

# **LARAVEL**

**AULA 01**

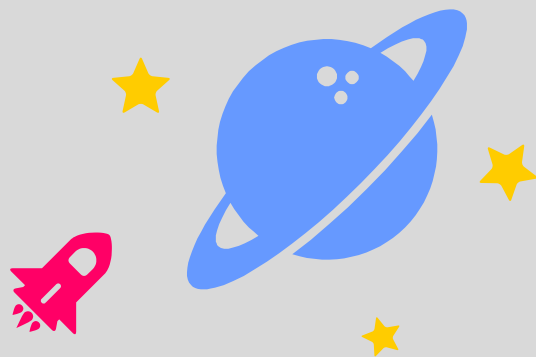


# FRAMEWORK

Os frameworks oferecem ambientes de desenvolvimento em que é possível resolver problemas comuns de forma mais rápida.

## Por que usar um framework?

- › Permite organizar o código e oferece um sistema de arquivos
- › Permite desenvolver mais rápido e escrever menos códigos
- › Facilita o trabalho em equipe e o uso de critérios comuns
- › Permite usar uma linguagem comum, independente do desenvolvedor



# BIBLIOTECA

As bibliotecas oferecem códigos para resolver problemas específicos sem precisar de muito contexto.

# COMPOSER

<https://getcomposer.org>

**COMPOSER** é um **gerenciador de dependências** para projetos **programados em PHP**. Isso quer dizer que ele permite instalar e manter atualizados os pacotes de software necessários para um projeto.

Acesse o **GUIA DE INSTALAÇÃO** do Composer:



- › [Instalação em LINUX](#)
- › [Instalação em MAC](#)
- › [Instalação em WINDOWS](#)

# LARAVEL

<https://laravel.com>

**LARAVEL é um `framework` de código aberto para desenvolver aplicativos e serviços web de forma muito mais rápida, fácil e segura.**

**Para poder começar a trabalhar, é necessário criar um projeto novo através do composer.**

## INSTALAÇÃO do Laravel:



- › [Instalação](#) em LINUX / MAC / WINDOWS



**HORA DE PRATICAR!**



Fazer o exercício 1

# FILE SYSTEM

## PHP

Para cada página de um site, costumamos criar um arquivo novo, ou seja, outro ponto de entrada.

## LARAVEL

Com Laravel, não é assim. Temos um só ponto de entrada para o aplicativo e diferentes Classes, que aceitam o pedido de acordo com o caminho.

# FILE SYSTEM

**Ou seja, temos que abandonar a ideia de que é possível identificar o arquivo que controla o site através do que aparece na URL e no file system.**

The background features abstract geometric shapes in the corners. On the left, there are overlapping shapes in shades of green, blue, orange, and purple. On the right, there are overlapping shapes in shades of green, blue, purple, and orange. The central text is white and bold.

# ESTRUTURA DE DIRETÓRIOS

Vamos conhecer a estrutura de  
diretórios de Laravel

Abstract geometric shapes in the top-left corner, including a light blue parallelogram, a green parallelogram, a red parallelogram, and a purple parallelogram, all overlapping and tilted at various angles.

# ROUTING

Vamos conhecer o sistema  
de roteamento de Laravel



# MÉTODOS HTTP

## MÉTODO

## FUNÇÃO

### GET

Serve para **consultar** elementos no banco de dados.

### POST

Serve para **criar** elementos no banco de dados.

### PUT / PATCH

Serve para **modificar** elementos no banco de dados.

### DELETE

Serve para **eliminar** elementos no banco de dados.

> routes/web.php

<?php

```
/*
|-----
| Caminhos SIMPLES
|-----
|
| Aqui, é possível registrar todos os caminhos de um aplicativo.
| É muito fácil. Basta dizer ao Laravel a qual URIs deve responder
| e dar a ele o controle para invocar quando essa URI é solicitada.
|
*/
```

```
Route::get('home', function () {

    return 'Hello World';

});
```

```
<?php
```

```
/*  
|-----  
| Caminhos com PARÂMETROS  
|-----  
|  
| Aqui, é possível registrar todos os caminhos de um aplicativo.  
| É muito fácil. Basta dizer ao Laravel a qual URIs deve responder  
| e dar a ele o controle para invocar quando essa URI é solicitada.  
|  
*/
```

```
Route::get('home/{name}', function ($name) {  
  
    return 'Hello ' . $name;  
  
});
```



```
<?php
```

```
/*
```

```
-----  
Caminhos com PARÂMETROS OPCIONAIS  
-----
```

```
Aqui, é possível registrar todos os caminhos de um aplicativo.  
É muito fácil. Basta dizer ao Laravel a qual URIs deve responder  
e dar a ele o controle para invocar quando essa URI é solicitada.
```

```
*/
```

```
Route::get('post/{post_id}/comments/{comment_id?}', function ($post_id, $comment_id =  
null) {
```

```
    if ($comment_id !== null) {  
        return $comment_id;  
    } else {  
        return 'Não há comentários';  
    }  
}
```

```
});
```

> routes/web.php

```
<?php
/*
|-----
| Caminhos para CONTROLADORES
|-----
|
| Aqui, é possível registrar todos os caminhos de um aplicativo.
| É muito fácil. Basta dizer ao Laravel a qual URIs deve responder
| e dar a ele o controle para invocar quando essa URI é solicitada.
|
*/

Route::get('/filmes', 'FilmesController@lista');           // Consulta (lista)

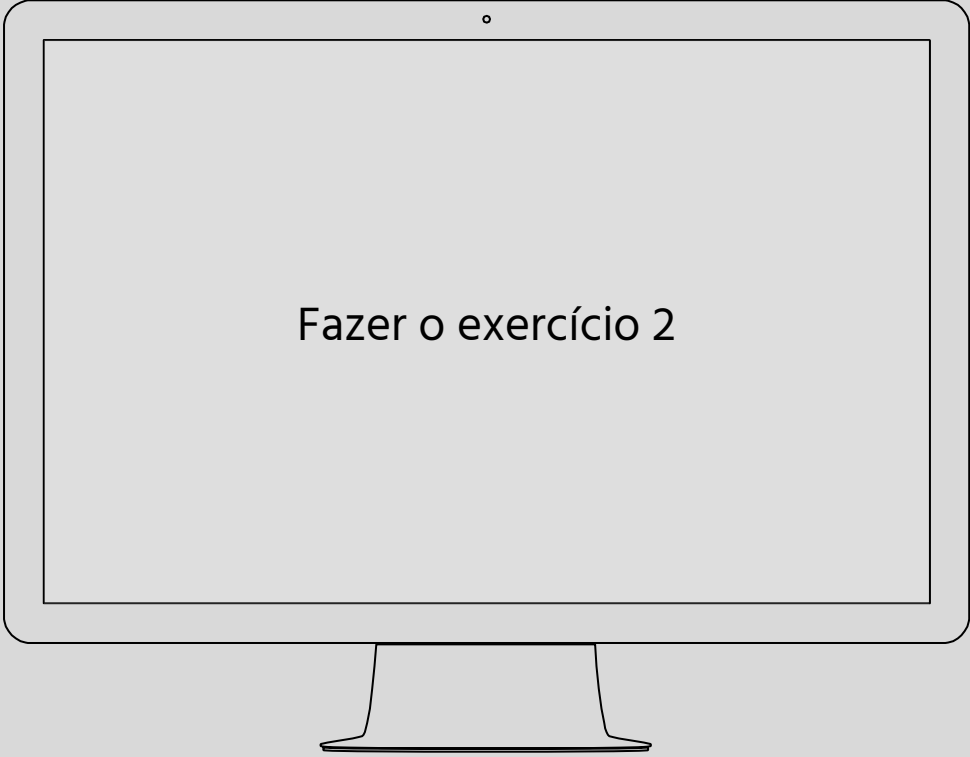
Route::get('/filmes/{id}', 'FilmesController@ver');        // Consulta (item)

Route::post('/atores', 'AtoresController@criar');          // Registro

Route::patch('/filmes/{id}', 'FilmesController@editar');  // Modificação

Route::delete('/atores/{id}', 'AtoresController@eliminar'); // Eliminação
```

**HORA DE PRATICAR!**



Fazer o exercício 2

The background features abstract, overlapping geometric shapes in various colors including light blue, green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, framing the central text.

**ATÉ A  
PRÓXIMA!**