

Curso de Laravel 8

Hora de Codar (Matheus Battisti)

<https://www.youtube.com/watch?v=qH7rsZBENJo&list=PLnDvRpP8BnewYKI1n2chQrrR4EYiJKbUG>

Resumo feito por Roberto Pinheiro

Aula 02 - Instalação

Instale inicialmente o XAMPP e posteriormente o Laravel. Se ocorrer erro na instalação do Laravel versão 8, no arquivo php.ini descomente a seguinte linha:

```
extension=fileinfo
```

Instalação global

```
composer global require laravel/installer
```

Remoção do Laravel instalado de forma global

```
composer global remove laravel/installer
```

Instalação local

```
composer create-project --prefer-dist laravel/laravel hdcevents
```

Executando o servidor

- Rode o servidor apache no Xampp
- Na pasta do projeto entre com o seguinte comando:

```
php artisan serve
```

```
\xampp\htdocs\hdcevents>php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[at Nov 20 11:45:08 2021] PHP 7.4.25 Development Server (http://127.0.0.1:8000) started
```

<http://127.0.0.1:8000/>

The screenshot shows a web browser window with the URL 127.0.0.1:8000. The page is the official Laravel website. At the top, there's a navigation bar with icons for back, forward, search, and other browser functions. Below the bar, the Laravel logo is displayed next to the word "Laravel". The main content area is divided into four sections:

- Documentation**: A section with a book icon, describing Laravel's documentation which covers every aspect of the framework.
- Laracasts**: A section with a video camera icon, describing Laracasts' thousands of video tutorials on Laravel, PHP, and JavaScript development.
- Laravel News**: A section with a news icon, describing Laravel News as a community-driven portal for news in the Laravel ecosystem.
- Vibrant Ecosystem**: A section with a globe icon, describing Laravel's robust library of tools and libraries like Forge, Vapor, Nova, and Envoyer.

At the bottom left, there are links for "Shop" and "Sponsor". At the bottom right, it says "Laravel v8.73.0 (PHP v7.4.25)".

Aula 03 - Rotas e Views

Rotas

`routes\web.php`

```
<?php

use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/


Route::get('/', function () {
    return view('welcome');
});

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    return view('products');
});
```

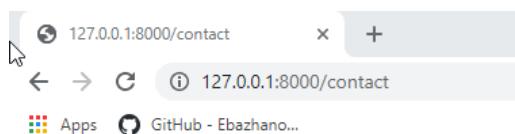
Views

`resources\views\contact.blade.php`

```
<h1>Esta é a página de contato</h1>
<a href="/">Voltar para home</a>
```

`resources\views\products.blade.php`

```
<h1>Tela de produtos</h1>
```



Esta é a página de contato

[Voltar para home](#)

Aula 04 - Conhecendo o Blade

- Blade é a template engine do Laravel;
- Com ele, vamos deixar as nossas views dinâmicas;
- Inserindo tags de HTML e também dados que são fornecidos pelo banco;
- Podemos dizer que as Views serão responsabilidade do Blade.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    $nome = "Matheus";
    $idade = 29;
    return view('welcome', ['nome' => $nome, 'idade' => $idade, 'profissao' =>
"Programador"]);
});

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    return view('products');
});
```

Diretivas do Blade

resources\views\welcome.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Laravel</title>
</head>

<body>
    <h1>Template Blade e diretivas</h1>
    <p>{{ $nome }}</p>
    @if($nome == "Pedro")
        <p>O nome é Pedro</p>
    @elseif($nome == "Matheus")
        <p>O nome é {{ $nome }}. Ele tem {{ $idade }} anos, e trabalha como {{ $profissao }}.</p>
    @else
        <p>O nome não é Pedro</p>
    @endif
</body>
</html>
```



Template Blade e diretivas

Matheus

O nome é Matheus. Ele tem 29 anos, e trabalha como Programador.

Aula 05 - Avançando na template engine Blade (loop)

- Podemos criar `estruturas de repetição` no Blade;
- Executar `PHP puro`;
- Escrever `comentários` nos arquivos de view;
- O Blade realmente é muito versátil e nos permite chegar num resultado excelente de renderização de views.

`routes\web.php`

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    $nome = "Matheus";
    $idade = 29;
    $arr = [10, 20, 30, 40, 50];
    $nomes = ["Matheus", "Maria", "João", "Saulo"];

    return view('welcome', [
        'nome' => $nome,
        'idade' => $idade,
        'profissao' => "Programador",
        'arr' => $arr,
        'nomes' => $nomes
    ]);
});

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    return view('products');
});
```

resources\views\welcome.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Laravel</title>
</head>

<body>

    <h1>Template Blade e diretivas</h1>

    <p>{{ $nome }}</p>

    @if($nome == "Pedro")
        <p>O nome é Pedro</p>
    @elseif($nome == "Matheus")
        <p>O nome é {{ $nome }}. Ele tem {{ $idade }} anos, e trabalha como {{ $profissao }}.</p>
    @else
        <p>O nome não é Pedro</p>
    @endif

    @for($i = 0; $i < count($arr); $i++)
        <p>{{$arr[$i]}}</p>
    @endfor

    @foreach($nomes as $nome)
        <p>{{$loop->index}} - {{ $nome }}</p>
    @endforeach

    @php
        $name = "João";
        echo $name;
    @endphp

    <!-- Comentário do HTML -->
    {{{-- Este é o comentário do Blade --}}
```

</body>

</html>

Apps GitHub - Ebazhano...

Template Blade e diretivas

Matheus

O nome é Matheus. Ele tem 29 anos, e trabalha como Programador.

10

20

30

40

50

0 - Matheus

1 - Maria

2 - João

3 - Saulo

João

Aula 06 - Adicionando arquivos estáticos

- Uma aplicação web normalmente tem arquivos de **CSS, JS e imagens**;
- O Laravel proporciona uma **maneira muito fácil de inserir estes arquivos no projeto**;
- Todos os recursos ficam na pasta **public**, e tem acesso direto nas tags que trabalham com arquivos estáticos.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    $nome = "Matheus";
    $idade = 29;
    $arr = [10, 20, 30, 40, 50];
    $nomes = ["Matheus", "Maria", "João", "Saulo"];

    return view('welcome', [
        'nome' => $nome,
        'idade' => $idade,
        'profissao' => "Programador",
        'arr' => $arr,
        'nomes' => $nomes
    ]);
});

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    return view('products');
});
```

```
resources\views\welcome.blade.php

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="/css/styles.css">
    <script src="/js/scripts.js"></script>
    <title>Laravel</title>
</head>

<body>

    <h1>Template Blade e diretivas</h1>

    <p>{{ $nome }}</p>

    @if($nome == "Pedro")
        <p>O nome é Pedro</p>
    @elseif($nome == "Matheus")
        <p>O nome é {{ $nome }}. Ele tem {{ $idade }} anos, e trabalha como
{{ $profissao }}.</p>
    @else
        <p>O nome não é Pedro</p>
    @endif

    @for($i = 0; $i < count($arr); $i++)
        <p>{{$arr[$i]}}</p>
    @endfor

    @foreach($nomes as $nome)
        <p>{{$loop->index}} - {{ $nome }}</p>
    @endforeach

    @php
        $name = "João";
        echo $name;
    @endphp

    <!-- Comentário do HTML -->
    {{-- Este é o comentário do Blade --}}
```

public\css\styles.css

```
h1 {
    color: red;
}
img {
    width: 500px;
}
```

```
public\js\scripts.js
```

```
console.log("Está funcionando!");
```

Template Blade e diretivas



Matheus

O nome é Matheus. Ele tem 29 anos, e trabalha como Programador.

10

20

30

40

50

0 - Matheus

1 - Maria

2 - João

3 - Saulo

João

Aula 07 - Criando layout com blade

- A funcionalidade de criar um layout permite o **reaproveitamento de código**;
- Por exemplo: Podemos utilizar o **mesmo header e footer** em todas as páginas sem repetir código;
- Mas o layout do blade não se limita a isso, podemos criar seções do site por meio do layout e também mudar o title da página.

`resources\views\layouts\main.blade.php`

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Fonte do Google -->
    <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

    <!-- CSS Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <!-- CSS da aplicação -->
    <link rel="stylesheet" href="/css/styles.css">

    <script src="/js/scripts.js"></script>
    <title>@yield('title')</title>
</head>

<body>

    @yield('content')

    <footer>
        <p>HDC Events &copy; 2021</p>
    </footer>

</body>

</html>
```

```
resources\views\welcome.blade.php
```

```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')

    <h1>Template Blade e diretivas</h1>

    <p>{{ $nome }}</p>

    @if($nome == "Pedro")
        <p>O nome é Pedro</p>
    @elseif($nome == "Matheus")
        <p>O nome é {{ $nome }}. Ele tem {{ $idade }} anos, e trabalha como
{{ $profissao }}.</p>
    @else
        <p>O nome não é Pedro</p>
    @endif

    @for($i = 0; $i < count($arr); $i++)
        <p>{{$arr[$i]}}</p>
    @endfor

    @foreach($nomes as $nome)
        <p>{{$loop->index}} - {{ $nome }}</p>
    @endforeach

    @php
        $name = "João";
        echo $name;
    @endphp

    <!-- Comentário do HTML -->
    {-- Este é o comentário do Blade --}

@endsection
```

```
resources\views\contact.blade.php
```

```
@extends('layouts.main')

@section('title', 'Contato')

@section('content')
    <h1>Esta é a página de contato</h1>
    <a href="/">Voltar para home</a>
@endsection
```

```
resources\views\products.blade.php
```

```
@extends('layouts.main')

@section('title', 'Produtos')

@section('content')
    <h1>Tela de produtos</h1>
@endsection
```

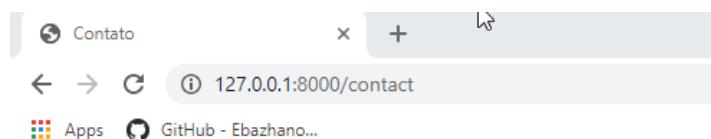
```
public\css\styles.css
```

```
* {
    font-family: 'Roboto';
}
```



Tela de produtos

HDC Events © 2021



Esta é a página de contato

[Voltar para home](#)

HDC Events © 2021

Template Blade e diretivas



Matheus

O nome é Matheus. Ele tem 29 anos, e trabalha como Programador.

10

20

30

40

50

0 - Matheus

1 - Maria

2 - João

3 - Saulo

João

HDC Events © 2021

Aula 08 - Resgatando parâmetros de URL

Parâmetros nas rotas

- Podemos mudar como uma view nos responde adicionando parâmetros a uma rota;
- Ao definir a rota devemos colocar o parâmetro desta maneira: `{id}`;
- Podemos ter parâmetros opcionais também, adicionando uma `?`;
- O Laravel aceita também `query parameters`, utilizando a seguinte sintaxe:
`?nome=Matheus&idade=29`

`routes\web.php`

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    $nome = "Matheus";
    $idade = 29;
    $arr = [10, 20, 30, 40, 50];
    $nomes = ["Matheus", "Maria", "João", "Saulo"];

    return view('welcome', [
        'nome' => $nome,
        'idade' => $idade,
        'profissao' => "Programador",
        'arr' => $arr,
        'nomes' => $nomes
    ]);
});

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    $busca = request('search');
    return view('products', ['busca' => $busca]);
});

Route::get('/produtos_teste/{id?}', function ($id = null) {
    return view('product', ['id' => $id]);
});
```

```
resources\views\product.blade.php
```

```
@extends('layouts.main')

@section('title', 'Produto')

@section('content')

@if($id != null)
    <p>Exibindo produto id: {{ $id }}</p>
@endif

@endsection
```

```
resources\views\products.blade.php
```

```
@extends('layouts.main')

@section('title', 'Produtos')

@section('content')

<h1>Tela de produtos</h1>

@if($busca != '')
    <p>O usuário está buscando por {{ $busca }}</p>
@endif

@endsection
```

```
public\css\styles.css
```

```
* {
    font-family: 'Roboto';
}
#navbar img{
    width: 50px;
}
```

```
resources\views\layouts\main.blade.php
```

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Fonte do Google -->
    <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

    <!-- CSS Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <!-- CSS da aplicação -->
    <link rel="stylesheet" href="/css/styles.css">

    <script src="/js/scripts.js"></script>
    <title>@yield('title')</title>
</head>

<body>

    <header>
        <nav class="navbar navbar-expand-lg navbar-light">
            <div id="navbar" class="collapse navbar-collapse">
                <a href="" class="navbar-brand">
                    
                </a>
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Criar Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Entrar</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Cadastrar</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>

    @yield('content')

    <footer>
        <p>HDC Events &copy; 2021</p>
    </footer>

</body>

</html>
```

```
resources\views\welcome.blade.php
```

```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')

<h1>HDC Events</h1>



<p>{{ $nome }}</p>

@if($nome == "Pedro")
    <p>O nome é Pedro</p>
@elseif($nome == "Matheus")
    <p>O nome é {{ $nome }}. Ele tem {{ $idade }} anos, e trabalha como
{{ $profissao }}.</p>
@else
    <p>O nome não é Pedro</p>
@endif

@for($i = 0; $i < count($arr); $i++)
    <p>{{$arr[$i]}}</p>
@endfor

@foreach($nomes as $nome)
    <p>{{$loop->index}} - {{ $nome }}</p>
@endforeach

@php
    $name = "João";
    echo $name;
@endphp

<!-- Comentário do HTML -->
{-- Este é o comentário do Blade --} }

@endsection
```



HDC Events



Matheus

O nome é Matheus. Ele tem 29 anos, e trabalha como Programador.

10

20

30

40

50

0 - Matheus

1 - Maria

2 - João

3 - Sául

João

HDC Events © 2021

Aula 09 - Criando Controllers

- Os **Controllers** são parte fundamental de toda aplicação em Laravel;
- Geralmente condensam a maior parte da lógica;
- Tem o papel de **enviar e esperar resposta do banco de dados**;
- E também **receber e enviar alguma resposta para as views**;
- Os Controller podem ser criados via **artisan**;
- É comum retornar uma view ou redirecionar para uma URL pelo Controller.

```
php artisan make:controller EventController
```

```
C:\xampp\htdocs\hdcevents>php artisan make:controller EventController
Controller created successfully.
```

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create']);

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/produtos', function () {
    $busca = request('search');
    return view('products', ['busca' => $busca]);
});

Route::get('/produtos_teste/{id?}', function ($id = null) {
    return view('product', ['id' => $id]);
});
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class EventController extends Controller
{
    public function index() {
        $nome = "Matheus";
        $idade = 29;
        $arr = [10, 20, 30, 40, 50];
        $nomes = ["Matheus", "Maria", "João", "Saulo"];

        return view('welcome', [
            'nome' => $nome,
            'idade' => $idade,
            'profissao' => "Programador",
            'arr' => $arr,
            'nomes' => $nomes
        ]);
    }

    public function create(){
        return view('events.create');
    }
}
```

resources\views\events\create.blade.php

```
@extends('layouts.main')

@section('title', 'Criar evento')

@section('content')

    <h1>Crie um evento</h1>

@endsection
```

```
resources\views\layouts\main.blade.php
```

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Fonte do Google -->
    <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

    <!-- CSS Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EqzqF4YVdCiI4qQz0EwqISJLWUk+DfXOvKZlXyWZlDyZjIW3" crossorigin="anonymous">

    <!-- CSS da aplicação -->
    <link rel="stylesheet" href="/css/styles.css">

    <script src="/js/scripts.js"></script>
    <title>@yield('title')</title>
</head>

<body>

    <header>
        <nav class="navbar navbar-expand-lg navbar-light">
            <div id="navbar" class="collapse navbar-collapse">
                <a href="#" class="navbar-brand">
                    
                </a>
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/events/create">Criar Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Entrar</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Cadastrar</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>

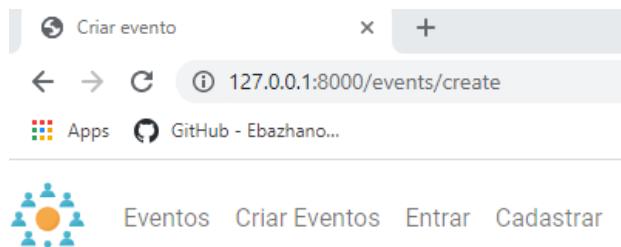
    @yield('content')

    <footer>
        <p>HDC Events &copy; 2021</p>
    </footer>

    <script src="https://unpkg.com/ionicons@4.5.10-0/dist/ionicons.js"></script>

</body>

</html>
```



Aula 10 - Conexão ao banco de dados MySQL

- A conexão do Laravel com o banco é configurada pelo arquivo `.env`;
- Isso nos proporciona maior liberdade e também segurança na aplicação;
- O Laravel utiliza um **ORM** (Object-Relational Mapping) chamada **Eloquent**;
- E também para a criação de tabelas as **migrations**.

- No PhpMyAdmin crie um banco de dados chamado `hdcevents`.

`.env`

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:MXMU5+qAuDOAJNnqxVVW/ABRWxrL++zmQiNBvLD+7/A=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=hdcevents
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DRIVER=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"
```

```
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
```

```
PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1
```

```
MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

php artisan migrate

```
Migrated: 2014_10_12_000000_create_users_table (761.48ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (527.53ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (692.98ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (859.83ms)
PS C:\xampp\htdocs\hdcevents> █
```

Tabela	Acções	Registos	Agrupamento (Collation)	Tamanho	Suspenso
		(	Tipo		
<input type="checkbox"/> failed_jobs	      	0	InnoDB	utf8mb4_unicode_ci	32.0 KB
<input type="checkbox"/> migrations	      	4	InnoDB	utf8mb4_unicode_ci	16.0 KB
<input type="checkbox"/> password_resets	      	0	InnoDB	utf8mb4_unicode_ci	32.0 KB
<input type="checkbox"/> personal_access_tokens	      	0	InnoDB	utf8mb4_unicode_ci	48.0 KB
<input type="checkbox"/> users	      	0	InnoDB	utf8mb4_unicode_ci	32.0 KB
5 tabelas	Soma			160.0 KB	0 Bytes

 Marcar todos ▾

Aula 11 - Introdução a migrations do Laravel

- As **migrations** funcionam como um versionamento de banco de dados;
- Podemos **avançar e retroceder** a qualquer momento;
- **Adicionar colunas e remover** de forma facilitada;
- Fazer o setup de DB de uma nova instalação em apenas um comando;
- Podemos verificar as migrations com **migrate:status**.

```
php artisan make:migration create_products_table
```

```
PS C:\xampp\htdocs\hdcevents> php artisan make:migration create_products_table
Created Migration: 2021_11_20_183946_create_products_table
PS C:\xampp\htdocs\hdcevents> 
```

```
php artisan migrate:status
```

```
PS C:\xampp\htdocs\hdcevents> php artisan migrate:status
+-----+-----+-----+
| Yes | 2014_10_12_000000_create_users_table | 1 |
| Yes | 2014_10_12_100000_create_password_resets_table | 1 |
| Yes | 2019_08_19_000000_create_failed_jobs_table | 1 |
| Yes | 2019_12_14_000001_create_personal_access_tokens_table | 1 |
| No  | 2021_11_20_183946_create_products_table | 1 |
+-----+-----+-----+
```

```
php artisan migrate
```

```
PS C:\xampp\htdocs\hdcevents> php artisan migrate
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (406.97ms)
PS C:\xampp\htdocs\hdcevents> 
```

```
php artisan migrate:status
```

```
PS C:\xampp\htdocs\hdcevents> php artisan migrate:status
| Ran? | Migration | Batch |
+-----+-----+-----+
| Yes | 2014_10_12_000000_create_users_table | 1 |
| Yes | 2014_10_12_100000_create_password_resets_table | 1 |
| Yes | 2019_08_19_000000_create_failed_jobs_table | 1 |
| Yes | 2019_12_14_000001_create_personal_access_tokens_table | 1 |
| Yes | 2021_11_20_183946_create_products_table | 2 |
+-----+-----+-----+
PS C:\xampp\htdocs\hdcevents> 
```

database\migrations\2021_11_20_183946_create_products_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('products');
    }
}
```

php artisan migrate:fresh

```
C:\xampp\htdocs\hdcevents>php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (747.98ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (602.11ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (913.52ms)
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (603.74ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina Mais
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	3 created_at	timestamp			Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	4 updated_at	timestamp			Sim	NULL			Muda Elimina Mais

Marcar todos Com os seleccionados: Procurar Muda Elimina Primária Único Índice Espacial
 Texto Completo Adicionar à(s) coluna(s) central(is) Remover da(s) coluna(s) central(is)

database\migrations\2021_11_20_183946_create_products_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('name', 100);
            $table->integer('qty');
            $table->text('description');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('products');
    }
}
```

php artisan migrate:fresh

```
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (631.73ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (391.75ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (690.95ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (1,139.92ms)
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (631.58ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
1	id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina Mais
2	name	varchar(100)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina Mais
3	qty	int(11)			Não	Nenhum			Muda Elimina Mais
4	description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina Mais
5	created_at	timestamp			Sim	NULL			Muda Elimina Mais
6	updated_at	timestamp			Sim	NULL			Muda Elimina Mais

Marcar todos Com os seleccionados:

Aula 12 - Avançando em migrations do Laravel

- Quando preceisamos adicionar um novo campo a uma tabela, **devemos criar uma nova migration**;
- Porém devemos **tomar cuidado** para não rodar o comando **fresh**, e apagar os dados já existentes;
- O comando **rollback** pode ser utilizado para voltar uma migration;
- Para voltar todas podemos utilizar o **reset**;
- Para voltar todas e rodar o migrate novamente utilizamos o **refresh**;

Adicionando uma nova coluna na tabela

```
php artisan make:migration add_category_to_products_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration add_category_to_products_table
Created Migration: 2021_11_20_222225_add_category_to_products_table
□
```

```
database\migrations\2021_11_20_222225_add_category_to_products_table.php
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddCategoryToProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('products', function (Blueprint $table) {
            $table->string('category', 100);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('products', function (Blueprint $table) {
            $table->dropColumn('category');
        });
    }
}
```

php artisan migrate

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2021_11_20_222225_add_category_to_products_table
Migrated: 2021_11_20_222225_add_category_to_products_table (466.61ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
<input type="checkbox"/>	2 name	varchar(100)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	3 qty	int(11)			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	4 description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	5 created_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	6 updated_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	7 category	varchar(100)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais

↑ Marcar todos Com os seleccionados: Muda Elimina Primária Índice Adicionar à(s) coluna(s) central(is) Remover da(s) coluna(s) central(is)

Rollback da última migration

php artisan migrate:rollback

```
C:\xampp\htdocs\hdcevents>php artisan migrate:rollback
Rolling back: 2021_11_20_222225_add_category_to_products_table
Rolled back: 2021_11_20_222225_add_category_to_products_table (796.89ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
<input type="checkbox"/>	2 name	varchar(100)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	3 qty	int(11)			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	4 description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	5 created_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	6 updated_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais

Rollback de todas as migrations

php artisan migrate:reset

```
C:\xampp\htdocs\hdcevents>php artisan migrate:reset
Rolling back: 2021_11_20_222225_add_category_to_products_table
Rolled back: 2021_11_20_222225_add_category_to_products_table (108.21ms)
Rolling back: 2021_11_20_183946_create_products_table
Rolled back: 2021_11_20_183946_create_products_table (133.36ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (239.88ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (357.00ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (224.27ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (188.56ms)
```

php artisan migrate:status

Ran?	Migration	Batch
No	2014_10_12_000000_create_users_table	
No	2014_10_12_100000_create_password_resets_table	
No	2019_08_19_000000_create_failed_jobs_table	
No	2019_12_14_000001_create_personal_access_tokens_table	
No	2021_11_20_183946_create_products_table	
No	2021_11_20_222225_add_category_to_products_table	

php artisan migrate

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (2,033.52ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (414.90ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (500.59ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (767.97ms)
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (780.08ms)
Migrating: 2021_11_20_222225_add_category_to_products_table
Migrated: 2021_11_20_222225_add_category_to_products_table (87.93ms)
```

php artisan migrate:status

Ran?	Migration	Batch
Yes	2014_10_12_000000_create_users_table	1
Yes	2014_10_12_100000_create_password_resets_table	1
Yes	2019_08_19_000000_create_failed_jobs_table	1
Yes	2019_12_14_000001_create_personal_access_tokens_table	1
Yes	2021_11_20_183946_create_products_table	1
Yes	2021_11_20_222225_add_category_to_products_table	1

R rollback de todas migrations e posterior migrate

```
php artisan migrate:refresh
```

```
C:\xampp\htdocs\hdcevents>php artisan migrate:refresh
Rolling back: 2021_11_20_222225_add_category_to_products_table
Rolled back: 2021_11_20_222225_add_category_to_products_table (179.80ms)
Rolling back: 2021_11_20_183946_create_products_table
Rolled back: 2021_11_20_183946_create_products_table (333.96ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (1,117.05ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (365.91ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (286.74ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (385.50ms)
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (1,762.58ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (1,646.25ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (1,320.56ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (1,489.13ms)
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (647.06ms)
Migrating: 2021_11_20_222225_add_category_to_products_table
Migrated: 2021_11_20_222225_add_category_to_products_table (397.41ms)
```

Apagando todas as tabelas e fazendo um posterior migrate

```
php artisan migrate:fresh
```

```
C:\xampp\htdocs\hdcevents>php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (762.58ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (405.78ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (707.15ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (811.47ms)
Migrating: 2021_11_20_183946_create_products_table
Migrated: 2021_11_20_183946_create_products_table (246.06ms)
Migrating: 2021_11_20_222225_add_category_to_products_table
Migrated: 2021_11_20_222225_add_category_to_products_table (122.58ms)
```

Aula 13 - Utilizando o Eloquent do Laravel

- Eloquent é a **ORM** do Laravel;
- Cada tabela tem um **Model** que é responsável pela interação entre as requisições ao banco;
- A convenção para o Model é o nome da entidade em singular, enquanto a tabela é a entidade no plural: Event e events;
- No Model faremos poucas alterações dos arquivos, geralmente **configurações específicas**.

- Inicialmente, apague as migrations de Products. Não iremos utilizá-las.

Criando a migration para Eventos

```
php artisan make:migration create_events_table

database\migrations\2021_11_20_232254_create_events_table.php

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateEventsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('events', function (Blueprint $table) {
            $table->id();
            $table->string("title");
            $table->text("description");
            $table->string("city");
            $table->boolean("private");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('events');
    }
}
```

```
php artisan migrate:status
```

Ran?	Migration	Batch
Yes	2014_10_12_000000_create_users_table	1
Yes	2014_10_12_100000_create_password_resets_table	1
Yes	2019_08_19_000000_create_failed_jobs_table	1
Yes	2019_12_14_000001_create_personal_access_tokens_table	1
No	2021_11_20_232254_create_events_table	

```
php artisan migrate
```

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2021_11_20_232254_create_events_table
Migrated: 2021_11_20_232254_create_events_table (678.98ms)
```

Criando o model para eventos

```
php artisan make:model Event
```

```
C:\xampp\htdocs\hdcevents>php artisan make:model Event
Model created successfully.
```

- Insira manualmente alguns dados na tabela events:

+ Opções										
		id	title	description	city	private	created_at	updated_at		
<input type="checkbox"/>	Edita Copiar Apagar	1	Evento teste	Descrição teste	Floripa	0	2021-11-20 20:47:42	2021-11-20 20:47:42		
<input type="checkbox"/>	Edita Copiar Apagar	2	Evento de Javascript	Aprender muito JS	São Paulo	0	2021-11-20 20:51:15	2021-11-20 20:51:15		
<input type="checkbox"/>	Edita Copiar Apagar	3	Evento de PHP	Aprender PHP	São José	0	2021-11-20 20:52:46	2021-11-20 20:52:46		

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create']);

Route::get('/contact', function () {
    return view('contact');
});
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }
}
```

resources\views\welcome.blade.php

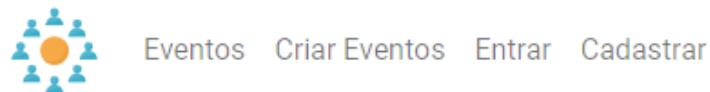
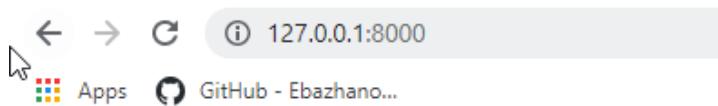
```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')

    @foreach($events as $event)
        <p>{{ $event->title }} - {{ $event->description}} </p>
    @endforeach

@endsection
```



Evento teste - Descrição teste

Evento de Javascript - Aprender muito JS

Evento de PHP - Aprender PHP

HDC Events © 2021

Aula 14 - Finalizando a home

`resources\views\welcome.blade.php`

```

@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')

    <div id="search-container" class="col-md-12">
        <h1>Busque um evento</h1>
        <form action="#">
            <input type="text" id="search" name="search" class="form-control"
placeholder="Procurar...">
        </form>
    </div>

    <div id="events-container" class="col-md-12">
        <h2>Próximos eventos</h2>
        <p class="subtitle">Veja os eventos dos próximos dias</p>
        <div id="cards-container" class="row">
            @foreach ($events as $event)
                <div class="card col-md-3">
                    title }}">
                    <div class="card-body">
                        <p class="card-date">10/09/2020</p>
                        <h5 class="card-title">{{ $event->title }}</h5>
                        <p class="card-participants">X participantes</p>
                        <a href="#" class="btn btn-primary">Saber mais</a>
                    </div>
                </div>
            @endforeach
        </div>
    </div>

@endsection

```

```
public\css\styles.css

/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

/* navbar */

#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

/* Search Container */

#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}
```

```
#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 0;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
    margin: 5px;
}

#events-container img{
    max-height: 150px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.card .card-date{
    font-size: 12px;
    color: #ccc;
}

.card .card-title{
    color: #212121;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 30px;
}

.card-participants{
    font-size: 14px;
    color: #757575;
}

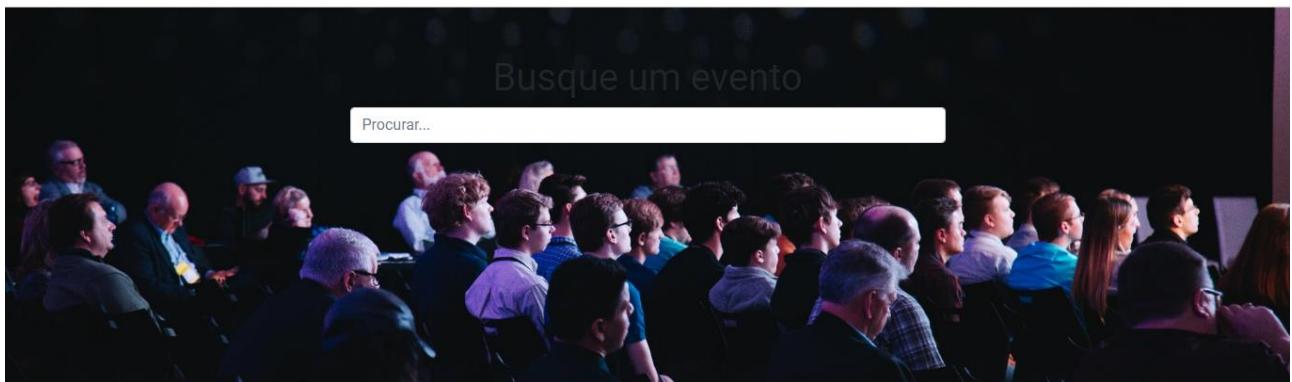
/* footer */

footer{
    text-align: center;
    background-color: #353e48;
    color: #fff;
    padding: 30px;
    position: absolute;
    bottom: 0;
    width: 100%;
}

footer p{
    margin-bottom: 0;
}
```



Eventos Criar Eventos Entrar Cadastrar



Próximos eventos

Veja os eventos dos próximos dias



10/09/2020

Evento teste

X participantes

[Saber mais](#)



10/09/2020

Evento de Javascript

X participantes

[Saber mais](#)



10/09/2020

Evento de PHP

X participantes

[Saber mais](#)

HDC Events © 2021

Aula 15 - Salvando dados no banco MySQL

Adicionando registro ao banco

- No Laravel é comum ter uma action específica para o post chamada de `store`;
- Lá vamos criar o objeto e compor ele com base nos dados enviados pelo `POST`;
- Com o objeto formado utilizamos o método `save` para persistir os dados.

`app\Http\Controllers\EventController.php`

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;

        $event->save();
        return redirect('/');
    }
}
```

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create']);
Route::post('/events', [EventController::class, 'store']);

Route::get('/contact', function () {
    return view('contact');
});
```

resources\views\events\create.blade.php

```
@extends('layouts.main')

@section('title', 'Criar evento')

@section('content')



# Crie o seu evento


@csrf


<label for="title">Evento:</label>
<input type="text" class="form-control" id="title" name="title" placeholder="Nome do evento">



<label for="city">Cidade:</label>
<input type="text" class="form-control" id="city" name="city" placeholder="Local do evento">



<label for="private">O evento é privado?</label>
<select class="form-control" name="private" id="private">
<option value="0">Não</option>
<option value="1">Sim</option>
</select>



<label for="description">Descrição:</label>
<textarea class="form-control" name="description" id="description" placeholder="O que vai acontecer no evento?"></textarea>


<input class="btn btn-primary" type="submit" value="Criar evento">


```

@endsection

← → ⌛ 127.0.0.1:8000/events/create

Apps GitHub - Ebazhano...

Lista de leitura

Eventos Criar Eventos Entrar Cadastrar

Crie o seu evento

Evento:

Cidade:

O evento é privado?

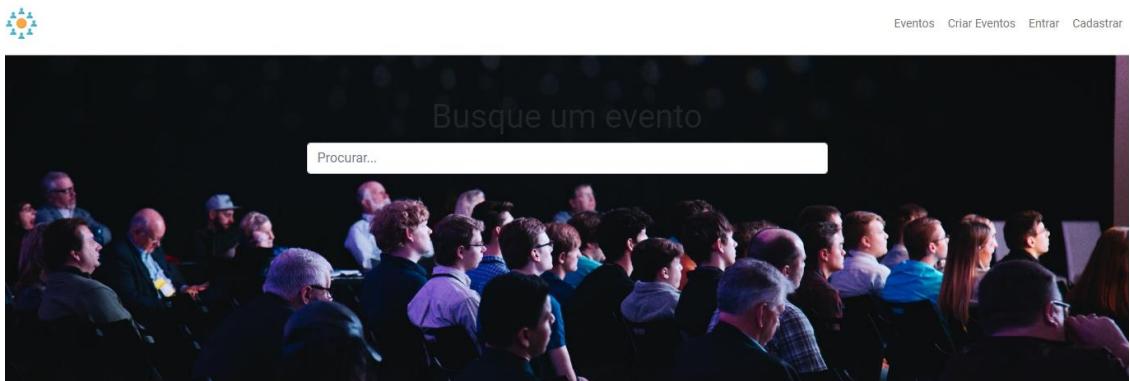
Descrição:

Criar evento

HDC Events © 2021

+ Opções

		id	title	description	city	private	created_at	updated_at
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	1	Evento teste	Descrição teste	Floripa	0 2021-11-20 20:47:42 2021-11-20 20:47:42
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	2	Evento de Javascript	Aprender muito JS	São Paulo	0 2021-11-20 20:51:15 2021-11-20 20:51:15
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	3	Evento de PHP	Aprender PHP	São José	0 2021-11-20 20:52:46 2021-11-20 20:52:46
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	4	Evento Java	Serão abordados detalhes importantes sobre a lingu...	Rio de Janeiro	0 2021-11-21 13:04:16 2021-11-21 13:04:16



Próximos eventos

Veja os eventos dos próximos dias

			
10/09/2020	10/09/2020	10/09/2020	10/09/2020
Evento teste	Evento de Javascript	Evento de PHP	Evento Java
X participantes	X participantes	X participantes	X participantes
Saber mais	Saber mais	Saber mais	Saber mais

public\css\styles.css

```
/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

/* navbar */

#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

/* Search Container */
```

```
#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}

#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 0;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
    margin: 5px;
}

#events-container img{
    max-height: 150px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.card .card-date{
    font-size: 12px;
    color: #ccc;
}

.card .card-title{
    color: #212121;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 30px;
}
```

```
.card-participants{
  font-size: 14px;
  color: #757575;
}

/* footer */

footer{
  text-align: center;
  background-color: #353e48;
  color: #fff;
  padding: 30px;
  position: absolute;
  bottom: 0;
  width: 100%;
}

/* Event create page */

#event-create-container{
  padding: 30px;
}

#event-create-container label{
  font-weight: bold;
}

#event-create-container input,
#event-create-container select,
#event-create-container textarea
{
  font-size: 12px;
  margin-bottom: 10px;
}

footer p{
  margin-bottom: 0;
}
```

Crie o seu evento

Evento:

Cidade:

O evento é privado?

Sim

Não

Descrição:

Criar evento

Aula 16 - Conhecendo as Flash Messages

- Podemos adicionar mensagens ao usuário por `session`;
- Esta mensagens são conhecidas por `flash messages`;
- Podemos adicionar com o método `with` nos Controllers;
- Utilizadas para apresentar um feedback ao usuário;
- No blade podemos verificar a presença da mensagem pela diretiva `@session`.

`app\Http\Controllers\EventController.php`

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }
}
```

resources\views\layouts\main.blade.php

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Fonte do Google -->
    <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

    <!-- CSS Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <!-- CSS da aplicação -->
    <link rel="stylesheet" href="/css/styles.css">

    <script src="/js/scripts.js"></script>
    <title>@yield('title')</title>
</head>

<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light">
            <div id="navbar" class="collapse navbar-collapse">
                <a href="#" class="navbar-brand">
                    
                </a>
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/events/create">Criar Eventos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Entrar</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Cadastrar</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>
    <main>
        <div class="container-fluid">
            <div class="row">
                @if(session('msg'))
                    <p class="msg">{{ session('msg') }}</p>
                @endif
                @yield('content')
            </div>
        </div>
    </main>
    <footer>
        <p>HDC Events &copy; 2021</p>
    </footer>

    <script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
</body>
</html>

```

public\css\styles.css

```

/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

.msg{
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
    width: 100%;
    margin-bottom: 0px;
    text-align: center;
    padding: 10px;
}

/* navbar */

#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

```

```
/* Search Container */

#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}

#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 0;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
    margin: 5px;
}

#events-container img{
    max-height: 150px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.card .card-date{
    font-size: 12px;
    color: #ccc;
}

.card .card-title{
    color: #212121;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 30px;
}

.card-participants{
    font-size: 14px;
    color: #757575;
}

/* footer */

footer{
    text-align: center;
```

```

background-color: #353e48;
color: #fff;
padding: 30px;
position: absolute;
bottom: 0;
width: 100%;
}

/* Event create page */

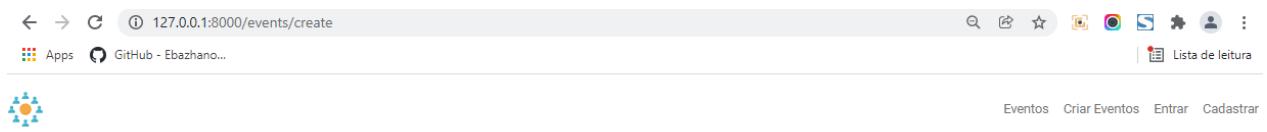
#event-create-container{
    padding: 30px;
}

#event-create-container label{
    font-weight: bold;
}

#event-create-container input,
#event-create-container select,
#event-create-container textarea
{
    font-size: 12px;
    margin-bottom: 10px;
}

footer p{
    margin-bottom: 0;
}

```



Crie o seu evento

Evento:
Dicas de CSS

Cidade:
Maceió

O evento é privado?
Não

Descrição:
Dicas para uma formatação de páginas web bem feita utilizando CSS.

Criar evento

HDC Events © 2021

Opções								
	← T →	id	title	description	city	private	created_at	updated_at
<input type="checkbox"/>	Edita	1	Evento teste	Descrição teste	Floripa	0	2021-11-20 20:47:42	2021-11-20 20:47:42
<input type="checkbox"/>	Edita	2	Evento de Javascript	Aprender muito JS	São Paulo	0	2021-11-20 20:51:15	2021-11-20 20:51:15
<input type="checkbox"/>	Edita	3	Evento de PHP	Aprender PHP	São José	0	2021-11-20 20:52:46	2021-11-20 20:52:46
<input type="checkbox"/>	Edita	4	Evento Java	Serão abordados detalhes importantes sobre a lingu...	Rio de Janeiro	0	2021-11-21 13:04:16	2021-11-21 13:04:16
<input type="checkbox"/>	Edita	5	Dicas de CSS	Dicas para uma formatação de páginas web bem feita...	Maceió	0	2021-11-21 13:59:12	2021-11-21 13:59:12



Evento criado com sucesso!

Busque um evento



Próximos eventos

Veja os eventos dos próximos dias



10/09/2020

Evento teste

X participantes

[Saber mais](#)



10/09/2020

Evento de Javascript

X participantes

[Saber mais](#)



10/09/2020

Evento de PHP

X participantes

[Saber mais](#)



10/09/2020

Evento Java

X participantes

[Saber mais](#)



10/09/2020

Dicas de CSS

X participantes

[Saber mais](#)

Aula 17 - Upload de imagens com Laravel

Salvando imagem no Laravel

- Para fazer o upload de imagens precisamos mudar o `enctype` do form e também `criar um input` de envio das mesmas;
- No Controller vamos fazer um `tratamento de verificação da imagem` que foi enviada;
- E depois vamos salvar ela com um `nome único` em um diretório do projeto;
- No banco salvamos apenas o `path` para a imagem;

Criando a migration para adicionar a imagem na tabela events

```
php artisan make:migration add_image_to_events_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration add_image_to_events_table
Created Migration: 2021_11_21_163304_add_image_to_events_table
```

```
database\migrations\2021_11_21_163304_add_image_to_events_table.php
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddImageToEventsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->string('image');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->dropColumn('image');
        });
    }
}
```

```
php artisan migrate
```

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2021_11_21_163304_add_image_to_events_table
Migrated: 2021_11_21_163304_add_image_to_events_table (1,696.26ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
1	<code>id</code>	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
2	<code>title</code>	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
3	<code>description</code>	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
4	<code>city</code>	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
5	<code>private</code>	tinyint(1)			Não	Nenhum			Muda Elimina ▾ Mais
6	<code>created_at</code>	timestamp			Sim	NULL			Muda Elimina ▾ Mais
7	<code>updated_at</code>	timestamp			Sim	NULL			Muda Elimina ▾ Mais
8	<code>image</code>	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {
            $requestImage = $request->image;
            $extension = $requestImage->extension();
            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) .
            "." . $extension;
            $requestImage->move(public_path('img/events'), $imageName);
            $event->image = $imageName;
        }

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }
}
```

```
resources\views\events\create.blade.php
```

```
@extends('layouts.main')

@section('title', 'Criar evento')

@section('content')



# Crie o seu evento


<form action="/events" method="POST" enctype="multipart/form-data">
    @csrf
    <div class="form-group">
        <label for="image">Imagen do Evento:</label>
        <input type="file" id="image" name="image" class="form-control-file">
    </div>
    <div class="form-group">
        <label for="title">Evento:</label>
        <input type="text" class="form-control" id="title" name="title"
placeholder="Nome do evento">
    </div>
    <div class="form-group">
        <label for="city">Cidade:</label>
        <input type="text" class="form-control" id="city" name="city"
placeholder="Local do evento">
    </div>
    <div class="form-group">
        <label for="private">O evento é privado?</label>
        <select class="form-control" name="private" id="private">
            <option value="0">Não</option>
            <option value="1">Sim</option>
        </select>
    </div>
    <div class="form-group">
        <label for="description">Descrição:</label>
        <textarea class="form-control" name="description" id="description"
placeholder="O que vai acontecer no evento?"></textarea>
    </div>
    <input class="btn btn-primary" type="submit" value="Criar evento">
</form>
</div>

@endsection


```

```
resources\views\welcome.blade.php
```

```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')



# Busque um evento



## Próximos eventos



Veja os eventos dos próximos dias



@foreach ($events as $event)


![{{ $event->title }}](/img/events/{{ $event->image }})

10/09/2020



##### {{ $event->title }}



X participantes

Saber mais


@endforeach


```

public\css\styles.css

```

/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

.msg{
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
    width: 100%;
    margin-bottom: 0px;
    text-align: center;
    padding: 10px;
}

/* navbar */

#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

/* Search Container */

#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}

```

```

#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 24%;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
    margin: .5%;

}

#events-container img{
    max-height: 150px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.card .card-date{
    font-size: 12px;
    color: #ccc;
}

.card .card-title{
    color: #212121;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 30px;
}

.card-participants{
    font-size: 14px;
    color: #757575;
}

/* footer */

footer{
    text-align: center;
    background-color: #353e48;
    color: #fff;
    padding: 30px;
    position: absolute;
    bottom: 0;
    width: 100%;
}

/* Event create page */

#event-create-container{
    padding: 30px;
}

#event-create-container label{
    font-weight: bold;
}

#event-create-container input,

```

```
#event-create-container select,
#event-create-container textarea
{
    font-size: 12px;
    margin-bottom: 10px;
}

footer p{
    margin-bottom: 0;
}
```

Crie o seu evento

Imagen do Evento: evento2.jpg

Evento:
Evento sobre Linguagens de Programação

Cidade:
Porto Alegre

O evento é privado?
Não

Descrição:
Linguagens de programação mais utilizadas na atualidade.



Próximos eventos

Veja os eventos dos próximos dias

10/09/2020 Evento teste X participantes <input type="button" value="Saber mais"/>	10/09/2020 Evento de Javascript X participantes <input type="button" value="Saber mais"/>	10/09/2020 Evento de PHP X participantes <input type="button" value="Saber mais"/>	10/09/2020 Evento Java X participantes <input type="button" value="Saber mais"/>
10/09/2020 Dicas de CSS X participantes <input type="button" value="Saber mais"/>	10/09/2020 Evento sobre Linguagens de Programação X participantes <input type="button" value="Saber mais"/>		

Aula 18 - Resgatando um registro do banco de dados

- Vamos fazer a view do evento nesta aula;
- E com isso aprender a resgatar **apenas um registro** pelo Eloquent;
- Vamos utilizar o método **findOrFail**;
- E também criar uma nova view e rota para esta função;
- Esta tela tem a função de exibir todas as informações do evento e também o botão de participar.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create']);
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);

Route::get('/contact', function () {
    return view('contact');
});
```

resources\views\welcome.blade.php

```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')



<h1>Busque um evento</h1>
    <form action="#">
        <input type="text" id="search" name="search" class="form-control" placeholder="Procurar...">
    </form>



<h2>Próximos eventos</h2>
    <p class="subtitle">Veja os eventos dos próximos dias</p>
    <div id="cards-container" class="row">
        @foreach ($events as $event)
            <div class="card col-md-3">
                title }}">
                <div class="card-body">
                    <p class="card-date">10/09/2020</p>
                    <h5 class="card-title">{{ $event->title }}</h5>
                    <p class="card-participants">X participantes</p>
                    <a href="/events/{{ $event->id }}" class="btn btn-primary">Saber
mais</a>
                </div>
            </div>
        @endforeach
    </div>


@endsection
```

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {

            $requestImage = $request->image;

            $extension = $requestImage->extension();

            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) .
". ." . $extension;

            $requestImage->move(public_path('img/events'), $imageName);

            $event->image = $imageName;
        }

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }

    public function show($id){
        $event = Event::findOrFail($id);
        return view('events.show', ['event' => $event]);
    }
}

```

```
resources\views\events\show.blade.php
```

```

@extends('layouts.main')

@section('title', $event->title)

@section('content')

    <div class="col-md-10 offset-md-1">
        <div class="row">
            <div id="image-container" class="col-md-6">
                title }}">
            </div>
            <div id="info-container" class="col-md-6">
                <h1>{{ $event->title }}</h1>
                <p class="event-city"><ion-icon name="pin"></ion-icon> {{ $event->city }}</p>
                <p class="events-participants"><ion-icon name="people"></ion-icon> X Participantes</p>
                <p class="event-owner"><ion-icon name="star-outline"></ion-icon> Dono do evento</p>
                <a href="#" class="btn btn-primary" id="event-submit">Confirmar presença</a>
            </div>
            <div class="col-md-12" id="description-container">
                <h3>Sobre o evento:</h3>
                <p class="event-description">{{ $event->description }}</p>
            </div>
        </div>
    </div>

@endsection

```

```

public\css\styles.css

/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

.msg{
    background-color: #d4edda;
    color: #155724;
    border: 1px solid #c3e6cb;
    width: 100%;
    margin-bottom: 0px;
    text-align: center;
    padding: 10px;
}

/* navbar */

#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

/* Search Container */

#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}

```

```

#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 24%;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
    margin: .5%;
}

#events-container img{
    max-height: 150px;
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.card .card-date{
    font-size: 12px;
    color: #ccc;
}

.card .card-title{
    color: #212121;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 30px;
}

.card-participants{
    font-size: 14px;
    color: #757575;
}

/* footer */

footer{
    text-align: center;
    background-color: #353e48;
    color: #fff;
    padding: 30px;
    position: absolute;
    bottom: 0;
    width: 100%;
}

/* Event create page */

#event-create-container{
    padding: 30px;
}

#event-create-container label{
    font-weight: bold;
}

#event-create-container input,

```

```

.event-create-container select,
.event-create-container textarea
{
    font-size: 12px;
    margin-bottom: 10px;
}

/* Event page */

.image-container,
.info-container{
    margin-top: 30px;
}

.image-container img{
    border-radius: 10px;
}

.info-container h1{
    font-size: 36px;
    font-weight: 900;
}

.info-container p{
    margin: 0;
    font-size: 14px;
    display: flex;
}

.info-container h3,
.description-container h3{
    font-size: 20px;
    font-weight: bold;
}

.info-container ion-icon{
    font-size: 20px;
    color: #f2a340;
    margin-right: 5px;
}

.info-container #event-submit{
    margin: 20px 0;
}

footer p{
    margin-bottom: 0;
}

```



[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)



Sobre o evento:

Linguagens de programação mais utilizadas na atualidade.

Evento sobre Linguagens de Programação

📍 Porto Alegre
👤 X Participantes
⭐ Dono do evento

[Confirmar presença](#)

Aula 19 - Salvando JSON no banco de dados

- Podemos salvar um conjunto de dados no banco para **itens de múltipla escolha**;
- Vamos criar um campo determinado de **json** via migrations;
- No front-end podemos utilizar **inputs com checkbox**;
- Após o envio para Controller, apenas recebemos do request os itens e resto do processo é igual ao dos outros dados do nosso Model.

```
php artisan make:migration add_items_to_events_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration add_items_to_events_table
Created Migration: 2021_11_21_231732_add_items_to_events_table
```

database\migrations\2021_11_21_231732_add_items_to_events_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddItemsToEventTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->json('items');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->dropColumn('items');
        });
    }
}
```

php artisan migrate

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2021_11_21_231732_add_items_to_events_table
Migrated: 2021_11_21_231732_add_items_to_events_table (822.83ms)
```

+ Opções									
	← T →	id	title	description	city	private	created_at	updated_at	image
<input type="checkbox"/>	Edita Copiar Apagar	1	Evento teste	Descrição teste	Floripa	0	2021-11-20 20:47:42	2021-11-20 20:47:42	
<input type="checkbox"/>	Edita Copiar Apagar	2	Evento de Javascript	Aprender muito JS	São Paulo	0	2021-11-20 20:51:15	2021-11-20 20:51:15	
<input type="checkbox"/>	Edita Copiar Apagar	3	Evento de PHP	Aprender PHP	São José	0	2021-11-20 20:52:46	2021-11-20 20:52:46	
<input type="checkbox"/>	Edita Copiar Apagar	4	Evento Java	Serão abordados detalhes importantes sobre a lingu...	Rio de Janeiro	0	2021-11-21 13:04:16	2021-11-21 13:04:16	
<input type="checkbox"/>	Edita Copiar Apagar	5	Dicas de CSS	Dicas para uma formatação de páginas web bem feita...	Maceió	0	2021-11-21 13:59:12	2021-11-21 13:59:12	
<input type="checkbox"/>	Edita Copiar Apagar	6	Evento sobre Linguagens de Programação	Linguagens de programação mais utilizadas na atual...	Porto Alegre	0	2021-11-21 17:08:41	2021-11-21 17:08:41	72c4c3b2c60ffffc6b78a45d1e0de7e9.jpg

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	4 city	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	5 private	tinyint(1)			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	6 created_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	7 updated_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	8 image	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	9 items	longtext	utf8mb4_bin		Não	Nenhum			Muda Elimina ▾ Mais

resources\views\events\create.blade.php

```

@extends('layouts.main')

@section('title', 'Criar evento')

@section('content')



# Crie o seu evento


<form action="/events" method="POST" enctype="multipart/form-data">
    @csrf
    <div class="form-group">
        <label for="image">Imagen do Evento:</label>
        <input type="file" id="image" name="image" class="form-control-file">
    </div>
    <div class="form-group">
        <label for="title">Evento:</label>
        <input type="text" class="form-control" id="title" name="title" placeholder="Nome do evento">
    </div>
    <div class="form-group">
        <label for="city">Cidade:</label>
        <input type="text" class="form-control" id="city" name="city" placeholder="Local do evento">
    </div>
    <div class="form-group">
        <label for="private">O evento é privado?</label>
        <select class="form-control" name="private" id="private">
            <option value="0">Não</option>
            <option value="1">Sim</option>
        </select>
    </div>
    <div class="form-group">
        <label for="description">Descrição:</label>
        <textarea class="form-control" name="description" id="description" placeholder="O que vai acontecer no evento?"></textarea>
    </div>
    <div class="form-group">
        <label for="items">Adicione itens de infra-estrutura</label>
        <div class="form-group">
            <input type="checkbox" name="items[]" value="Cadeiras"> Cadeiras
        </div>
        <div class="form-group">
            <input type="checkbox" name="items[]" value="Palco"> Palco
        </div>
        <div class="form-group">
            <input type="checkbox" name="items[]" value="Cerveja grátis"> Cerveja grátis
        </div>
        <div class="form-group">
            <input type="checkbox" name="items[]" value="Open Food"> Open Food
        </div>
        <div class="form-group">
            <input type="checkbox" name="items[]" value="Brindes"> Brindes
        </div>
    </div>
    <input class="btn btn-primary" type="submit" value="Criar evento">
</form>
</div>


```

@endsection

```
resources\views\events\show.blade.php
```

```
@extends('layouts.main')

@section('title', $event->title)

@section('content')



![{{ $event->title }}](/img/events/{{ $event->image }})



# {{ $event->title }}



<ion-icon name="pin"></ion-icon> {{ $event->city }}</p>


<ion-icon name="people"></ion-icon> X Participantes</p>


<ion-icon name="star-outline"></ion-icon> Dono do evento</p>
Confirmar presença</a>



### O evento conta com:




@foreach ($event->items as $item)
- <ion-icon name="play"></ion-icon> <span>{{ $item }}</span></li>
@endforeach



### Sobre o evento:



{{ $event->description }}


```

```
@endsection
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {

            $requestImage = $request->image;

            $extension = $requestImage->extension();

            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . " ." . $extension;

            $requestImage->move(public_path('img/events'), $imageName);

            $event->image = $imageName;
        }

        $event->save();
        return redirect('/')->with('msg','Evento criado com sucesso!');
    }

    public function show($id){
        $event = Event::findOrFail($id);
        return view('events.show', ['event' => $event]);
    }
}
```

app\Models\Event.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Event extends Model
{
    use HasFactory;

    protected $casts = [
        'items' => 'array'
    ];
}
```

public\css\styles.css

```
/* General */

* {
    font-family: 'Roboto';
}

body{
    min-height: 100vh;
    padding-bottom: 100px;
    position: relative;
}

header{
    background-color: #fff;
    border-bottom: 1px solid #ccc;
}

.row{
    margin: 0;
}

.container-fluid{
    padding: 0;
}

.btn-primary{
    background-color: #f2a340;
    border: 2px solid #c28008
}

.btn-primary:hover{
    background-color: #c28008;
    border-color: #c28008;
}

.msg{
    background-color: #d4edda;
    color: #155724;
```

```

border: 1px solid #c3e6cb;
width: 100%;
margin-bottom: 0px;
text-align: center;
padding: 10px;
}

/* navbar */

#navbar img{
  width: 40px;
}

#navbar{
  justify-content: space-between;
  background-color: #fff;
  padding: 5px 0;
}

#navbar a{
  font-size: 14px;
  color: #757575;
}

#navbar a:hover{
  color: #f2a340;
}

/* Search Container */

#search-container{
  background-image: url('../img/banner.jpg');
  background-size: cover;
  background-position: center;
  height: 400px;
  padding: 50px;
  text-align: center;
}

#search-container form{
  width: 50%;
  margin: 0 auto;
}

/* Events list */

#events-container{
  padding: 50px;
}

#events-container h2{
  margin-bottom: 10px;
}

#events-container .subtitle{
  color: #757575;
  margin-bottom: 30px;
}

#cards-container{
  display: flex;
}

#events-container .card{
  flex: 1 1 24%;
}

```

```

max-width: 25%;
border-radius: 10px;
padding: 0;
margin: .5%;
}

#events-container img{
  max-height: 150px;
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
}

.card .card-date{
  font-size: 12px;
  color: #ccc;
}

.card .card-title{
  color: #212121;
  font-size: 20px;
  font-weight: bold;
  margin-bottom: 30px;
}

.card-participants{
  font-size: 14px;
  color: #757575;
}

/* footer */

footer{
  text-align: center;
  background-color: #353e48;
  color: #fff;
  padding: 30px;
  position: absolute;
  bottom: 0;
  width: 100%;
}

/* Event create page */

#event-create-container{
  padding: 30px;
}

#event-create-container label{
  font-weight: bold;
}

#event-create-container input,
#event-create-container select,
#event-create-container textarea
{
  font-size: 12px;
  margin-bottom: 10px;
}

/* Event page */

#image-container,
#info-container{
  margin-top: 30px;
}

```

```
#image-container img{
  border-radius: 10px;
}

#info-container h1{
  font-size: 36px;
  font-weight: 900;
}

#info-container p{
  margin: 0;
  font-size: 14px;
  display: flex;
}

#info-container h3,
#description-container h3{
  font-size: 20px;
  font-weight: bold;
}

#info-container ion-icon{
  font-size: 20px;
  color: #f2a340;
  margin-right: 5px;
}

#info-container #event-submit{
  margin: 20px 0;
}

#items-list{
  list-style: none;
  padding-left: 0;
}

#items-list li{
  display: flex;
}

footer p{
  margin-bottom: 0;
}
```

I Crie o seu evento

Imagen do Evento: evento3.jpg

Evento:

Evento de TI

Cidade:

São Paulo

O evento é privado?

Não

Descrição:

Grande evento de tecnologia da informação.

Adiciona itens de infra-estrutura

- Cadeiras
- Palco
- Cerveja grátis
- Open Food
- Brindes

<input type="checkbox"/>	<input type="button" value="Procurar"/>	<input type="button" value="Estrutura"/>	<input type="button" value="SQL"/>	<input type="button" value="Pesquisar"/>	<input type="button" value="Insere"/>	<input type="button" value="Exportar"/>	<input type="button" value="Importar"/>	<input type="button" value="Privilégios"/>	<input type="button" value="Operações"/>	<input type="button" value="Rastreando"/>	<input type="button" value="Acionadores"/>	
												items
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	1	Evento teste	Descrição teste	Floripa	0	2021-11-20 20:47:42	2021-11-20 20:47:42		
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	2	Evento de Javascript	Aprender muito JS	São Paulo	0	2021-11-20 20:51:15	2021-11-20 20:51:15		
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	3	Evento de PHP	Aprender PHP	São José	0	2021-11-20 20:52:46	2021-11-20 20:52:46		
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	4	Evento Java	Serão abordados detalhes importantes sobre a lingu...	Rio de Janeiro	0	2021-11-21 13:04:16	2021-11-21 13:04:16		
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	5	Dicas de CSS	Dicas para uma formatação de páginas web bem feita...	Maceió	0	2021-11-21 13:59:12	2021-11-21 13:59:12		
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	6	Evento sobre Linguagens de Programação	Linguagens de programação mais utilizadas na atual...	Porto Alegre	0	2021-11-21 17:08:41	2021-11-21 17:08:41	72c4c3b2c60fffc6b78a45d1e0de7e9.jpg	
<input type="checkbox"/>	<input type="button" value="Edita"/>	<input type="button" value="Copiar"/>	<input type="button" value="Apagar"/>	7	Evento de TI	Grande evento de tecnologia da informação.	São Paulo	0	2021-11-21 23:58:11	2021-11-21 23:58:11	b950a2c28fd4aad30f21e0f61e4d67e6.jpg	[{"Cadeiras","Palco","Open Food"]



Evento criado com sucesso!

Busque um evento

Próximos eventos

Veja os eventos dos próximos dias

<p>10/09/2020</p> <p>Evento teste</p> <p>X participantes</p> <p>Saber mais</p>	<p>10/09/2020</p> <p>Evento de Javascript</p> <p>X participantes</p> <p>Saber mais</p>	<p>10/09/2020</p> <p>Evento de PHP</p> <p>X participantes</p> <p>Saber mais</p>	<p>10/09/2020</p> <p>Evento Java</p> <p>X participantes</p> <p>Saber mais</p>
<p>10/09/2020</p> <p>Dicas de CSS</p> <p>X participantes</p> <p>Saber mais</p>	<p>10/09/2020</p> <p>Evento sobre Linguagens de Programação</p> <p>X participantes</p> <p>Saber mais</p>	<p>10/09/2020</p> <p>Evento de TI</p> <p>X participantes</p> <p>Saber mais</p>	

HDC Events © 2021



Evento de TI

📍 São Paulo
👤 X Participantes
⭐ Dono do evento

[Confirmar presença](#)

O evento conta com:

- Cadeiras
- Palco
- Open Food

Sobre o evento:

Grande evento de tecnologia da informação.

HDC Events © 2021

Aula 20 - Salvando datas no banco de dados

- Para salvar datas no Laravel é muito fácil, precisamos criar um **input do tipo date** na nossa view;
- Um campo de **dateTime** via migrations;
- E processar o envio dos dados via Controller pelo request de POST.

```
php artisan make:migration add_date_to_events_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration add_date_to_events_table
Created Migration: 2021_11_22_003554_add_date_to_events_table
```

database\migrations\2021_11_22_003554_add_date_to_events_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddDateToEventsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->dateTime('date');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->dropColumn('date');
        });
    }
}
```

php artisan migrate:fresh

```
C:\xampp\htdocs\hdcevents>php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (624.34ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (576.41ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (551.50ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (1,974.42ms)
Migrating: 2021_11_20_232254_create_events_table
Migrated: 2021_11_20_232254_create_events_table (879.48ms)
Migrating: 2021_11_21_163304_add_image_to_events_table
Migrated: 2021_11_21_163304_add_image_to_events_table (308.05ms)
Migrating: 2021_11_21_231732_add_items_to_events_table
Migrated: 2021_11_21_231732_add_items_to_events_table (137.52ms)
Migrating: 2021_11_22_003554_add_date_to_events_table
Migrated: 2021_11_22_003554_add_date_to_events_table (149.08ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	4 city	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	5 private	tinyint(1)			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	6 created_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	7 updated_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	8 image	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	9 items	longtext	utf8mb4_bin		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	10 date	datetime			Não	Nenhum			Muda Elimina ▾ Mais

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $events = Event::all();
        return view('welcome', ['events' => $events]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {

            $requestImage = $request->image;

            $extension = $requestImage->extension();

            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;

            $requestImage->move(public_path('img/events'), $imageName);

            $event->image = $imageName;
        }

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }

    public function show($id){
        $event = Event::findOrFail($id);
        return view('events.show', ['event' => $event]);
    }
}

```

app\Models\Event.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Event extends Model
{
    use HasFactory;

    protected $casts = [
        'items' => 'array'
    ];

    protected $dates = ['date'];
}
```

resources\views\welcome.blade.php

```
@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')


<h1>Busque um evento</h1>
    <form action="#">
        <input type="text" id="search" name="search" class="form-control" placeholder="Procurar...">
    </form>



<h2>Próximos eventos</h2>
    <p class="subtitle">Veja os eventos dos próximos dias</p>
    <div id="cards-container" class="row">
        @foreach ($events as $event)
            <div class="card col-md-3">
                title }}">
                <div class="card-body">
                    <p class="card-date">{{ date('d/m/Y', strtotime($event->date)) }}</p>
                    <h5 class="card-title">{{ $event->title }}</h5>
                    <p class="card-participants">X participantes</p>
                    <a href="/events/{{ $event->id }}" class="btn btn-primary">Saber mais</a>
                </div>
            </div>
        @endforeach
        @if(count($events) == 0)
            <p>Não há eventos disponíveis</p>
        @endif
    </div>


@endsection
```



Crie o seu evento

Imagen do Evento: [Escolher arquivo](#) evento_.php.png

Evento:

PHP Conference Brasil 2021

Data do evento:

19/12/2019

Cidade:

Porto Alegre

O evento é privado?

Não

Descrição:

Será a 16ª edição do evento, que se consagra como o Principal evento de PHP da América Latina

Adicione itens de infra-estrutura

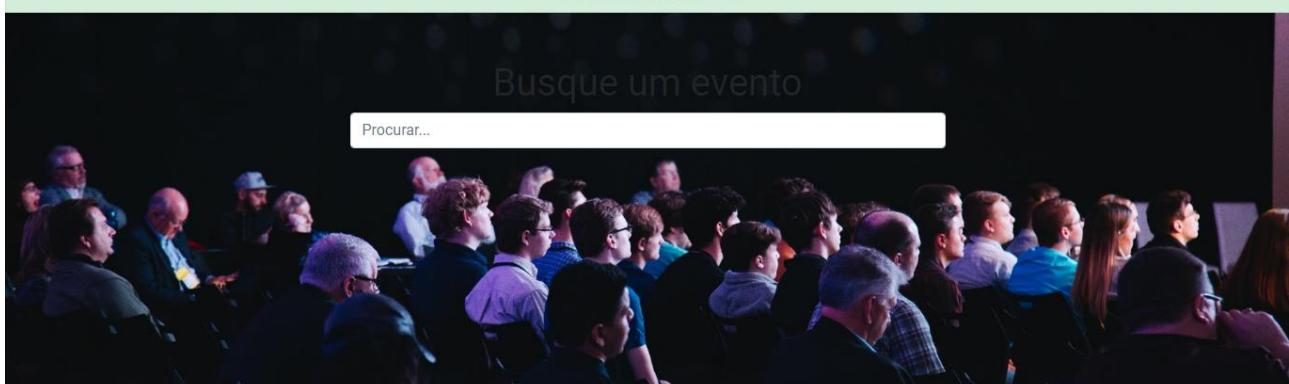
- Cadeiras
- Palco
- Cerveja grátis
- Open Food
- Brindes

[Criar evento](#)


[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)

Evento criado com sucesso!

Busque um evento



Próximos eventos

Veja os eventos dos próximos dias



19/12/2019

PHP Conference Brasil 2021

X participantes

[Saber mais](#)

HDC Events © 2021


[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)


PHP Conference Brasil 2021

📍 Porto Alegre
👤 X Participantes
⭐ Dono do evento

[Confirmar presença](#)

O evento conta com:

- Cadeiras
- Palco

Sobre o evento:

Será a 16ª edição do evento, que se consagra como o Principal evento de PHP da América Latina

HDC Events © 2021



Crie o seu evento

Imagen do Evento: evento_js.jpg

Evento:

Brazil JS Conference

Data do evento:

21/08/2015

Cidade:

Porto Alegre

O evento é privado?

Não

Descrição:

Em comemoração aos 20 anos de [JavaScript](#) a 5ª edição do evento [Brazil JS Conference](#) traz ao Brasil vários dos maiores desenvolvedores da atualidade.

Adicione itens de infra-estrutura

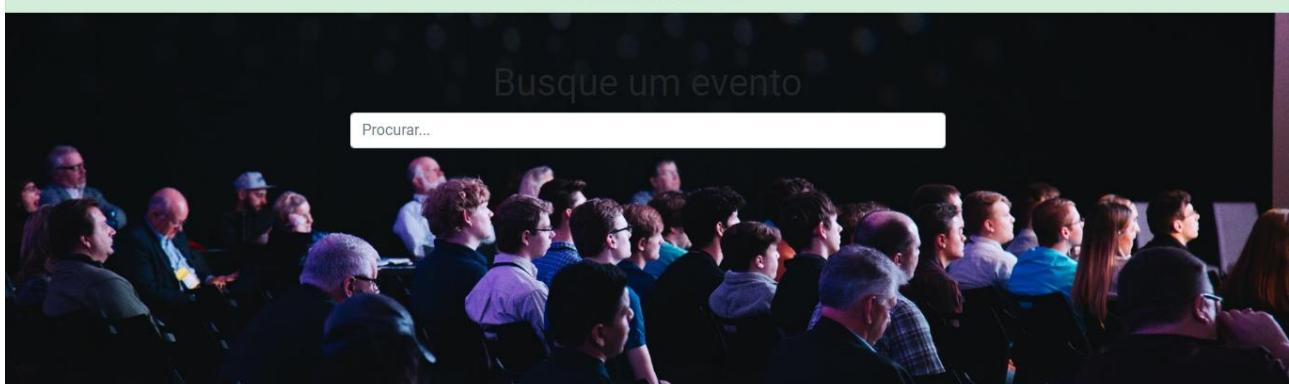
- Cadeiras
- Palco
- Cerveja grátis
- Open Food
- Brindes


[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)

Evento criado com sucesso!

Busque um evento

Procurar...



Próximos eventos

Veja os eventos dos próximos dias



19/12/2019

PHP Conference Brasil 2021

X participantes

[Saber mais](#)


21/08/2015

Brazil JS Conference

X participantes

[Saber mais](#)

HDC Events © 2021


[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)


Brazil JS Conference

📍 Porto Alegre
👤 X Participantes
⭐ Dono do evento

[Confirmar presença](#)

O evento conta com:

- Cadeiras
- Palco
- Brindes

Sobre o evento:

Em comemoração aos 20 anos de JavaScript a 5° edição do evento Brazil JS Conference traz ao Brasil vários dos maiores desenvolvedores da atualidade.

HDC Events © 2021

Aula 21 - Busca no Laravel

- Para criar uma busca no Laravel é muito simples, vamos utilizar o **Eloquent**;
- O método **where** vai identificar os registros que precisamos, **fazendo um filtro e nos enviando para a view**;
- Podemos fazer a exibição da busca na própria Home, mudando o layout com **diretivas de Blade**.

resources\views\welcome.blade.php

```
@extends('layouts.main')
@section('title', 'HDC Events')

@section('content')
<div id="search-container" class="col-md-12">
    <h1>Busque um evento</h1>
    <form action="/" method="GET">
        <input type="text" id="search" name="search" class="form-control" placeholder="Procurar...">
    </form>
</div>

<div id="events-container" class="col-md-12">
    @if($search)
        <h2>Buscando por {{ $search }}</h2>
    @else
        <h2>Próximos eventos</h2>
        <p class="subtitle">Veja os eventos dos próximos dias</p>
    @endif

    <div id="cards-container" class="row">
        @foreach ($events as $event)
            <div class="card col-md-3">
                title }}">
                <div class="card-body">
                    <p class="card-date">{{ date('d/m/Y', strtotime($event->date)) }}</p>
                    <h5 class="card-title">{{ $event->title }}</h5>
                    <p class="card-participants">X participantes</p>
                    <a href="/events/{{ $event->id }}" class="btn btn-primary">Saber mais</a>
                </div>
            </div>
        @endforeach
        @if(count($events) == 0 && $search)
            <p>Não foi possível encontrar nenhum evento com {{ $search }}! <a href="/">Ver todos!</a></p>
        @elseif(count($events) == 0)
            <p>Não há eventos disponíveis</p>
        @endif
    </div>
</div>
@endsection
```

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

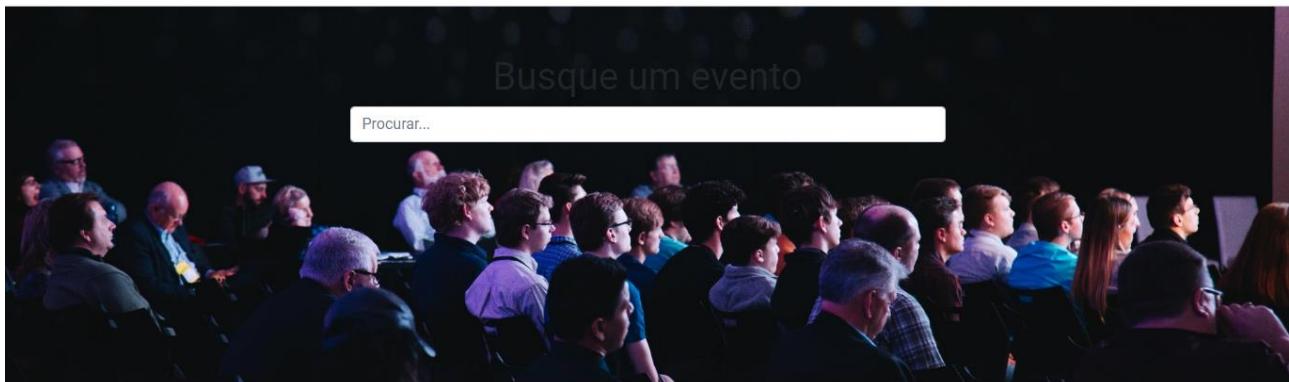
    public function store(Request $request) {
        $event = new Event;
        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {
            $requestImage = $request->image;
            $extension = $requestImage->extension();
            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
            $requestImage->move(public_path('img/events'), $imageName);
            $event->image = $imageName;
        }

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }

    public function show($id){
        $event = Event::findOrFail($id);
        return view('events.show', ['event' => $event]);
    }
}

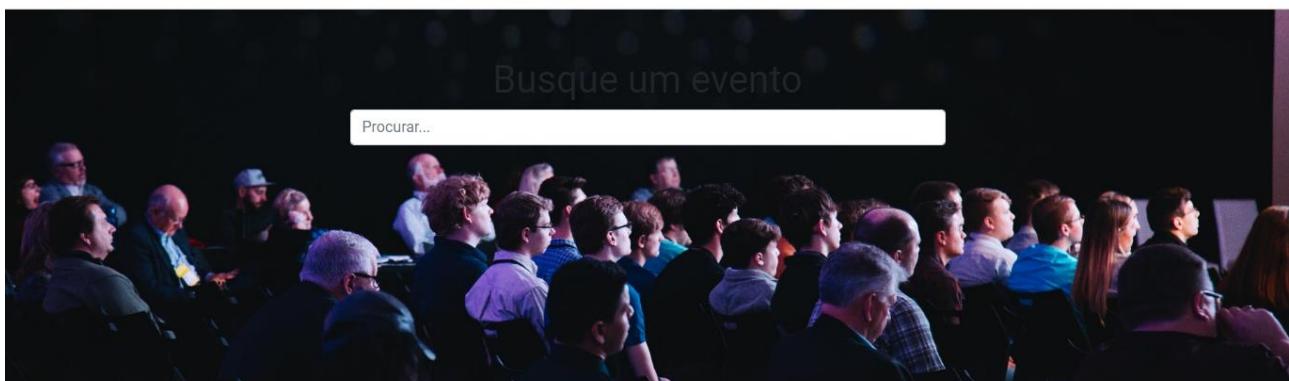
```

[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)

Buscando por php

19/12/2019
PHP Conference Brasil 2021
X participantes
[Saber mais](#)

HDC Events © 2021

[Eventos](#) [Criar Eventos](#) [Entrar](#) [Cadastrar](#)

Buscando por python

Não foi possível encontrar nenhum evento com python! [Ver todos!](#)

HDC Events © 2021

Aula 22 - Autenticação no Laravel

- Podemos utilizar o **Jetstream** para aplicar uma autenticação de modo rápido no nosso sistema;
- Primeiramente instalamos o pacote via Composer;
- Depois instalamos o **Liveware**, que são componentes de autenticação para o Blade;
- E por fim rodamos as migrations.

Instalação do jetstream

`composer require laravel/jetstream`

```
C:\xampp\htdocs\hdcevents>composer require laravel/jetstream
Using version ^2.4 for laravel/jetstream
./composer.json has been updated
Running composer update laravel/jetstream
Loading composer repositories with package information
Updating dependencies
Lock file operations: 9 installs, 0 updates, 0 removals
- Locking bacon/bacon-qr-code (2.0.4)
- Locking dasprid/enum (1.0.3)
- Locking jaybizzle/crawler-detect (v1.2.108)
- Locking jenssegers/agent (v2.6.4)
- Locking laravel/fortify (v1.8.3)
- Locking laravel/jetstream (v2.4.4)
- Locking mobiledetect/mobiledetectlib (2.8.37)
- Locking paragonie/constant_time_encoding (v2.4.0)
- Locking pragmarx/google2fa (8.0.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 9 installs, 0 updates, 0 removals
- Downloading dasprid/enum (1.0.3)
- Downloading bacon/bacon-qr-code (2.0.4)
- Downloading jaybizzle/crawler-detect (v1.2.108)
- Downloading paragonie/constant_time_encoding (v2.4.0)
- Downloading pragmarx/google2fa (8.0.0)
- Downloading laravel/fortify (v1.8.3)
- Downloading mobiledetect/mobiledetectlib (2.8.37)
- Downloading jenssegers/agent (v2.6.4)
- Downloading laravel/jetstream (v2.4.4)
- Installing dasprid/enum (1.0.3): Extracting archive
- Installing bacon/bacon-qr-code (2.0.4): Extracting archive
- Installing jaybizzle/crawler-detect (v1.2.108): Extracting archive
- Installing paragonie/constant_time_encoding (v2.4.0): Extracting archive
- Installing pragmarx/google2fa (8.0.0): Extracting archive
- Installing laravel/fortify (v1.8.3): Extracting archive
- Installing mobiledetect/mobiledetectlib (2.8.37): Extracting archive
- Installing jenssegers/agent (v2.6.4): Extracting archive
- Installing laravel/jetstream (v2.4.4): Extracting archive
1 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
```

Instalação do Liveware

`php artisan jetstream:install liveware`

```
C:\xampp\htdocs\hdcevents>php artisan jetstream:install liveware
Migration created successfully!
```

php artisan migrate:status

Ran?	Migration	Batch
Yes	2014_10_12_000000_create_users_table	1
Yes	2014_10_12_100000_create_password_resets_table	1
No	2014_10_12_200000_add_two_factor_columns_to_users_table	
Yes	2019_08_19_000000_create_failed_jobs_table	1
Yes	2019_12_14_000001_create_personal_access_tokens_table	1
Yes	2021_11_20_232254_create_events_table	1
Yes	2021_11_21_163304_add_image_to_events_table	1
Yes	2021_11_21_231732_add_items_to_events_table	1
Yes	2021_11_22_003554_add_date_to_events_table	1
No	2021_11_22_094331_create_sessions_table	

php artisan migrate

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2014_10_12_200000_add_two_factor_columns_to_users_table
Migrated: 2014_10_12_200000_add_two_factor_columns_to_users_table (1,922.45ms)
Migrating: 2021_11_22_094331_create_sessions_table
Migrated: 2021_11_22_094331_create_sessions_table (2,184.66ms)
```

php artisan jetstream:install livewire

```
C:\xampp\htdocs\hdcevents>php artisan jetstream:install livewire
./composer.json has been updated
Running composer update livewire/livewire
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking livewire/livewire (v2.7.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Downloading livewire/livewire (v2.7.2)
- Installing livewire/livewire (v2.7.2): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fruitcake/laravel-cors
Discovered Package: jenssegers/agent
Discovered Package: laravel/fortify
Discovered Package: laravel/jetstream
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: livewire/livewire
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
No publishable resources for tag [laravel-assets].
Publishing complete.
Copied Directory [\vendor\laravel\sanctum\database\migrations] To [\database\migrations]
Copied File [\vendor\laravel\sanctum\config\sanctum.php] To [\config\sanctum.php]
Publishing complete.

Livewire scaffolding installed successfully.
Please execute "npm install && npm run dev" to build your assets.
```

npm install

```
C:\xampp\htdocs\hdcevents>npm install
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. New code should use the URLSearchParams API instead.
npm WARN deprecated uid@0.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic
. See https://v8.dev/blog/math-random for details.
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
added 810 packages from 460 contributors in 420.249s

83 packages are looking for funding
  run `npm fund` for details
```

npm run dev

```
C:\xampp\htdocs\hdcevents>npm run dev
> @ dev C:\xampp\htdocs\hdcevents
> npm run development

> @ development C:\xampp\htdocs\hdcevents
> mix

• Mix [██████████] done (99%) plugins
BuildCallbackPlugin

warn - You have enabled the JIT engine which is currently in preview.
warn - Preview features are not covered by semver, may introduce breaking changes, and can change at any time.

• Mix [██████████] done (99%) plugins
WebpackBar:done

warn - You have enabled the JIT engine which is currently in preview.
warn - Preview features are not covered by semver, may introduce breaking changes, and can change at any time.
```

```
v Mix
Compiled successfully in 24.31s

warn - You have enabled the JIT engine which is currently in preview.
warn - Preview features are not covered by semver, may introduce breaking changes, and can change at any time.

v Mix
Compiled successfully in 24.31s

warn - You have enabled the JIT engine which is currently in preview.
warn - Preview features are not covered by semver, may introduce breaking changes, and can change at any time.

Laravel Mix v6.0.39

Compiling successfully in 23578ms
```

	File	Size
/js/app.js	710 Kib	
css/app.css	43 Kib	

```
• Mix [██████████] done (99%)
plugins

webpack compiled successfully
```

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create']);
Route::get('/events/{id}', [EventController::class, 'show']);
```

```

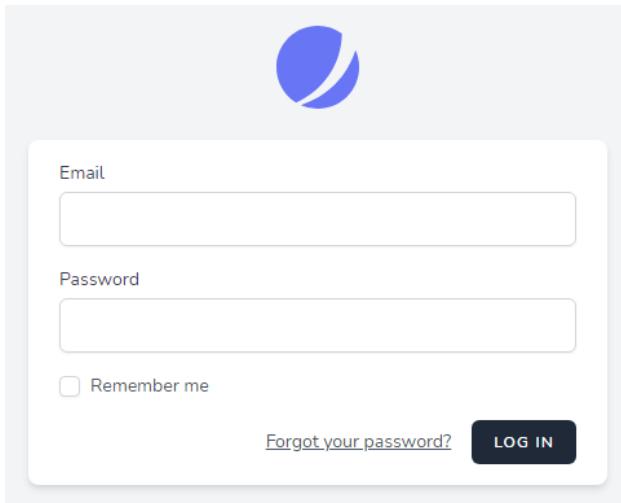
Route::post('/events', [EventController::class, 'store']);

Route::get('/contact', function () {
    return view('contact');
});

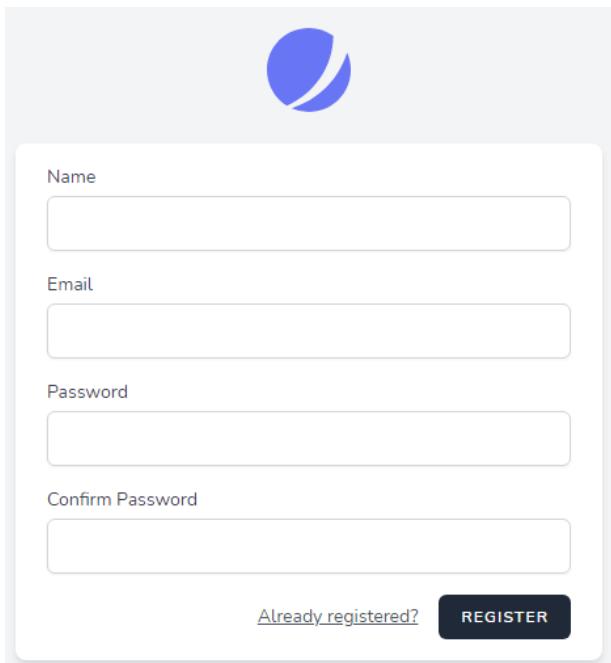
Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
    return view('dashboard');
})->name('dashboard');

```

<http://127.0.0.1:8000/login>



<http://127.0.0.1:8000/register>



resources\views\layouts\main.blade.php

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Fonte do Google -->
    <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

    <!-- CSS Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jiW3"
crossorigin="anonymous">

    <!-- CSS da aplicação -->
    <link rel="stylesheet" href="/css/styles.css">

    <script src="/js/scripts.js"></script>
    <title>@yield('title')</title>
</head>

<body>

<header>
    <nav class="navbar navbar-expand-lg navbar-light">
        <div id="navbar" class="collapse navbar-collapse">
            <a href="" class="navbar-brand">
                
            </a>
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/">Eventos</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/events/create">Criar Eventos</a>
                </li>
                @auth
                    <li class="nav-item">
                        <a class="nav-link" href="/dashboard">Meus eventos</a>
                    </li>
                    <li class="nav-item">
                        <form action="/logout" method="POST">
                            @csrf
                            <a href="/logout" class="nav-link"
                                onclick="event.preventDefault();
                                this.closest('form').submit();">
                                Sair
                            </a>
                        </form>
                    </li>
                @endauth
                @guest
                    <li class="nav-item">
                        <a class="nav-link" href="/login">Entrar</a>
                    </li>
                @endguest
            </ul>
        </div>
    </nav>
</header>

```

```

<li class="nav-item">
  <a class="nav-link" href="/register">Cadastrar</a>
</li>
</ul>
</div>
</nav>
</header>

<main>
<div class="container-fluid">
<div class="row">
  @if(session('msg'))
    <p class="msg">{{ session('msg') }}</p>
  @endif
  @yield('content')
</div>
</div>
</main>

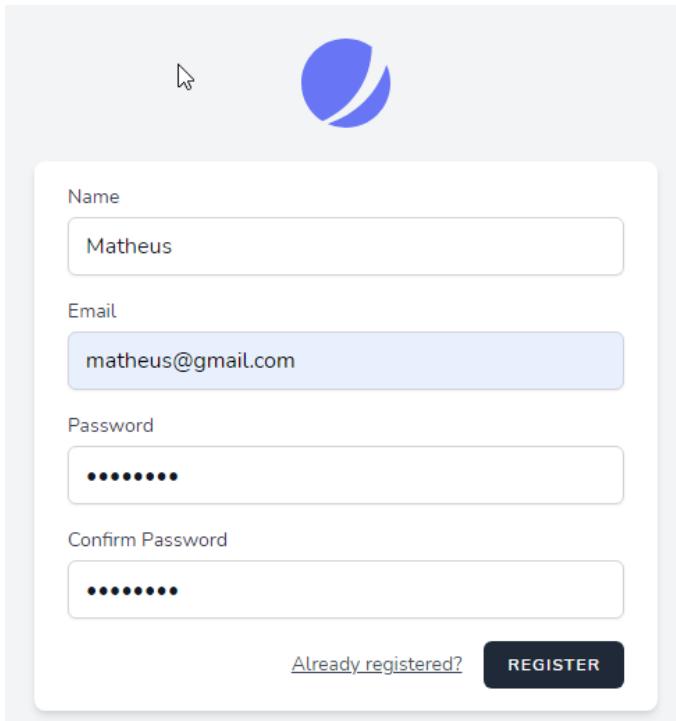
<footer>
  <p>HDC Events © 2021</p>
</footer>

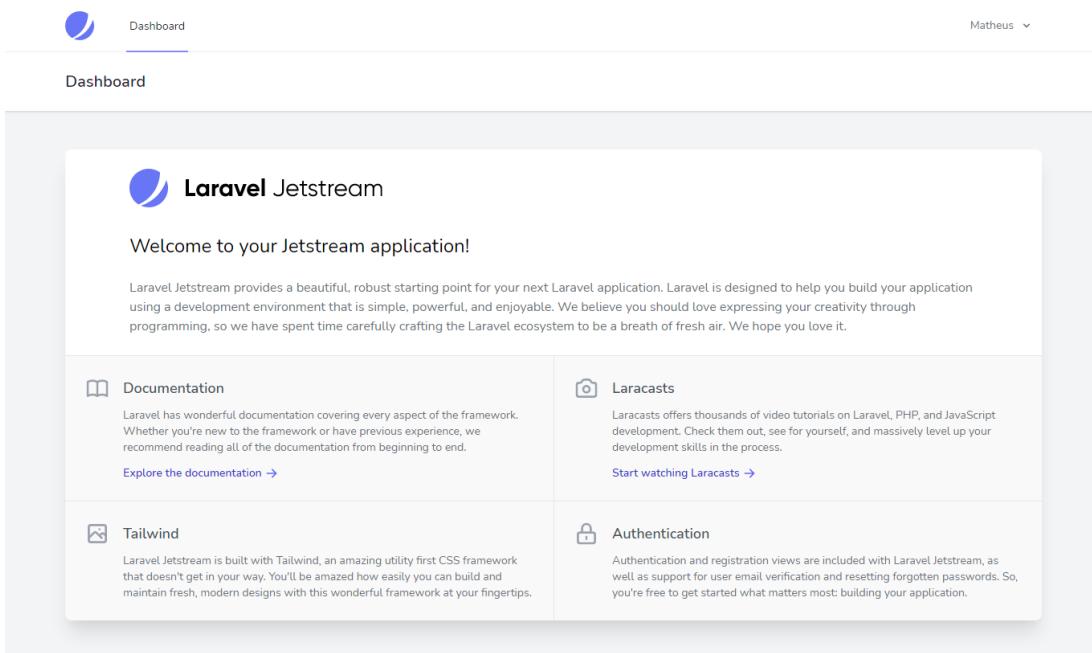
<script src="https://unpkg.com/ionicons@4.5.10-0/dist/ionicons.js"></script>

</body>
</html>

```

Password: 12345678



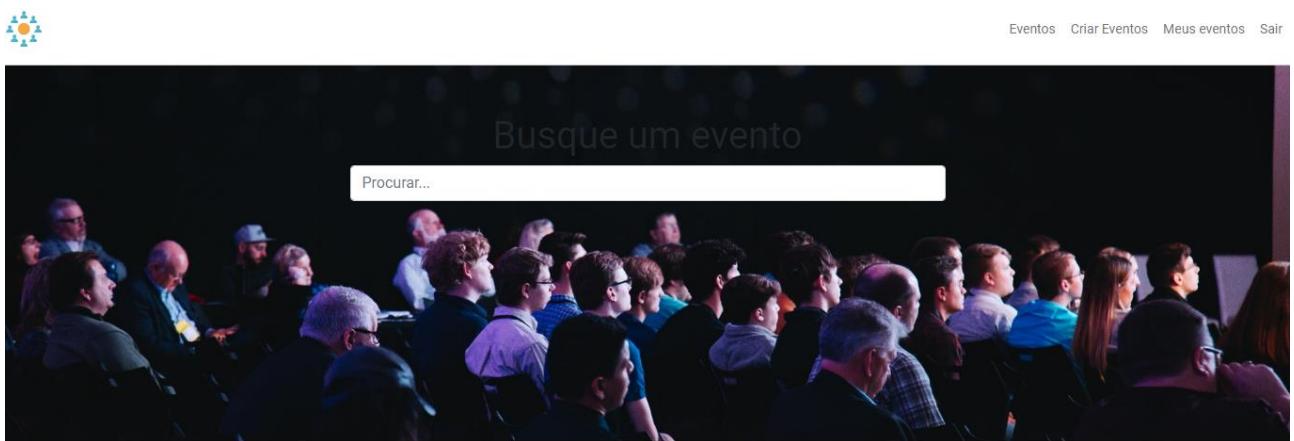


Laravel Jetstream

Welcome to your Jetstream application!

LaravelJetstream provides a beautiful, robust starting point for your next Laravel application. Laravel is designed to help you build your application using a development environment that is simple, powerful, and enjoyable. We believe you should love expressing your creativity through programming, so we have spent time carefully crafting the Laravel ecosystem to be a breath of fresh air. We hope you love it.

 Documentation	 Laracasts
Laravel has wonderful documentation covering every aspect of the framework. Whether you're new to the framework or have previous experience, we recommend reading all of the documentation from beginning to end. Explore the documentation →	Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process. Start watching Laracasts →
 Tailwind	 Authentication
Laravel Jetstream is built with Tailwind, an amazing utility first CSS framework that doesn't get in your way. You'll be amazed how easily you can build and maintain fresh, modern designs with this wonderful framework at your fingertips.	Authentication and registration views are included with Laravel Jetstream, as well as support for user email verification and resetting forgotten passwords. So, you're free to get started what matters most: building your application.



Busque um evento

Procurar...

Eventos Criar Eventos Meus eventos Sair

Próximos eventos

Veja os eventos dos próximos dias



19/12/2019

PHP Conference Brasil 2021

X participantes

[Saber mais](#)



21/08/2015

Brazil JS Conference

X participantes

[Saber mais](#)

Aula 23 - Relação One to Many

- Relações são essenciais para sistemas de **bancos de dados relacionais**;
- Vamos criar uma relação de **um para muitos**, entre usuário e eventos;
- Isso tornará **um usuário dono de um evento** e também a possibilidade dele possuir vários;
- Precisaremos alterar as migrations, adicionando uma chave estrangeira no model **Event**.

```
php artisan make:migration add_user_id_to_events_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration add_user_id_to_events_table
Created Migration: 2021_11_22_133917_add_user_id_to_events_table
```

database\migrations\2021_11_22_094331_create_sessions_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddUserIdToEventsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->foreignId('user_id')->constrained();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('events', function (Blueprint $table) {
            $table->foreignId('user_id')->constrained()->onDelete('cascade');
        });
    }
}
```

php artisan migrate:status

Ran?	Migration	Batch
Yes	2014_10_12_00000_create_users_table	1
Yes	2014_10_12_100000_create_password_resets_table	1
Yes	2014_10_12_200000_add_two_factor_columns_to_users_table	2
Yes	2019_08_19_000000_create_failed_jobs_table	1
Yes	2019_12_14_000001_create_personal_access_tokens_table	1
Yes	2021_11_20_232254_create_events_table	1
Yes	2021_11_21_163304_add_image_to_events_table	1
Yes	2021_11_21_231732_add_items_to_events_table	1
Yes	2021_11_22_003554_add_date_to_events_table	1
Yes	2021_11_22_094331_create_sessions_table	2
No	2021_11_22_133917_add_user_id_to_events_table	

app\Models\Event.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Event extends Model
{
    use HasFactory;

    protected $casts = [
        'items' => 'array'
    ];

    protected $dates = ['date'];

    public function user(){
        return $this->belongsTo('App\Models\User');
    }
}
```

app\Models\User.php

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Fortify\TwoFactorAuthenticatable;
use Laravel\Jetstream\HasProfilePhoto;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
```

```
{
use HasApiTokens;
use HasFactory;
use HasProfilePhoto;
use Notifiable;
use TwoFactorAuthenticatable;

/**
 * The attributes that are mass assignable.
 *
 * @var string[]
 */
protected $fillable = [
    'name',
    'email',
    'password',
];

/**
 * The attributes that should be hidden for serialization.
 *
 * @var array
 */
protected $hidden = [
    'password',
    'remember_token',
    'two_factor_recovery_codes',
    'two_factor_secret',
];

/**
 * The attributes that should be cast.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];

/**
 * The accessors to append to the model's array form.
 *
 * @var array
 */
protected $appends = [
    'profile_photo_url',
];

public function events(){
    return $this->hasMany('App\Models\Event');
}

}
```

php artisan migrate:fresh

```
C:\xampp\htdocs\hdcevents>php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (413.19ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (542.56ms)
Migrating: 2014_10_12_200000_add_two_factor_columns_to_users_table
Migrated: 2014_10_12_200000_add_two_factor_columns_to_users_table (163.98ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (524.34ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (2,270.25ms)
Migrating: 2021_11_20_232254_create_events_table
Migrated: 2021_11_20_232254_create_events_table (230.96ms)
Migrating: 2021_11_21_163304_add_image_to_events_table
Migrated: 2021_11_21_163304_add_image_to_events_table (144.09ms)
Migrating: 2021_11_21_231732_add_items_to_events_table
Migrated: 2021_11_21_231732_add_items_to_events_table (97.83ms)
Migrating: 2021_11_22_003554_add_date_to_events_table
Migrated: 2021_11_22_003554_add_date_to_events_table (87.15ms)
Migrating: 2021_11_22_094331_create_sessions_table
Migrated: 2021_11_22_094331_create_sessions_table (1,426.90ms)
Migrating: 2021_11_22_133917_add_user_id_to_events_table
Migrated: 2021_11_22_133917_add_user_id_to_events_table (1,439.37ms)
```

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Elimina ▾ Mais
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	4 city	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	5 private	tinyint(1)			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	6 created_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	7 updated_at	timestamp			Sim	NULL			Muda Elimina ▾ Mais
<input type="checkbox"/>	8 image	varchar(255)	utf8mb4_unicode_ci		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	9 items	longtext	utf8mb4_bin		Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	10 date	datetime			Não	Nenhum			Muda Elimina ▾ Mais
<input type="checkbox"/>	11 user_id	bigint(20)		UNSIGNED	Não	Nenhum			Muda Elimina ▾ Mais

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {

            $requestImage = $request->image;

            $extension = $requestImage->extension();

            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;

            $requestImage->move(public_path('img/events'), $imageName);

            $event->image = $imageName;
        }

        $user = auth()->user();
        $event->user_id = $user->id;
    }
}

```

```

$event->save();
return redirect('/')->with('msg', 'Evento criado com sucesso!');
}

public function show($id){
    $event = Event::findOrFail($id);
    return view('events.show', ['event' => $event]);
}

}

```

routes\web.php

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);

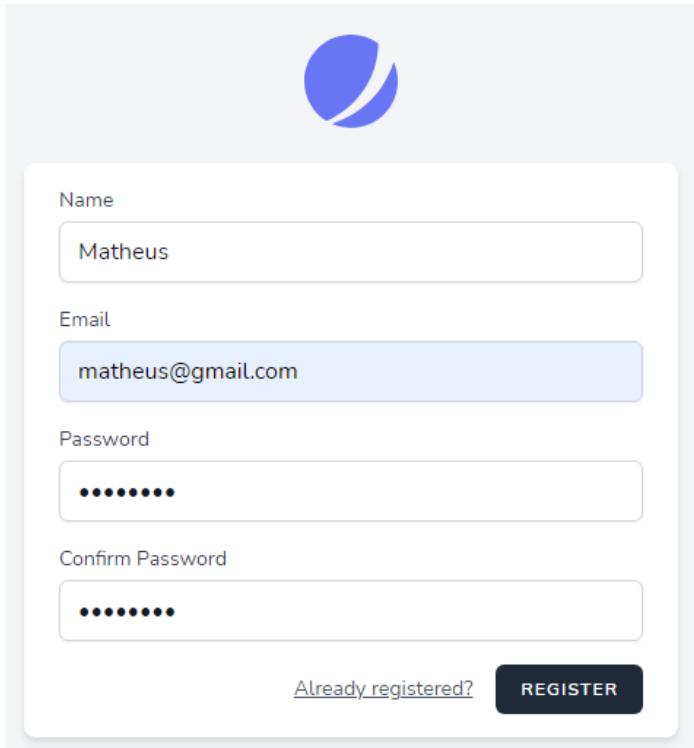
Route::get('/contact', function () {
    return view('contact');
});

Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
    return view('dashboard');
})->name('dashboard');

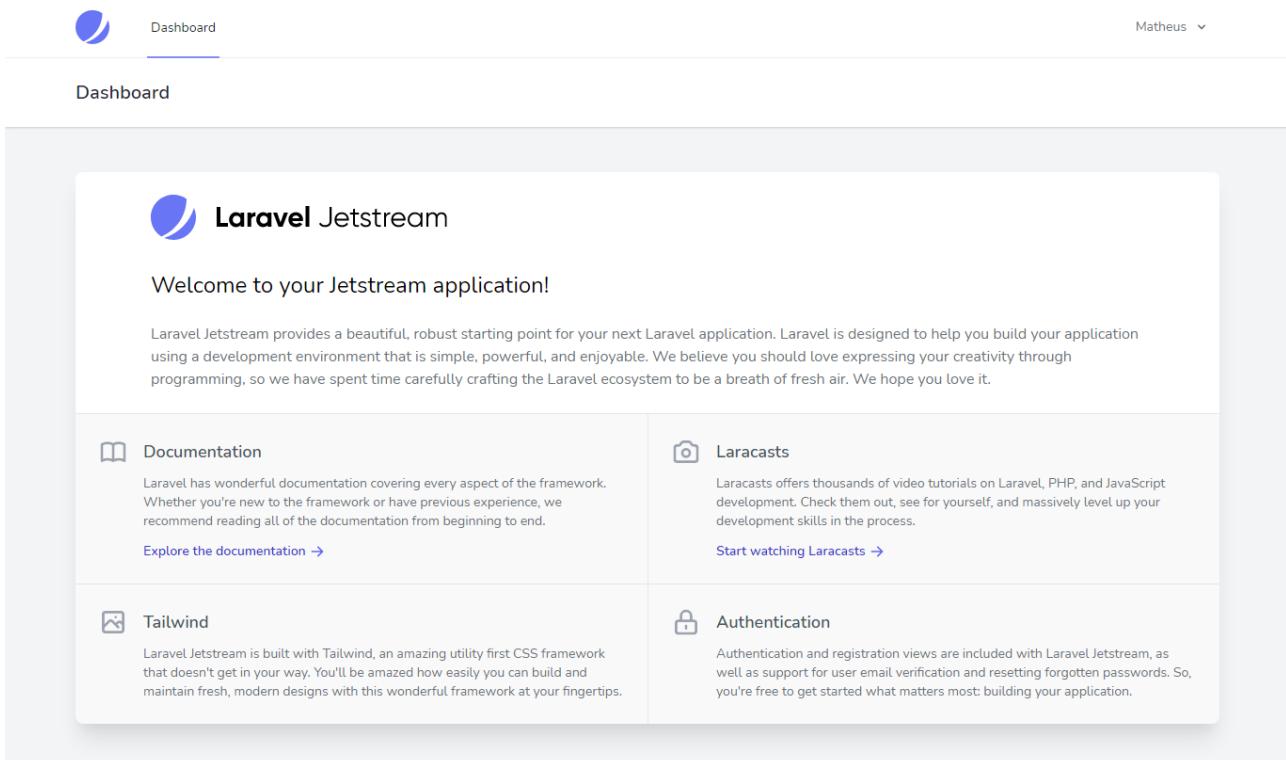
```

<http://127.0.0.1:8000/register>

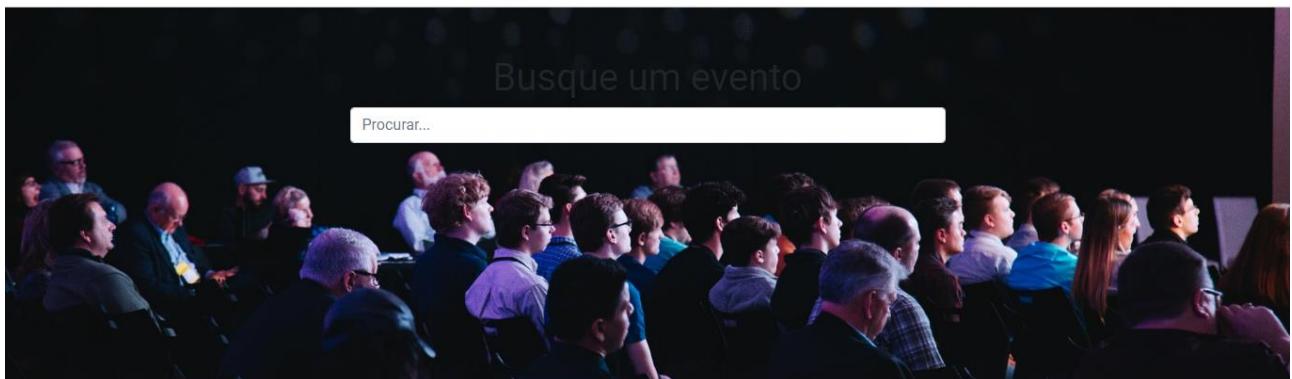
Password: 12345678



The image shows a registration form for a Laravel Jetstream application. At the top is a blue circular logo. Below it are four input fields: 'Name' containing 'Matheus', 'Email' containing 'matheus@gmail.com', 'Password' containing '*****', and 'Confirm Password' also containing '*****'. Below the inputs are two buttons: a blue link-like button 'Already registered?' and a dark blue 'REGISTER' button.



The dashboard page features a header with a profile icon, 'Dashboard', and 'Matheus'. The main content area is titled 'Laravel Jetstream' and includes a welcome message: 'Welcome to your Jetstream application!'. It explains that Laravel Jetstream provides a robust starting point for building Laravel applications. Below this, there are four cards: 'Documentation' (with a book icon), 'Tailwind' (with a photo icon), 'Laracasts' (with a camera icon), and 'Authentication' (with a lock icon). Each card has a brief description and a link to explore further.



Próximos eventos

Veja os eventos dos próximos dias

Não há eventos disponíveis

HDC Events © 2021

- Deslogado, ao tentar criar o evento, o usuário é redirecionado à tela de login.



Crie o seu evento

Imagen do Evento: [Escolher arquivo](#) evento_.php.png

Evento:

PHP Conference Brasil 2021

Data do evento:

19/12/2019



Cidade:

Porto Alegre

O evento é privado?

Não

Descrição:

Será a 16ª edição do evento, que se consagra como o Principal evento de PHP da América Latina

Adicione itens de infra-estrutura

Cadeiras

Palco

Cerveja grátis

Open Food

Brindes

[Criar evento](#)

HDC Events © 2021



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)

Evento criado com sucesso!



Próximos eventos

[Veja os eventos dos próximos dias](#)

A wide-angle photograph of a large lecture hall. The room is filled with rows of blue upholstered chairs, all facing towards a stage area. On the stage, a person, likely a professor, stands behind a podium. The hall has high ceilings and fluorescent lighting. In the background, there are green flags hanging from the ceiling.

19/12/2019

PHP Conference Brasil 2021

X participantes

HDC Events © 2021

Aula 24 - Exibindo dados do usuário na view

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                ['title', 'like', '%' . $search . '%']
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {

            $requestImage = $request->image;
            $extension = $requestImage->extension();

            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;

            $requestImage->move(public_path('img/events'), $imageName);

            $event->image = $imageName;
        }
    }
}
```

```

}

$user = auth()->user();
$event->user_id = $user->id;

$event->save();
return redirect('/')->with('msg','Evento criado com sucesso!');
}

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

}

```

resources\views\events\show.blade.php

```

@extends('layouts.main')

@section('title', $event->title)

@section('content')

<div class="col-md-10 offset-md-1">
    <div class="row">
        <div id="image-container" class="col-md-6">
            title }}"/>
        </div>
        <div id="info-container" class="col-md-6">
            <h1>{{ $event->title }}</h1>
            <p class="event-city"><ion-icon name="pin"></ion-icon> {{ $event->city }}</p>
            <p class="events-participants"><ion-icon name="people"></ion-icon> X Participantes</p>
            <p class="event-owner"><ion-icon name="star-outline"></ion-icon> {{ $eventOwner['name'] }}</p>
            <a href="#" class="btn btn-primary" id="event-submit">Confirmar presença</a>
            <h3>O evento conta com:</h3>
            <ul class="" id="items-list">
                @foreach ($event->items as $item)
                    <li><ion-icon name="play"></ion-icon> <span>{{ $item }}</span></li>
                @endforeach
            </ul>
        </div>
        <div class="col-md-12" id="description-container">
            <h3>Sobre o evento: </h3>
            <p class="event-description">{{ $event->description }}</p>
        </div>
    </div>
</div>

@endsection

```

public\css\styles.css

Inserir:

```
#description-container{  
    margin-top: 30px;  
}
```



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)



PHP Conference Brasil 2021

📍 Porto Alegre
👤 X Participantes
⭐ Matheus

[Confirmar presença](#)

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 16ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021

Aula 25 - Criando uma dashboard

- Agora que um usuário está ligado a um evento podemos criar uma dashboard!
- Ela vai informar todos os eventos que o usuário possui;
- Futuramente também vamos adicionar as ações de edição e de deletar evento nesta view.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/dashboard', [EventController::class, 'dashboard'])->middleware('auth');
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }
}
```

```

public function create()
{
    return view('events.create');
}

public function store(Request $request) {
    $event = new Event;

    $event->title = $request->title;
    $event->date = $request->date;
    $event->city = $request->city;
    $event->private = $request->private;
    $event->description = $request->description;
    $event->items = $request->items;

    // Image Upload
    if($request->hasFile('image') && $request->file('image')->isValid()) {

        $requestImage = $request->image;

        $extension = $requestImage->extension();

        $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;

        $requestImage->move(public_path('img/events'), $imageName);

        $event->image = $imageName;
    }

    $user = auth()->user();
    $event->user_id = $user->id;

    $event->save();
    return redirect('/')->with('msg','Evento criado com sucesso!');
}

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard(){
    $user = auth()->user();
    $events = $user->events;

    return view('events.dashboard', ['events' => $events]);
}
}

```

resources\views\events\dashboard.blade.php

```

@extends('layouts.main')

@section('title', 'Dashboard')

@section('content')


# Meus eventos



@if(count($events) > 0)


| #                       | Nome                                                               | Participantes | Ações                                          |
|-------------------------|--------------------------------------------------------------------|---------------|------------------------------------------------|
| {{ \$loop->index + 1 }} | <a href="/events/{{ \$event-&gt;id }}">{{ \$event-&gt;title }}</a> | 0             | <a href="#">Editar</a> <a href="#">Deletar</a> |


@else


Você ainda não tem eventos, Criar evento


@endif


```

@endsection

public\css\styles.css

Inserir o código:

```

.dashboard-title-container{
    margin-bottom: 30px;
    margin-top: 30px;
}

.dashboard-events-container th{
    /* width: 25%; */
}

```

- Crie mais um evento:

Crie o seu evento

Imagen do Evento: evento_nodejs.png

Evento:
Trilha Node JS

Data do evento:
19/07/2017

Cidade:
São Paulo

O evento é privado?
Não

Descrição:
Node.js ganhou o mundo, e hoje é usado pela pequena startup e pelo grande banco. A plataforma amadureceu, mas segue evoluindo rápido. Venha descobrir as novidades e ver como ela pode agregar na sua carreira.

Adicione itens de infra-estrutura

Cadeiras
 Palco
 Cerveja grátis
 Open Food
 Brindes



Evento criado com sucesso!

Busque um evento

Próximos eventos

Veja os eventos dos próximos dias



19/12/2019

PHP Conference Brasil 2021

X participantes

[Saber mais](#)



19/07/2017

Trilha Node JS

X participantes

[Saber mais](#)

HDC Events © 2021



Meus eventos

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	0	Editar Deletar
2	Trilha Node JS	0	Editar Deletar

HDC Events © 2021

Aula 26 - Deletando dados (CRUD Delete)

Deletando eventos

- Vamos agora **deletar um evento**;
- Utilizaremos uma **nova rota** para isso;
- Outro **verbo HTTP** também será preciso: o **DELETE**;
- E ainda faremos a lógica no nosso **Controller** para esta alteração persistir no banco de dados.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);
Route::delete('/events/{id}', [EventController::class, 'destroy'])->middleware('auth');

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/dashboard', [EventController::class, 'dashboard'])->middleware('auth');
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }
    }
}
```

```

    return view('welcome', ['events' => $events, 'search' => $search]);
}

public function create()
{
    return view('events.create');
}

public function store(Request $request) {
    $event = new Event;

    $event->title = $request->title;
    $event->date = $request->date;
    $event->city = $request->city;
    $event->private = $request->private;
    $event->description = $request->description;
    $event->items = $request->items;

    // Image Upload
    if($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $event->image = $imageName;
    }

    $user = auth()->user();
    $event->user_id = $user->id;

    $event->save();
    return redirect('/')->with('msg', 'Evento criado com sucesso!');
}

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard(){
    $user = auth()->user();
    $events = $user->events;

    return view('events.dashboard', ['events' => $events]);
}

public function destroy($id){
    Event::findOrFail($id)->delete();
    return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}
}

```

resources\views\events\dashboard.blade.php

```

@extends('layouts.main')

@section('title', 'Dashboard')

@section('content')


<h1>Meus eventos</h1>



@if(count($events) > 0)
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">#</th>
                    <th scope="col">Nome</th>
                    <th scope="col">Participantes</th>
                    <th scope="col">Ações</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($events as $event)
                    <tr>
                        <td scope="row">{{ $loop->index + 1 }}</td>
                        <td><a href="/events/{{ $event->id }}"/>{{ $event->title }}</a></td>
                        <td>0</td>
                        <td>
                            <a href="#" class="btn btn-info edit-btn"><ion-icon name="create"></ion-icon> Editar</a>
                            <form action="/events/{{ $event->id }}" method="POST">
                                @csrf
                                @method('delete');
                                <button type="submit" class="btn btn-danger delete-btn"><ion-icon name="trash"></ion-icon>
                                Deletar</button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    @else
        <p>Você ainda não tem eventos, <a href="/events/create">Criar evento</a></p>
    @endif


```

```
</div>

@endsection
```

public\css\styles.css

Inserir o código:

```
.dashboard-events-container form {
    display: inline-block;
}
```

Meus eventos

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	0	Editar ; Deletar
2	Trilha Node JS	0	Editar ; Deletar

HDC Events © 2021

- Exclua o evento 2:

Evento excluído com sucesso!

Meus eventos

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	0	Editar ; Deletar

HDC Events © 2021

Aula 27 - Atualização de dados (CRUD Update)

Editando eventos

- Vamos fazer a lógica de **editar um registro**;
- Preceisaremos criar uma **nova rota e também a view** que mostra o form com os dados preenchidos;
- Além disso será necessário aplicar um outro verbo HTTP: o **PUT**;
- E persistir todas as alterações por meio de uma action no nosso **Controller**.

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);
Route::delete('/events/{id}', [EventController::class, 'destroy'])->middleware('auth');
Route::get('/events/edit/{id}', [EventController::class, 'edit'])->middleware('auth');
Route::put('/events/update/{id}', [EventController::class, 'update'])->middleware('auth');

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/dashboard', [EventController::class, 'dashboard'])->middleware('auth');
```

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                ['title', 'like', '%' . $search . '%']
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {
            $requestImage = $request->image;
            $extension = $requestImage->extension();
            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
            $requestImage->move(public_path('img/events'), $imageName);
            $event->image = $imageName;
        }

        $user = auth()->user();
        $event->user_id = $user->id;

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }
}

```

```

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();
    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard(){
    $user = auth()->user();
    $events = $user->events;

    return view('events.dashboard', ['events' => $events]);
}

public function edit($id){
    $event = Event::findOrFail($id);
    return view('events.edit', ['event' => $event]);
}

public function update(Request $request){
    $data = $request->all();

    // Image Upload
    if ($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName()) . strtotime("now") . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $data['image'] = $imageName;
    }

    Event::findOrFail($request->id)->update($data);
    return redirect('/dashboard')->with('msg', 'Evento editado com sucesso!');
}

public function destroy($id){
    Event::findOrFail($id)->delete();
    return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}

```

resources\views\events\edit.blade.php

```

@extends('layouts.main')

@section('title', 'Editando: ' . $event->title)

@section('content')

<div id="event-create-container" class="col-md-6 offset-md-3">
    <h1>Editando: {{ $event->title }}</h1>
    <form action="/events/update/{{ $event->id }}" method="POST" enctype="multipart/form-data">
        @csrf
        @method('put')
        <div class="form-group">
            <label for="image">Imagen do Evento:</label>
            <input type="file" id="image" name="image" class="form-control-file">
            title }}" class="img-preview">
    
```

```

</div>
<div class="form-group">
    <label for="title">Evento:</label>
    <input type="text" class="form-control" id="title" name="title" placeholder="Nome do evento" value="{{ $event->title }}">
</div>
<div class="form-group">
    <label for="date">Data do evento:</label>
    <input type="date" class="form-control" id="date" name="date" value="{{ $event->date->format('Y-m-d') }}">
</div>
<div class="form-group">
    <label for="city">Cidade:</label>
    <input type="text" class="form-control" id="city" name="city" placeholder="Local do evento" value="{{ $event->city }}">
</div>
<div class="form-group">
    <label for="private">O evento é privado?</label>
    <select class="form-control" name="private" id="private">
        <option value="0">Não</option>
        <option value="1" {{ $event->private == 1 ? "selected='selected'" : "" }}>Sim</option>
    </select>
</div>
<div class="form-group">
    <label for="description">Descrição:</label>
    <textarea class="form-control" name="description" id="description" placeholder="O que vai acontecer no evento?">{{ $event->description }}</textarea>
</div>
<div class="form-group">
    <label for="items">Adicione itens de infra-estrutura</label>
    <div class="form-group">
        <input type="checkbox" name="items[]" value="Cadeiras"> Cadeiras
    </div>
    <div class="form-group">
        <input type="checkbox" name="items[]" value="Palco"> Palco
    </div>
    <div class="form-group">
        <input type="checkbox" name="items[]" value="Cerveja grátis"> Cerveja grátis
    </div>
    <div class="form-group">
        <input type="checkbox" name="items[]" value="Open Food"> Open Food
    </div>
    <div class="form-group">
        <input type="checkbox" name="items[]" value="Brindes"> Brindes
    </div>
</div>
<input class="btn btn-primary" type="submit" value="Editar evento">

</form>
</div>

```

@endsection

resources\views\events\dashboard.blade.php

```

@extends('layouts.main')

@section('title', 'Dashboard')

@section('content')


<h1>Meus eventos</h1>



@if(count($events) > 0)


| #                       | Nome                                                               | Participantes | Ações                                                                                                                                                                                                                                                                                                                         |
|-------------------------|--------------------------------------------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ \$loop->index + 1 }} | <a href="/events/{{ \$event-&gt;id }}">{{ \$event-&gt;title }}</a> | 0             | <a class="btn btn-info edit-btn" href="/events/edit/{{ \$event-&gt;id }}">&lt;ion-icon name="create"&gt;</a> <form action="/events/{{ \$event->id }}" method="POST">     @csrf     @method('delete');     <button type="submit" class="btn btn-danger delete-btn"><ion-icon name="trash"></ion-icon> Deletar</button> </form> |


@else


Você ainda não tem eventos, <a href="/events/create">Criar evento</a>


@endif


```

public\css\styles.css

Insira:

```

/* edit page */
.img-preview{
    width: 200px;
    margin: 20px 0px;
    display: block;
}

```

app\Models\Event.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Event extends Model
{
    use HasFactory;

    protected $casts = [
        'items' => 'array'
    ];

    protected $dates = ['date'];

    protected $guarded = [];

    public function user(){
        return $this->belongsTo('App\Models\User');
    }
}
```

Crie o seu evento

Imagen do Evento: evento_.php.png

Evento:

Data do evento:

Cidade:

O evento é privado?

Descrição:

Adicione Itens de infra-estrutura

Cadeiras
 Palco
 Cerveja grátis
 Open Food
 Brindes

Evento: PHP Conference Brasil 2021

Data do evento: 19/12/2021

Cidade: Porto Alegre

O evento é privado? Não

Descrição: Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

Itens de infra-estrutura: Cadeiras, Palco.



Evento editado com sucesso!

Meus eventos

#	Nome	Participantes	Ações
1	Evento PHP Conference Brasil 2021	0	Editar ; Deletar

HDC Events © 2021



Evento PHP Conference Brasil 2021

📍 Porto Alegre
👥 X Participantes
⭐ Matheus

[Confirmar presença](#)

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021

Aula 28 - Relação many to many

- Vamos aprender uma nova relação de registro pelo laravel: a **many to many**;
- Vamos aplicar esta **lógica a participação de eventos**, onde um usuário pode ter vários eventos e um evento vários participantes;
- Precisaremos **criar uma nova tabela** que administra estas relações, seguindo também a convenção do Laravel.

```
php artisan make:migration create_event_user_table
```

```
C:\xampp\htdocs\hdcevents>php artisan make:migration create_event_user_table
Created Migration: 2021_11_23_041050_create_event_user_table
```

database\migrations\2021_11_23_041050_create_event_user_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateEventUserTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('event_user', function (Blueprint $table) {
            $table->foreignId('event_id')->constrained();
            $table->foreignId('user_id')->constrained();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('event_user');
    }
}
```

app\Models\Event.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Event extends Model
{
    use HasFactory;

    protected $casts = [
        'items' => 'array'
    ];

    protected $dates = ['date'];

    protected $guarded = [];

    public function user(){
        return $this->belongsTo('App\Models\User');
    }

    public function users(){
        return $this->hasMany('App\Models\User');
    }
}
```

app\Models\User.php

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Fortify\TwoFactorAuthenticatable;
use Laravel\Jetstream\HasProfilePhoto;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens;
    use HasFactory;
    use HasProfilePhoto;
    use Notifiable;
    use TwoFactorAuthenticatable;
```

```

/**
 * The attributes that are mass assignable.
 *
 * @var string[]
 */
protected $fillable = [
    'name',
    'email',
    'password',
];

/**
 * The attributes that should be hidden for serialization.
 *
 * @var array
 */
protected $hidden = [
    'password',
    'remember_token',
    'two_factor_recovery_codes',
    'two_factor_secret',
];

/**
 * The attributes that should be cast.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];

/**
 * The accessors to append to the model's array form.
 *
 * @var array
 */
protected $appends = [
    'profile_photo_url',
];

public function events(){
    return $this->hasMany('App\Models\Event');
}

public function eventsAsParticipant()
{
    return $this->belongsToMany('App\Models\Event');
}
}

```

php artisan migrate:status

Ran?	Migration	Batch
Yes	2014_10_12_00000_create_users_table	1
Yes	2014_10_12_100000_create_password_resets_table	1
Yes	2014_10_12_200000_add_two_factor_columns_to_users_table	1
Yes	2019_08_19_000000_create_failed_jobs_table	1
Yes	2019_12_14_000001_create_personal_access_tokens_table	1
Yes	2021_11_20_232254_create_events_table	1
Yes	2021_11_21_163304_add_image_to_events_table	1
Yes	2021_11_21_231732_add_items_to_events_table	1
Yes	2021_11_22_003554_add_date_to_events_table	1
Yes	2021_11_22_094331_create_sessions_table	1
Yes	2021_11_22_133917_add_user_id_to_events_table	1
No	2021_11_23_041050_create_event_user_table	

php artisan migrate

```
C:\xampp\htdocs\hdcevents>php artisan migrate
Migrating: 2021_11_23_041050_create_event_user_table
Migrated: 2021_11_23_041050_create_event_user_table (12,285.18ms)
```

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);
Route::delete('/events/{id}', [EventController::class, 'destroy'])->middleware('auth');
Route::get('/events/edit/{id}', [EventController::class, 'edit'])->middleware('auth');
Route::put('/events/update/{id}', [EventController::class, 'update'])->middleware('auth');

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/dashboard', [EventController::class, 'dashboard'])->middleware('auth');

Route::post('/events/join/{id}', [EventController::class, 'joinEvent'])->middleware('auth');
```

app\Http\Controllers\EventController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {
            $requestImage = $request->image;
            $extension = $requestImage->extension();
            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
            $requestImage->move(public_path('img/events'), $imageName);
            $event->image = $imageName;
        }

        $user = auth()->user();
        $event->user_id = $user->id;

        $event->save();
        return redirect('/')->with('msg', 'Evento criado com sucesso!');
    }
}

```

```

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard(){
    $user = auth()->user();
    $events = $user->events;

    return view('events.dashboard', ['events' => $events]);
}

public function edit($id){
    $event = Event::findOrFail($id);
    return view('events.edit', ['event' => $event]);
}

public function update(Request $request){
    $data = $request->all();

    // Image Upload
    if ($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName()) . strtotime("now") . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $data['image'] = $imageName;
    }

    Event::findOrFail($request->id)->update($data);
    return redirect('/dashboard')->with('msg', 'Evento editado com sucesso!');
}

public function destroy($id){
    Event::findOrFail($id)->delete();
    return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}

public function joinEvent($id){
    $user = auth()->user();
    $user->eventsAsParticipant()->attach($id);

    $event = Event::findOrFail($id);

    return redirect('/dashboard')->with('msg', 'Sua presença está confirmada no evento' . $event->title);
}

```

resources\views\events\show.blade.php

```

@extends('layouts.main')

@section('title', $event->title)

@section('content')



![{{ $event->title }}](/img/events/{{ $event->image }})



# {{ $event->title }}



<ion-icon name="pin"></ion-icon> {{ $event->city }}</p>


<ion-icon name="people"></ion-icon> X Participantes</p>


<ion-icon name="star-outline"></ion-icon> {{ $eventOwner['name'] }}</p>



<form action="/events/join/{{ $event->id }}" method="POST">
    @csrf
    <a href="/events/join/{{ $event->id }}" class="btn btn-primary" id="event-submit"
    onclick="event.preventDefault(); this.closest('form').submit();">Confirmar presença</a>


```

<h3>O evento conta com:</h3>

@foreach (\$event->items as \$item)
 <ion-icon name="play"></ion-icon> {{ \$item }}

@endforeach

</div>

<div class="col-md-12" id="description-container">

<h3>Sobre o evento: </h3>

<p class="event-description">{{ \$event->description }}</p>

</div>

</div>

</div>

@endsection

- Clique no botão "Confirmar presença":



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)



PHP Conference Brasil 2021

Porto Alegre
X Participantes
Matheus

[Confirmar presença](#)

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)

Sua presença está confirmada no evento [PHP Conference Brasil 2021](#)

Meus eventos

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	0	 Editar Deletar

HDC Events © 2021

- Cadastre um novo usuário:

Password: 12345678



Name

Email

Password

Confirm Password

[Already registered?](#)



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)

Meus eventos

Você ainda não tem eventos, [Criar evento](#)

- Confirme a presença desse usuário no evento:



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)



PHP Conference Brasil 2021

📍 Porto Alegre
👤 X Participantes
⭐ Matheus

[Confirmar presença](#)

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)

Sua presença está confirmada no evento **PHP Conference Brasil 2021**

Meus eventos

Você ainda não tem eventos, [Criar evento](#)

HDC Events © 2021

event_id	user_id	created_at	updated_at
1	1	NULL	NULL
1	2	NULL	NULL

resources\views\events\show.blade.php

```

@extends('layouts.main')

@section('title', $event->title)

@section('content')



![{{ $event->title }}](/img/events/{{ $event->image }})



# {{ $event->title }}



<ion-icon name="pin"></ion-icon> {{ $event->city }}</p>


<ion-icon name="people"></ion-icon> {{ count($event->users) }}</p>



Participantes</p>


<p class="event-owner"><ion-icon name="star-outline"></ion-icon> {{ $eventOwner['name'] }}</p>
<form action="/events/join/{{ $event->id }}" method="POST">
    @csrf
    <a href="/events/join/{{ $event->id }}" class="btn btn-primary" id="event-submit"
    onclick="event.preventDefault(); this.closest('form').submit();">Confirmar presença</a>
</form>
<h3>O evento conta com:</h3>
<ul class="" id="items-list">
    @foreach ($event->items as $item)
        <li><ion-icon name="play"></ion-icon> <span>{{ $item }}</span></li>
    @endforeach
</ul>
</div>
<div class="col-md-12" id="description-container">
    <h3>Sobre o evento: </h3>
    <p class="event-description">{{ $event->description }}</p>
</div>
</div>
</div>


```

@endsection

resources\views\welcome.blade.php

```

@extends('layouts.main')

@section('title', 'HDC Events')

@section('content')



# Busque um evento


<form action="/" method="GET">
    <input type="text" id="search" name="search" class="form-control" placeholder="Procurar... ">
</form>
</div>



@if($search)


```

```

<h2>Buscando por {{ $search }}</h2>
@else
<h2>Próximos eventos</h2>
<p class="subtitle">Veja os eventos dos próximos dias</p>
@endif

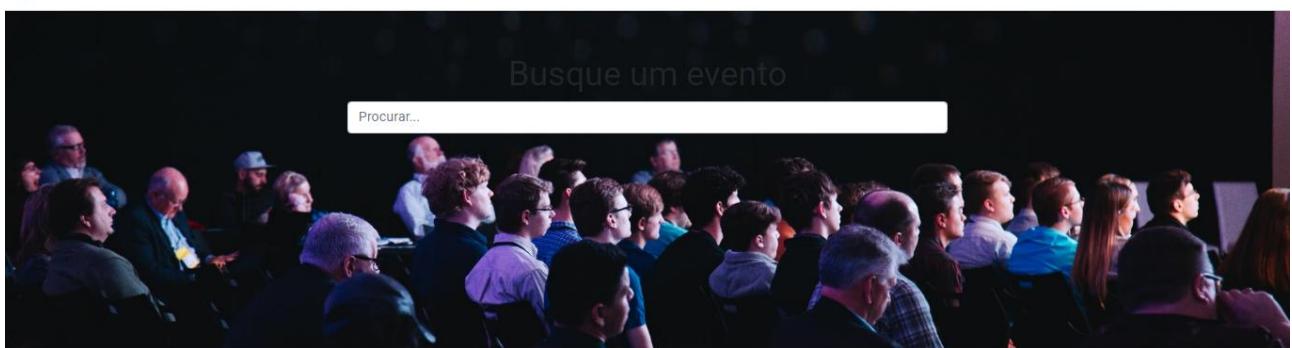
<div id="cards-container" class="row">
@foreach ($events as $event)
<div class="card col-md-3">
title }}">
<div class="card-body">
<p class="card-date">{{ date('d/m/Y', strtotime($event->date)) }}</p>
<h5 class="card-title">{{ $event->title }}</h5>
<p class="card-participants">{{ count($event->users) }} participantes</p>
<a href="/events/{{ $event->id }}" class="btn btn-primary">Saber mais</a>
</div>
</div>
@endforeach
@if(count($events) == 0 && $search)
<p>Não foi possível encontrar nenhum evento com {{ $search }}! <a href="/">Ver todos!</a></p>
@elseif(count($events) == 0)
<p>Não há eventos disponíveis</p>
@endif
</div>
</div>

@endsection

```



Eventos Criar Eventos Meus eventos Sair



Próximos eventos

Veja os eventos dos próximos dias



19/12/2021

PHP Conference Brasil 2021

2 participantes

[Saber mais](#)



PHP Conference Brasil 2021

📍 Porto Alegre
👤 2 Participantes
⭐ Matheus

[Confirmar presença](#)

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021

Aula 29 - Exibindo eventos do usuário

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request) {
        $event = new Event;

        $event->title = $request->title;
        $event->date = $request->date;
        $event->city = $request->city;
        $event->private = $request->private;
        $event->description = $request->description;
        $event->items = $request->items;

        // Image Upload
        if($request->hasFile('image') && $request->file('image')->isValid()) {
            $requestImage = $request->image;
            $extension = $requestImage->extension();
            $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
            $requestImage->move(public_path('img/events'), $imageName);
            $event->image = $imageName;
        }

        $user = auth()->user();
        $event->user_id = $user->id;
    }
}
```

```

$event->save();
return redirect('/')->with('msg', 'Evento criado com sucesso!');
}

public function show($id){
$event = Event::findOrFail($id);
$eventOwner = User::where('id', $event->user_id)->first()->toArray();

return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard()
{
$user = auth()->user();
$events = $user->events;
$eventsAsParticipant = $user->eventsAsParticipant;

return view(
    'events.dashboard',
    ['events' => $events, 'eventsasparticipant' => $eventsAsParticipant]
);
}

public function edit($id){

$user = auth()->user();
$event = Event::findOrFail($id);

if($user->id != $event->user_id){
    return redirect('/dashboard');
}

return view('events.edit', ['event' => $event]);
}

public function update(Request $request){
$data = $request->all();

// Image Upload
if ($request->hasFile('image') && $request->file('image')->isValid()) {
    $requestImage = $request->image;
    $extension = $requestImage->extension();
    $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
    $requestImage->move(public_path('img/events'), $imageName);
    $data['image'] = $imageName;
}

Event::findOrFail($request->id)->update($data);
return redirect('/dashboard')->with('msg', 'Evento editado com sucesso!');
}

public function destroy($id){
Event::findOrFail($id)->delete();
return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}

public function joinEvent($id){
$user = auth()->user();

```

```

$user->eventsAsParticipant()->attach($id);
$event = Event::findOrFail($id);

return redirect('/dashboard')->with('msg', 'Sua presença está confirmada no evento ' . $event->title);
}

}

```

resources\views\events\dashboard.blade.php

```

@extends('layouts.main')

@section('title', 'Dashboard')

@section('content')


# Meus eventos



@if(count($events) > 0)


| col">#                       | Nome                                                               | Participantes               | Ações                                                                                                                                                                                                                                                                                         |
|------------------------------|--------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| row">{{ \$loop->index + 1 }} | <a href="/events/{{ \$event-&gt;id }}">{{ \$event-&gt;title }}</a> | {{ count(\$event->users) }} | <a class="btn btn-info edit-btn" href="/events/edit/{{ \$event-&gt;id }}"><ion-icon name="create"></ion-icon> Editar</a> <form action="/events/{{ \$event-&gt;id }}" method="POST"> @csrf @method('delete'); <input class="btn btn-danger delete-btn" type="submit" value="Deletar"/> </form> |


@else


Você ainda não tem eventos, Criar evento


@endif


```

```

<div class="col-md-10 offset-md-1 dashboard-title-container">
    <h1>Eventos que estou participando</h1>
</div>
<div class="col-md-10 offset-md-1 dashboard-events-container">
    @if(count($eventsasparticipant) > 0):

        <table class="table">
            <thead>
                <tr>
                    <th scope="col">#</th>
                    <th scope="col">Nome</th>
                    <th scope="col">Participantes</th>
                    <th scope="col">Ações</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($eventsasparticipant as $event)
                    <tr>
                        <td scope="row">{{ $loop->index + 1 }}</td>
                        <td><a href="/events/{{ $event->id }}"/>{{ $event->title }}</a></td>
                        <td>{{ count($event->users) }}</td>
                        <td>
                            <a href="#">Sair do evento</a>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>

    @else
        <p>Você ainda não está participando de nenhum evento, <a href="/">Veja todos os
        eventos</a></p>
    @endif
</div>

@endsection

```

- Crie um novo evento:

Crie o seu evento

Imagen do Evento: evento_nodejs.png

Evento:

Trilha Node.JS

Data do evento:

09/12/2021

Cidade:

São Paulo

O evento é privado?

Não

Descrição:

Node.JS ganhou o mundo e hoje é usado pela pequena startup e pelo grande banco. A plataforma amadureceu, mas segue evoluindo rápido. Venha descobrir as novidades e ver como ela pode agregar na sua carreira.

Adicione itens de infra-estrutura

- Cadeiras
- Palco
- Cerveja grátis
- Open Food
- Brindes

Evento: Trilha Node.JS

Data do evento: 09/12/2021

Cidade: São Paulo

O evento é privado? Não

Descrição: Node.JS ganhou o mundo e hoje é usado pela pequena startup e pelo grande banco. A plataforma amadureceu, mas segue evoluindo rápido. Venha descobrir as novidades e ver como ela pode agregar na sua carreira.

Itens de infra-estrutura: Cadeiras, Palco, Brindes.



Evento criado com sucesso!

Busque um evento

Próximos eventos

Veja os eventos dos próximos dias



19/12/2021

PHP Conference Brasil 2021

2 participantes

[Saber mais](#)



09/12/2021

Trilha NODE.JS

0 participantes

[Saber mais](#)

HDC Events © 2021

- Confirme presença:



Trilha Node.JS

São Paulo
1 Participantes
Roberto Pinheiro

[Confirmar presença](#)

O evento conta com:

- Cadeiras
- Palco
- Brindes

Sobre o evento:

Node.js ganhou o mundo e hoje é usado pela pequena startup e pelo grande banco. A plataforma amadureceu, mas segue evoluindo rápido. Venha descobrir as novidades e ver como ela pode agregar na sua carreira.

HDC Events © 2021



Meus eventos

#	Nome	Participantes	Ações
1	Trilha Node.JS	1	 Editar ; Deletar

Eventos que estou participando

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	2	Sair do evento
2	Trilha Node.JS	1	Sair do evento

Aula 30 - Removendo presença do evento

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\EventController;

Route::get('/', [EventController::class, 'index']);
Route::get('/events/create', [EventController::class, 'create'])->middleware('auth');
Route::get('/events/{id}', [EventController::class, 'show']);
Route::post('/events', [EventController::class, 'store']);
Route::delete('/events/{id}', [EventController::class, 'destroy'])->middleware('auth');
Route::get('/events/edit/{id}', [EventController::class, 'edit'])->middleware('auth');
Route::put('/events/update/{id}', [EventController::class, 'update'])->middleware('auth');

Route::get('/contact', function () {
    return view('contact');
});

Route::get('/dashboard', [EventController::class, 'dashboard'])->middleware('auth');

Route::post('/events/join/{id}', [EventController::class, 'joinEvent'])->middleware('auth');

Route::delete('/events/leave/{id}', [EventController::class, 'leaveEvent'])->middleware('auth');
```

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                ['title', 'like', '%' . $search . '%']
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }
}
```

```

}

public function create()
{
    return view('events.create');
}

public function store(Request $request) {
    $event = new Event;

    $event->title = $request->title;
    $event->date = $request->date;
    $event->city = $request->city;
    $event->private = $request->private;
    $event->description = $request->description;
    $event->items = $request->items;

    // Image Upload
    if($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName()) . strtotime("now") . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $event->image = $imageName;
    }

    $user = auth()->user();
    $event->user_id = $user->id;

    $event->save();
    return redirect('/')->with('msg', 'Evento criado com sucesso!');
}

public function show($id){
    $event = Event::findOrFail($id);
    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner]);
}

public function dashboard()
{
    $user = auth()->user();
    $events = $user->events;
    $eventsAsParticipant = $user->eventsAsParticipant;

    return view(
        'events.dashboard',
        ['events' => $events, 'eventsasparticipant' => $eventsAsParticipant]
    );
}

public function edit($id){

    $user = auth()->user();
    $event = Event::findOrFail($id);

    if($user->id != $event->user_id){
        return redirect('/dashboard');
    }
}

```

```

}

return view('events.edit', ['event' => $event]);
}

public function update(Request $request){
    $data = $request->all();

    // Image Upload
    if ($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $data['image'] = $imageName;
    }

    Event::findOrFail($request->id)->update($data);
    return redirect('/dashboard')->with('msg', 'Evento editado com sucesso!');
}

public function destroy($id){
    Event::findOrFail($id)->delete();
    return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}

public function joinEvent($id){
    $user = auth()->user();
    $user->eventsAsParticipant()->attach($id);

    $event = Event::findOrFail($id);

    return redirect('/dashboard')->with('msg', 'Sua presença está confirmada no evento ' . $event->title);
}

public function leaveEvent($id)
{
    $user = auth()->user();
    $user->eventsAsParticipant()->detach($id);
    $event = Event::findOrFail($id);

    return redirect('/dashboard')->with('msg', 'Você saiu com sucesso do evento: ' . $event->title);
}
}

```

resources\views\events\dashboard.blade.php

```

@extends('layouts.main')

@section('title', 'Dashboard')

@section('content')


<h1>Meus eventos</h1>



@if(count($events) > 0)
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Nome</th>
                <th scope="col">Participantes</th>
                <th scope="col">Ações</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($events as $event)
                <tr>
                    <td scope="row">{{ $loop->index + 1 }}</td>
                    <td><a href="/events/{{ $event->id }}">{{ $event->title }}</a></td>
                    <td>{{ count($event->users) }}</td>
                    <td>
                        <a href="/events/edit/{{ $event->id }}" class="btn btn-info edit-btn"><ion-icon name="create"></ion-icon> Editar</a>
                        <form action="/events/{{ $event->id }}" method="POST">
                            @csrf
                            @method('delete');
                            <button type="submit" class="btn btn-danger delete-btn"><ion-icon name="trash"></ion-icon> Deletar</button>
                        </form>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
@else
    <p>Você ainda não tem eventos, <a href="/events/create">Criar evento</a></p>
@endif



<h1>Eventos que estou participando</h1>



@if(count($eventsasparticipant) > 0):
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Nome</th>
                <th scope="col">Participantes</th>
                <th scope="col">Ações</th>
            </tr>
        </thead>
        <tbody>


```

```

</tr>
</thead>
<tbody>
    @foreach ($eventsasparticipant as $event)
        <tr>
            <td scope="row">{{ $loop->index + 1 }}</td>
            <td><a href="/events/{{ $event->id }}">{{ $event->title }}</a></td>
            <td>{{ count($event->users) }}</td>
            <td>
                <form action="/events/leave/{{ $event->id }}" method="POST">
                    @csrf
                    @method("DELETE")
                    <button type="submit" class="btn btn-danger delete-btn">
                        <ion-icon name="trash"></ion-icon>
                        Sair do Evento
                    </button>
                </form>
            </td>
        </tr>
    @endforeach
</tbody>
</table>

@else
    <p>Você ainda não está participando de nenhum evento, <a href="/">Veja todos os eventos</a></p>
@endif
</div>

@endsection

```

The screenshot shows a user profile page with the following sections:

- Meus eventos**: A table listing the user's events, showing one event titled "Trilha Node JS" with 1 participant.
- Eventos que estou participando**: A table listing the user's participated events, showing two events: "PHP Conference Brasil 2021" (2 participants) and "Trilha Node JS" (1 participant), each with a "Sair do Evento" button.

At the bottom, a footer bar displays the text "HDC Events © 2021".

#	Nome	Participantes	Ações
1	Trilha Node JS	1	Editar ; Deletar

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	2	Sair do Evento
2	Trilha Node JS	1	Sair do Evento

- Sair do evento Trilha Node.JS:

#	Nome	Participantes	Ações
1	Trilha Node.JS	0	Editar ; Deletar

#	Nome	Participantes	Ações
1	PHP Conference Brasil 2021	2	Sair do Evento

HDC Events © 2021

app\Http\Controllers\EventController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Event;
use App\Models\User;

class EventController extends Controller
{
    public function index()
    {
        $search = request('search');

        if($search){
            $events = Event::where([
                'title', 'like', '%' . $search . '%'
            ])->get();
        } else {
            $events = Event::all();
        }

        return view('welcome', ['events' => $events, 'search' => $search]);
    }

    public function create()
    {
        return view('events.create');
    }

    public function store(Request $request)
    {
        $event = new Event;
    }
}
```

```

$event->title = $request->title;
$event->date = $request->date;
$event->city = $request->city;
$event->private = $request->private;
$event->description = $request->description;
$event->items = $request->items;

// Image Upload
if($request->hasFile('image') && $request->file('image')->isValid()) {
    $requestImage = $request->image;
    $extension = $requestImage->extension();
    $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
    $requestImage->move(public_path('img/events'), $imageName);
    $event->image = $imageName;
}

$user = auth()->user();
$event->user_id = $user->id;

$event->save();
return redirect('/')->with('msg','Evento criado com sucesso!');
}

public function show($id)
{
    $event = Event::findOrFail($id);

    $user = auth()->user();
    $hasUserJoined = false;

    if ($user) {

        $userEvents = $user->eventsAsParticipant->toArray();

        foreach ($userEvents as $userEvent) {
            if ($userEvent['id'] == $id) {
                $hasUserJoined = true;
            }
        }
    }

    $eventOwner = User::where('id', $event->user_id)->first()->toArray();

    return view('events.show', ['event' => $event, 'eventOwner' => $eventOwner, 'hasUserJoined' => $hasUserJoined]);
}

public function dashboard()
{
    $user = auth()->user();
    $events = $user->events;
    $eventsAsParticipant = $user->eventsAsParticipant;

    return view(
        'events.dashboard',
        ['events' => $events, 'eventsasparticipant' => $eventsAsParticipant]
    );
}

```

```

public function edit($id){

    $user = auth()->user();
    $event = Event::findOrFail($id);

    if($user->id != $event->user_id){
        return redirect('/dashboard');
    }

    return view('events.edit', ['event' => $event]);
}

public function update(Request $request){
    $data = $request->all();

    // Image Upload
    if ($request->hasFile('image') && $request->file('image')->isValid()) {
        $requestImage = $request->image;
        $extension = $requestImage->extension();
        $imageName = md5($requestImage->getClientOriginalName() . strtotime("now")) . "." . $extension;
        $requestImage->move(public_path('img/events'), $imageName);
        $data['image'] = $imageName;
    }

    Event::findOrFail($request->id)->update($data);
    return redirect('/dashboard')->with('msg', 'Evento editado com sucesso!');
}

public function destroy($id){
    Event::findOrFail($id)->delete();
    return redirect('/dashboard')->with('msg', 'Evento excluído com sucesso!');
}

public function joinEvent($id){
    $user = auth()->user();
    $user->eventsAsParticipant()->attach($id);

    $event = Event::findOrFail($id);

    return redirect('/dashboard')->with('msg', 'Sua presença está confirmada no evento ' . $event->title);
}

public function leaveEvent($id)
{
    $user = auth()->user();
    $user->eventsAsParticipant()->detach($id);
    $event = Event::findOrFail($id);

    return redirect('/dashboard')->with('msg', 'Você saiu com sucesso do evento: ' . $event->title);
}

```

resources\views\events\show.blade.php

```

@extends('layouts.main')

@section('title', $event->title)

@section('content')



![{{ $event->title }}](/img/events/{{ $event->image }})



# {{ $event->title }}



<ion-icon name="pin"></ion-icon> {{ $event->city }}</p>


<ion-icon name="people"></ion-icon> {{ count($event->users) }}</p>



Participantes</p>


<ion-icon name="star-outline"></ion-icon> {{ $eventOwner['name'] }}</p>
@if(!$hasUserJoined)
<form action="/events/join/{{ $event->id }}" method="POST">
@csrf
Você já está participando deste evento!</p>
@endif
<h3>O evento conta com:</h3>
<ul class="" id="items-list">
@foreach \(\$event->items as \$item\)
<li><ion-icon name="play"></ion-icon> <span>{{ \$item }}</span></li>
@endforeach
</ul>
</div>
<div class="col-md-12" id="description-container">
<h3>Sobre o evento: </h3>
<p class="event-description">{{ \$event->description }}</p>
</div>
</div>
</div>

@endsection


```

public\css\styles.css

```

/* General */

* {
  font-family: 'Roboto';
}

body{
  min-height: 100vh;
  padding-bottom: 100px;
  position: relative;
}

header{
  background-color: #fff;
  border-bottom: 1px solid #ccc;
}

.row{
  margin: 0;
}

.container-fluid{
  padding: 0;
}

.btn-primary{
  background-color: #f2a340;
  border: 2px solid #c28008
}

.btn-primary:hover{
  background-color: #c28008;
  border-color: #c28008;
}

.msg{
  background-color: #d4edda;
  color: #155724;
  border: 1px solid #c3e6cb;
  width: 100%;
  margin-bottom: 0px;
  text-align: center;
  padding: 10px;
}

#info-container .already-joined-msg{
  margin: 15px 0;
  color: #004085;
  background-color: #cce5ff;
  border: 1px solid #b8daff;
  padding: 10px;
  width: 275px;
}

/* navbar */

```

```
#navbar img{
    width: 40px;
}

#navbar{
    justify-content: space-between;
    background-color: #fff;
    padding: 5px 0;
}

#navbar a{
    font-size: 14px;
    color: #757575;
}

#navbar a:hover{
    color: #f2a340;
}

/* Search Container */

#search-container{
    background-image: url('../img/banner.jpg');
    background-size: cover;
    background-position: center;
    height: 400px;
    padding: 50px;
    text-align: center;
}

#search-container form{
    width: 50%;
    margin: 0 auto;
}

/* Events list */

#events-container{
    padding: 50px;
}

#events-container h2{
    margin-bottom: 10px;
}

#events-container .subtitle{
    color: #757575;
    margin-bottom: 30px;
}

#cards-container{
    display: flex;
}

#events-container .card{
    flex: 1 1 24%;
    max-width: 25%;
    border-radius: 10px;
    padding: 0;
```

```
margin: .5%;  
}  
  
#events-container img{  
    max-height: 150px;  
    border-top-left-radius: 10px;  
    border-top-right-radius: 10px;  
}  
  
.card .card-date{  
    font-size: 12px;  
    color: #555;  
}  
  
.card .card-title{  
    color: #212121;  
    font-size: 20px;  
    font-weight: bold;  
    margin-bottom: 30px;  
}  
  
.card-participants{  
    font-size: 14px;  
    color: #757575;  
}  
  
/* Dashboard */  
  
.dashboard-title-container{  
    margin-bottom: 30px;  
    margin-top: 30px;  
}  
  
.dashboard-events-container th{  
    /* width: 25%; */  
}  
  
.dashboard-events-container form {  
    display: inline-block;  
}  
  
/* edit page */  
  
.img-preview{  
    width: 200px;  
    margin: 20px 0px;  
    display: block;  
}  
  
/* footer */  
  
footer{  
    text-align: center;  
    background-color: #353e48;  
    color: #fff;  
    padding: 30px;  
    position: absolute;  
    bottom: 0;  
    width: 100%;
```

```

}

/* Event create page */

#event-create-container{
  padding: 30px;
}

#event-create-container label{
  font-weight: bold;
}

#event-create-container input,
#event-create-container select,
#event-create-container textarea
{
  font-size: 12px;
  margin-bottom: 10px;
}

/* Event page */

#image-container,
#info-container{
  margin-top: 30px;
}

#image-container img{
  border-radius: 10px;
}

#info-container h1{
  font-size: 36px;
  font-weight: 900;
}

#info-container p{
  margin: 0;
  font-size: 14px;
  display: flex;
}

#info-container h3,
#description-container h3{
  font-size: 20px;
  font-weight: bold;
}

#info-container ion-icon{
  font-size: 20px;
  color: #f2a340;
  margin-right: 5px;
}

#info-container #event-submit{
  margin: 20px 0;
}

#items-list{

```

```
list-style: none;
padding-left: 0;
}

#items-list li{
  display: flex;
}

#description-container{
  margin-top: 30px;
}

footer p{
  margin-bottom: 0;
}
```



[Eventos](#) [Criar Eventos](#) [Meus eventos](#) [Sair](#)



PHP Conference Brasil 2021

📍 Porto Alegre
👤 2 Participantes
⭐ Matheus

Você já está participando deste evento!

O evento conta com:

- ▶ Cadeiras
- ▶ Palco

Sobre o evento:

Será a 18ª edição do evento, que se consagra como o principal evento de PHP da América Latina.

HDC Events © 2021