# Passo a Passo Detalhado — Laravel Maps App (OpenStreetMap + Leaflet)

**Este documento apresenta um passo a passo detalhado e explicado para criar um projeto em Laravel que utiliza um banco MySQL e integra mapas com OpenStreetMap (via Leaflet)**. O foco principal está em explicar como usar o OpenStreetMap, suas APIs (tiles e Nominatim) e boas práticas para desenvolvimento e produção.

## 1) Pré-requisitos

Antes de começar, verifique que você tem:

* XAMPP (Apache + MySQL) ou outro ambiente com MySQL instalado.
* Composer (para instalar Laravel).
* PHP 8.2+ recomendado (algumas versões mais recentes do Laravel exigem PHP 8.2 ou superior).
* Node.js e npm (opcional — usamos CDNs, mas útil para asset build).
* MySQL Workbench ou phpMyAdmin para administrar o banco.

## 2) Criar o projeto Laravel

Abra o terminal e execute os comandos abaixo. Neste exemplo usamos o XAMPP (Windows) e o diretório htdocs:

cd C:\xampp\htdocs  
composer create-project laravel/laravel laravel-maps-app  
cd laravel-maps-app

Aguarde a instalação do Laravel e, ao final, abra o diretório do projeto.

## 3) Configurar o ambiente (.env)

Copie o arquivo de exemplo e gere a chave da aplicação (Se nao tiver sido criada):

copy .env.example .env # Windows  
php artisan key:generate

Edite o arquivo .env e configure os dados do banco MySQL criado no passo seguinte (DB\_DATABASE, DB\_USERNAME, DB\_PASSWORD).

## 4) Criar o banco de dados (MySQL)

Abra o MySQL Workbench ou phpMyAdmin e crie um banco de dados. Exemplo SQL:

CREATE DATABASE laravel\_maps\_app CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

No arquivo .env, configure:

DB\_CONNECTION=mysql  
DB\_HOST=127.0.0.1  
DB\_PORT=3306  
DB\_DATABASE=laravel\_maps\_app  
DB\_USERNAME=root  
DB\_PASSWORD=

## 5) Criar migration para a tabela de locations

Crie a migration:

php artisan make:migration create\_locations\_table --create=locations

Edite o arquivo de migration em database/migrations/ 0001\_01\_01\_000000\_create\_users\_table e ajuste o método up():

public function up(): void  
{  
 Schema::create('locations', function (Blueprint $table) {  
 $table->id();  
 $table->string('name');  
 $table->text('description')->nullable();  
 $table->decimal('latitude', 10, 8);  
 $table->decimal('longitude', 11, 8);  
 $table->string('address')->nullable();  
 $table->string('category')->nullable();  
 $table->timestamps();  
 });  
}

Observação: é comum usar decimal(10,7) para lat/lng — os valores acima seguem sua sugestão e funcionam normalmente.

## 6) Executar migration

Rode o comando para criar a tabela no banco:

php artisan migrate

## 7) Criar modelo (Model) Location

Crie o model com comando:

php artisan make:model Location

No arquivo app/Models/Location.php, defina os campos fillable e casts (para garantir que lat/lng saiam como números no JSON):

class Location extends Model  
{  
 protected $fillable = ['name','description','latitude','longitude','address','category'];  
  
 protected $casts = [  
 'latitude' => 'float',  
 'longitude' => 'float',  
 ];  
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\*Model*;

*class* Location extends *Model*

{

        protected $fillable = ['name','description','latitude','longitude','address','category'];

    protected $casts = [

        'latitude' => 'float',

        'longitude' => 'float',

    ];

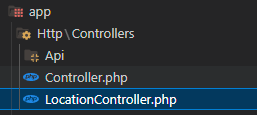
}

## 8) Criar controllers

Crie o controller principal (resource) e o controller da API:

php artisan make:controller LocationController --resource  
php artisan make:controller Api/LocationController

Implemente os métodos (index, create, store, show, edit, update, destroy) no LocationController conforme seu roteiro. No controller da API implemente os endpoints REST (index, store, show, update, destroy).



LocationController:  
<?php

namespace App\Http\Controllers;

use App\Models\*Location*;

use Illuminate\Http\*Request*;

*class* LocationController extends *Controller*

{

    /\*\*

     \* Display a listing of the resource.

     \*/

    public *function* index()

    {

        $locations = *Location*::all();

        return view('locations.index', compact('locations'));

    }

    /\*\*

     \* Show the form for creating a new resource.

     \*/

    public *function* create()

    {

        return view('locations.create');

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public *function* store(*Request* $request)

    {

        $request->validate([

            'name' => 'required|string|max:255',

            'description' => 'nullable|string',

            'latitude' => 'required|numeric|between:-90,90',

            'longitude' => 'required|numeric|between:-180,180',

            'address' => 'nullable|string|max:255',

            'category' => 'nullable|string|max:100'

        ]);

*Location*::create($request->all());

        return redirect()->route('locations.index')

            ->with('success', 'Local criado com sucesso!');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public *function* show(string $id)

    {

        $location = *Location*::findOrFail($id);

        return view('locations.show', compact('location'));

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*/

    public *function* edit(string $id)

    {

        $location = *Location*::findOrFail($id);

        return view('locations.edit', compact('location'));

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public *function* update(*Request* $request, string $id)

    {

        $request->validate([

            'name' => 'required|string|max:255',

            'description' => 'nullable|string',

            'latitude' => 'required|numeric|between:-90,90',

            'longitude' => 'required|numeric|between:-180,180',

            'address' => 'nullable|string|max:255',

            'category' => 'nullable|string|max:100'

        ]);

        $location = *Location*::findOrFail($id);

        $location->update($request->all());

        return redirect()->route('locations.index')

            ->with('success', 'Local atualizado com sucesso!');

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public *function* destroy(string $id)

    {

        $location = *Location*::findOrFail($id);

        $location->delete();

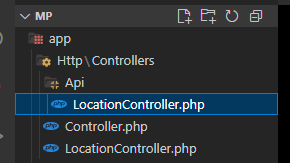
        return redirect()->route('locations.index')

            ->with('success', 'Local excluído com sucesso!');

    }

}

Agora em LocationController dentro de API



<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\*Controller*;

use App\Models\*Location*;

use Illuminate\Http\*Request*;

*class* LocationController extends *Controller*

{

    public *function* index()

    {

        $locations = *Location*::all();

        return response()->json($locations);

    }

    public *function* store(*Request* $request)

    {

        $request->validate([

            'name' => 'required|string|max:255',

            'description' => 'nullable|string',

            'latitude' => 'required|numeric|between:-90,90',

            'longitude' => 'required|numeric|between:-180,180',

            'address' => 'nullable|string|max:255',

            'category' => 'nullable|string|max:100'

        ]);

        $location = *Location*::create($request->all());

        return response()->json($location, 201);

    }

    public *function* show($id)

    {

        $location = *Location*::findOrFail($id);

        return response()->json($location);

    }

    public *function* update(*Request* $request, $id)

    {

        $request->validate([

            'name' => 'required|string|max:255',

            'description' => 'nullable|string',

            'latitude' => 'required|numeric|between:-90,90',

            'longitude' => 'required|numeric|between:-180,180',

            'address' => 'nullable|string|max:255',

            'category' => 'nullable|string|max:100'

        ]);

        $location = *Location*::findOrFail($id);

        $location->update($request->all());

        return response()->json($location);

    }

    public *function* destroy($id)

    {

        $location = *Location*::findOrFail($id);

        $location->delete();

        return response()->json(null, 204);

    }

}

## 9) Trecho exemplo: store (LocationController) Verificar

public function store(Request $request)  
{  
 $request->validate([  
 'name' => 'required|string|max:255',  
 'latitude' => 'required|numeric|between:-90,90',  
 'longitude' => 'required|numeric|between:-180,180',  
 ]);  
  
 Location::create($request->all());  
  
 return redirect()->route('locations.index')->with('success','Local criado com sucesso!');  
}

## 10) Rotas (web e api)

routes/web.php:

Route::get('/', function() { return redirect()->route('locations.index'); });  
Route::resource('locations', LocationController::class);

<?php

use App\Http\Controllers\*LocationController*;

use Illuminate\Support\Facades\*Route*;

*Route*::get('/', *function* () {

    return view('welcome');

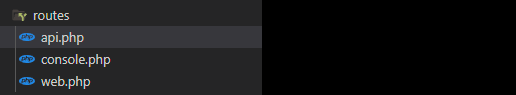
});

*Route*::get('/', *function*() { return redirect()->route('locations.index'); });

*Route*::resource('locations', *LocationController*::class);

routes/api.php (API REST):

Route::apiResource('locations', App\Http\Controllers\Api\LocationController::class);



<?php

use Illuminate\Http\*Request*;

use Illuminate\Support\Facades\*Route*;

use App\Http\Controllers\Api\*LocationController*;

*Route*::get('/user', *function* (*Request* $request) {

    return $request->user();

})->middleware('auth:sanctum');

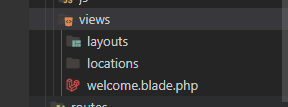
*Route*::apiResource('locations', *LocationController*::class);

## 11) Views principais (layouts + index)

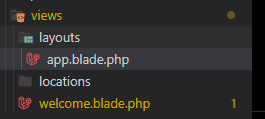
Você pode usar Bootstrap + Leaflet via CDN (sem build). No layout acerte as referências para Leaflet CSS/JS e para os assets do Bootstrap.

No index (resources/views/locations/index.blade.php) carregue a variável $locations e adicione o mapa (div #map). Em scripts, inicialize o Leaflet (exemplo no próximo tópico).

Dentro de views criar as pastas layouts, locations



Dentro da pasta layouts criar o arquivo app.blade.php



app.blade.php

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>@yield('title', 'Laravel Maps App')</title>

    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Leaflet CSS -->

    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />

    <!-- Font Awesome -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <style>

        #map {

*height*: 500px;

*width*: 100%;

*border-radius*: 8px;

        }

        .location-card {

*transition*: transform 0.2s;

        }

        .location-card:hover {

*transform*: translateY(-2px);

        }

    </style>

    @yield('styles')

</head>

<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">

        <div class="container">

            <a class="navbar-brand" href="{{ route('locations.index') }}">

                <i class="fas fa-map-marker-alt me-2"></i>

                Laravel Maps App

            </a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarNav">

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('locations.index') }}">

                            <i class="fas fa-list me-1"></i>Locais

                        </a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="{{ route('locations.create') }}">

                            <i class="fas fa-plus me-1"></i>Adicionar Local

                        </a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <main class="container my-4">

        @if(session('success'))

            <div class="alert alert-success alert-dismissible fade show" role="alert">

                {{ session('success') }}

                <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

            </div>

        @endif

        @if(session('error'))

            <div class="alert alert-danger alert-dismissible fade show" role="alert">

                {{ session('error') }}

                <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

            </div>

        @endif

        @yield('content')

    </main>

    <!-- Bootstrap JS -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

    <!-- Leaflet JS -->

    <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

    @yield('scripts')

</body>

</html>

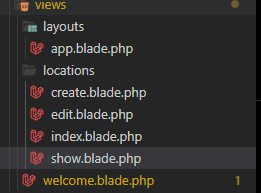
Dentro da pasta views locations criar os arquivos:

create.blade.php

edit.blade.php

index.blade.php

show.blade.php



create.blade.php

@extends('layouts.app')

@section('title', 'Adicionar Local - Laravel Maps App')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h1><i class="fas fa-plus me-2"></i>Adicionar Novo Local</h1>

            <a href="{{ route('locations.index') }}" class="btn btn-outline-secondary">

                <i class="fas fa-arrow-left me-1"></i>Voltar

            </a>

        </div>

    </div>

</div>

<div class="row">

    <!-- Mapa para seleção de coordenadas -->

    <div class="col-md-8">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-map me-2"></i>Selecionar Localização

                </h5>

            </div>

            <div class="card-body">

                <div id="map" style="height: 400px;"></div>

                <div class="mt-3">

                    <small class="text-muted">

                        <i class="fas fa-info-circle me-1"></i>

                        Clique no mapa para selecionar as coordenadas do local

                    </small>

                </div>

            </div>

        </div>

    </div>

    <!-- Formulário -->

    <div class="col-md-4">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-edit me-2"></i>Informações do Local

                </h5>

            </div>

            <div class="card-body">

                <form action="{{ route('locations.store') }}" method="POST">

                    @csrf

                    <div class="mb-3">

                        <label for="name" class="form-label">Nome do Local \*</label>

                        <input type="text" class="form-control @error('name') is-invalid @enderror"

                               id="name" name="name" value="{{ old('name') }}" required>

                        @error('name')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="mb-3">

                        <label for="description" class="form-label">Descrição</label>

                        <textarea class="form-control @error('description') is-invalid @enderror"

                                  id="description" name="description" rows="3">{{ old('description') }}</textarea>

                        @error('description')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="row">

                        <div class="col-6">

                            <div class="mb-3">

                                <label for="latitude" class="form-label">Latitude \*</label>

                                <input type="number" step="any" class="form-control @error('latitude') is-invalid @enderror"

                                       id="latitude" name="latitude" value="{{ old('latitude') }}" required>

                                @error('latitude')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                        <div class="col-6">

                            <div class="mb-3">

                                <label for="longitude" class="form-label">Longitude \*</label>

                                <input type="number" step="any" class="form-control @error('longitude') is-invalid @enderror"

                                       id="longitude" name="longitude" value="{{ old('longitude') }}" required>

                                @error('longitude')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                    </div>

                    <div class="mb-3">

                        <label for="address" class="form-label">Endereço</label>

                        <input type="text" class="form-control @error('address') is-invalid @enderror"

                               id="address" name="address" value="{{ old('address') }}">

                        @error('address')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="mb-3">

                        <label for="category" class="form-label">Categoria</label>

                        <select class="form-select @error('category') is-invalid @enderror" id="category" name="category">

                            <option value="">Selecione uma categoria</option>

                            <option value="Restaurante" {{ old('category') == 'Restaurante' ? 'selected' : '' }}>Restaurante</option>

                            <option value="Hotel" {{ old('category') == 'Hotel' ? 'selected' : '' }}>Hotel</option>

                            <option value="Shopping" {{ old('category') == 'Shopping' ? 'selected' : '' }}>Shopping</option>

                            <option value="Parque" {{ old('category') == 'Parque' ? 'selected' : '' }}>Parque</option>

                            <option value="Museu" {{ old('category') == 'Museu' ? 'selected' : '' }}>Museu</option>

                            <option value="Teatro" {{ old('category') == 'Teatro' ? 'selected' : '' }}>Teatro</option>

                            <option value="Hospital" {{ old('category') == 'Hospital' ? 'selected' : '' }}>Hospital</option>

                            <option value="Escola" {{ old('category') == 'Escola' ? 'selected' : '' }}>Escola</option>

                            <option value="Outro" {{ old('category') == 'Outro' ? 'selected' : '' }}>Outro</option>

                        </select>

                        @error('category')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="d-grid">

                        <button type="submit" class="btn btn-primary">

                            <i class="fas fa-save me-1"></i>Salvar Local

                        </button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

    // Inicializar o mapa

*var* map = L.map('map').setView([-23.5505, -46.6333], 10); // São Paulo como centro inicial

    // Adicionar camada do OpenStreetMap

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

        attribution: '© OpenStreetMap contributors'

    }).addTo(map);

*var* marker = null;

    // Evento de clique no mapa

    map.on('click', *function*(*e*) {

*var* lat = e.latlng.lat;

*var* lng = e.latlng.lng;

        // Preencher os campos de latitude e longitude

        document.getElementById('latitude').value = lat.toFixed(8);

        document.getElementById('longitude').value = lng.toFixed(8);

        // Remover marcador anterior se existir

        if (marker) {

            map.removeLayer(marker);

        }

        // Adicionar novo marcador

        marker = L.marker([lat, lng]).addTo(map);

        marker.bindPopup(`

            <div class="text-center">

                <strong>Coordenadas selecionadas:</strong><br>

                Lat: ${lat.toFixed(8)}<br>

                Lng: ${lng.toFixed(8)}

            </div>

        `).openPopup();

    });

});

</script>

@endsection

edit.blade.php

@extends('layouts.app')

@section('title', 'Editar ' . $location->name . ' - Laravel Maps App')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h1><i class="fas fa-edit me-2"></i>Editar Local</h1>

            <a href="{{ route('locations.show', $location->id) }}" class="btn btn-outline-secondary">

                <i class="fas fa-arrow-left me-1"></i>Voltar

            </a>

        </div>

    </div>

</div>

<div class="row">

    <!-- Mapa para seleção de coordenadas -->

    <div class="col-md-8">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-map me-2"></i>Atualizar Localização

                </h5>

            </div>

            <div class="card-body">

                <div id="map" style="height: 400px;"></div>

                <div class="mt-3">

                    <small class="text-muted">

                        <i class="fas fa-info-circle me-1"></i>

                        Clique no mapa para atualizar as coordenadas do local

                    </small>

                </div>

            </div>

        </div>

    </div>

    <!-- Formulário -->

    <div class="col-md-4">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-edit me-2"></i>Editar Informações

                </h5>

            </div>

            <div class="card-body">

                <form action="{{ route('locations.update', $location->id) }}" method="POST">

                    @csrf

                    @method('PUT')

                    <div class="mb-3">

                        <label for="name" class="form-label">Nome do Local \*</label>

                        <input type="text" class="form-control @error('name') is-invalid @enderror"

                               id="name" name="name" value="{{ old('name', $location->name) }}" required>

                        @error('name')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="mb-3">

                        <label for="description" class="form-label">Descrição</label>

                        <textarea class="form-control @error('description') is-invalid @enderror"

                                  id="description" name="description" rows="3">{{ old('description', $location->description) }}</textarea>

                        @error('description')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="row">

                        <div class="col-6">

                            <div class="mb-3">

                                <label for="latitude" class="form-label">Latitude \*</label>

                                <input type="number" step="any" class="form-control @error('latitude') is-invalid @enderror"

                                       id="latitude" name="latitude" value="{{ old('latitude', $location->latitude) }}" required>

                                @error('latitude')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                        <div class="col-6">

                            <div class="mb-3">

                                <label for="longitude" class="form-label">Longitude \*</label>

                                <input type="number" step="any" class="form-control @error('longitude') is-invalid @enderror"

                                       id="longitude" name="longitude" value="{{ old('longitude', $location->longitude) }}" required>

                                @error('longitude')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                    </div>

                    <div class="mb-3">

                        <label for="address" class="form-label">Endereço</label>

                        <input type="text" class="form-control @error('address') is-invalid @enderror"

                               id="address" name="address" value="{{ old('address', $location->address) }}">

                        @error('address')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="mb-3">

                        <label for="category" class="form-label">Categoria</label>

                        <select class="form-select @error('category') is-invalid @enderror" id="category" name="category">

                            <option value="">Selecione uma categoria</option>

                            <option value="Restaurante" {{ old('category', $location->category) == 'Restaurante' ? 'selected' : '' }}>Restaurante</option>

                            <option value="Hotel" {{ old('category', $location->category) == 'Hotel' ? 'selected' : '' }}>Hotel</option>

                            <option value="Shopping" {{ old('category', $location->category) == 'Shopping' ? 'selected' : '' }}>Shopping</option>

                            <option value="Parque" {{ old('category', $location->category) == 'Parque' ? 'selected' : '' }}>Parque</option>

                            <option value="Museu" {{ old('category', $location->category) == 'Museu' ? 'selected' : '' }}>Museu</option>

                            <option value="Teatro" {{ old('category', $location->category) == 'Teatro' ? 'selected' : '' }}>Teatro</option>

                            <option value="Hospital" {{ old('category', $location->category) == 'Hospital' ? 'selected' : '' }}>Hospital</option>

                            <option value="Escola" {{ old('category', $location->category) == 'Escola' ? 'selected' : '' }}>Escola</option>

                            <option value="Outro" {{ old('category', $location->category) == 'Outro' ? 'selected' : '' }}>Outro</option>

                        </select>

                        @error('category')

                            <div class="invalid-feedback">{{ $message }}</div>

                        @enderror

                    </div>

                    <div class="d-grid">

                        <button type="submit" class="btn btn-warning">

                            <i class="fas fa-save me-1"></i>Atualizar Local

                        </button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

    // Inicializar o mapa

*var* map = L.map('map').setView([{{ $location->latitude }}, {{ $location->longitude }}], 15);

    // Adicionar camada do OpenStreetMap

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

        attribution: '© OpenStreetMap contributors'

    }).addTo(map);

    // Adicionar marcador inicial

*var* marker = L.marker([{{ $location->latitude }}, {{ $location->longitude }}]).addTo(map);

    // Evento de clique no mapa

    map.on('click', *function*(*e*) {

*var* lat = e.latlng.lat;

*var* lng = e.latlng.lng;

        // Atualizar campos de latitude e longitude

        document.getElementById('latitude').value = lat.toFixed(8);

        document.getElementById('longitude').value = lng.toFixed(8);

        // Mover marcador para nova posição

        marker.setLatLng([lat, lng]);

        marker.bindPopup(`

            <div class="text-center">

                <strong>Nova localização:</strong><br>

                Lat: ${lat.toFixed(8)}<br>

                Lng: ${lng.toFixed(8)}

            </div>

        `).openPopup();

    });

});

</script>

@endsection

index.blade.php

@extends('layouts.app')

@section('title', 'Locais - Laravel Maps App')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h1><i class="fas fa-map-marker-alt me-2"></i>Locais Cadastrados</h1>

            <a href="{{ route('locations.create') }}" class="btn btn-primary">

                <i class="fas fa-plus me-1"></i>Adicionar Local

            </a>

        </div>

    </div>

</div>

<div class="row">

    <div class="col-md-8">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-map me-2"></i>Mapa Interativo

                </h5>

            </div>

            <div class="card-body">

                <div id="map"></div>

            </div>

        </div>

    </div>

    <div class="col-md-4">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-list me-2"></i>Lista de Locais

                </h5>

            </div>

            <div class="card-body">

                @if($locations->count() > 0)

                    <div class="list-group">

                        @foreach($locations as $location)

                            <div class="list-group-item location-card">

                                <div class="d-flex justify-content-between align-items-start">

                                    <div>

                                        <h6 class="mb-1">{{ $location->name }}</h6>

                                        @if($location->category)

                                            <span class="badge bg-secondary mb-2">{{ $location->category }}</span>

                                        @endif

                                        @if($location->address)

                                            <p class="mb-1 text-muted small">

                                                <i class="fas fa-map-pin me-1"></i>{{ $location->address }}

                                            </p>

                                        @endif

                                        <p class="mb-1 text-muted small">

                                            <i class="fas fa-location-arrow me-1"></i>

                                            {{ $location->latitude }}, {{ $location->longitude }}

                                        </p>

                                    </div>

                                    <div class="btn-group btn-group-sm">

                                        <a href="{{ route('locations.show', $location->id) }}"

                                           class="btn btn-outline-primary btn-sm">

                                            <i class="fas fa-eye"></i>

                                        </a>

                                        <a href="{{ route('locations.edit', $location->id) }}"

                                           class="btn btn-outline-warning btn-sm">

                                            <i class="fas fa-edit"></i>

                                        </a>

                                        <form action="{{ route('locations.destroy', $location->id) }}"

                                              method="POST" class="d-inline">

                                            @csrf

                                            @method('DELETE')

                                            <button type="submit" class="btn btn-outline-danger btn-sm"

                                                    onclick="return confirm('Tem certeza que deseja excluir este local?')">

                                                <i class="fas fa-trash"></i>

                                            </button>

                                        </form>

                                    </div>

                                </div>

                            </div>

                        @endforeach

                    </div>

                @else

                    <div class="text-center py-4">

                        <i class="fas fa-map-marker-alt fa-3x text-muted mb-3"></i>

                        <h5 class="text-muted">Nenhum local cadastrado</h5>

                        <p class="text-muted">Clique em "Adicionar Local" para começar!</p>

                    </div>

                @endif

            </div>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

    // Inicializar o mapa

*var* map = L.map('map').setView([-23.5505, -46.6333], 10); // São Paulo como centro inicial

    // Adicionar camada do OpenStreetMap

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

        attribution: '© OpenStreetMap contributors'

    }).addTo(map);

    // Array para armazenar os marcadores

*var* markers = [];

    // Dados dos locais do PHP

*var* locations = {!! json\_encode(*$locations*) !!};

    // Adicionar marcadores para cada local

    locations.forEach(*function*(*location*) {

*var* marker = L.marker([location.latitude, location.longitude])

            .addTo(map)

            .bindPopup(`

                <div class="text-center">

                    <h6><strong>${location.name}</strong></h6>

                    ${location.category ? `<p class="badge bg-secondary">${location.category}</p>` : ''}

                    ${location.address ? `<p><i class="fas fa-map-pin"></i> ${location.address}</p>` : ''}

                    ${location.description ? `<p>${location.description}</p>` : ''}

                    <div class="mt-2">

                        <a href="/locations/${location.id}" class="btn btn-sm btn-primary">Ver Detalhes</a>

                    </div>

                </div>

            `);

        markers.push(marker);

    });

    // Ajustar o zoom para mostrar todos os marcadores

    if (markers.length > 0) {

*var* group = new L.featureGroup(markers);

        map.fitBounds(group.getBounds().pad(0.1));

    }

});

</script>

@endsection

show.blade.php

@extends('layouts.app')

@section('title', $location->name . ' - Laravel Maps App')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h1><i class="fas fa-map-marker-alt me-2"></i>{{ $location->name }}</h1>

            <div>

                <a href="{{ route('locations.edit', $location->id) }}" class="btn btn-warning me-2">

                    <i class="fas fa-edit me-1"></i>Editar

                </a>

                <a href="{{ route('locations.index') }}" class="btn btn-outline-secondary">

                    <i class="fas fa-arrow-left me-1"></i>Voltar

                </a>

            </div>

        </div>

    </div>

</div>

<div class="row">

    <!-- Mapa com localização -->

    <div class="col-md-8">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-map me-2"></i>Localização no Mapa

                </h5>

            </div>

            <div class="card-body">

                <div id="map" style="height: 400px;"></div>

            </div>

        </div>

    </div>

    <!-- Detalhes do local -->

    <div class="col-md-4">

        <div class="card">

            <div class="card-header">

                <h5 class="card-title mb-0">

                    <i class="fas fa-info-circle me-2"></i>Informações do Local

                </h5>

            </div>

            <div class="card-body">

                <div class="mb-3">

                    <label class="form-label fw-bold">Nome:</label>

                    <p class="mb-0">{{ $location->name }}</p>

                </div>

                @if($location->category)

                <div class="mb-3">

                    <label class="form-label fw-bold">Categoria:</label>

                    <p class="mb-0">

                        <span class="badge bg-secondary">{{ $location->category }}</span>

                    </p>

                </div>

                @endif

                @if($location->description)

                <div class="mb-3">

                    <label class="form-label fw-bold">Descrição:</label>

                    <p class="mb-0">{{ $location->description }}</p>

                </div>

                @endif

                @if($location->address)

                <div class="mb-3">

                    <label class="form-label fw-bold">Endereço:</label>

                    <p class="mb-0">

                        <i class="fas fa-map-pin me-1"></i>{{ $location->address }}

                    </p>

                </div>

                @endif

                <div class="mb-3">

                    <label class="form-label fw-bold">Coordenadas:</label>

                    <p class="mb-0">

                        <i class="fas fa-location-arrow me-1"></i>

                        {{ $location->latitude }}, {{ $location->longitude }}

                    </p>

                </div>

                <div class="mb-3">

                    <label class="form-label fw-bold">Data de Criação:</label>

                    <p class="mb-0">{{ $location->created\_at->format('d/m/Y H:i') }}</p>

                </div>

                @if($location->updated\_at != $location->created\_at)

                <div class="mb-3">

                    <label class="form-label fw-bold">Última Atualização:</label>

                    <p class="mb-0">{{ $location->updated\_at->format('d/m/Y H:i') }}</p>

                </div>

                @endif

                <hr>

                <div class="d-grid gap-2">

                    <a href="{{ route('locations.edit', $location->id) }}" class="btn btn-warning">

                        <i class="fas fa-edit me-1"></i>Editar Local

                    </a>

                    <form action="{{ route('locations.destroy', $location->id) }}" method="POST">

                        @csrf

                        @method('DELETE')

                        <button type="submit" class="btn btn-danger w-100"

                                onclick="return confirm('Tem certeza que deseja excluir este local?')">

                            <i class="fas fa-trash me-1"></i>Excluir Local

                        </button>

                    </form>

                </div>

            </div>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

    // Inicializar o mapa

*var* map = L.map('map').setView([{{ $location->latitude }}, {{ $location->longitude }}], 15);

    // Adicionar camada do OpenStreetMap

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

        attribution: '© OpenStreetMap contributors'

    }).addTo(map);

    // Adicionar marcador do local

*var* marker = L.marker([{{ $location->latitude }}, {{ $location->longitude }}])

        .addTo(map)

        .bindPopup(`

            <div class="text-center">

                <h6><strong>{{ $location->name }}</strong></h6>

                @if($location->category)

                    <p class="badge bg-secondary">{{ $location->category }}</p>

                @endif

                @if($location->address)

                    <p><i class="fas fa-map-pin"></i> {{ $location->address }}</p>

                @endif

                @if($location->description)

                    <p>{{ $location->description }}</p>

                @endif

            </div>

        `);

    // Abrir popup automaticamente

    marker.openPopup();

});

</script>

@endsection

## 12) OpenStreetMap (O que é e como usar)

OpenStreetMap (OSM) é um projeto colaborativo que produz dados geográficos livres. Existem duas coisas diferentes que você pode usar em um aplicativo web:

* Os dados OSM (os bancos com ruas, pontos de interesse) — são conteúdos sob licença ODbL.
* Os tiles (imagens de mapa) que desenham o mapa em tela — normalmente disponibilizados por servidores de tiles (por exemplo: tile.openstreetmap.org).

Principais pontos para entender antes de usar:

1. 1) Tiles e servidores: os tiles que o site https://tile.openstreetmap.org/ (ou https://{s}.tile.openstreetmap.org/) fornece são recursos de doadores e não são destinados a uso intensivo em produção. Se a sua aplicação for escolar ou de baixo tráfego, usar os tiles diretamente pode ser aceitável, mas sempre observe as regras de uso. Para uso pesado, escolha um provedor de tiles ou hospede seu próprio servidor de tiles se for necessário servir muitos usuários.

*Referência: Tile Usage Policy (OSM Foundation). Fonte: https://operations.osmfoundation.org/policies/tiles/*

1. 2) Atribuição (attribution): quando usar tiles ou dados do OSM, você deve exibir a atribuição adequada (por exemplo “© OpenStreetMap contributors”) de forma legível no mapa.

*Referência: Guia de Atribuição OSM. Fonte: https://www.openstreetmap.org/copyright*

1. 3) Geocoding e Nominatim: para converter endereços em coordenadas (geocoding) ou o contrário (reverse geocoding) você pode usar o serviço Nominatim (nominatim.openstreetmap.org). A instância pública tem uma política de uso: não faça requests em massa, identifique sua aplicação via 'User-Agent' ou 'referer' e considere rodar sua própria instância se for uso pesado. Sempre leia e respeite a Nominatim Usage Policy.

*Referência: Nominatim Usage Policy. Fonte: https://operations.osmfoundation.org/policies/nominatim/*

## 13) Como usar OpenStreetMap com Leaflet (exemplos práticos)

No front-end usamos a biblioteca Leaflet (JS). Exemplo básico para iniciar o mapa e adicionar uma camada de tiles do OpenStreetMap:

var map = L.map('map').setView([-23.5505, -46.6333], 12);  
  
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
 maxZoom: 19,  
 attribution: '&copy; OpenStreetMap contributors'  
}).addTo(map);

Este trecho cria o mapa centralizado em São Paulo e carrega as imagens (tiles) do servidor público do OpenStreetMap. A string de 'attribution' é obrigatória sempre que usar os tiles públicos.

*Referência: Leaflet quick start. Fonte: https://leafletjs.com/examples/quick-start/*

## 14) Geocoding com Nominatim (exemplos e boas práticas)

A instância pública do Nominatim pode ser usada para pequenas consultas. Exemplos:

* A) Geocoding (buscar por endereço) — request via fetch (front-end):

fetch('https://nominatim.openstreetmap.org/search?format=json&q=Av.+Paulista+1578&limit=1')  
 .then(res => res.json())  
 .then(data => console.log(data[0]));

* Observação: requests directos do navegador têm o header Referer set automaticamente. Para requests server-side (PHP), informe um User-Agent explicito contendo contato (ex: MinhaApp/1.0 contato@exemplo.com) para cumprir a política do serviço.

// Exemplo em PHP (server-side) usando cURL  
$ch = curl\_init('https://nominatim.openstreetmap.org/search?format=json&q=' . urlencode($address) . '&limit=1');  
curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);  
curl\_setopt($ch, CURLOPT\_USERAGENT, 'MinhaApp/1.0 (contato@exemplo.com)');  
$result = curl\_exec($ch);  
curl\_close($ch);  
$data = json\_decode($result, true);

* Boas práticas Nominatim: adicionar atraso entre requests, cachear resultados (não repetir consultas idênticas), e para uso intenso, instale sua própria instância do Nominatim ou use um serviço comercial.

*Referência: Nominatim homepage/documentação. Fonte: https://nominatim.org/*

## 15) Solução de problemas comuns

• Erro 1045 (Access denied): verifique usuário/senha no .env.  
• Erro 1049 (Unknown database): confirme se o banco foi criado no Workbench.  
• Tiles não carregam: verifique internet, console do navegador e se o URL de tiles está correto.  
• Erros na API Nominatim: confira User-Agent e verifique se não está excedendo limites.

## Anexos: links úteis e referências

Leaflet Quick Start: https://leafletjs.com/examples/quick-start/

Leaflet API reference: https://leafletjs.com/reference.html

OSM Tile Usage Policy: https://operations.osmfoundation.org/policies/tiles/

Nominatim Usage Policy: https://operations.osmfoundation.org/policies/nominatim/

Nominatim (home/documentation): https://nominatim.org/

OSM Attribution & Copyright: <https://www.openstreetmap.org/copyright>

Erro na tabela e Migração

php artisan make:migration fix\_locations\_table --table=locations

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

public function up(): void

{

Schema::table('locations', function (Blueprint $table) {

if (!Schema::hasColumn('locations', 'name')) {

$table->string('name')->after('id');

}

if (!Schema::hasColumn('locations', 'description')) {

$table->text('description')->nullable()->after('name');

}

if (!Schema::hasColumn('locations', 'latitude')) {

$table->decimal('latitude', 10, 8)->after('description');

}

if (!Schema::hasColumn('locations', 'longitude')) {

$table->decimal('longitude', 11, 8)->after('latitude');

}

if (!Schema::hasColumn('locations', 'address')) {

$table->string('address')->nullable()->after('longitude');

}

if (!Schema::hasColumn('locations', 'category')) {

$table->string('category')->nullable()->after('address');

}

});

}

public function down(): void

{

Schema::table('locations', function (Blueprint $table) {

$table->dropColumn(['name', 'description', 'latitude', 'longitude', 'address', 'category']);

});

}

};