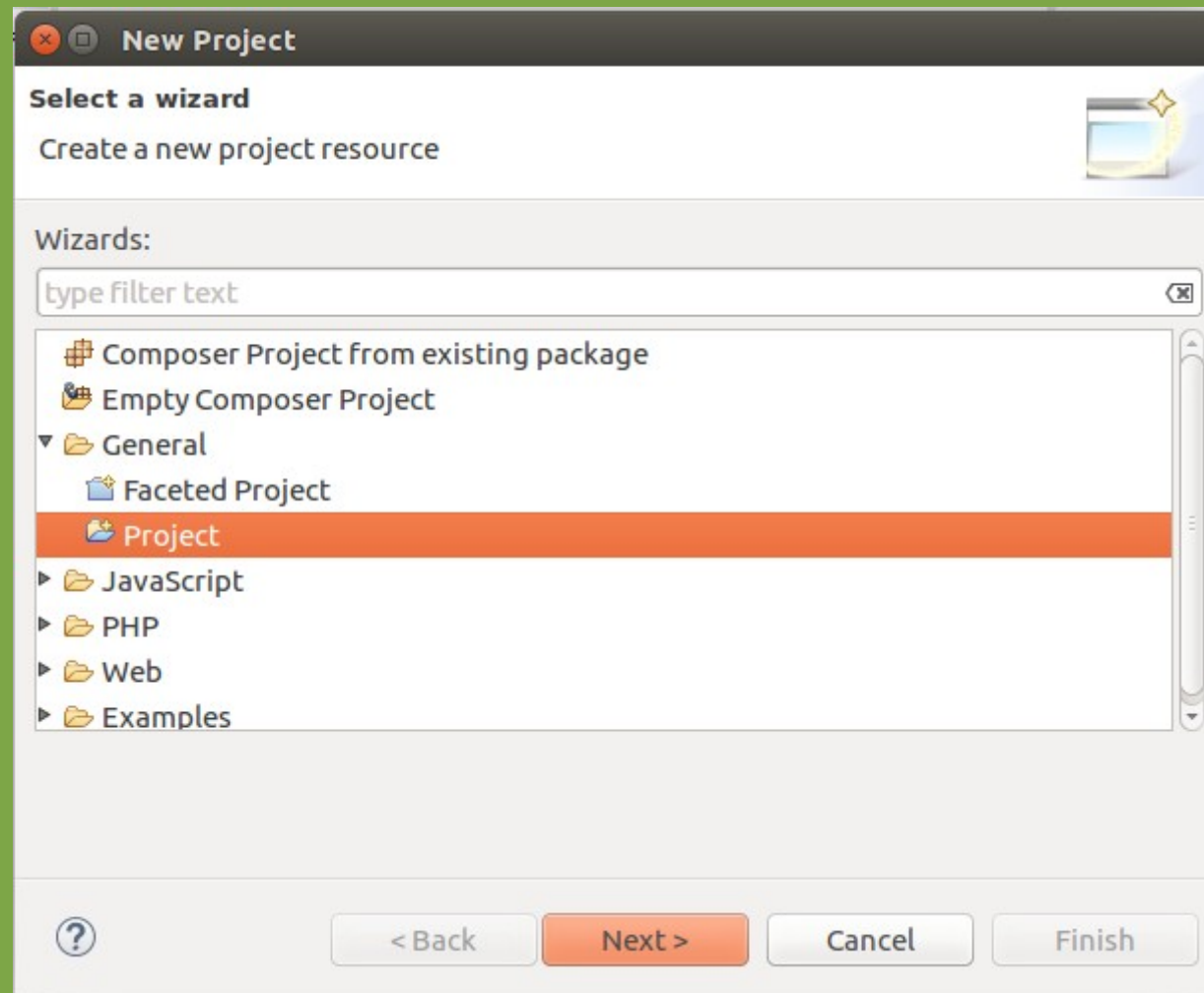
Abertura do projeto no Zend Eclipse PDT

Crie um novo projeto:



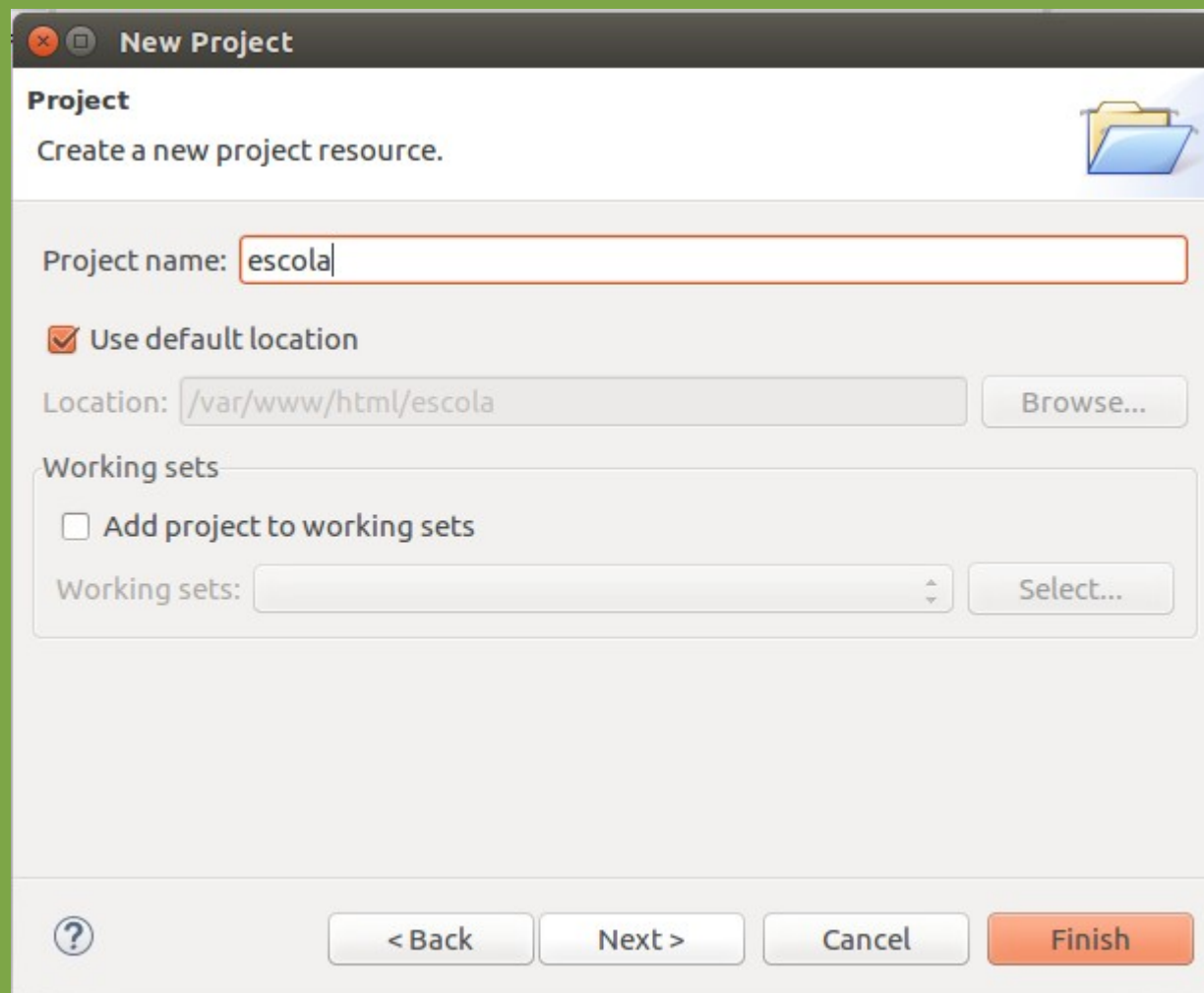
Abertura do projeto no Zend Eclipse PDT

Crie um novo projeto:



Abertura do projeto no Zend Eclipse PDT

Atribua o nome do esqueleto criado pelo Composer:



The screenshot shows the 'New Project' dialog box in the Eclipse IDE. The title bar reads 'New Project'. The main heading is 'Project' with the instruction 'Create a new project resource.' and a folder icon. The 'Project name' field contains the text 'escola'. The 'Use default location' checkbox is checked. The 'Location' field shows the path '/var/www/html/escola' with a 'Browse...' button to its right. Under the 'Working sets' section, the 'Add project to working sets' checkbox is unchecked. Below this, there is a 'Working sets:' label followed by a dropdown menu and a 'Select...' button. At the bottom of the dialog, there is a help icon (question mark) on the left and four buttons: '< Back', 'Next >', 'Cancel', and 'Finish'.

New Project

Project
Create a new project resource.

Project name:

☒ Use default location

Location:

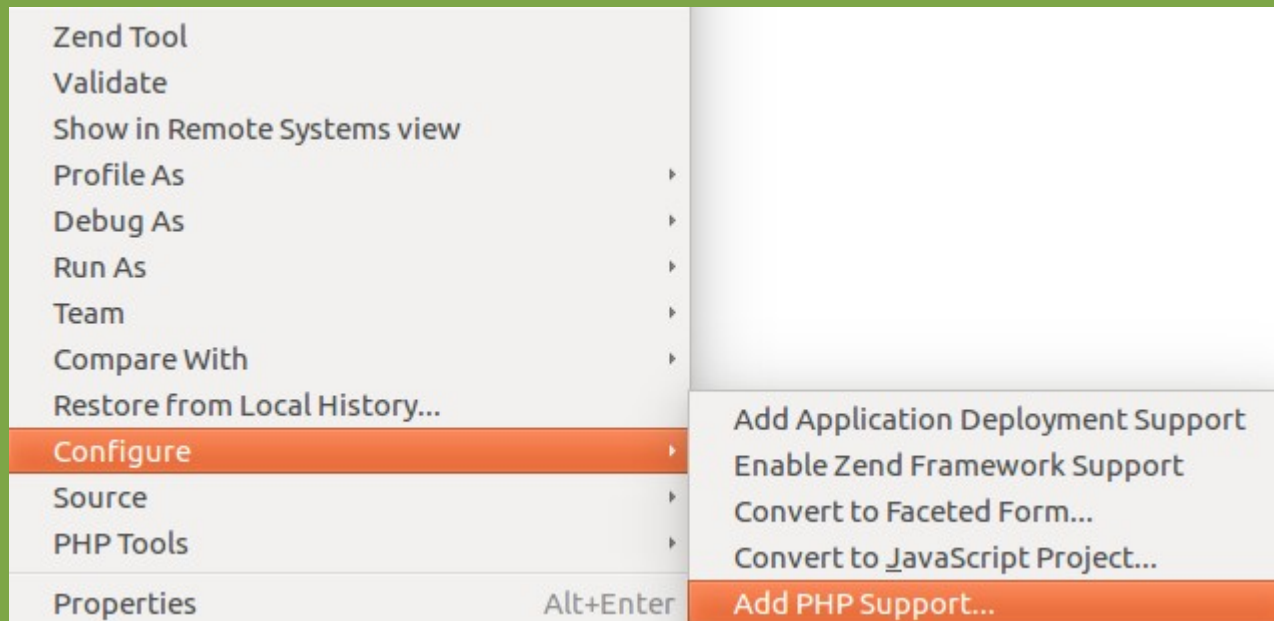
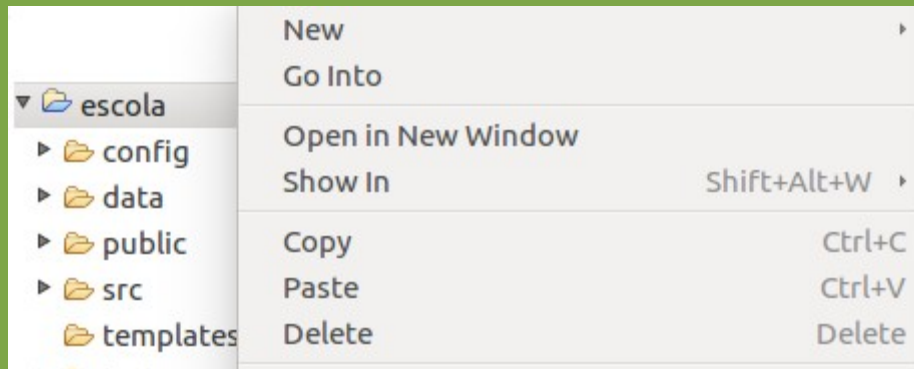
Working sets

☐ Add project to working sets

Working sets:

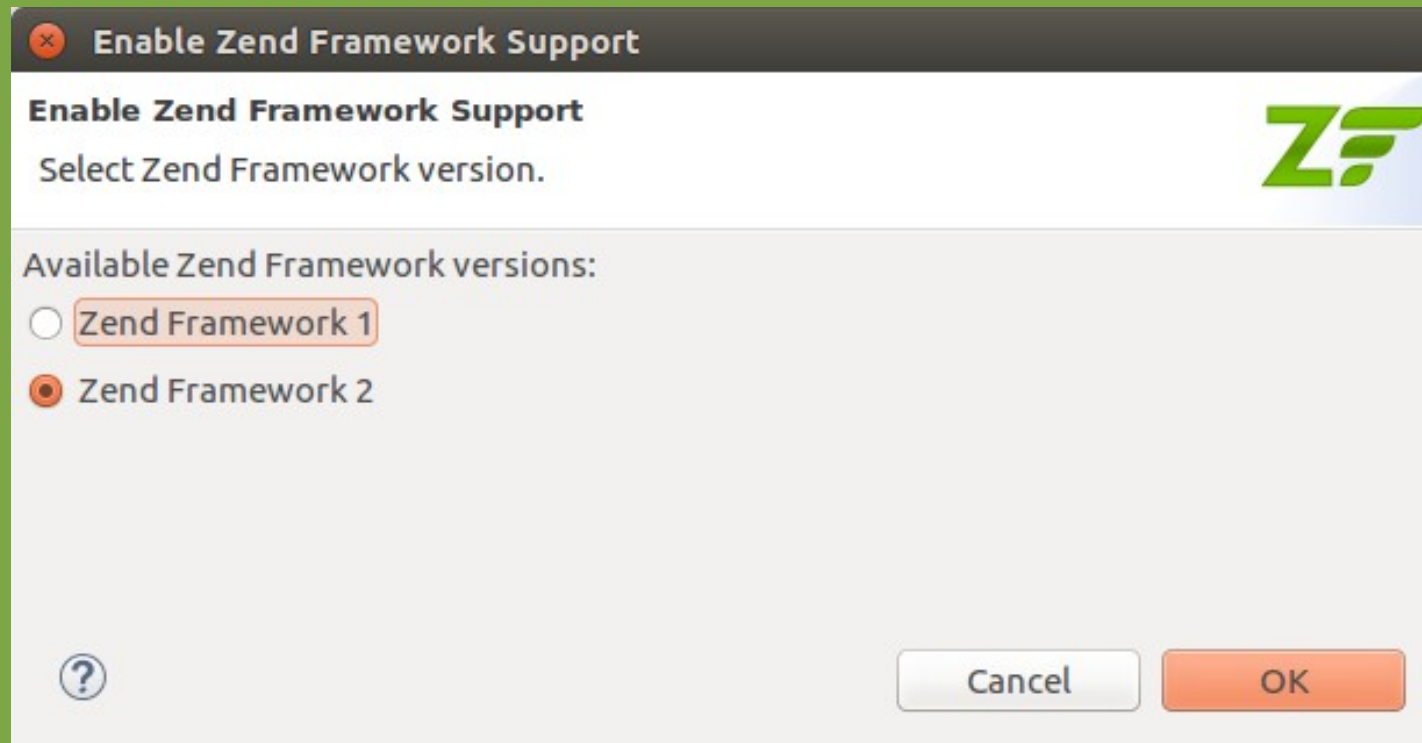
Abertura do projeto no Zend Eclipse PDT

Configure o suporte PHP ao projeto:

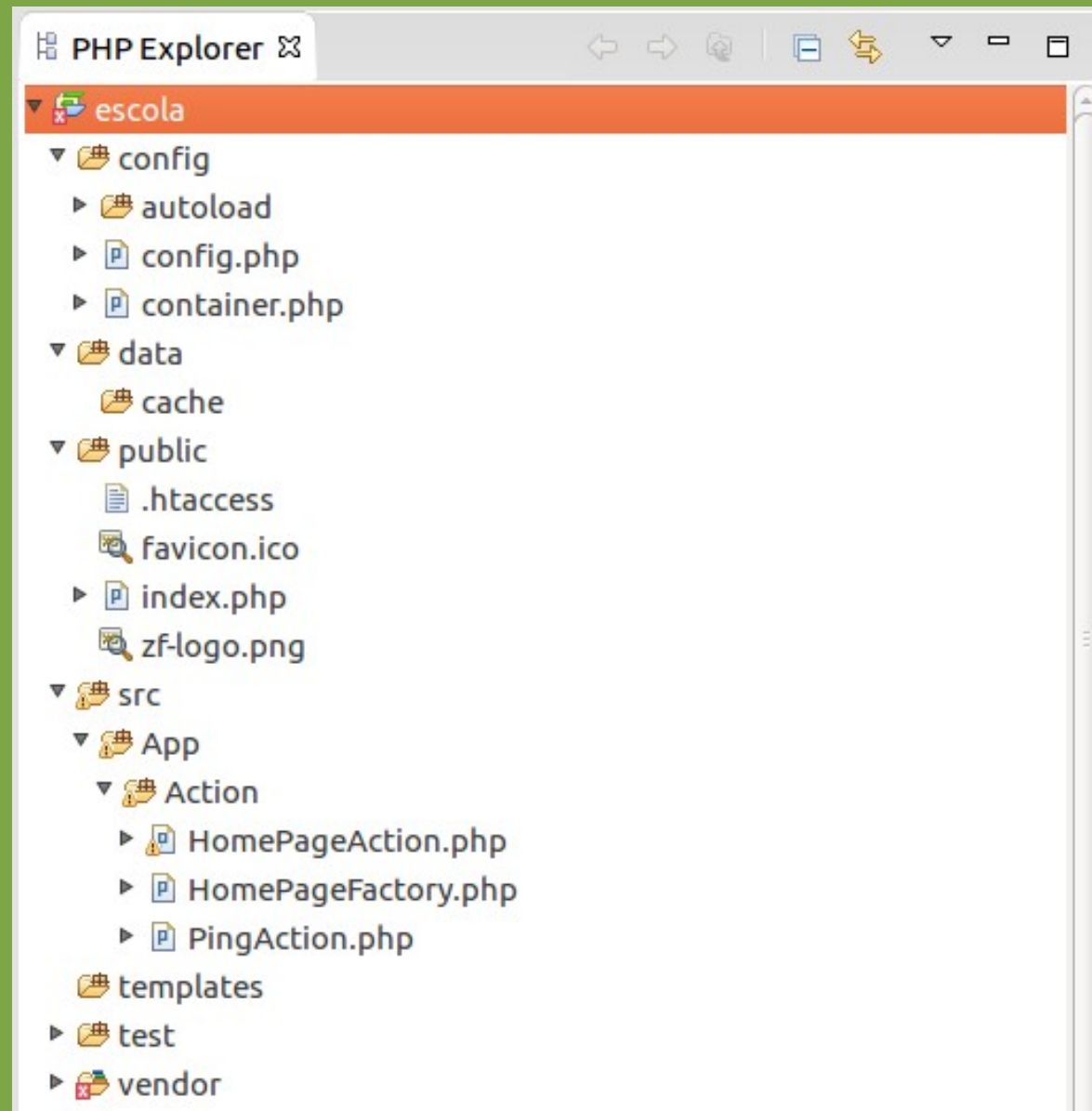


Abertura do projeto no Zend Eclipse P D T

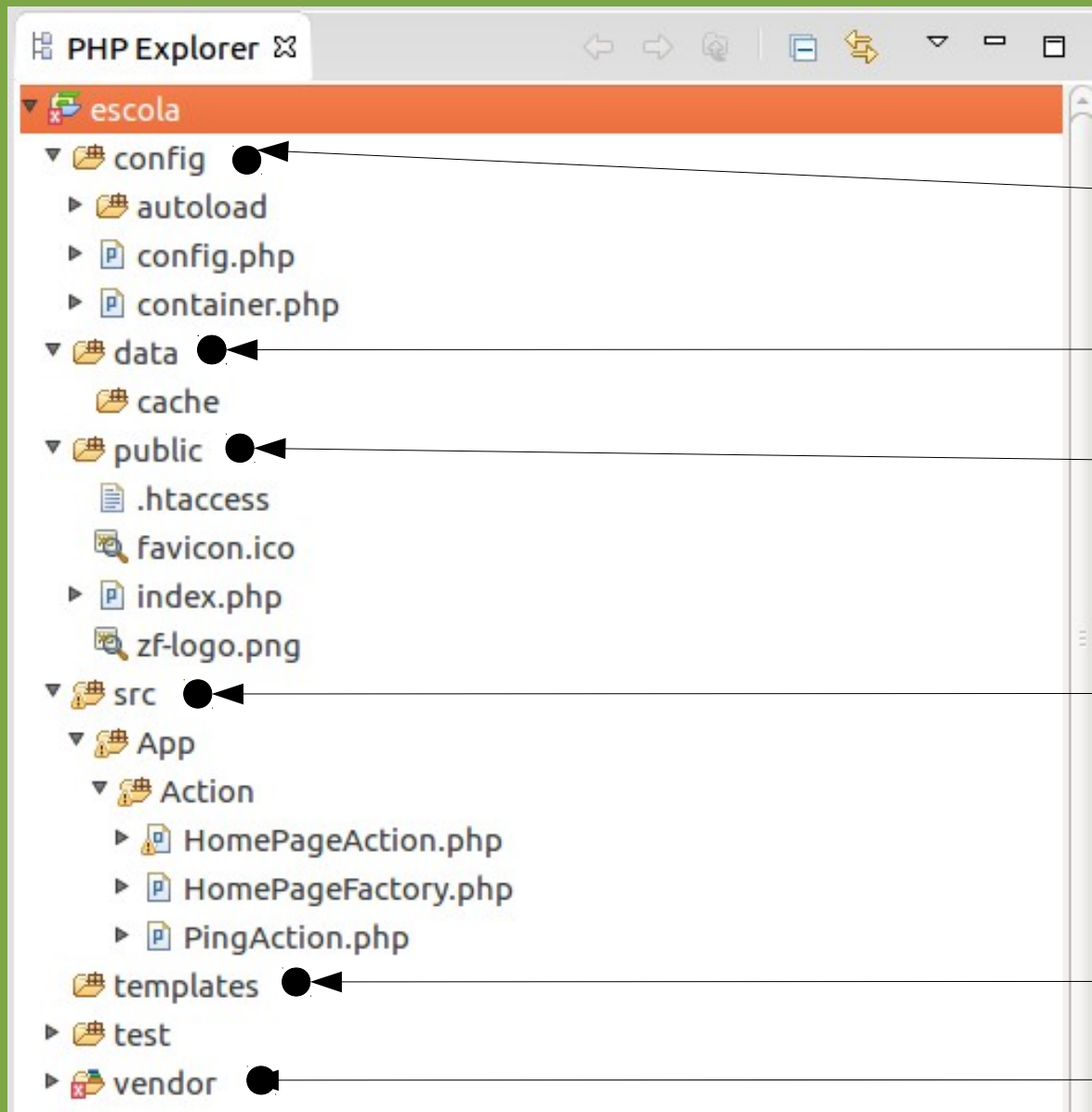
Configure o suporte Zend Framework ao projeto:



Abertura do projeto no Zend Eclipse PDT



Estrutura do projeto Zend Expressive



Configuração

Dados internos

Ponto único de acesso

Classes

Interface com o usuário

Dependências

Rodando a aplicação

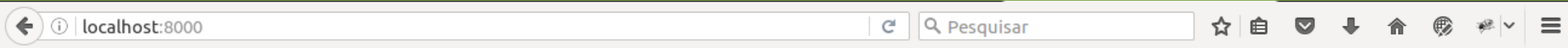
O esqueleto de projeto do Zend Expressive foi criado com a premissa de que o diretório web root seja a pasta public.

A menos que o servidor esteja dedicado para a aplicação, é necessário configurar um virtual host, que direcione as requisições HTTP para a pasta public.

Rodando a aplicação

Alternativa mais simples, para desenvolvimento:

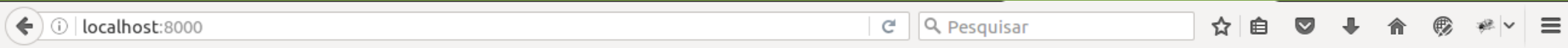
```
php -S localhost:8000 -t escola/public/
```



```
{"welcome": "Congratulations! You have installed the  
zend-expressive skeleton application.", "docsUrl": "zend-  
expressive.readthedocs.org"}
```

Rodando a aplicação

A resposta veio no formato JSON porque não definimos um *template engine*.



```
{"welcome": "Congratulations! You have installed the  
zend-expressive skeleton application.", "docsUrl": "zend-  
expressive.readthedocs.org"}
```


Criação do banco de dados e da tabela para o exercício



Bem-vindo ao phpMyAdmin

Linguagem - *Language*

Português - Brazilian portuguese ▼

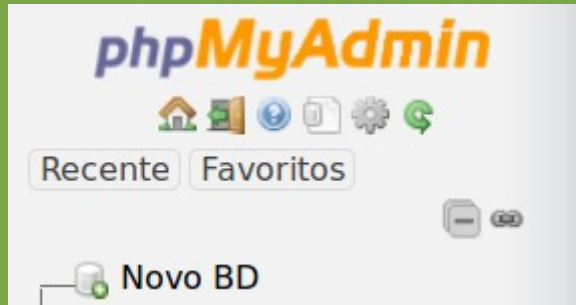
Entrar ⓘ

Usuário:

Senha:

Executar

Criação do banco de dados e da tabela para o exercício



Criação do banco de dados e da tabela para o exercício

Servidor: localhost » Banco de dados: escola

Estrutura SQL Procurar Consulta Exportar Importar Operações

⚠ Nenhuma tabela encontrada no banco de dados.

Criar tabela

Nome: Número de colunas:

Criação do banco de dados e da tabela para o exercício

Servidor: localhost » Banco de dados: escola » Tabela: alunos

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Privilégios Operações Gatilhos

Nome da Tabela: alunos Add 1 column(s) Executar

Estrutura

Nome	Tipo	Tamanho/Valores	Padrão	Colaço	Atributos	Nulo	Índice	A.I.
matricula	INT		Padrão: none			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
nome	VARCHAR	80	Padrão: none			<input type="checkbox"/>	—	<input type="checkbox"/>

PRIMARY

Comentários de tabela:

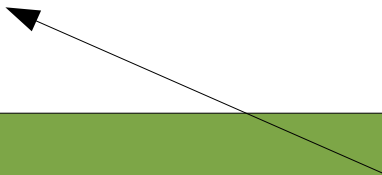
Colaço:

Mecanismo de armazenamento: MyISAM

Acesso a banco de dados com injeção de dependências

Adicione a dependência ao **Zend\ Db** no composer.json:

```
"require": {  
    "php": "^5.5 || ^7.0",  
    "roave/security-advisories": "dev-master",  
    "zendframework/zend-expressive": "^1.0",  
    "zendframework/zend-expressive-helpers": "^2.0",  
    "zendframework/zend-stdlib": "^2.7 || ^3.0",  
    "zendframework/zend-expressive-fastroute": "^1.0",  
    "zendframework/zend-servicemanager": "^2.7.3 || ^3.0",  
    "zendframework/zend-db": "^2.8.2"  
},
```



Acesso a banco de dados com injeção de dependências

Execute o composer update:

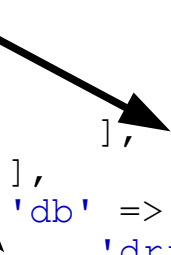
```
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing zendframework/zend-db (2.8.2)
  Downloading: 100%

zendframework/zend-db suggests installing zendframework/zend-eventmanager (Zend\EventManager component)
zendframework/zend-db suggests installing zendframework/zend-hydrator (Zend\Hydrator component for using HydratingResultSets)
Writing lock file
Generating autoload files
```

Acesso a banco de dados com injeção de dependências

Edite o arquivo dependencies.global.php:

```
'dependencies' => [
    // Use 'invokables' for constructor-less services, or services that do
    // not require arguments to the constructor. Map a service name to the
    // class name.
    'invokables' => [
        // Fully\Qualified\InterfaceName::class => Fully\Qualified\ClassName::class,
        Helper\ServerUrlHelper::class => Helper\ServerUrlHelper::class,
    ],
    // Use 'factories' for services provided by callbacks/factory classes.
    'factories' => [
        Application::class => ApplicationFactory::class,
        Helper\UrlHelper::class => Helper\UrlHelperFactory::class,
        'DbAdapter' => Zend\Db\Adapter\AdapterServiceFactory::class,
    ],
],
'db' => [
    'driver' => 'Pdo',
    'driver_options' => [
        PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES \'UTF8\''
    ],
    'dsn' => 'mysql:host=localhost;dbname=escola',
    'username' => 'root',
    'password' => 'mysql'
],
```



Acesso a banco de dados com injeção de dependências

Edite a classe `HomePageAction`:

Crie o atributo privado `dbAdapter`:

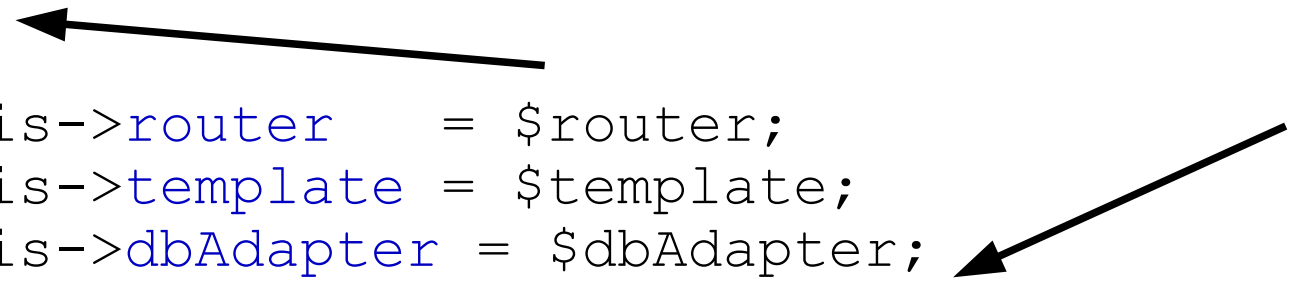
```
/**
 * @var AdapterInterface
 */
private $dbAdapter;
```

Acesso a banco de dados com injeção de dependências

Edite a classe `HomePageAction`:

Altere o construtor:

```
public function  
__construct(Router\RouterInterface $router,  
Template\TemplateRendererInterface $template = null,  
$dbAdapter)  
{  
    $this->router    = $router;  
    $this->template  = $template;  
    $this->dbAdapter = $dbAdapter;  
}
```




Acesso a banco de dados com injeção de dependências

Edite a classe HomePageFactory:

Altere o método `__invoke`:

```
public function __invoke(ContainerInterface $container)
{
    $router    = $container->get(RouterInterface::class);
    $template = ($container->has(TemplateRendererInterface::class))
        ? $container->get(TemplateRendererInterface::class)
        : null;

    $dbAdapter = $container->get('DbAdapter');
    return new HomePageAction($router, $template, $dbAdapter);
}
```



Acesso a banco de dados com injeção de dependências

Recarregue a página. Se algo estiver errado, o Whoops renderizará uma tela indicando a causa:

The screenshot displays a Whoops error page. On the left, a stack trace shows the error path: `Zend\ServiceManager\Exception\ServiceNotFoundException` at line 670 of `src/ServiceManager.php`, triggered by `getFactory` at line 746, `doCreate` at line 195, and `get` at line 18 of `src/App/Action/HomePageFactory.php`. The main error message reads: "Unable to resolve service 'dbAdapter' to a factory; are you certain you provided it during configuration?". On the right, a code editor shows the PHP code for `ServiceManager.php`, specifically the `createDelegateFromName` method. The code attempts to resolve the service 'dbAdapter' by iterating through abstract factories. Since no factory can create it, a `ServiceNotFoundException` is thrown with the message: "Unable to resolve service '%s' to a factory; are you certain you provided it during configuration?". Below the code editor, the environment details are listed: HTTP Method is GET, and the URL is `http://localhost:8000/`.

Zend\ServiceManager\Exception\ServiceNotFoundException

Unable to resolve service "dbAdapter" to a factory; are you certain you provided it during configuration?

Stack frames (32)

- 31 Zend\ServiceManager\Exception\ServiceNotFoundException
.../vendor/zendframework/zend-servicemanager/src/ServiceManager.php:670
- 30 Zend\ServiceManager\ServiceManager getFactory
.../vendor/zendframework/zend-servicemanager/src/ServiceManager.php:746
- 29 Zend\ServiceManager\ServiceManager doCreate
.../vendor/zendframework/zend-servicemanager/src/ServiceManager.php:195
- 28 Zend\ServiceManager\ServiceManager get
.../src/App/Action/HomePageFactory.php:18
- 27 App\Action\HomePageFactory __invoke
.../vendor/zendframework/zend-servicemanager/src/ServiceManager.php:195

```
660.         return $factory;
661.     }
662.
663.     // Check abstract factories
664.     foreach ($this->abstractFactories as $abstractFactory) {
665.         if ($abstractFactory->canCreate($this->creationContext, $name)) {
666.             return $abstractFactory;
667.         }
668.     }
669.
670.     throw new ServiceNotFoundException(sprintf(
671.         'Unable to resolve service "%s" to a factory; are you certain you provided
it during configuration?',
672.         $name
673.     ));
674. }
675.
676. /**
677.  * @param string $name
678.  * @param null|array $options
679.  * @return object
680.  */
681. private function createDelegateFromName($name, array $options = null)
682. {
```

No comments for this stack frame.

Environment & details:

Expressive Application Request

HTTP Method	GET
URL	http://localhost:8000/

Roteamento para URLs amigáveis

Nossa aplicação terá as seguintes rotas:

- / **alunos** - mostrará a lista de alunos
- / **aluno/ edit[/ matricula]** – abrirá um formulário para inclusão ou alteração
- / **aluno/ save** – gravará um registro na tabela
- / **aluno/ delete[/ matricula]** – removerá um registro

Roteamento para URLs amigáveis

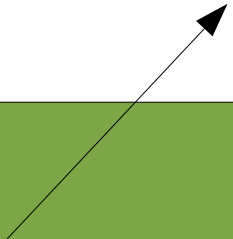
Edite o arquivo routes.global.php:

```
[
    'name' => 'list',
    'path' => '/alunos',
    'middleware' => App\Action\AlunoAction::class,
    'allowed_methods' => ['GET'],
],
[
    'name' => 'edit',
    'path' => '/aluno/edit[{matricula: \d+}]',
    'middleware' => App\Action\AlunoAction::class,
    'allowed_methods' => ['GET'],
],
[
    'name' => 'save',
    'path' => '/aluno/save',
    'middleware' => App\Action\AlunoAction::class,
    'allowed_methods' => ['POST'],
],
[
    'name' => 'delete',
    'path' => '/aluno/delete[{matricula: \d+}]',
    'middleware' => App\Action\AlunoAction::class,
    'allowed_methods' => ['GET'],
],
```

Interface com usuário com padrões MVC + Two Step View

Adicione a dependência ao **Plates** no composer.json:

```
"require": {  
    "php": "^5.5 || ^7.0",  
    "roave/security-advisories": "dev-master",  
    "zendframework/zend-expressive": "^1.0",  
    "zendframework/zend-expressive-helpers": "^2.0",  
    "zendframework/zend-stdlib": "^2.7 || ^3.0",  
    "zendframework/zend-expressive-fastroute": "^1.0",  
    "zendframework/zend-servicemanager": "^2.7.3 || ^3.0",  
    "zendframework/zend-db": "^2.8.2",  
    "zendframework/zend-expressive-platesrenderer": "^1.1.0"  
},
```



Interface com usuário com padrões MVC + Two Step View

Execute o composer update:

```
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing league/plates (3.1.1)
  Downloading: 100%


- Installing zendframework/zend-expressive-platesrenderer (1.1.0)
  Downloading: 100%

zendframework/zend-expressive-platesrenderer suggests installing mouf/pimple-interop (^1.0 to use Pimple for dependency injection)
zendframework/zend-expressive-platesrenderer suggests installing aura/di (3.0.*@beta to make use of Aura.Di dependency injection container)
Writing lock file
Generating autoload files
```

Interface com usuário com padrões MVC + Two Step View

Edite o arquivo dependencies.global.php

```
'factories' => [  
    Application::class => ApplicationFactory::class,  
    Helper\UrlHelper::class =>  
Helper\UrlHelperFactory::class,  
    'DbAdapter' =>  
Zend\Db\Adapter\AdapterServiceFactory::class,  
    'Zend\Expressive\Template\TemplateRendererInterface' =>  
    Zend\Expressive\Plates\PlatesRendererFactory::class  
],
```



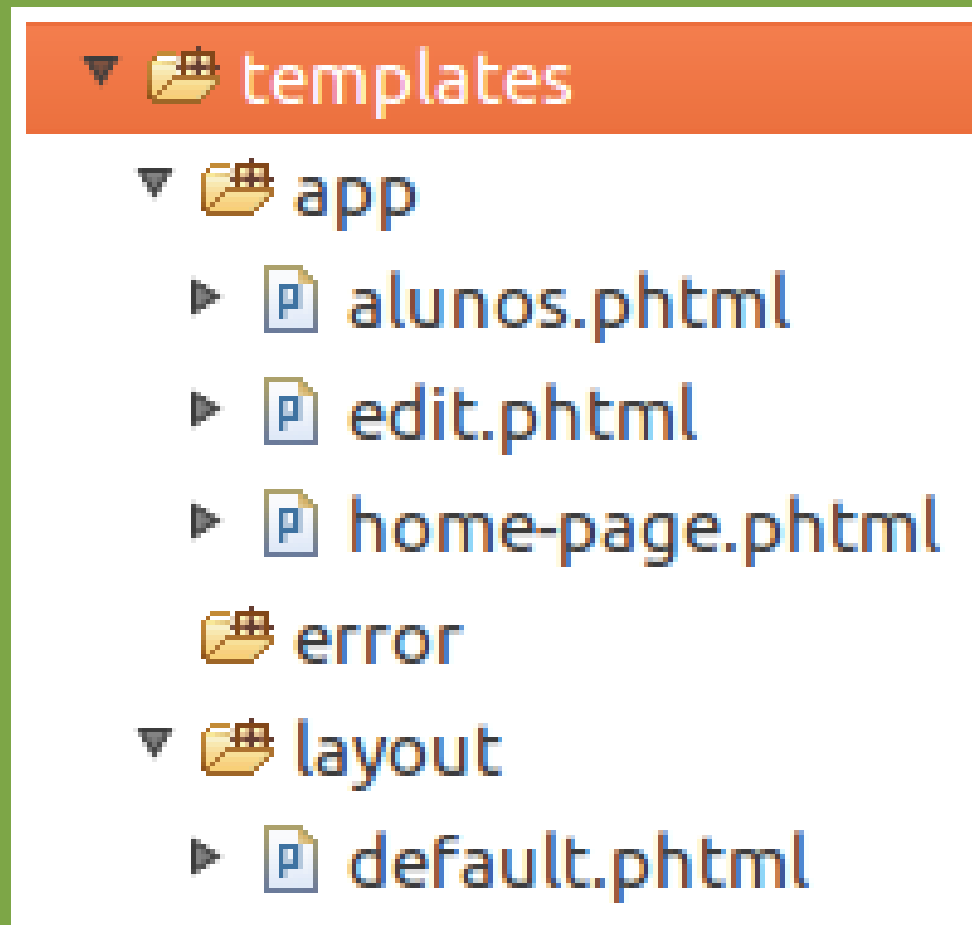
Interface com usuário com padrões MVC + Two Step View

Edite o arquivo dependencies.global.php

```
'templates' => [  
    'extension' => 'phtml',  
    'layout' => 'layout/default',  
    'map' => [  
        'layout/default' =>  
'templates/layout/default.phtml',  
        'error/error' =>  
'templates/error/error.phtml',  
        'error/404' => 'templates/error/404.phtml',  
    ],  
    'paths' => [  
        'app' => ['templates/app'],  
        'layout' => ['templates/layout'],  
        'error' => ['templates/error'],  
    ],  
]
```

Interface com usuário com padrões MVC + Two Step View

Crie a estrutura de pastas e os arquivos de *templating*:



Interface com usuário com padrões MVC + Two Step View

Edite o arquivo alunos.phtml:

```
<h1>Cadastro de Alunos</h1>
<a href="<?=$this->e($editLink) ?
>">Incluir aluno</a>
```

Interface com usuário com padrões MVC + Two Step View

Edite o arquivo edit.phtml:

```
<h1>Cadastro de Aluno</h1>
<form>
<label for="nome">Nome</label>
<input type="text" name="nome" value=""
autofocus="autofocus">
<input type="hidden" name="matricula">
<input type="submit" value="gravar">
</form>
<a href="<?=$this->e($returnLink) ?
">Voltar</a>
```

O middleware

Crie a classe AlunoAction:

```
namespace App\Action;

use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Zend\Diactoros\Response\HtmlResponse;
use Zend\Expressive\Router;
use Zend\Expressive\Template;
use Zend\Expressive\Plates\PlatesRenderer;
use Zend\Db\Adapter\AdapterInterface;
```


O middleware

Crie a classe AlunoAction:

```
class AlunoAction
{
    private $router;

    private $template;

    /**
     * @var AdapterInterface
     */
    private $dbAdapter;

    public function __construct(Router\RouterInterface
    $router, Template\TemplateRendererInterface $template =
    null, $dbAdapter)
    {
        $this->router    = $router;
        $this->template = $template;
        $this->dbAdapter = $dbAdapter;
    }
}
```

○ middleware

Crie a classe AlunoAction:

```
public function __invoke(ServerRequestInterface $request, ResponseInterface $response,
callable $next = null)
{
    $data = [];

    $path = $request->getUri()->getPath();
    switch ($path) {
        case '/alunos':
            $page = 'alunos';
            $data['editLink'] = $this->router->generateUri('edit');
            break;
        case '/aluno/edit':
            $page = 'edit';
            $data['returnLink'] = $this->router->generateUri('list');
            break;
        default:
            $page = 'home-page';
    }

    return new HtmlResponse($this->template->render('app::' . $page, $data));
}
```

Navegação

Acesse as URLs:

- <http://localhost:8000/alunos>
- <http://localhost:8000/aluno/edit>



O modelo

Crie a classe Aluno:

```
namespace App\Model;

use Zend\Db\Adapter\AdapterInterface;
use Zend\Db\Adapter\Adapter;
use Zend\Db\Sql\Sql;
class Aluno
{
    private $matricula;
    private $nome;

    /**
     * @var ContainerInterface
     */
    private static $dbAdapter;

    /**
     * @param AdapterInterface $dbAdapter
     */
    public static function setDbAdapter(AdapterInterface
$dbAdapter)
    {
        self::$dbAdapter = $dbAdapter;
    }
}
```

O modelo

Crie a classe Aluno:

```
/**
 * @param array $array
 */
public function __construct(array $array)
{
    $this->matricula = $array['matricula'];
    $this->nome = $array['nome'];
}

public static function get($matricula)
{
    $results = self::getAll(['matricula' => $matricula]);
    return $results->current();
}

public static function getAll($where = null)
{
    $sql = new Sql(self::$dbAdapter);
    $select = $sql->select('alunos');
    if ($where !== null) {
        $select->where($where);
    }
    $statement = $sql->prepareStatementForSqlObject($select);
    $results = $statement->execute();
    return $results;
}
```

O modelo

Crie a classe Aluno:

```
public function save()
{
    $sql = new Sql(self::$dbAdapter);
    if (empty($this->matricula)) {
        $sqlObject = $sql->insert('alunos');
        $sqlObject->columns(['nome'])
        ->values([$this->nome]);
    } else {
        $sqlObject = $sql->update('alunos');
        $sqlObject->set([
            'matricula' => $this->matricula,
            'nome' => $this->nome
        ]);
        $sqlObject->where(['matricula' => $this->matricula]);
    }
    $statement = $sql->prepareStatementForSqlObject($sqlObject);
    return $statement->execute();
}

public static function delete($matricula)
{
    $sql = new Sql(self::$dbAdapter);
    $sqlObject = $sql->delete('alunos');
    $sqlObject->where(['matricula' => $matricula]);
    $statement = $sql->prepareStatementForSqlObject($sqlObject);
    return $statement->execute();
}
```

O modelo

Edite a classe AlunoAction:

```
Aluno::setDbAdapter($this->dbAdapter);
$path = $request->getUri()->getPath();
switch ($path){
    case '/aluno/save':
        $matricula = isset($_POST['matricula']) ? $_POST['matricula'] : null;
        $nome = isset($_POST['nome']) ? $_POST['nome'] : null;
        $aluno = new Aluno([
            'matricula' => $matricula,
            'nome' => $nome
        ]);
        $aluno->save();
    case '/alunos':
        $page = 'alunos';
        $data['saveLink'] = $this->router->generateUri('save');
        $data['editLink'] = $this->router->generateUri('edit');
        $data['alunos'] = Aluno::getAll();
        break;
    case '/aluno/edit':
        $page = 'edit';
        $data['saveLink'] = $this->router->generateUri('save');
        $data['returnLink'] = $this->router->generateUri('list');
        break;
    default:
        $page = 'home-page';
}
```

O middleware

Crie a classe AlunoFactory:

```
namespace App\Action;

use Interop\Container\ContainerInterface;
use Zend\Expressive Router\RouterInterface;
use Zend\Expressive\Template\RendererInterface;

class AlunoFactory
{
    public function __invoke(ContainerInterface $container)
    {
        $router = $container->get(RouterInterface::class);
        $template = ($container->has(RendererInterface::class))
            ? $container->get(RendererInterface::class)
            : null;

        return new AlunoAction($router, $template);
    }
}
```


Manipulação de erros na aplicação

Tente incluir um aluno. Ocorrerá um erro, mas não será apresentado pela falta do *template* de erro.

LogicException

The template name "error::error" is not valid. The default directory has not been defined.



COPY

Stack frames (38)

37 LogicException

.../vendor/league/plates/src/Template/Name.php
:194

36 League\Plates\Template\Name getDefaultDirectory

.../vendor/league/plates/src/Template/Name.php
:168

35 League\Plates\Template\Name getPath

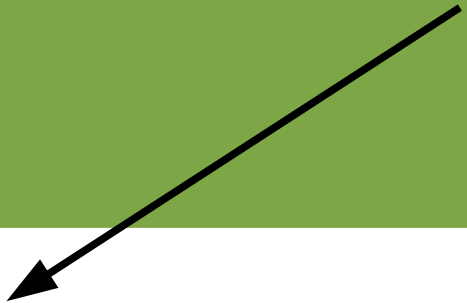
/var/www/html/escola/vendor/league/plates/src/Template/Name.php

```
179.     */
180.     public function doesPathExist()
181.     {
182.         return is_file($this->getPath());
183.     }
184.
185.     /**
186.      * Get the default templates directory.
187.      * @return string
188.      */
189.     protected function getDefaultDirectory()
190.     {
191.         $directory = $this->engine->getDirectory();
192.
193.         if (is_null($directory)) {
194.             throw new LogicException(
195.                 'The template name "' . $this->name . '" is not valid. '.
196.                 'The default directory has not been defined.'
197.             );
198.         }
199.
200.         return $directory;
201.     }
202. }
```

No comments for this stack frame.

Refatorando o formulário

Edite o arquivo edit.phtml:

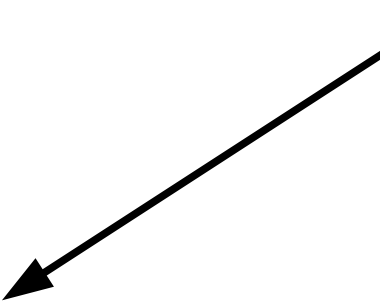


```
<h1>Cadastro de Aluno</h1>
<form action="<?=$this->e($saveLink) ?>"
method="post">
<label for="nome">Nome</label>
<input type="text" name="nome" value=""
autofocus="autofocus">
<input type="hidden" name="matricula">
<input type="submit" value="gravar">
</form>
<a href="<?=$this->e($returnLink) ?>">Voltar</a>
```

Refatorando a página inicial

Edite o arquivo alunos.phtml:

```
<h1>Cadastro de Alunos</h1>
<a href="<?=$this->e($editLink) ?>">Incluir aluno</a>
<table>
<thead>
<tr>
<th>Matrícula</th>
<th>Nome</th>
</tr>
</thead>
<tbody>
<?php
foreach ($alunos as $aluno):
?>
<tr>
<td><?=$aluno['matricula'] ?></td>
<td><?=$aluno['nome'] ?></td>
</tr>
<?php
endforeach;
?>
</tbody>
</table>
```



Agora, cadastre os alunos!



Espera aí...

E A URL DA GRAVAÇÃO DO
REGISTRO? SE A PÁGINA FOR
RECARREGADA, ELE VAI
INCLUIR O MESMO NOME
NOVAMENTE!



Redirecionando

Edite a classe AlunoAction:

```
        case '/aluno/save':  
            $matricula = isset($_POST['matricula']) ?  
$_POST['matricula'] : null;  
            $nome = isset($_POST['nome']) ? $_POST['nome']  
: null;  
  
            $aluno = new Aluno([  
                'matricula' => $matricula,  
                'nome' => $nome  
            ]);  
            $aluno->save();  
            return new RedirectResponse($this->router->  
>generateUri('list'));
```

Refatorando a listagem

Edite a classe AlunoAction:

```
        case '/alunos':  
            $page = 'alunos';  
            $data['editLink'] = $this->router->generateUri('edit');  
            $data['deleteLink'] = $this->router->generateUri('delete');  
            $data['alunos'] = Aluno::getAll();  
            break;
```

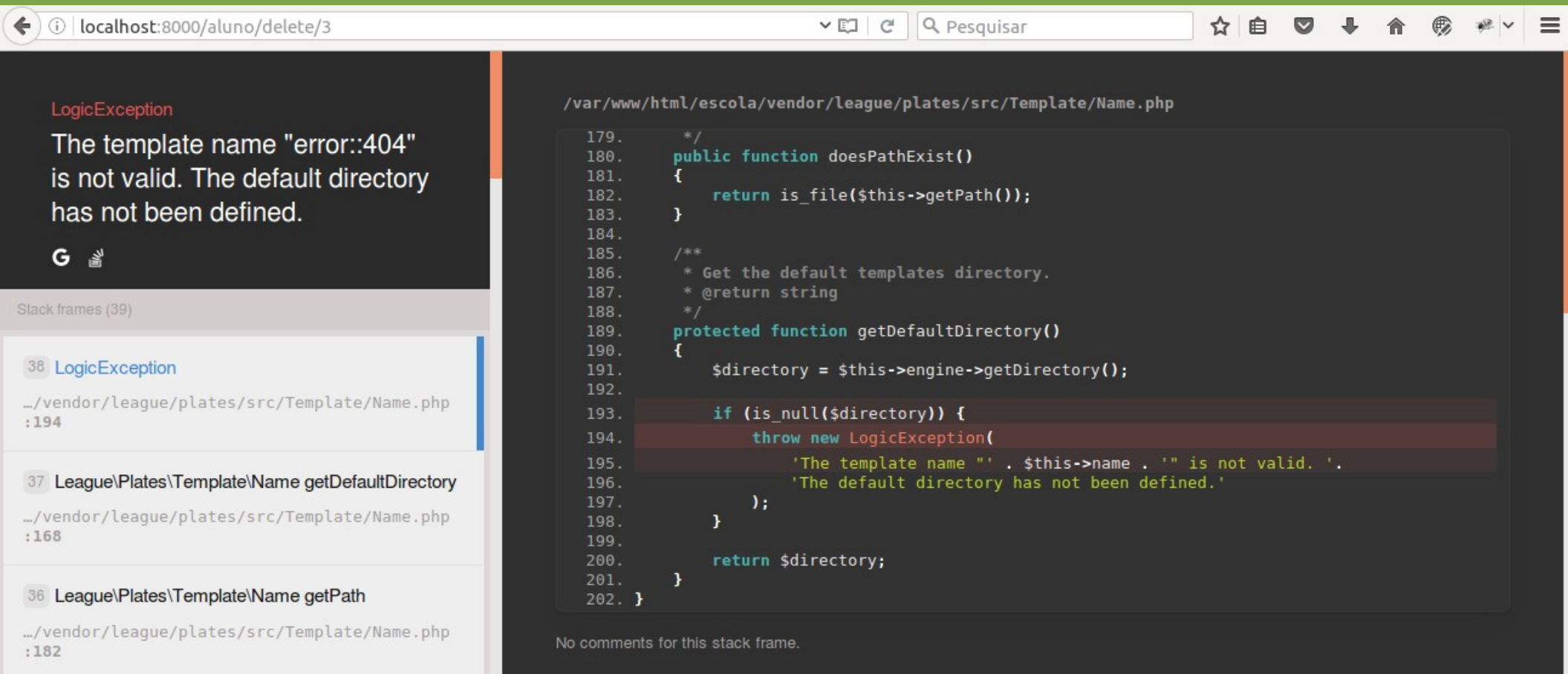
Refatorando a listagem

Edite o *template* alunos.phtml:

```
<?php
foreach ($alunos as $aluno):
?>
<tr>
<td><?=$aluno['matricula']?></td>
<td><?=$aluno['nome']?></td>
<td><a href="<?=$this->e($editLink) . '/' . $aluno['matricula']?>">E</td>
<td><a href="<?=$this->e($deleteLink) . '/' . $aluno['matricula']?>">X</td>
</tr>
<?php
endforeach;
?>
```


Excluindo registros

Não tente excluir ainda!



The image shows a web browser window with the address bar displaying `localhost:8000/aluno/delete/3`. The browser's developer tools are open, showing a `LogicException` error. The error message states: "The template name "error::404" is not valid. The default directory has not been defined." The stack trace shows the error occurred in `.../vendor/league/plates/src/Template/Name.php` at line 194.

The code editor on the right shows the source code of `/var/www/html/escola/vendor/league/plates/src/Template/Name.php`. The code includes a `doesPathExist()` method and a `getDefaultDirectory()` method. The `getDefaultDirectory()` method is highlighted, showing a `LogicException` being thrown when the default directory is not defined.

```
179.  */
180.  public function doesPathExist()
181.  {
182.      return is_file($this->getPath());
183.  }
184.
185.  /**
186.   * Get the default templates directory.
187.   * @return string
188.   */
189.  protected function getDefaultDirectory()
190.  {
191.      $directory = $this->engine->getDirectory();
192.
193.      if (is_null($directory)) {
194.          throw new LogicException(
195.              'The template name "' . $this->name . '" is not valid. '.
196.              'The default directory has not been defined.'
197.          );
198.      }
199.
200.      return $directory;
201.  }
202. }
```

Stack frames (39)

- 38 `LogicException`
`.../vendor/league/plates/src/Template/Name.php`
:194
- 37 `League\Plates\Template\Name` `getDefaultDirectory`
`.../vendor/league/plates/src/Template/Name.php`
:168
- 36 `League\Plates\Template\Name` `getPath`
`.../vendor/league/plates/src/Template/Name.php`
:182

No comments for this stack frame.

Manipulação de erros na aplicação

MAS ESSA TELA AÍ VAI
SEMPRE APARECER QUANDO
OCORRER UM ERRO?
MESMO EM PRODUÇÃO?



Manipulação de erros na aplicação

O arquivo `errorhandler.local.php` em produção deve ser diferente.

```
return [
    'dependencies' => [
        'invokables' => [
            'Zend\Expressive\Whoops' => Whoops\Run::class,
            'Zend\Expressive\WhoopsPageHandler' => Whoops\Handler\PrettyPageHandler::class,
        ],
        'factories' => [
            'Zend\Expressive\FinalHandler' =>
                Zend\Expressive\Container\WhoopsErrorHandlerFactory::class,
        ],
    ],

    'whoops' => [
        'json_exceptions' => [
            'display' => true,
            'show_trace' => true,
            'ajax_only' => true,
        ],
    ],
];
```

Excluindo registros

Edite a classe AlunoAction:

```
$path = $request->getUri()->getPath();  
$path = explode('/', $path);  
if (isset($path[3])) { // remove a  
matrícula  
    unset($path[3]);  
}  
$path = implode('/', $path);
```

Excluindo registros

DESTRUIR
SEMPRE É MAIS
FÁCIL!



Alterando registros

Edite a classe AlunoAction:

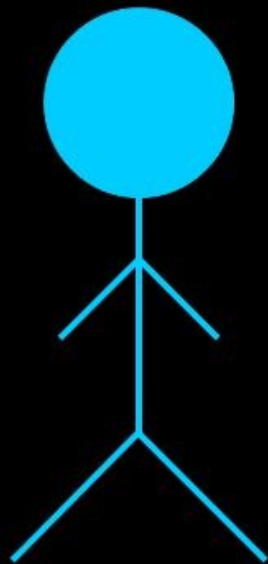
```
        case '/aluno/edit':  
            $page = 'edit';  
            $matricula = $request->  
>getAttribute('matricula', null);  
            if (is_null($matricula)) {  
                $aluno = new Aluno();  
            } else {  
                $aluno = Aluno::get($matricula);  
            }  
            $data['aluno'] = $aluno;  
            $data['saveLink'] = $this->router->  
>generateUri('save');  
            $data['returnLink'] = $this->router->  
>generateUri('list');  
            break;
```

Alterando registros

Edite a classe AlunoAction:

```
        case '/aluno/delete':  
            $matricula = $request-  
>getAttribute('matricula', null);  
            Aluno::delete($matricula);  
            return new RedirectResponse($this->router->  
>generateUri('list'));
```

Interface com usuário com padrões MVC + Two Step View



Mas peraí...

E O LAYOUT?

Interface com usuário com padrões MVC + Two Step View

Altere a classe AlunoAction:

```
$content = $this->template->render('app::' . $page, $data);  
$output = $this->template->render('layout::default');  
$output = str_replace('$content', $content, $output);  
return new HtmlResponse($output);
```

Interface com usuário com padrões MVC + Two Step View

Crie o arquivo de layout:

```
<!DOCTYPE html>
<html>
<head>
<title>Escola</title>
</head>
<body>
$content
</body>
</html>
```

E o MVC?

View

Templating Engine

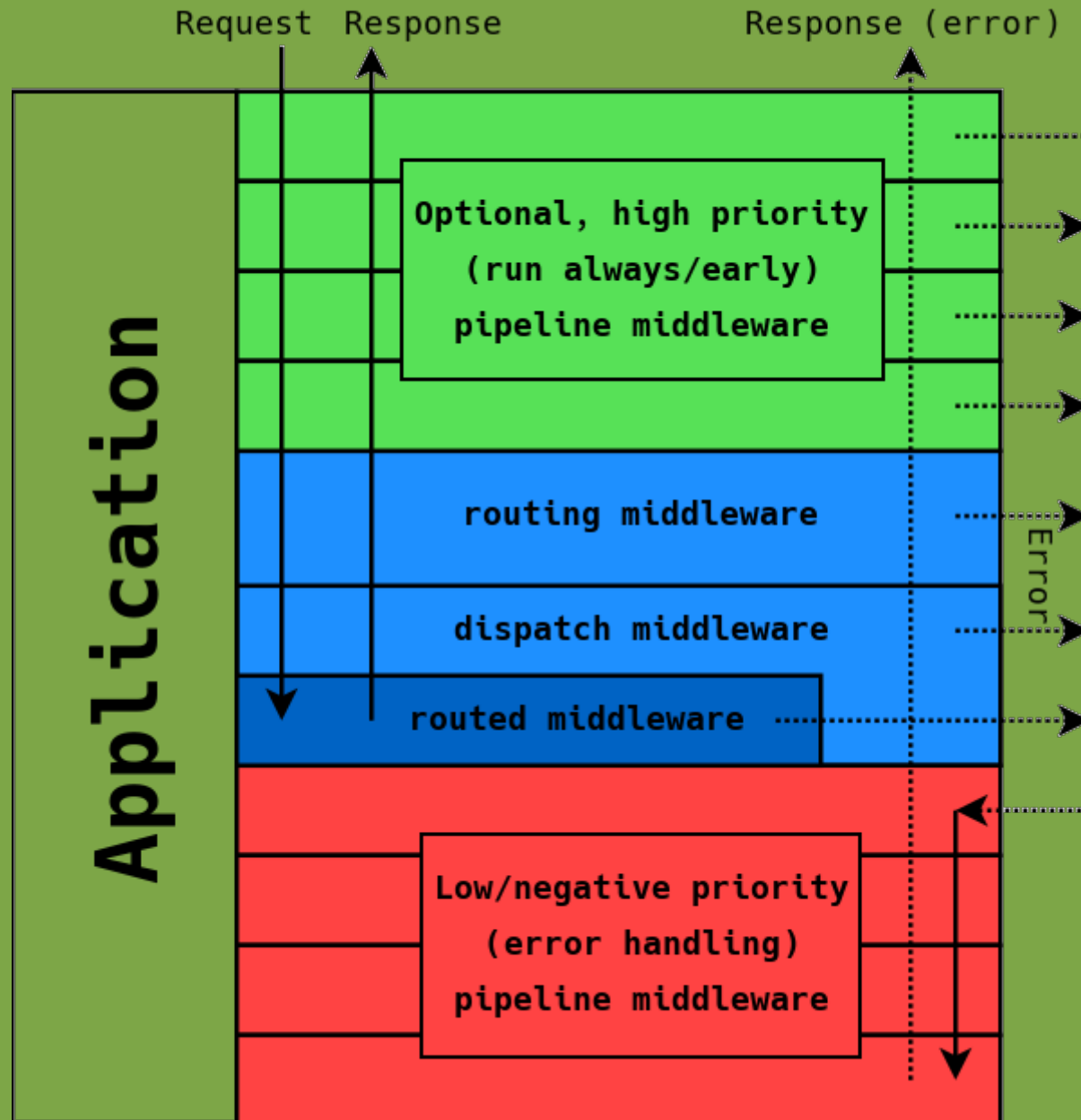
Controller

Middleware

Model

Só PHP (com PSRs, claro)

Diagrama de Fluxo do Zend Expressive



Obrigado!

www.fgsl.eti.br

www.phpconf.com.br

<http://romocavaleirodoespaco.blogspot.com.br/>

